Flexible and Smooth Trajectory Generation based on Parametric Clothoids for Nonholonomic Car-like Vehicles

Suhyeon Gim

N° d'ordre: D.U. 2818
EDSPIC: 801

UNIVERSITÉ CLERMONT AUVERGNE
ÉCOLE DOCTORALE
SCIENCES POUR L'INGÉNIEUR DE CLERMONT-FERRAND

THÈSE

Soutenue le 27 juin 2017

présentée par

Suhyeon GIM

Ingénieur Mécanique
Master Robotique

pour obtenir le grade de :

DOCTEUR DE L'UNIVERSITÉ CLERMONT AUVERGNE
Spécialité : VISION pour la ROBOTIQUE

Titre de la thèse :

Flexible and Smooth Trajectory Generation based on Parametric Clothoids for Nonholonomic Car-like Vehicles

COMPOSITION DU JURY

Woojin CHUNG	Rapporteur	Professeur, ISR-KU
Thierry FRAICHARD	Rapporteur	Chercheur à l'INRIA, Rhône-Alpes
Ouiddad LABBANI-IGBIDA	Examinateur	Professeur, XLIM-UL
Youcef MEZOUAR	Examinateur	Professeur, IP-SIGMA
Sukhan LEE	Directeur de thèse	Professeur, ISRI-SKKU
Lounis ADOUANE	Directeur de thèse	Maître de Conférences-HDR, IP-UCA

Remerciements

This thesis work has started from Double PhD Degree Program with cotutorship convention between University Blaise Pascal (France, currently University Clermont Auvergne) and SungKyunKwan University (South Korea). The program was supported by the Gyeonggi Province in Republic of Korea with the project of *Cognitive Personal Transport Service Robot*. This thesis is an accomplishment during the study in Intelligent Systems Research Institute of SungKyunKwan University and Institut Pascal of University Clermont Auvergne.

I actually give my great honor to this thesis directors and advisors, Prof. Lounis Adouane, Jean-Pierre Derutin and Sukhan Lee. Without their sincere guidance and supports, this thesis could not have finished yet. Especially for Lounis Adouane, his great elaborate collaboration and detailed proof reading for all through the manuscript enable this work to be end. Prof. Sukhan Lee who is one of the most celebrated scholar in this field, pull my work to go forward by his in-depth knowledge and research experiences. I have much to present my gratitude to Prof. Jean-Pierre Derutin for his full consideration and assistance during my study in Institute Pascal. But, it is unfortunate and regrettable that he could not participate to the jury of my thesis.

About this thesis and my major journal work, I could not leave out his/her excellent reference and elaborate review from Prof. Doran K. Wilde, Prof. Thierry Fraichard, Prof. Woojin Chung, Prof. Ouiddad Labbani-Igbida and Prof. Youcef Mezouar. In my future, I will endeavor my research capability to reach at such a glorious academy as much as I can. Then, I need to attribute my research benefits to my favorite colleagues, Mahmoud Khojet-Kesba, User Ferit and Zhao Xhinsuang who start this cotutorship program in the same time as well as Marko Jerkovic, Abdrakhmanov Rustem, Kwansuk Yang, Chunsung Nam and Kangwon Nam for their sincere collaborations. I also would like to thank to their great works for Jose Miguel Vilca Ventura, Ahmed Benzerrouk and Pierre Avanzini.

Lastly but not the least, I am perfectly appreciate my family, my only daughter Hanyoung and my wife Euiseon. Furthermore, I am always happy to share the news that this hard work has been successfully completed to the families of my father and brothers as well as my wife. And, I really devote this years-old work to my late mother.

Contents

List of Figures

List of Tables

Glossaire

ICR: **I**nstantaneous **C**enter of **R**otation.

RS path: **R**eeds-**S**hepp path.

FS path: **F**raichard-**S**cheur path.

RRT: **R**apidly-exploring **R**andom **T**rees.

CCP: **C**ontinuous **C**urvature **P**ath.

pCCP: **p**arametric **C**ontinuous **C**urvature **P**ath.

d-pCCP: **p**arametric **C**ontinuous **C**urvature **P**ath for **d**ynamic target.

h-CCP: **h**uman inspired **C**ontinuous **C**urvature **P**ath.

MSC: **M**inimax **S**harpness **C**onstraint.

C-S diagram: **C**urvature and **S**harpness diagram.

FA: **F**ront **A**voidance.

RA: **R**ear **A**voidance.

Introduction

Development of Autonomous navigated vehicles and current issues

Autonomous navigation has attracted much attention to research as well as industry for several decades. Some prototypes of autonomous navigated vehicles are currently on the road test for its performance verification and many vehicle companies have also appllied autonomous navigation technology to their commercialized models with several safety modules of ADAS (Advanced Driver Assistance System) such for instance LDWS (Lane Departure Warning System) and ACC (Adaptive Cruise Control) by integrated high-end sensor and control systems.

Several autonomous vehicles or self-driving cars for passengers and freight transportation are currently on the road test by Google (*Waymo*), Apple (*Apple Car*), Tesla (*Autopilot*), Ford (*Argo*), GM (*Chevy Bolt*)[1] and so on (cf. Fig. 1). Almost all main companies expect the fully autonomous navigated vehicle (Level 4, cf. **Appendix A**) in 2021 [Goodman 16, Abele 16, Blog 17]. Uber and Lyft have also taken part in the competition of autonomous navigation to provide driverless call-taxi service [Goldman 15, Blog 17].

In the control of autonomous navigated vehicle, there are still several unresolved issues as summarized in [Adouane 16] (Chapter 1) or [Muddhor 16]. Among them, two major subjects are important, the first corresponds to the perception/localization part and the second is related to planning/control.

In terms of localization of the vehicle, GPS (Global Positioning System)/IMU (Inertial Measurement Units) are largely used and fused with other sensors such

[1] Company (its project/vehicle name).

(a) Google Car (Renamed to Waymo)

(b) Tesla with Autopilot

(c) Uber's autonomous call-taxi

Figure 1: Autonomous navigated vehicles on the road

as LiDAR (Light Detection and Ranging sensor) and odometry encoders to estimate the vehicle's pose [Dudek 10, Siegwart 11, Jo 14]. However, accurate perception/localization (for safe driving) depend on the environment features, such as: the whether, structured or unstructured and the status of static or/and dynamic obstacles. As the number of sensors and the complexity of data processing systems increase, the safety and accuracy could be driven to higher stage, however the cost effectiveness or time-consumption also remains as an issue to overcome.

With regards to planning and control, several important works have been developed since the last decades [Siegwart 11, Siciliano 16]. This thesis manuscript will focus on the planning and re-planning aspects. This topic is very important in order to guarantee the vehicle safety (no collision with static/dynamic obstacles) and to ensure driving performance (passenger comfort for instance) during the navigation.

The path planning is closely related to the localization of the vehicle since the exact path and its following control are based on the exact localization of the vehicle itself [Thrun 06a][2]. Path planning depends on the environmental status and map scale and it is categorized as global and local. When a global path is required in the digitized macro-scale map, simple geometric segments (line, arc or spline) are sufficient to generate the path which passes through the given way points, otherwise for local path/trajectory planning since the data is only local, it is commont to define/specify detailed set of actual control commands (including its speed) to follow the path while maintaining acceptable position accuracy in the real road.

Prevailing topics in the commercialized vehicles are also concentrated on the localization and path planning in the structured road with huge data given for instance the surroundings from LiDAR (or Radar) and vision. However, currently there still remain several technological issues to overcome as follows. The first one is enhancement of localization beyond GPS/IMU/odometry sensors. The GPS-based driving suffers from signal disturbances, data receiving failure, large error from ground truth. Common examples are found as GPS outage or bad accuracy in a vally of tall building, tunnel and underground parking lot as well as bad weather conditions, and difficult loop closure detection by laser reflective data or image feature matching algorithm in incessantly varying boundary edges by walking people, moving vehicles or windy/rainy/blinking road surface[3]. The IMU and odometry with its data fusion can be an alternative or supportive system to the only GPS-based positioning of the vehicle, but obataining the absolute

[2]This topic has been largely and intensively dealt with in SLAM (Simultaneous Localization And Mapping) research.

[3]Recently, a number of those drawbacks for autonomous navigation of Uber or Tesla on service have been reported [Muoio 16, Ctvnews 17, Bhuiyan 17].

orientation is not an easy task due to magnetic fields corruption around electric system. It is also true that the vehicle system of fusing range sensor such as LiDAR, ultrasonic and vision processing could come into being blind when there is no salient feature to detect or crowded or noisy environment. Actually, a lot of methods using filtering and smoothing including visual odometry loop closure [Nister 06] have been developed so far, however there are still some problems to overcome in the GPS-denied region and accumulated heading error in IMU [Guo 14, Jo 14] as well as accumulated travel distance error on tyre's ground slip [Lee 01, Smith 08, Truax 08, Pacejka 92].

Another important issue can be found in the path following of the vehicle having nonholonomic constraint [Lavalle 06, Sprunk 08, Dudek 10]. The vehicle can follow the path exactly by controlling its steering wheel and accelerating/braking while overcoming the position errors coming from friction and slip betweeon tyres and ground surface which has nonlinearity and numerous exterior/interior factors. This issue is closely related to the localization issue (which is mentioned above) since only the exact positioning promises the satisfied localization performance on path following or trajectory tracking.

The last issue to be considered is about human factors in the path planning strategy. Even if the path planning is accomplished in the point of target reaching, shortest travel length (fuel economy) or minimized travel time, the path could deteriorate the human passenger's comfort and not well optimized for human aspects of transportation feelings. Additionally, when a vehicle is controlled to follow a path, the driving comfort could be ignored to human passenger and sometimes raise discomfort or even head ache. In this respect, it is important to consider some aspects of expert human drivers to the path generation methodology. This thesis is constructed from the considerations of the above three described issues as how to increase accurate perception/localization, how to increase path following performance and how to consider human aspect.

This thesis deals with path generation method using one or multiple clothoids element which has continuous curvature property [Gim 17]. The continuous curvature is regarded as a major chatacteristic to generate a path for nonholonomic car-like vehicles in overall chapters and the resulting continuous curvature paths are constructed from the geometric/kinetic compatibility and algorithmic procedures. The author presumes that the well-defined vehicle kinematics/dynamics model could benifit much from the generated continuous curvature path. More specifically, the curvature according to the travel length from the obtained path solution could provide a steering reference (or set-points) along the driving distance. This odometry-based navigation strategy which is also a kind of open loop strategy for the vehicle actuators with dead-reckoning and provides a *Virtual rail*[4]

[4]This conceptual terminology is not common yet in the field of path planning or autonomous

for the vehicle, could be effective especially in the GPS-denied region or when confronting with bad status or failure in the localization system. Obviously, this navigation requires to localize the vehicle periodically with absolute way during all the travel time, however it could lengthen the checking period or distance without localization, thus reduce the computational burden (compared to the other path generation methods).

Related works and background

This thesis has been progressed from the remarkable works that were previously performed by other researchers in the following literatures. At first, it requires to mention on the path generation methods (or trajectory generation by including velocity planning) and its categorization. According to the scale and covering range of the path generation scheme, two kinds of planners (or generation algorithms) are categorized as global or local [Solea 06, Villagra 12, Adouane 17].

The global path generation concerns about general navigational information (usually acquired from GPS) for a vehicle to travel from its initial position to a final position through the successive multiple way points [Macek 09, Vilca 15]. This global planner is also aware of the overall map about the environment [Elfes 89] which is represented by topological information (road networks) or gridy tessellation, but it lacks specific local data for a driving vehicle, such as vehicle size, lanes, speed bumps, obstacles, and road boundaries. Representative examples in the global planner could be found in the navigation software equiped in the commercial vehicle and GIS (Geographic Information System) algorithms with high-precision map [Abele 16].

The local path generation, on the contrary, uses physical scale information taken from on-board sensors, and it focuses more on its actual driving in the short distance range from the vehicle's current configuration to the target configuration. This planner provides sufficiently precise path for the vehicle's feasible following and it is usually applied to lane change or obstacle avoidance maneuvers. Artificial potential field method [Khatib 86a] is one of the most representative works for local path planning. Authors in [Mouad 12] used this method for planning and re-planning one mission for a group of mobile robots which evolve in structured environment to attain different way-points. Such local planner is more reactive [Khatib 86b, Kelly 03, Likhachev 09, Chebly 15] in the sense that it can deal with dynamic obstacles [Chakravarthy 98, Giesbrecht 04, Fulgenzi 07, Chung 09]. A multitude of local paths are integrated to obtain the global path for navigation

navigated vehicles, thus it requires to be specified and concretized with further analytical and experimental investigations in near future.

[Adouane 13, Adouane 17] or locally path smoothing on the given global path [Yang 13]. As an another local planner, road map (or topological map) approach [Dudek 10] constructs the path by lines or curves, where visibility graph and voronoi diagram are two different kinds of the approach. To cope with obstacle avoidance path planning, the visibility graph has closest line for obstacle polygon vertex, but voronoi path maintains far away as possible from the obstacle polygonal vertices [Siegwart 11, Marie 13]. However, a defect of these paths are that they lack curvature continuity.

There also has been hybrid path generation methods which integrate global and local planning in autonomous navigated vehicles [Adouane 17, Lozenguez 11]. The authors in [Thrun 06b] used a search algorithm minimizing a cost fuction with a vehicle kinematic constraint, and [Urmson 06] used the A^* search to find the local path in the graph. The authors in [Trepagnier 06] utilized a cubic B-spline for smoothing the obtained trajectory, while [Hundelshausen 08] applied its local path as a number of clothoids, and a similar clothoid-like arc path was constructed in [Ziegler 08] using the A^* search heuristic algorithm.

One of largely adopted hybrid methods is the *Rapidly-exploring Random Trees* (**RRT**) [Lavalle 99]. The **RRT** finds the path in a growing tree-like structure and covers a large domain, thus it works well as a global path planner; the drawback of the method is that it often provides a non-smooth path (in term of curvature) to be followed by the vehicle. To make up for the smoothness in local path planning, advances have been demonstrated mainly by applying the optimization function [Kelly 03, Dolgov 10, Zucker 10, Xu 12] or integrating spline methods [Lan 15]. The work of [Lin 14] applied **RRT**-based path planning for moving obstacle avoidance with unmanned aerial vehicle system, where each node is connected by propagating along Dubins curve in tree expansion. In [Lan 15], the **RRT** path was constructed by G^2 continuous cubic Bézier spiral which was developed from [Yang 13].

Scope and contribution of the proposed work

The proposed path generation method in this thesis begins with the local path generation using clothoid segment, but it can facilitate the global path generation by connecting consecutive local paths. Thus, the proposed method can cover both the local and global path generation.

The previous researches using clothoids as the path component are summarized as follows. The work of [Walton 09] is performed to describe methods for G^1 *Hermite* interpolation with a single clothoid segment, where the interpolation problem is reduced to a single nonlinear equation that was solved nu-

merically with *Newton-Raphson* method. By a graphical inspection, this work named its problem as *C*-shape and *S*-shape by the geometric condition, where *C*-shape curve is generated by one clothoid and *S*-shape is from two (elementary) clothoids of positive curvature to negative curvature case (or vice versa). However, solutions of [Walton 09] which enhances also the work of [Kimia 03] were not satisfying the curvature continuity, but only position and orientation continuity conditions. To overcome these drawbacks, various smoothing techniques for the obtained path have been used such as cubic spiral, B-spline, trigonometric splines [Lamiraux 01, Montes 07, Avanzini 12]. As non-spline based method, Dubins [Dubins 57] (*Dubins* path) and Reeds-Shepp [Reeds 90] (**RS** path) proposed smooth path generation methods for nonholonomic vehicles which are composed by line and arc segments and yield to the shortest travel length; however, their models lack curvature continuity. In the work of [Macek 06], the line segment path is interpolated with a B-spline curve in order to generate a feasible trajectory that takes into account nonholonomic constraints.

To generate a smooth path for a nonholonomic car-like vehicle, the curvature continuity is an important part to be considered since it is closely related to the steering smoothness and reduction of the undesirable jerk. There have also been continuous curvature path generation methods since it has a close relationship with vehicle steering and driving performance [Montes 07, Labakhua 08, Choi 10]. However, each proposed path is effective only to limited configurations cases.

The authors in [Fraichard 04] proposed a continuous curvature path (**FS** path) in the absence of obstacles. The **FS** path originates from the **RS** path, but it includes clothoid having a fixed sharpness (rate of curvature) so that the vehicle does not need to stop to reorient its front wheels. The authors in [Parlangeli 10] proposed an infinitely differentiable smooth path that approximates *Dubins* paths with bounded curvature and sharpness. In [Montes 07], the overtaking problem was addressed using clothoidal trajectories by approximating the rational *Bézier* curves. The study used the homothetic factor adjusted by scalable parameters and control points; however, the solution needs a large amount of calculation for the multi-dimensional linear algebraic equation to determine a number of *Bézier* control points [Chevallereau 02], and it is also difficult to acquire the curvature information from the resultant curve. The authors of [Walton 05] addressed the multiple clothoids problem to satisfy two boundary configurations, and proved its solvability and solution uniqueness through rigorous mathematical analysis. However, the solution and its algorithm are not sufficient for curvature or sharpness parameter analysis and thus, it is hard to get curvature diagram from the solution. Moreover, paired clothoids require control points that must be provided manually by a human designer. The work of [Wilde 09] (*Wilde*'s path) presented a simple and fast path generation method with continuous curvature of minimum

sharpness that is close to human driving. The solution is fast, computationally effective and considers human aspect, but it is limited to a straight lane change maneuver example.

To attain the goal of autonomous navigation for nonholonomic vehicle, the path following is also an important task, where the task largely depends on the smoothness of the given path. For the path following (or trajectory tracking), the work of [Solea 06] produced a smooth path in the respect to the human comfort standard while following the path by sliding mode control. The path came from the predefined quintic polynomial but, it resulted in some discontinuous steering angle while keeping its robust and fast error diminuation. As for the result, authors expected that the system might suffer from other errors in the real vehicle experiment caused from discontinuous steering control. Authors in [Biagiotti 08] proposed kinematic model of the car like robot and its general trajectory tacking. Authors in [Labakhua 08] made smooth trajectory by cubic and trigonometric splines and its control results were compared to the clothoidal planning with consideration of human comfort criteria. It concluded that the use of the clothoid curves is better for the performance but it is more complex and hard to use the curves for trajectory change. In the paper of [Brambley 11], to endow the physical feasibility to the aerospace path planning, a hyperbola is considered to connect two straight lines where it approximates the double clothoidal segments. It tries to facilitate the key clothoidal properties by hyperbolic parameters which include the path length and acceleration but it is only applicable for the symmetric clothoids case. The work of [Ghita 12] deals with autonomous navigation covering from the path generation to the following control with the elaborating work for the feasible path using cell decomposition. It tries to connect *Dubins* path to control nonholonomic vehicles for the known map with static obstacles. In the work of [Avanzini 12], automatic multi-vehicle navigation for urban transportation, multi-vehicle system shares a common reference trajectory with the compensation of the localization by outward sensors where the trajectory convoying was locally optimized by *B*-spline interpolation method.

As described above, the research of path generation has been developed for its smoothness in order to be followed by nonholonomic vehicles. One of the main contributions in this thesis is the path generation methods by ensuring the curvature continuity and that purpose is attained by algorithmic solution for the clothoids composition. Several algorithms are proposed to solve the inverse clothoid problem, which finds the clothoids' parameters satisfying boundary configurations, i.e., initial and final posture to reach while having always continuous curvature between the two configurations.

Another main objective of this thesis is to provide a general solution for local path generation problem, where the problem is defined by two boundary config-

urations and the solution path is extended to global path generation problem. There exist a lot of literatures that deal with *Continuous Curvature Path* (**CCP**); however, there has been few works that provides a general solution for various configurations. In deriving solution for each **CCP** problem, several specific properties for clothoid are used, where the properties are identified from parametric variation or regulation for simple clothoid segment. From these known properties and composing rules of clothoids, algorithms for general **CCP** problem are constructed and proposed. Furthermore, the proposed algorithm for parametric (or parameter-regulated) **CCP** (**pCCP**) could be useful for generating a local path for static or dynamic obstacle avoidance as well as lane change maneuver for car-like vehicle.

As a local path planner, each problem is defined by both ends configurations specified by pose and curvature, and the solution is found by parametrically adjusted algorithmic procedure. Using clothoids as major components, the proposed algorithm gives out parametric solutions for multiple clothoids composition of which both ends meet. The resultant curvature diagram gives a reference on continuity-assured smooth steering control for the vehicles. Among the solutions, the minimax sharpness constraint (**MSC**) is addressed to obtain unique solution as well as its effectivenss for passenger comfort. The proposed solutions are applied to local path planner and global path planner examples to varify their usefulness and effectiveness. It is also applied to global path smoothing example, where the original path is given from real road datasets and then it is solved by multiple local paths which are divided by each boundary configurations and solved by the proposed algorithms. The resultant paths are tested on the *Lyapunov*-based controller [Canudas 96, DeLuca 98, Adouane 08, Benzerrouk 14] for highlighting the performance effectiveness on path following and compared with other smooth path generation methods such as quintic polynomial, cubic spline, Bézier path for lane change maneuver.

The proposed **pCCP** has different contributions compared to previous works such as *Dubins* path, **RS** path, and **FS** path; the proposed solution allows the sharpness as a variable to determine the clothoid solution and it is found by iterative convergence with constraints. The **FS** path tackles the problem of zero-curvature at both boundary configurations *line-to-line* by fixing the sharpness in symmetric clothoids paths where the arc segment in the *Dubins* path is replaced by a maximum sharpness clothoid. Different to **FS path**, the algorithm doesn't tackles the problem with any fixed parameter constraint such as maximal sharpness or maximal curvature. It is differentiated also from **RRT** method in that while the solution accuracy in **RRT** depends on the grid resolution setting, the proposed algorithm guarantees the solution to converge within pre-defined accuracy. And the generated path could not satisfy the curvature continuity since it

uses line or arc segment to construct the path.

Another advantage of **pCCP** is that the problem of non-zero curvature (or *arc-to-arc*) configurations on both ends is addressed, where the solution can be used for obstacle avoidance or path replanning maneuver, where initial and final curvature values are determined from current steering and obstacle boundary modeling as well as road profile. The nonzero initial curvature configuration is useful to replan a pre-defined path at a nonzero steering angle. The nonzero final curvature is also important for avoiding obstacles modeled in a bounded radius circle by generating the path surrounding this circle [Chakravarthy 98, DenBerg 08].

The **pCCP** provides curvature diagram along the whole travel length and it has several advantages for the vehicle path following performance. The diagram can reproduce its odometry records of the trajectory only by inter-vehicular parameters, i.e., steering variation according to the rear wheels movement. Thus, the vehicle follows the curvature diagram while it moves and its velocity according to time, which is related to the travel length. Another benifit can be found in its usefulness as a reference path for multi-vehicle formation where, for instance, the leader can generate a reference path to be followed. The reference path could be used in autonomous navigation systems such as public transportation or platooning guidance [Villagra 12, Solea 06, Scheuren 12]. Each vehicle can follow the path with smooth steering, thus enhancing the path following performance where the trajectory could be well tracked by inter-vehicular control by steering front wheels and driving rear wheels, with a little dependency on its current pose accuracy from the exterior position sensors. The proposed **pCCP** can be used for designing the whole autonomous navigated vehicle system as shown in Fig. 2.

The system could acquire the geometric data from the global digital map, and the geometric data comprised of a number of way points or consecutive line/arc segments in order to be converted into smooth trajectory. The trajectory includes both path and velocity profiles and it is followed and tracked by the vehicle controller. Since the control result as well as path following necessarily contains pose error, thus it requires error correction by calibrating procedure. These procedures presented in Fig. 2 are performed in closed loop to ensure safe autonomous navigation. In the global framework described above, the proposed **pCCP** could be useful for making a reference trajectory [Villagra 12, Solea 06, Scheuren 12, Avanzini 12, Bae 13] or obstacle avoidance maneuver by including velocity planning as performed in [Vilca 13]. In addition to the local and global path generation methods, this thesis includes an additional problem to deal with dynamic target/obstacle. In a cluttered/dynamic environment or when obstacles are moving, path generation needs to cope with collision case and expectation on the future status. To satisfy those requirements, **pCCP** for dynamic target

(**d-pCCP**) is proposed and evaluated by simple demonstrations. This thesis also contains path generation inspired from human driving pattern, where the solution constraints are defined from human aspects such as comfort, steering speed and steering work. These considerations enhance the performance of the proposed solution and expand its applicability and interest to real experimental vehicle navigation system. This thesis is organized as follows.

Chapter 1

The path planning for nonholonomic car-like vehicle is modelled and the path following model is addressed with clothoid based representation. It is described that some clothoid properties are found to represent efficiently nonholonomic path and those properties are utilized to derive iterative solution procedure of parametric regulation.

Chapter 2

The **pCCP** is introduced and described where some basic problems are defined and analyzed using clothoid formulations. Three problems are classified by

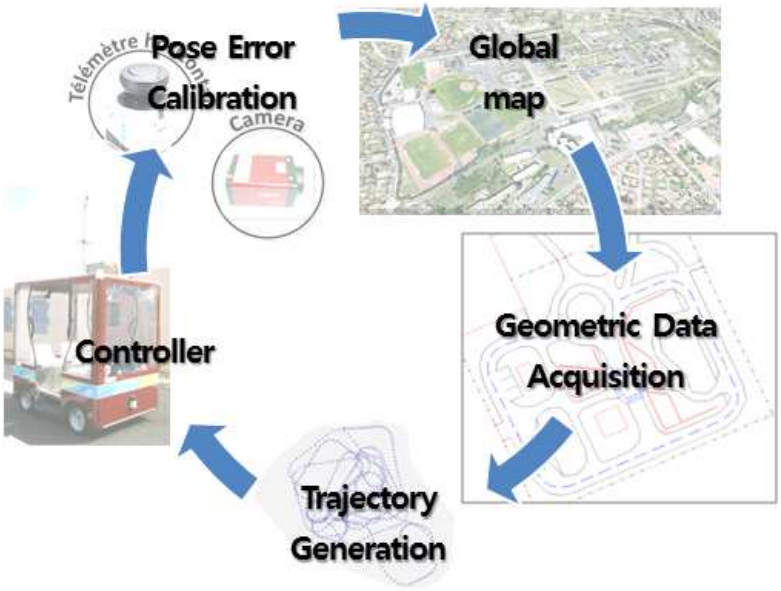


Figure 2: Global framework in autonomous navigated vehicle

boundary conditions which are differently treated to facilitate multiple clothoids composition. Each problem is solved by different algorithm using common properties and the results are investigated with path and curvature/sharpness diagram.

Chapter 3

The **d-pCCP** is addressed to extend the **pCCP** to dynamic target. The proposed path is integrated in a dynamic obstacle avoidance problem by adopting a 4D configuration space. In this space, the avoidance pose and the velocity replanning are performed for safe maneuver of the considered nonholonomic vehicle.

Chapter 4

The obtained path solutions are applied to local and global path generation examples and its velocity planning is also integrated with the proposed path in order to consider dynamic environment or several obstacle avoidance maneuvers.

Chapter 5

A methodology of human driver inspired path generation is introduced and examined with well-defined experiments to build up an optimization guidance in the **pCCP** solution. For the human comfort, the proposed path, i.e., human inspired continuous curvature path (**h-CCP**) is generated and compared with other smooth paths. Then it is evaluated by vehicle dynamics simulation in order to be implemented finally in a real vehicle system.

Followed by above chapters, this manuscript finishes with a general conclusion and several prospects.

Chapter 1

Path planning for autonomous navigated vehicle

1.1 Introduction

Path planning for nonholonomic car-like vehicle is an important requisite in autonomous navigated vehicles. In this chapter, a mathematical background about *clothoid* segment and its derivation into a nonholonomic path is performed. At first, it is focused on the relation between the nonholonomic path and the continuous curvature path (**CCP**), and then mathematical representation and integrated formulation of the clothoid is linked to apply **CCP** problem. To generate **CCP** using clothoid segment, geometric composition of the segments is presented and some basic properties are summarized to facilitate parametric variation for the composed clothoids by geometric constraints. Under some conventions which are implemented through overall chapters, parametric variation is designed to achieve algorithmic convergence; thus the concept of parametric continuous curvature path generation problem is drawn out. The described properties with conventions are used to represent and analyze the obtained results in the following overall chapters.

1.2 Path following model for nonholonomic car-like vehicle

Nonholonomic car-like vehicle is driven by acceleration/brake mechanism at rear wheels while it is steered by steering actuator at front wheels. This vehicle has a *nonholonomic* kinematics model where it is assumed that there exists no rolling contact-slip on the ground surface and the steering angle corresponds to its curvature.[1] When the vehicle is modeled as a point of the motion center, the nonholonomic vehicle kinematics are represented by following differential form.

$$\dot{x}(t) = v(t) \cdot cos\ \theta(t), \tag{1.1a}$$
$$\dot{y}(t) = v(t) \cdot sin\ \theta(t), \tag{1.1b}$$
$$\dot{\theta}(t) = \frac{v(t)}{L} \cdot tan\ \gamma(t). \tag{1.1c}$$

where, $v(t)$ and $\theta(t)$ are the linear driving velocity and orientation angle at time t respectively and $x(t)$, $y(t)$ ($x \in \mathbb{R}^2, y \in \mathbb{R}^2$) represent the position of the vehicle, middle position of rear wheels. $\gamma(t)$ is the steering angle of the vehicle at time t and it is equal to the curvature $\kappa(t)$ where the radius of curvature $\rho(t)$ is defined by the two axis interconnection at the *Instantaneous Center of Rotation* (**ICR**). L is the wheelbase between frontal and rear axis.

For the description of the whole trajectory, above equations require the whole information on the velocity and steering angle through the time. Different to the trajectory tracking, the path following requires for the vehicle to converge and follow the path without time specification where the major control is performed by its steering with some forward speed for its orientation to drive on to the path. For smoother convergence to the desired path, *virtual target* projected of the actual vehicle on the path is tackled to bypass some problem of the feedback controller [Soetanto 98, Avanzini 09] where the target is defined on *Frenet-Serret* frame $\mathcal{F}$. Fig. 1.1 depicts the nonholonomic car-like vehicle based on virtual target for the path following on the global coordinate X-Y.

In Fig. 1.1, the local frame of the vehicle is located on its motion center, $\boldsymbol{P_v}$ with orientation θ_v and its virtual target frame places on $\boldsymbol{P_t}$ with θ_t. The vehicle has the wheelbase L and steers γ with **ICR**. At $\boldsymbol{P_t}$ on the path, the pose is described by *curvilinear abscissa* s_t and it has curvature κ_t and radius of curvature

[1] This model is in accord with *Bicycle model* which is widely used in the field of nonholonomic constraint model [Webster 06, Rill 11, Ulsoy 12].

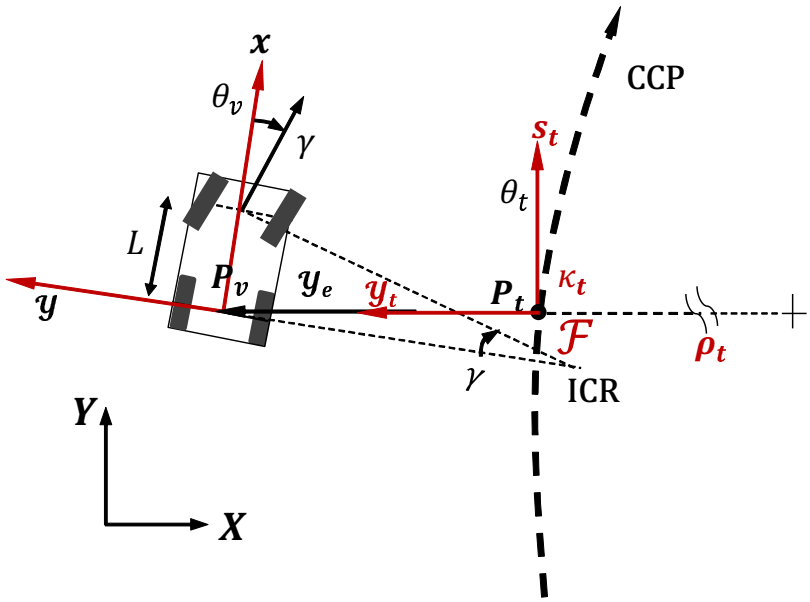


Figure 1.1: Path following model for the car-like vehicle

ρ_t. From above definition, following equations present the kinematic relations on the Frenet frame $\mathcal{F}$.

$$\begin{cases} \dot{s}_e = v\frac{\cos\theta_e}{1-y_e\kappa(s_e)}, \\ \dot{y}_e = v\sin\theta_e, \\ \dot{\theta}_e = v\left(\frac{\tan\gamma}{L} - \frac{\kappa(s_e)\cos\theta_e}{1-y_e\kappa(s_e)}\right). \end{cases} \tag{1.2}$$

In Eq.(1.2), $\mathcal{F}$ is represented as $(s_t,\ y_t)$ at $\boldsymbol{P_t}$ and $\theta_e = \theta_v - \theta_t$ where θ_v and θ_t is measured counterclockwise from X-axis on the global coordinate. The control strategy for path following must permit is to have y_e, θ_e to converge to 0 while keeping $1 - y_e\kappa(s_e) \neq 0$. For the strategy of trajectory following based on the *virtual target*, *Lyapunov* based controller are augmented to extend kinematic control to a dynamic consideration as well as its robustness to the parametrical modeling uncertainties [Samson 95].

1.3 Parametric clothoid and convergence property

A clothoid has the property to have continuous curvature which is either increasing or decreasing through the length. There are some kinds of formulations for the rate of the curvature (or *sharpness*) such as polynomials, exponential

or trigonemetric function, however, the 1^{st} order form, relying on the constant sharpness, is well-known not only for its simplicity for computation but also for physical similarity to the real vehicle actuation system. To be more specific, the curvature of a point on the path corresponds to the steering angle of the vehicle which follows the path at the point and the sharpness signifies the rate of the steering change at the point. Thus, the sharpness is related to the steering actuation which exerted by human handling motion or steering motor system.

A continuous curvature path or **CCP** can be simply represented using *clothoids*. A path defined by clothoid is efficient for analyzing and controlling the car-like vehicles in that it provides information on the curvature along the length and can also supply maneuvering set-points information to the vehicle control system.

Using the definition of clothoid (or *Euler* spiral) [Walton 05], the coordinates x, y are consecutively defined along the length s. If a basic formulation for clothoid curvature $\kappa(s)$ is assumed using a simple first-order polynomial with initially zero, the path is determined by integration procedures with length variable s as follows;

$$\kappa(s) = \alpha s \tag{1.3}$$

$$\theta(s) = \int_0^s \kappa(u)\, du \tag{1.4}$$

$$x(s) = \int_0^s \cos\ \theta(u)\, du \tag{1.5}$$

$$y(s) = \int_0^s \sin\ \theta(u)\, du \tag{1.6}$$

where α is the sharpness and $\kappa(s)$ is the curvature. Equation (1.3) determines if the curvature increases or decreases by constant sharpness α, and the orientation θ in Eq. (1.4) changes with the integration of curvature by s in Eq. (1.3). To specify the problem, it is useful to consider another formulation to describe the clothoid. By integrating all equations from Eqs. (1.3) to (1.6) in a forward manner, the clothoid in the *Cartesian* coordinate is defined as follows [Wilde 09, Labakhua 08].

$$x(s) = \sqrt{\pi/\alpha} \cdot \mathcal{FC}\left(\sqrt{\alpha/\pi}\ s\right), \tag{1.7}$$

$$y(s) = \sqrt{\pi/\alpha} \cdot \mathcal{FS}\left(\sqrt{\alpha/\pi}\ s\right). \tag{1.8}$$

where, $\mathcal{FC}$ and $\mathcal{FS}$ mean *Fresnel Cosine and Sine Integral* as,

$$\mathcal{FC}(s') = \int_0^{s'} cos\ \left(\frac{\pi}{2}u^2\right) du, \tag{1.9}$$

$$\mathcal{FS}(s') = \int_0^{s'} sin\left(\frac{\pi}{2}u^2\right)du. \tag{1.10}$$

It is notable that in Eqs. (1.7) and (1.8), a clothoid is defined only by its sharpness through the length. This formulation is represented by other cltohoid parameters as described in next subsection. To compute Eqs. (1.9) and (1.10), it requires numerical integration and thus, there have been many assumed methods as given in [Kimia 03, Montes 07, Wilde 09, Brambley 11, Brezak 13]. However, in this work, any assumed method is not considered to ensure the accuracies of the clothoid representation.

When a clothoid is generated from Eqs.(1.3) to (1.6) to have the coordinates Eqs.(1.7) and (1.8), the parameter values at the end point have following relations with each other.

$$\kappa = \alpha s = \frac{2\delta}{s} = \sqrt{2\delta\alpha}, \tag{1.11}$$

$$\delta = \frac{\kappa s}{2} = \frac{\alpha s^2}{2} = \frac{\kappa^2}{2\alpha}, \tag{1.12}$$

$$s = \frac{\kappa}{\alpha} = \frac{2\delta}{\kappa} = \sqrt{\frac{2\delta}{\alpha}}, \tag{1.13}$$

where α, δ, and κ are the values at the end points of the clothoid, and resulting in

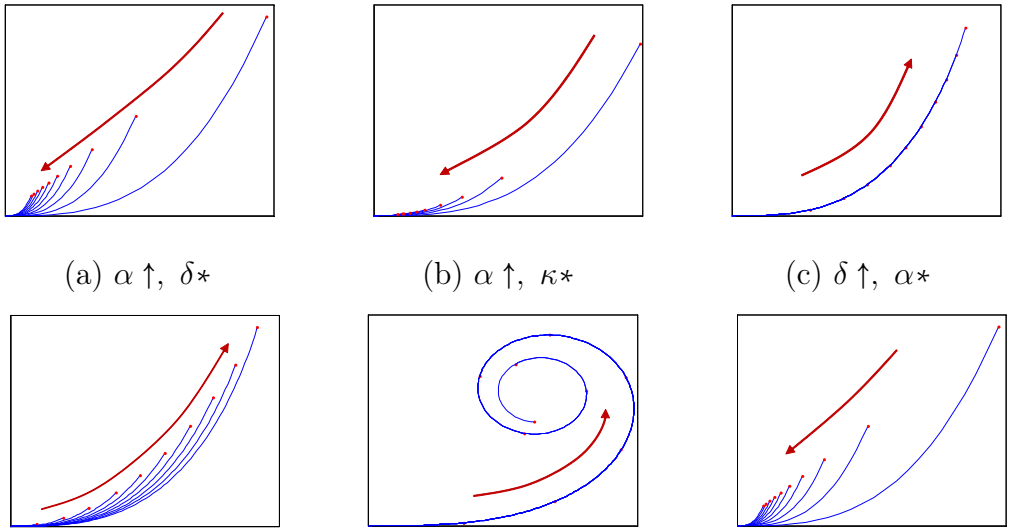
$$\kappa = \sqrt{2\delta\alpha}\ (or\ \delta = \frac{\kappa^2}{2\alpha}),\ s = \sqrt{\frac{2\delta}{\alpha}}. \tag{1.14}$$

The parameter δ is the amount of orientation change between both ends (*deflection*, > 0), also described in [Labakhua 08, Wilde 09].

In fact, the exact *Cartesian* position at the clothoid end is difficult to determine without full integration procedures or any approximation functions, however the pattern for the clothoid generation can be derived from the above parametric relations and utilized to formulate algorithmic procedure. As the parameter varies (sharpness α, deflection δ and curvature κ), the shape and the end point of the clothoid also varies, where the following patterns are observed as shown in Fig. 1.3.

From the observed patterns in Fig. 1.2, the geometric properties for clothoids are summarized as follows.

Property 1 [Clothoid geometric property on parameter variation] *Among the three clothoid parameters (α, δ, and κ),*



(a) $\alpha \uparrow,\ \delta*$ (b) $\alpha \uparrow,\ \kappa*$ (c) $\delta \uparrow,\ \alpha*$

(d) $\delta \uparrow,\ \kappa*$ (e) $\kappa \uparrow,\ \alpha*$ (f) $\kappa \uparrow,\ \delta*$

Figure 1.2: Basic properties on a clothoid parameters variation

i. As α increases with another parameter constant, the clothoid shrinks (cf. Figs. 1.2(a) and (b)).

ii. As δ increases with another parameter constant, the clothoid expands (cf. Figs. 1.2(c) and (d)).

iii. As κ increases with δ constant, the clothoid shrinks (cf. Fig. 1.2(f)).

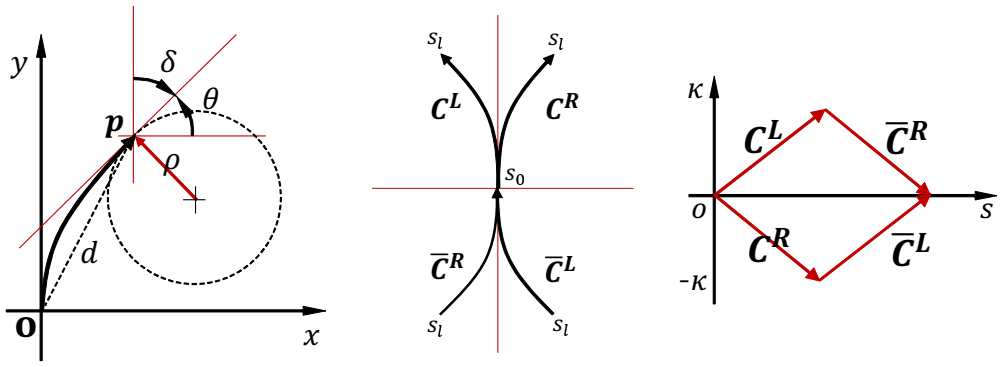
Using the properties given in **Property 1**[2], multiple clothoids are generated and composed by parametric regulation. Before entering into the problem definition and its solution, it is require to mention about some conventions and notations.

Property 2 [Notation for curvature positivity] *A clothoid is generated so that its length s increases from zero s_0 through its length s_l, i.e., as $s \rightarrow [s_0,\ s_l]$, $\boldsymbol{C} \rightarrow [\boldsymbol{C}(s_0),\ \boldsymbol{C}(s_l)]$. The $\kappa(s)$ is positive when the vehicle steering angle is in the left hand side from its center and negative for the right hand side. Also, $\alpha(s)$ is positive when the vehicle turns counter-clockwise, as $\boldsymbol{C^L}$, and negative for clockwise rotation as $\boldsymbol{C^R}$. The $\theta(s)$ is obtained by integrating curvature by clockwise direction on the Cartesian coordinate, and the δ is calculated by the*

[2]This property can be analytically derived as given in **Appendix B.1**.

orientation changes from initial to final pose, thus $\delta > 0$ *for* $s \to [s_0,\ s_l]$. *These geometric patterns could be obtained by mathematical analogy, and described in* **Appendix B.1**.

Property 3 [Clothoids composition] *To compose* $\boldsymbol{C_1}$ *and* $\boldsymbol{C_2}$ *with geometric continuities (orientation* G^1 *and curvature* G^2*), the operator* $\oplus$ *is used (i.e.* $\boldsymbol{C_1 \oplus C_2}$*). If one is* $\boldsymbol{C^R = C_1}$, *then the other is* $\boldsymbol{C_2 = \bar{C}^L}$, *or vice versa., where the superscript* **R**, **L** *stand for right, left turning respectively, and* $\bar{\boldsymbol{C}}$ *presents the reverse shape of* $\boldsymbol{C}$ *which is generated from the final to the initial by* $s \to [s_l,\ s_0]$.



(a) Parameters notation (b) Path shape (c) Curvature shape

Figure 1.3: Notation and convention for an elementary clothoid generation

Here, *elementary* clothoid is defined that its initial curvature is *zero* and its deflection is below $\frac{\pi}{2}$. [3] Figure 1.3 describes about parameters notations and its shape convention for an elementary clothoid.

In Fig. 1.3(a), the clothoid is generated from **o** to $\boldsymbol{p}$ point (the shortest distance d) by **Algorithm 21** in **Appendix B.2**, where orientation θ and deflection δ (cf. orientation difference between initial and final positions) are denoted. The radius of curvature ρ is shown with a tangential circle at $\boldsymbol{p}$, which presents the curvature of $\frac{1}{\rho}$. Figures. 1.3(b) and (c) depict the clothoid path and shape conventions as noted in **Property 2** and **3**, respectively. The arrow of (b) indicates the path generating direction, and each clothoid is generated from s_0 to s_l. The corresponding curvature can be matched in the curvature diagram shown in Fig. 1.3(c), where a clothoid is generated, then the next clothoid should be its reverse shape like $\boldsymbol{C_1^R \bar{C}_2^L}$ or $\boldsymbol{C_1^L \bar{C}_2^R}$.

[3] whereas, the *non-elementary* clothoid has non-zero initial curvature with the deflection below $\frac{\pi}{2}$.

From the properties mentioned above, two clothoids are composed as follows. For the first clothoid $\boldsymbol{C_1}$, the sharpness α_1 and the deflection δ_1 are assumed, and then the κ_1 is also determined by Eq.(1.14). To satisfy the curvature continuity (G^2), the second clothoid $\boldsymbol{C_2}$ has the curvature $\kappa_2 = \kappa_1$ at its end, and then the deflection δ_2 is determined from the orientation continuity (G^1). For shape conformity, if $\boldsymbol{C_1}$ has right turnining shape as $\boldsymbol{C_1^R}$, then $\boldsymbol{C_2}$ conforms to right or left according to the configuration. With known parameters of δ_2 and κ_2, the remaining parameter α_2 is also determined from Eq. (1.11) or (1.13), thus $\boldsymbol{C_2}$ is composed with $\boldsymbol{C_1}$ while fulfilling the geometric continuity constraints.

In the procedure described above, it is notable that two parameters such as α_1 or δ_1 require to be assumed before the composition. However, for the symmetric clothoids, the parameter to be assumed could be reduced to only one variable since another constraint is given as $\delta_1 = \delta_2$. Therefore, the problem of symmetric two clothoids is simpler in that it requires only one parameter variation to find the solution satisfying the configuration. The parametric variation is depicted for the two clothoids in the following figure.

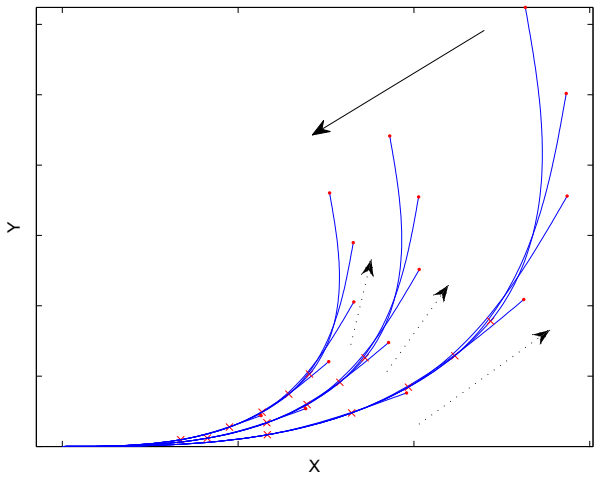


Figure 1.4: Variation of composed two clothoids by increasing $\alpha_{1,2}(\rightarrow)$ and increasing $\delta_{1,2}(\dashrightarrow)$

In Fig. 1.4, parametric variation is performed on its two different sharpness with deflection. The first clothoid $\boldsymbol{C_1}$, where its end is marked by $\times$ and the second clothoid $\boldsymbol{C_2}$ is linked at the connection point. The line arrow ($\rightarrow$) indicates that the shape with pose varies according to its sharpnesses α_1, α_2 with

different changes and the dashed arrow (⇢) explains on the variation of same deflections $\delta_1 = \delta_2$. Note that even in the two clothoids, the composed one shrinks according to increasing sharpness as **Property 1** and it also follows **Property 2** by expanding its shape from increased deflection. The properties and constraints described above are utilized to adjust clothoid parameters in each iterative loop to converge to the solution of each defined problem (cf. **Chapter 2**).

Another important property is the geometric convergence by parameter variation in the iterative algorithm. For each problem or subcase which is addressed in **Chapter 2**, geometric composition is performed using one or multiple number of elementary clothoid and different convergence criteria is applied into the iterative algorithmic loop. Among them, the convergence criteria for two clothoids composition with zero curvatures at both ends (cf. **Case 1A** in **Section 2.1.1**) is described here since this case is the one of the most fundamental logics in all the proposed algorithms. Additionally, the basic convergence criteria for an elementary clothoid is described in **Appendix B.4**.

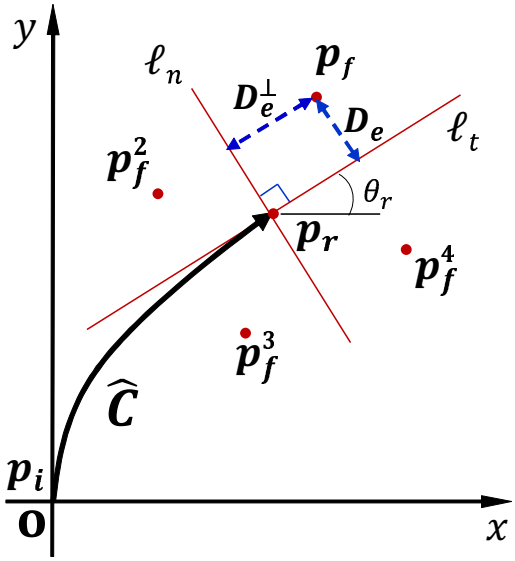


Figure 1.5: Geometric representation for convergence criteria

Figure 1.5 depicts a geometric representation of the convergence into the solution by clothoid parameter variation for the given problem. A composed clothoids $\widehat{\boldsymbol{C}}$ by $\boldsymbol{C_1^R} \oplus \bar{\boldsymbol{C}}_{\boldsymbol{2}}^{\boldsymbol{L}}$ is generated to have its end point, (a reference point) $\boldsymbol{p_r}$ under the boundary condition of $\kappa_r = 0$ and θ_r. The objective of the algorithmic iteration is that $\widehat{\boldsymbol{C}}$ connects $\boldsymbol{p_i}$ to $\boldsymbol{p_f}$ and, thus to design the parameter variation rule

for the point of $\boldsymbol{p_r}$ to reach a target position, $\boldsymbol{p_f}$. Two reference lines ℓ_t, ℓ_n are drawn from $\boldsymbol{p_r}$ by its tangential and normal direction respectively. When $\boldsymbol{p_f}$ is the target position, the convergence criteria is formulated as,

$$|\boldsymbol{D_e}| < \varepsilon,\ \boldsymbol{D_e^{\perp}} > 0, \tag{1.15}$$

where $\boldsymbol{D_e}$ and $\boldsymbol{D_e^{\perp}}$ are the minimum distance from $\boldsymbol{p_f}$ to ℓ_t and ℓ_n respectively, and ℓ_t has the slope of $\tan\theta_r$. The criteria given in Eq. (1.15) indicates that the convergence is attained when the absolute distance error $|\boldsymbol{D_e}|$ is less than a designed threshold value ε while holding a positive value for $\boldsymbol{D_e^{\perp}}$. Here, the condition $\boldsymbol{D_e^{\perp}} \geq 0$ allows the inclusion of a line segment to complete the path solution by connecting $\boldsymbol{p_r}$ to $\boldsymbol{p_f}$ after the condition of $|\boldsymbol{D_e}| \leq \varepsilon$ is satisfied.

The parameter variation rule in each algorithmic iteration is based on the geometric relation between ℓ_t and $\boldsymbol{p_f}$, as well as the relation between ℓ_n and $\boldsymbol{p_f}$. A determinant function to guide parameter variation is defined as follows:

$$\begin{cases} \lambda(\boldsymbol{p_f}) = D_e > 0\ (<0) & : \boldsymbol{p_f} \text{ is on the upper (lower) side of } \ell_t. \\ \lambda^{\perp}(\boldsymbol{p_f}) = D_e^{\perp} > 0\ (<0) & : \boldsymbol{p_f} \text{ is on the right (left) side of } \ell_n. \end{cases} \tag{1.16}$$

When a pair of symmetric clothoids which has same clothoids parameters is considered, the determinant function $\lambda(\boldsymbol{p_f})$, which checks the geometric relation between a target point ($\boldsymbol{p_f}$) and the end point $\boldsymbol{p_r}$, [4] can be used to change clothoid parameter by using one of the following rule.

Property 4 *Parameter variation rule for two clothoids problem*

a. If $\lambda(\boldsymbol{p_f}) > 0\ (<0)$, *then* $\boldsymbol{\alpha}$ *should decrease (increase) with constant* $\boldsymbol{\kappa}$,

b. If $\lambda^{\perp}(\boldsymbol{p_f}) > 0\ (<0)$, *then* $\boldsymbol{\delta}$ *should increase (decrease) with constant* $\boldsymbol{\kappa}$,

c. If $\lambda(\boldsymbol{p_f}) > 0\ (<0)$, *then* $\boldsymbol{\alpha}$ *should decrease (increase) with constant* $\boldsymbol{\delta}$.

In Fig. 1.5, target positions in other three quadrants are denoted from $\boldsymbol{p_f^1}$ to $\boldsymbol{p_f^3}$ in counter-clockwise. Based on **Property 4**, both $\boldsymbol{p_f}$ and $\boldsymbol{p_f^2}$ hold the condition of $\boldsymbol{D_e} > 0$, thus α should decrease, whereas $\boldsymbol{p_f^3}$ and $\boldsymbol{p_f^4}$ are under the conditions of $\boldsymbol{D_e} < 0$, thus α should increase. For $\boldsymbol{p_f^2}$ and $\boldsymbol{p_f^3}$, $\boldsymbol{D_e^{\perp}} < 0$, δ should decrease. For convergence into the algorithmic solution, two iterative variation for α and δ could be regulated until the convergence criteria is satisfied within

[4] $\lambda(\boldsymbol{p_f}) = -\tan(\theta_r)x_f + \tan(\theta_r)x_r + y_f - y_r$, where $\boldsymbol{p_r}(x_r, y_r)$ and $\boldsymbol{p_f}(x_f, y_f)$.

an error bound. According to the given target position, it is not possible to reach by composed clothoids. Such case can be analyzed also by the empirical study and finding regular patterns for the generated clothoids which are mentioned in **Appendix B.4**.

1.4 Conclusion

Before entering into **pCCP** problem, some important properties about clothoid parameters and empirical studies on the parametric variations have been performed. The clothoid inherits curvature continuity as well as orientation and position continuities, therefore it efficiently represents the path driven by nonholonomic vehicle. From the differential equations of the nonholonomic vehicle motion, the clothoid equations are related to design algorithmic procedure. To formulate the algorithm that each iteration gets converged into **pCCP** solution, some basic properties and convergence criteria were introduced and derived. Even though a simple form of clothoid which has 1^{st} order of curvature equation or a constant sharpness (α), is utilized to generate the path, it is assumed that the nonholonomic path could be represented better than other path generation methods without high order of algorithmic complexity such as spline or polynomials. In the following chapters, **pCCP** problems using one or multiple number of clothoids with composition method are addressed in more details and investigated to apply the proposed paths to various road environments and obstacle avoidance maneuvers.

Chapter 2

Local pCCP generation for static target

2.1 Introduction

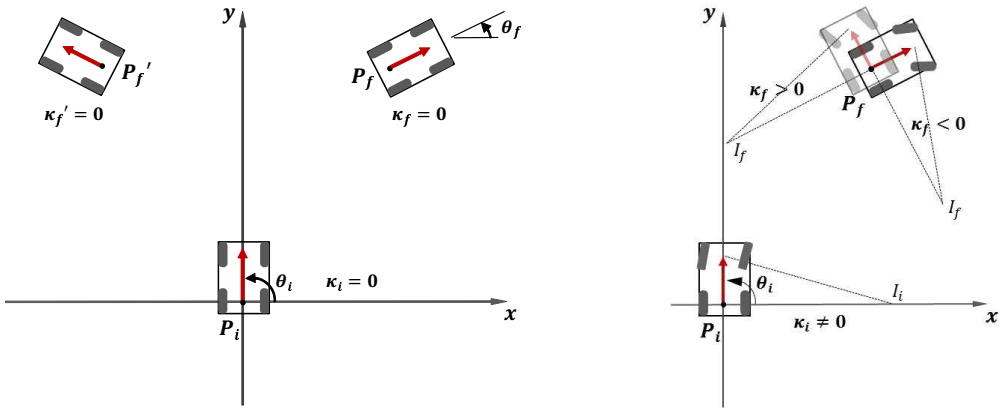
This chapter addresses a general **pCCP** problem and solution [1] on local path generation for nonholonomic car-like vehicle, where the problem is defined by two boundary configurations at both ends. The solution path is comprised of parameter-adjusted multiple clothoids. There exists a lot of literature that deals with *Continuous Curvature Path* (**CCP**); however there has been only few works that provide a general solution for various configurations. In subsequent sections, the problem is categorized into three sub problems and each sub problem is also divided into several subcases to deal with the algorithmic procedure differently according to the condition such that how many number of clothoids are used or how to compose the clothoids under geometric continuity constraints. The proposed algorithmic solutions for parametric (or parameter-adjusted) **CCP** (**pCCP**) could be useful for generating a local path in obstacle avoidance maneuvers or smoothing/reconstructing a given raw noisy path.

[1] The solution covers various boundary configurations as much as possible by general iterative procedures.

2.2 Problems definition and clothoids composition

This section deals with the problem definition of parametric Continuous Curvature Path (**pCCP**) and describes clothoids composition method for each problem. The vehicle is assumed to have only forward motion[2], and the problem is defined by two boundary configurations which is applicable to local path planning scheme. (i.e., only initial and final pose and curvature are given, and other way point or way pose are not fixed.) For **pCCP** generation, multiple clothoids with line or arc are composed to satisfy the boundary configuration. The obtained path should be continuous in its geomety as well as curvature through overall length.

Three basic problems will be shown here. Each problem is defined by initial and final *configuration* (pose and curvature) noted by $\boldsymbol{P_i}$ and $\boldsymbol{P_f}$ where index i, f denotes the initial or final configuration, respectively. Each configuration includes four variables (or four tuples) of (x, y, θ, κ).



(a) Line-to-line problem (b) Arc-to-arc problem

Figure 2.1: Problem descriptions by boundary configurations

Figure 2.1 shows the representative cases according to the curvature at both ends configurations. The first, *line-to-line* problem is named from the condition that both boundary curvatures are equal to zeros, i.e. $\kappa_{i,f} = 0$, as shown in Fig. 2.1(a). Both configurations $\boldsymbol{P_i}$ and $\boldsymbol{P_f}$ indicate that a vehicle starts from the position of (x_i, y_i) with orientation θ_i and curvature κ_i and arrives at (x_f, y_f) with θ_f and κ_f, where θ_i is fixed to $\frac{\pi}{2}$ to avoid redundancy for the opposite side

[2]Different to [Reeds 90, Fraichard 04], the proposed path do not consider the backward motion, since the vehicle should stop to change its motion from forward to backward, or vice versa.

case $\boldsymbol{P_f'}$ which is located in a symmetric configuration to the x axis (cf. Fig. 2.1(a))). For example, if the solution to reach $\boldsymbol{P_f}$ is $\boldsymbol{C^R}$, then other solution to reach $\boldsymbol{P_f'}$ will be $\boldsymbol{C^L}$.

Another problem is shown in Fig. 2.1(b) where the curvatures at both boundaries are non-zeroes with I_f which corresponds to *ICR* (Instantaneous Center of Rotation); hence, this problem is named as *arc-to-arc* problem from the condition $\kappa_{i,f} \neq 0$. This problem involves two different cases of $\kappa_f > 0$ and $\kappa_f < 0$ (cf. **Section 2.3**). Indeed, the *arc-to-arc* problem with different final configuration ($\kappa_f > 0$ or < 0) cannot be addressed with symmetric reasoning like the first *line-to-line* problem since $\kappa_i \neq 0$. Furthermore, there is another case in which one end has zero curvature, and the other has non-zero curvature. This case, $\kappa_i = 0, \kappa_f \neq 0$ or vice versa, is named as *line-to-arc* problem. For the opposite condition of $\kappa_i \neq 0, \kappa_f = 0$, it is redundant so as to be replaced by the symmetric solution of the regular problem, i.e., $\kappa_i = 0, \kappa_f \neq 0$.

To sum up, the **pCCP** generation problem using clothoids can be decomposed into three different problems, according to initial and final configurations as shown in Table 2.1.

Table 2.1: Problems decomposition

Problem type	$\boldsymbol{P_i}(x_i, y_i, \theta_i, \kappa_i)$	$\boldsymbol{P_f}(x_f, y_f, \theta_f, \kappa_f)$
Problem 1	$\kappa_i = 0$	$\kappa_f = 0$
Problem 2	$\kappa_i = 0$	$\kappa_f \neq 0$
Problem 3	$\kappa_i \neq 0$	$\kappa_f \neq 0$

In Table 2.1, the problem type is determined from boundary configurations. The initial configuration corresponds to the vehicle's state in the local coordinate at which the local path generation begins with the vehicle's motion center (cf. Fig. 1.1) oriented toward the positive *y-axis*. The final configuration is specified on the first quadrant plane, i.e., $x > 0, y > 0$ while maintaining its general solvability by symmetric compatibility explained above.

Before beginning the specific analogy, it requires to define a general problem for the clothoid-based **pCCP**.

Problem (pCCP) *From* $\boldsymbol{P_i}(x_i, y_i, \theta_i, \kappa_i)$ *to* $\boldsymbol{P_f}(x_f, y_f, \theta_f, \kappa_f)$*, find a path composed of the minimum number of clothoids that satisfies both configurations with geometric and curvature continuities along the length.*

The objective for the **pCCP** problem is to find the clothoid-based path that satisfies both configurations with the possibility of including lines and circular arcs (the conformity to G^1 and G^2 constraints [3] must be verified/respected. For **pCCP** generation, multiple clothoids are composed to satisfy the boundary configuration. The obtained path should be continuous in overall curvature.

2.2.1 Problem 1: $\boldsymbol{P_i}\left(x_i, y_i, \frac{\pi}{2}, \boldsymbol{\kappa_i = 0}\right)$ to $\boldsymbol{P_f}\left(x_f, y_f, \theta_f, \boldsymbol{\kappa_f = 0}\right)$

The **pCCP** problem for *line-to-line* condition is addressed with two subcases. The reasoning for the decomposition is described as follows. In this problem, the minimum number of clothoids is two, however there is other case that it is not possible to satisfy the configurations using only two clothoids.

Let it be focused on the geometric condition given in Fig. 2.2. When ϕ is defined as the angle between $\boldsymbol{p_i}$ and $\boldsymbol{p_f}$, then according to the condition between θ_f and the reference line ℓ_r (cf. Fig. 2.2(a) or Fig. 2.3(a)) (i.e., whether $\theta_f > \phi$ or $\theta_f < \phi$), the minimum number of clothoids satisfying both configurations is either two or four. Hence, **Problem 1** should be decomposed into two subcases **1A, 1B** in the following two subsections.

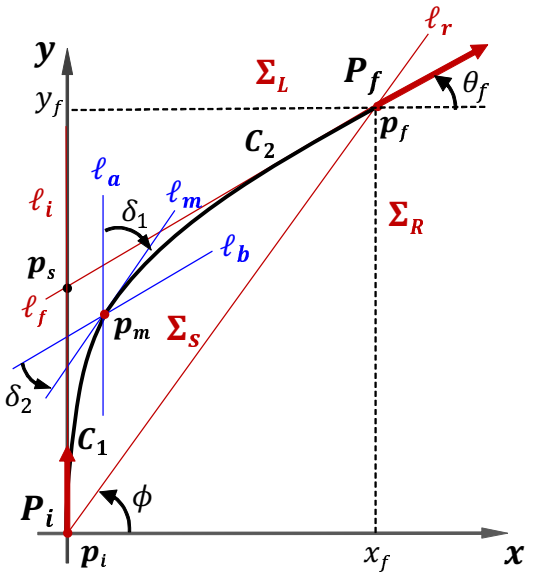
2.2.1.1 Case 1A ($\theta_f < \phi$)

In this case, two clothoids are sufficient to construct the feasible path. A clothoid $\boldsymbol{C_1}$ with the shape $\boldsymbol{C^R}$, and the other clothoid $\boldsymbol{C_2}$ with $\boldsymbol{C^L}$ (cf. **Property 2**), compose $\widehat{\boldsymbol{C}}$ by $\boldsymbol{C_1^R} \oplus \bar{\boldsymbol{C}}_{\boldsymbol{2}}^{\boldsymbol{L}}$ while holding G^1 and G^2.

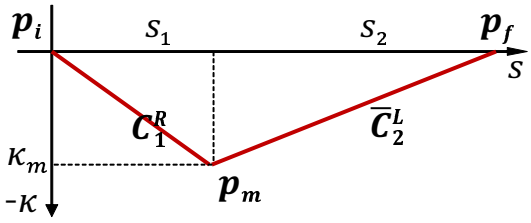
Figure 2.2 depicts the case $\theta_f < \phi$. In Fig. 2.2(a), two extension lines from both configurations are denoted as ℓ_i, ℓ_f, respectively and ℓ_r is drawn by connecting $\boldsymbol{P_i}$ with $\boldsymbol{P_f}$. When three intersection points $\boldsymbol{p_i}$, $\boldsymbol{p_f}$, and $\boldsymbol{p_s}$ are defined by ℓ_i, ℓ_f, and ℓ_r respectively ($\boldsymbol{p_s}$ by ℓ_i and ℓ_f), then the areas $\boldsymbol{\Sigma_R}$, $\boldsymbol{\Sigma_L}$ are the half-space to the right and the left of ℓ_r respectively, and $\boldsymbol{\Sigma_S}$ is the area inside the polygon $\triangle \boldsymbol{p_i p_f p_s}$. Thus for this case, all the points in the composed clothoids are inside $\boldsymbol{\Sigma_S}$. Note that if $\theta_f = \phi$, then no clothoid pairs satisfy both configurations except a straight line since $\ell_f = \ell_r$.

Under the defined geometric representation, $\boldsymbol{C_1}$ starts from $\boldsymbol{p_i}$ to $\boldsymbol{p_m}$ (*meeting* point), while $\boldsymbol{C_2}$ starts from $\boldsymbol{p_f}$ to $\boldsymbol{p_m}$. The shape of the composed path becomes $\boldsymbol{C_1^R} \oplus \bar{\boldsymbol{C}}_{\boldsymbol{2}}^{\boldsymbol{L}}$, where $s_1 \rightarrow [s_0, s_m]$ in $\boldsymbol{C_1}$, $s_2 \rightarrow [s_m, s_f]$ in $\boldsymbol{C_2}$ from **Property 3** and Fig. 2.2(b). Note that, since $\boldsymbol{P_f}$ heads toward $\boldsymbol{\Sigma_R}$, $\boldsymbol{C_2}$ is located inside $\boldsymbol{\Sigma_L}$, and $\boldsymbol{p_m}$ also resides in $\boldsymbol{\Sigma_S}$. The curvature of $\boldsymbol{C_1}$ grows from $\boldsymbol{p_i} = 0$ to $\boldsymbol{p_m}$ with a

[3] 1^{st}(tangential) and 2^{nd}(curvature) geometric continuity



(a) Geometric representation



(b) Curvature diagram

Figure 2.2: **Case 1A** ($C_1^R \bar{C}_2^L$)

constant sharpness, as determined in Eq.(1.4), and the curvature of $\boldsymbol{C_2}$ decreases from $\boldsymbol{p_m}$ to $\boldsymbol{p_f}$. The deflections for $\boldsymbol{C_1}$ and $\boldsymbol{C_2}$ reach δ_1 and δ_2, respectively, until $\boldsymbol{p_m}$. At this point $\boldsymbol{p_m}$, there exists an important geometric constraint about G^1 constraint.

$$\delta_1 + \delta_2 = \theta_i - \theta_f. \tag{2.1}$$

Figure 2.2(b) depicts the corresponding curvature diagram where $\boldsymbol{C_1}$ increases its curvature to the maximum $\boldsymbol{\kappa_m}$ at $\boldsymbol{p_m}$ with G^2 continuity constraint and $\boldsymbol{C_2}$ decreases to zero at $\boldsymbol{p_f}$. About Eq.(2.1), Fig. 2.2(b) informs us that δ_1 is the area of the triangle for $\boldsymbol{C_1^R}$ and δ_2 is the area of the triangle for $\bar{\boldsymbol{C}}_{\boldsymbol{2}}^{\boldsymbol{L}}$, thus the sum of these two areas equals the total deflection $\theta_i - \theta_f$.

In **Algorithm 1**, initial values of α_1 and δ_1 for $\boldsymbol{C_1}$ are assumed before entering the loop with required pre-setting values. From the lines [3-6], the parameters to compose two clothoids are determined by geometric constraints. The parameters are adjusted from the lines [16-23] (cf. **Property 4**). The parameter variation rules of lines [16-18] and lines [19-21] are same as the *bisection method* [Kiusalaas 07].[4]

To obtain solution $\boldsymbol{\alpha_{sol}}, \boldsymbol{\delta_{sol}}, \widehat{\boldsymbol{C}}$, the stopping criteria are defined in line 9 and line 10, where the ε should be larger than the sampling distance of *Fresnel* integration for bisection convergence, e.g. $\varepsilon > 10^{-3}[m]$.

2.2.1.2 Case 1B ($\theta_f > \phi$)

The condition used to decompose the **Problem 1** into two subcases is whether $\theta_f > \phi$ or not, (If $\theta_f = \phi$, only a line solution is feasible.) and this condition is determined from the geometric relation betweem $\boldsymbol{P_i}$ and $\boldsymbol{P_f}$.

Figure 2.3 depicts **Case 1B**, where the only difference from **Case 1A** is the orientation of $\boldsymbol{P_f}$, which is directed toward the area $\boldsymbol{\Sigma_L}$ (i.e., $\theta_f > \phi$). For this case, it is not possible to generate **pCCP** using only two clothoids; indeed, additional clothoids are required.

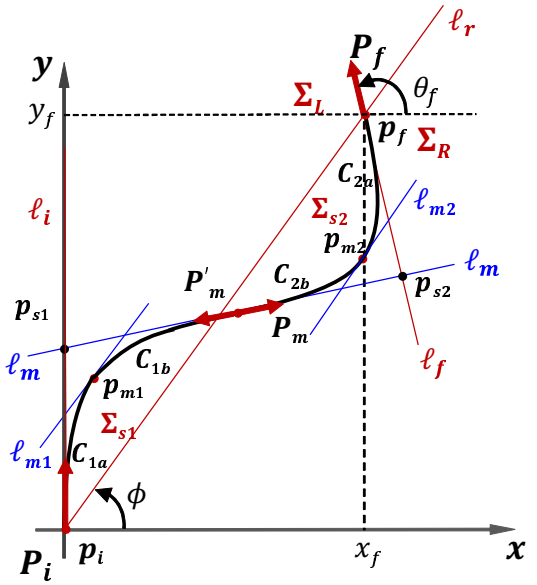
This condition is proved using the geometric representation as follows. In Fig. 2.3(a), four clothoids are generated from $\boldsymbol{p_i}$ to $\boldsymbol{p_f}$ as $\boldsymbol{C_{1a}}$, $\boldsymbol{C_{1b}}$, $\boldsymbol{C_{2b}}$, and $\boldsymbol{C_{2a}}$. In the same way as for **Case 1A**, the clothoid pair $\widehat{\boldsymbol{C_1}} = \boldsymbol{C_{1a}^R} \oplus \bar{\boldsymbol{C}}_{\boldsymbol{1b}}^{\boldsymbol{L}}$, and the other pair $\widehat{\boldsymbol{C_2}} = \bar{\boldsymbol{C}}_{\boldsymbol{2a}}^{\boldsymbol{R}} \oplus \boldsymbol{C_{2b}^L}$. Each pair has the connection point $\boldsymbol{p_{m1}}$, $\boldsymbol{p_{m2}}$, and it also has common tangential line ℓ_{m1}, ℓ_{m2}, where the lines provide the reference for both clothoids deflections.

[4]The bisection method guarantees the convergence.

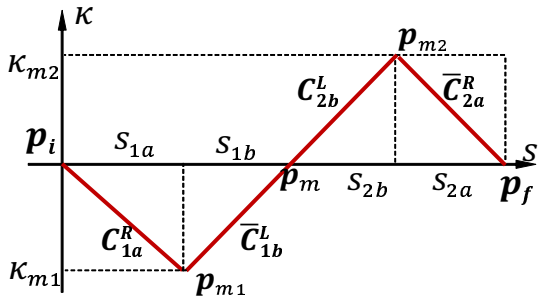
Algorithm 1 Case 1A: Two clothoids composition

Require: $\varepsilon, d\alpha, d\delta, \text{sol} = \text{FALSE}$ ▷ Pre-setting values for iterative convergence
Require: α_1, δ_1 ▷ Initial assumption

procedure CLOTHOID2LL($\boldsymbol{P_i}, \boldsymbol{P_f}$)
  while sol == FALSE **do**
    $\kappa_1 \leftarrow \alpha_1, \delta_1$ ▷ C_1 generation
    $\kappa_2 = \kappa_1$ ▷ G^2 continuity
    $\delta_2 \leftarrow \delta_1, \theta_f$ ▷ G^1 continuity
    $\alpha_2 \leftarrow \kappa_2, \delta_2$ ▷ C_2 generation
    $\widehat{C} \leftarrow C_1 \oplus C_2$ ▷ Two clothoids composition
    $D_e \leftarrow \ell_f, \widehat{C},\ D_e^{\perp} \leftarrow \ell_f^{\perp}, \widehat{C}$ ▷ Distance error parameters
    if $|D_e| < \varepsilon$ **then**
      if $D_e^{\perp} \geq 0$ **then**
        sol = TRUE ▷ Convergence achieved
        return $\boldsymbol{\alpha_{sol}, \delta_{sol}, \widehat{C}}$ ▷ Obtained solution
      end if
    end if
    $\lambda \leftarrow D_e, \lambda^{\perp} \leftarrow D_e^{\perp}$
    if $\lambda \cdot \lambda' < 0$ **then** ▷ Bisection method for α
      $d\alpha = \frac{d\alpha}{2}$
    end if
    if $\lambda^{\perp} \cdot \lambda'^{\perp} < 0$ **then** ▷ Bisection method for δ
      $d\delta = \frac{d\delta}{2}$
    end if
    $d\alpha = |d\alpha| \cdot \text{sign}(\lambda),\ d\delta = |d\delta| \cdot \text{sign}(\lambda^{\perp})$ ▷ α, δ variation criteria
    $\alpha_1 = (\alpha_1 + d\alpha),\ \delta_1 = (\delta_1 + d\delta)$
    $\lambda' = \lambda,\ \lambda'^{\perp} = \lambda^{\perp}$ ▷ For next iteration
  end while
end procedure



(a) Geometric representation



(b) Curvature diagram

Figure 2.3: **Case 1B** ($C^R_{1a}\bar{C}^L_{1b}C^L_{2b}\bar{C}^R_{2a}$)

Under the described notations, let us assume two clothoids which connecting boundary configurations as following. If $\boldsymbol{C_{1a}}$ meets $\bar{\boldsymbol{C}}_{\boldsymbol{2a}}$, the curvature signs must be opposite to each other. If $\bar{\boldsymbol{C}}_{\boldsymbol{2a}}$ is to be the same as $\boldsymbol{C_{1a}}$, $\bar{\boldsymbol{C}}_{\boldsymbol{2a}}$ should be located outside of $\boldsymbol{\Sigma_{s2}}$ (right side of ℓ_f), which could not satisfy the orientation continuity. This result is clearly seen from the corresponding curvature diagram in Fig. 2.3(b), $\boldsymbol{p_{m1}}$ can not meet $\boldsymbol{p_{m2}}$.

In Fig. 2.3(b), the curvature at the first pair $\widehat{\boldsymbol{C_1}}$ produces two curvature lines of $\boldsymbol{C_{1a}}$, $\boldsymbol{C_{1b}}$, and that for the second pair $\widehat{\boldsymbol{C_2}}$ is $\boldsymbol{C_{2b}}$ and $\boldsymbol{C_{2a}}$. All curvature lines are continuously connected along the travel length, where $\widehat{\boldsymbol{C_1}}$ retains in negative curvature and $\widehat{\boldsymbol{C_2}}$ remains in the positive.

The four clothoids composition in **Case 1B** are achieved using two consecutive pairs of **Case 1A** compositions. Hence, it is important to determine a common boundary condition at the intersection point of both composition pairs. To tackle this problem, a reference line ℓ_m is initially fixed at the end of the first clothoid pair $\widehat{\boldsymbol{C_1}}$ with slope $\tan\theta_m$ (θ_m is the slope angle of ℓ_m). The line ℓ_m gives an orientation constraint for both $\widehat{\boldsymbol{C_1}}$ and $\widehat{\boldsymbol{C_2}}$. If the common orientation constraint of $\tan\theta_m$ is given, two pairs of clothoids can each be solved as two clothoid problems, according to the resolved **Case 1A**.

In **Algorithm 2**, two procedures are performed according to convergence criteria (lines [10-11]) using **Property 4**. Note that this subcase can address two separated **Case 1A** constrained by a reference line ℓ_m which passes through $\boldsymbol{p_m}$ with the slope of $\tan\theta_m$.

Furthermore, $\boldsymbol{p_m}$ and $\boldsymbol{p_m'}$ are result of the procedure **clothoid2LL** in line [6] by given parameters and $\boldsymbol{p_m'}$ should be on ℓ_m with positive distance, i.e., $\boldsymbol{D_e^{\perp}} \geq 0$; This subcase has numerous solutions according to the slope of ℓ_m (cf. line 2) and the parameters of $\widehat{\boldsymbol{C}}_{\boldsymbol{1,2}}$; thus, additional constraint for the procedure is required. About this constraint, it is addressed specifically in **Section 4.1.2**.

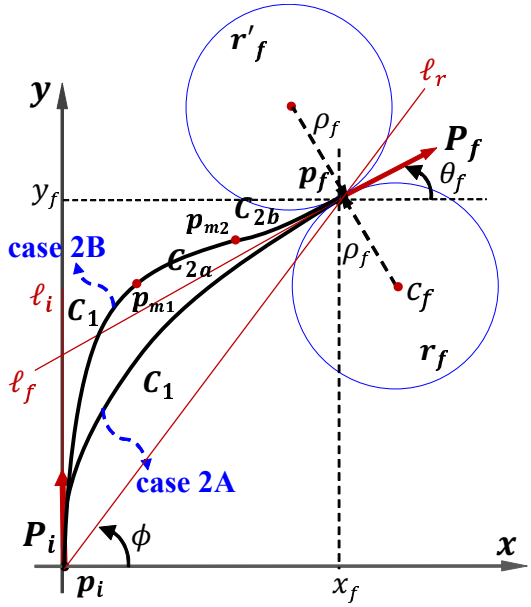
Algorithm 2 Case 1B: Four clothoids composition

Require: $\varepsilon, d\alpha, d\theta, \text{sol} = \text{FALSE}$ ▷ Pre-setting values for iterative convergence
Require: $\tan\theta_m$ ▷ Slope of ℓ_m
Require: $\alpha_1, \delta_1, \alpha_2, \delta_2$ ▷ Initial parameters assumption for C_1 and C_2

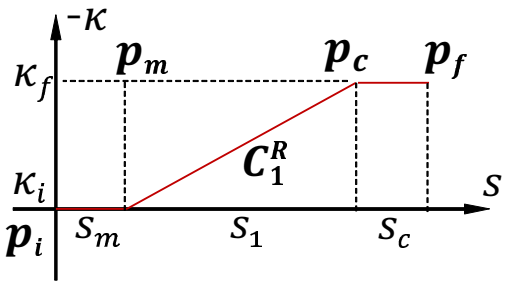
procedure CLOTHOID4LL($\boldsymbol{P_i}, \boldsymbol{P_f}$)
  for $\theta_m \leftarrow (\theta_m + d\theta)$ **do**
    while sol == FALSE **do**
      call clothoid2LL($\boldsymbol{P_i}$, $\boldsymbol{P_m}$)
      $\widehat{\boldsymbol{C_1}} \leftarrow C_{1a} \oplus C_{1b}$
      call clothoid2LL($\boldsymbol{P_f}$, $\boldsymbol{P'_m}$)
      $\widehat{\boldsymbol{C_2}} \leftarrow C_{2a} \oplus C_{2b}$
      $D_e \leftarrow \widehat{C_1}, \widehat{C_2}$
      $\lambda = D_e, \lambda^{\perp} = D_e^{\perp}$
      if $|D_e| < \varepsilon$ **then**
        if $D_e^{\perp} \geq 0$ **then**
          sol = TRUE ▷ Convergence achieved
          return $\bar{\boldsymbol{\alpha}}_{\boldsymbol{sol}}$,$\bar{\boldsymbol{\delta}}_{\boldsymbol{sol}}$, $\widehat{\boldsymbol{C}}_{\mathbf{1,2}}$ ▷ Obtained solution
        end if
      end if
      if $\lambda \cdot \lambda' < 0$ **then** ▷ Bisection method for α_1, α_2
        $d\alpha = \frac{d\alpha}{2}$
      end if
      if $\lambda^{\perp} \cdot \lambda'^{\perp} < 0$ **then** ▷ Bisection method for δ_1, δ_2
        $d\delta = \frac{d\delta}{2}$
      end if
      $d\alpha = |d\alpha| \cdot \text{sign}(\lambda)$, $d\delta = |d\delta| \cdot \text{sign}(\lambda^{\perp})$ ▷ α, δ variation criteria
      $\alpha_{1,2} = (\alpha_{1,2} + d\alpha)$, $\delta_{1,2} = (\delta_{1,2} + d\delta)$
      $\lambda' = \lambda$, $\lambda'^{\perp} = \lambda^{\perp}$ ▷ For next iteration
    end while
  end for
end procedure

2.2.2 Problem 2: $\boldsymbol{P_i}\left(x_i, y_i, \frac{\pi}{2}, \boldsymbol{\kappa_i = 0}\right)$ to $\boldsymbol{P_f}\left(x_f, y_f, \theta_f, \boldsymbol{\kappa_f \neq 0}\right)$

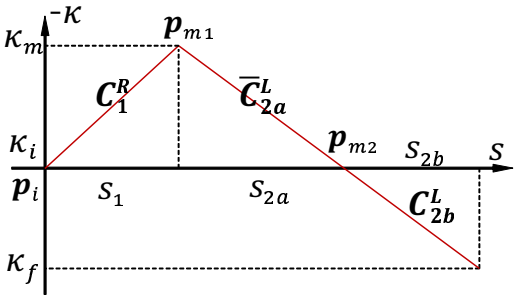
In this subsection, the *line-to-arc* problem is dealt with as $\kappa_i = 0$ and $\kappa_f \neq 0$. Its reverse formulation ($\kappa_i \neq 0$ and $\kappa_f = 0$) is symmetrically tackled by resolving the same problem. This problem is compensatory for the *line-to-line* problem, where the target position could be a meeting point between two clothoids like $\boldsymbol{p_m}$ of **Problem 1**.



(a) Geometric representation



(b) Curvature diagram (**Case 2A**)



(c) Curvature diagram (**Case 2B**)

Figure 2.4: **Problem 2** ($\boldsymbol{C_1^R}$, $\boldsymbol{C_1^R \bar{C}_{2a}^L C_{2b}^L}$)

This case is also divided into two subcases, $\kappa_f > 0$ and $\kappa_f < 0$. In Fig. 2.4(a), initial and final configurations are given as $\boldsymbol{P_i}$ and $\boldsymbol{P_f}$. Two lines ℓ_i and ℓ_f are drawn from $\boldsymbol{p_i}$ and $\boldsymbol{p_f}$, of which the orientations are $\frac{\pi}{2}$ and θ_f, respectively. In

the given configurations, an important factor to divide this problem into subcases is the sign of the final curvature κ_f. Namely, the sign of the final curvature determines the direction of the clothoid at $\boldsymbol{P_f}$. As is shown in Fig. 2.4(a), the clothoid directs downward or upward to ℓ_f depending on wheather $\kappa > 0$ or $\kappa < 0$, respectively, where the corresponding circles $\boldsymbol{r_f}$, $\boldsymbol{r_f'}$ are also symmetrically located to ℓ_f.

The case of $\kappa_f > 0$ is examined in Fig. 2.4(a) and its curvature diagram (b). A clothoid is generated so that its curvature increases; thus, the curvature positivity is consistent with κ_f through its length, i.e., $\kappa(s) > 0,\ \forall s > 0$. As shown in (b), an elementary clothoid could satisfy both configurations (an arc segment $\boldsymbol{p_f p_f'}$ is involved for further discussion in the following subsection).

For the case $\kappa_f < 0$, it is impossible to connect both ends using only one clothoid $\boldsymbol{C_1}$ since its curvature sign is inconsistent with that of κ_f. The curvature diagram of Fig. 2.4(c) more clearly shows that the first clothoid $\boldsymbol{C_1^R}$ goes to $\boldsymbol{p_{m1}}$ with κ_m, and two elementary clothoids are required to connect $\boldsymbol{p_{m1}}$ to the final position $\boldsymbol{p_f}$ with κ_f. Thus, in **Problem 2**, two subcases of **Case 2A** and **Case 2B** with $\kappa_f < 0$ and $\kappa_f > 0$ respectively, are dealt with separately.

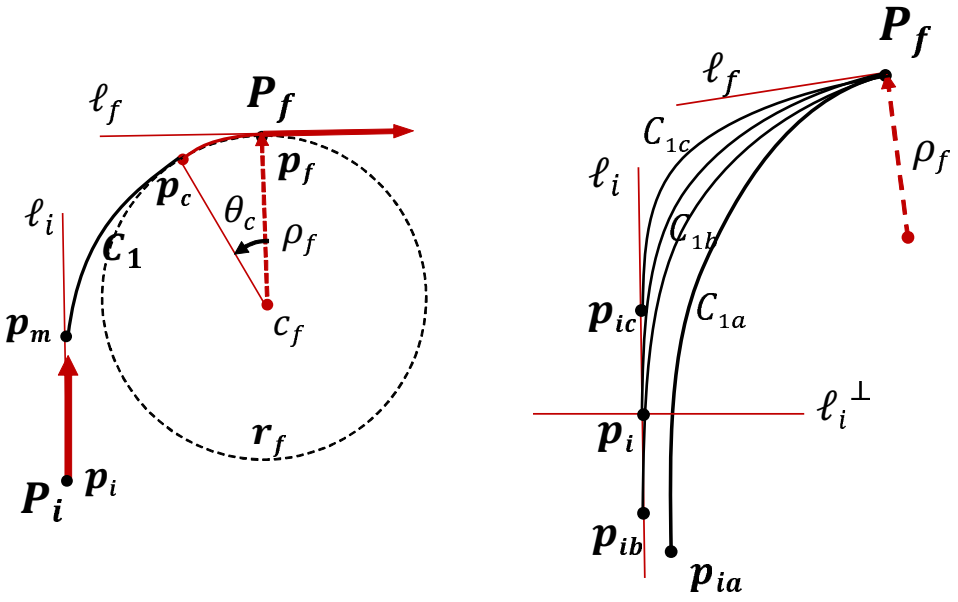
2.2.2.1 Case 2A ($\kappa_f < 0$)

In this subcase, one clothoid can satisfy both configurations. A negative κ_f is given; thus, the clothoid parameters are all defined (*frozen*).

In Fig. 2.5, a geometric representation of this subcase is depicted. At $\boldsymbol{p_f}$, the parameter δ_1 should be $\theta_i - \theta_f$ by the orientation constraint, and κ_1 is given as κ_f from configuration (cf. Eq. (1.7)); there is no room for adjusting clothoid parameter.

A possible solution it to include line or arc segment to satisfy both configurations with geometric constraint. For that purpose, a circular arc segment at $\boldsymbol{C_1}(s_l)$ and a line segment at $\boldsymbol{C_1}(s_0)$ are added, as shown in Fig. 2.5(a) of which the corresponding curvature diagram has the shape of Fig. 2.4(b). Thus the resultant path composes a line segment, $\overline{\boldsymbol{p_i p_m}}$ and a circular arc segment, $\widehat{\boldsymbol{p_c p_f}}$ which having its central angle θ_c.

A rule for this procedure is similar to **Algorithm 1** such that the $\boldsymbol{p_i}$ is adjusted to meet ℓ_i in the upper side of $\ell_i^{\perp}$, cf. Fig. 2.5(a). In Fig. 2.5(b), the procedure for adjusting arc parameter θ_c and line segment $\overline{\boldsymbol{p_i p_m}}$ is shown as the path varies from $\boldsymbol{C_{1a}}$ to $\boldsymbol{C_{1c}}$. Firstly, a fixed clothoid is is generated and shifted so that its end position $\boldsymbol{C_1}(s_\ell)$ meets $\boldsymbol{p_f}$. If the position $\boldsymbol{p_i}$ is located between ℓ_i and $\boldsymbol{p_f}$ (or inside ℓ_i), then θ_c increases for $\boldsymbol{C_{ia}}$ and $\boldsymbol{p_{ia}}$ to move outward. As θ_c increases, $\boldsymbol{p_{ib}}$ tends to move up so that it can converge to ℓ_i and $\ell_{ia}^{\perp}$. After convergence to



(a) Line or arc inclusion (b) Iterative procedure

Figure 2.5: Parametric regulation in **Problem 2**

ℓ_i, the conditional check is carried out to determine that $\boldsymbol{p_i}$ is in the upper or lower region of $\ell_{ia}^{\perp}$, i.e., $\boldsymbol{D_e^{\perp}} \geq 0$. When the resultant clothoid is $\boldsymbol{C_{1c}}$ with $\boldsymbol{p_{ic}}$, it could be the solution with an additional line $\boldsymbol{p_{ic}p_i}$ included. However, when $\boldsymbol{C_{ia}}$ is displaced to $\boldsymbol{p_{ib}}$, it could not be a solution since it is lower than $\ell_i^{\perp}$. Clothoid parameter adjustmant is not meaningful anymore.

In **Algorithm 3**, a clothoid is determined from the given κ_f, δ_1 by Eq. (1.7) (lines [3-5]). With the initial $\theta_c = 0$ (i.e., no arc), $\boldsymbol{C_1}$ is generated using and translated (shifted) for $\boldsymbol{C_1}(s_{l)}$ to meet $\boldsymbol{p_f}$. The parameter θ_c is adjusted with variation rule of lines [15-19] until the convergence criteria is passed in lines [8-13].

2.2.2.2 Case 2B ($\kappa_f > 0$)

This subcase is represented in Fig. 2.4(a) of applying to $\boldsymbol{r_f'}$, with the corresponding curvature diagram in (c). For this subcase, it is not possible to satisfy both configurations using only one clothoid since $\boldsymbol{C_1^R}$ cannot satisfy the final curvature $\kappa_f > 0$ (cf. Fig. 2.4(c)). Based on the final configuration of $\boldsymbol{P_f}$, clothoid $\boldsymbol{C_{2b}^L}$ should be located in the upper side region to ℓ_f, indicating that it is impossible to meet $\boldsymbol{p_i}$ without additional clothoids. Thus, in this subcase, the whole

Algorithm 3 Line-to-arc (one clothoid generation)

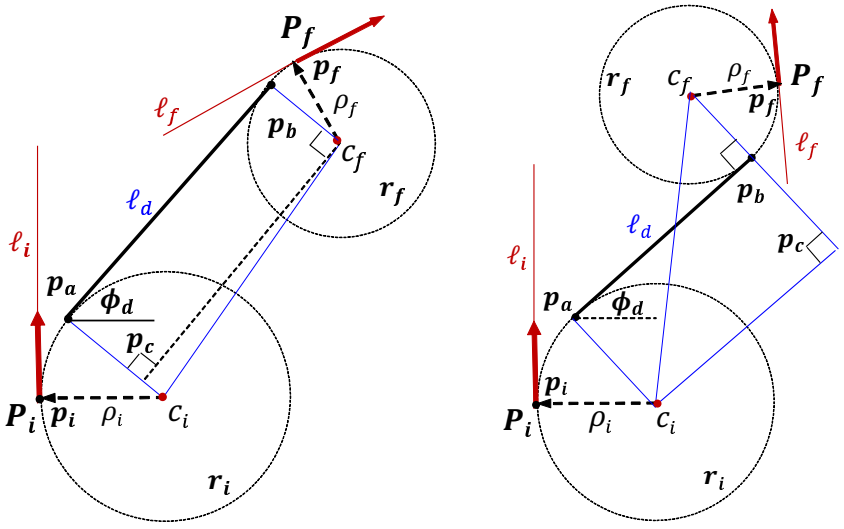
Require: $\varepsilon, d\alpha, d\theta, \text{sol} = \text{FALSE}$ ▷ Pre-setting values for iterative convergence
Require: $\theta_c = 0$ ▷ Initial parameter assumption

procedure CLOTHOID1LA($\boldsymbol{P_i}, \boldsymbol{P_f}$)
  while sol == FALSE **do**
    $\delta_1 \leftarrow \theta_i, \theta_f$ ▷ G^1 constraint
    $\kappa_1 = \kappa_f$ ▷ G^2 constraint
    $\alpha_1 \leftarrow \delta_1, \kappa_f$ ▷ C_1 generation by Eq. (1.14)
    $\widehat{\boldsymbol{C}} \leftarrow C_1 \oplus C_c$ ▷ Arc segment C_c with θ_c
    $\boldsymbol{D_e} \leftarrow \ell_i, \widehat{C}(s_0),\ \boldsymbol{D_e^\perp} \leftarrow \ell_i^\perp, \widehat{C}(s_0)$ ▷ Distance error parameters
    if $|D_e| < \varepsilon$ **then**
      if $D_e^\perp \geq 0$ **then**
        sol = TRUE ▷ Convergence achieved
        return $\widehat{\boldsymbol{C}}$ ▷ Obtained solution
      end if
    end if
    $\lambda = D_e, \lambda^\perp = D_e^\perp$
    if $\lambda\lambda' < 0$ **then** ▷ Bisection method for θ_c
      $d\theta = \frac{d\theta}{2}$
    end if
    $d\theta = -|d\theta| \cdot \text{sign}(\lambda)$
    $\theta_c = \theta_c + d\theta$
    $\lambda' = \lambda$ ▷ For next iteration
  end while ▷ Obtained solution
end procedure

path could be obtained by solving another *line-to-line* problem from $\boldsymbol{p_i}$ to $\boldsymbol{p_{m2}}$. The procedure is therefore the same as that of **Case 2A** and **Problem 1**; thus, the further description is omitted.

2.2.3 Problem 3: $\boldsymbol{P_i}\,(x_i, y_i, \theta_i, \boldsymbol{\kappa_i \neq 0})$ to $\boldsymbol{P_f}\,(x_f, y_f, \theta_f, \boldsymbol{\kappa_f \neq 0})$

In this subsection, the *arc-to-arc* problem is addressed where the curvatures at both ends are nonzero. This problem is meaningful when a vehicle needs to generate a path with a nonzero steering angle, i.e., nonzero curvature, and to arrive at a target position also with a nonzero curvature. This problem has been addressed to generate the shortest path between two positions [Dubins 57], naming the solution path the *Dubins* path, which is comprised only with line(s) and arc(s). Even if this path is not curvature-continuous, this solution is important to analyze the **Problem 3** by separated subcases. Before analyzing the subcases, two different cases of *Dubins* path are shown through geometric representation as follows.



(a) **Case 3A** ($\kappa_i \kappa_f > 0$) (b) **Case 3B** ($\kappa_i \kappa_f < 0$)

Figure 2.6: *Dubins* path representation for **Problem 3**

Dubins path is the shortest path between two configurations such that the path is like string or cable with infinite flexibility and it winds the given circles of radius of curvatures while satisfying the configurations. However, the path is not continuous in the curvature which uses only circular arc and/or line segments in composing the path [Reeds 90, Fraichard 04, Wilde 09]. The schematics of

Dubins path generation is described as follows. At first, the configuration circles for κ_i, κ_f are drawn. According to the sign of κ_f, tangential line and arcs are determined from the geometrical relation. That is, for the case $\kappa_f > 0$ as shown in Fig. 2.6(a), if $\theta_f < \phi_d$, then two clothoids can make feasible path and if not, four clothoids are necessary for the path. For the case $\kappa_f < 0$ as shown in Fig. 2.6(b), if $\theta_f > \phi_d$, then two clothoids can make feasible path and if not, four clothoids are necessary for the path.

Figures 2.6(a) and (b) show two paths generated by the *Dubins* approach for two nonzero curvatures at both boundaries ($\kappa_f < 0$ and $\kappa_f > 0$, respectively with $\kappa_i < 0$). The configurations $\boldsymbol{P_i}$, $\boldsymbol{P_f}$ have curvatures κ_i, κ_f, which correspond to the radii of two circles ρ_i, ρ_f centered respectively at c_i and c_f (cf. Fig. 2.6). In the two configurations, the path that connects two ends is the shortest [Dubins 57].

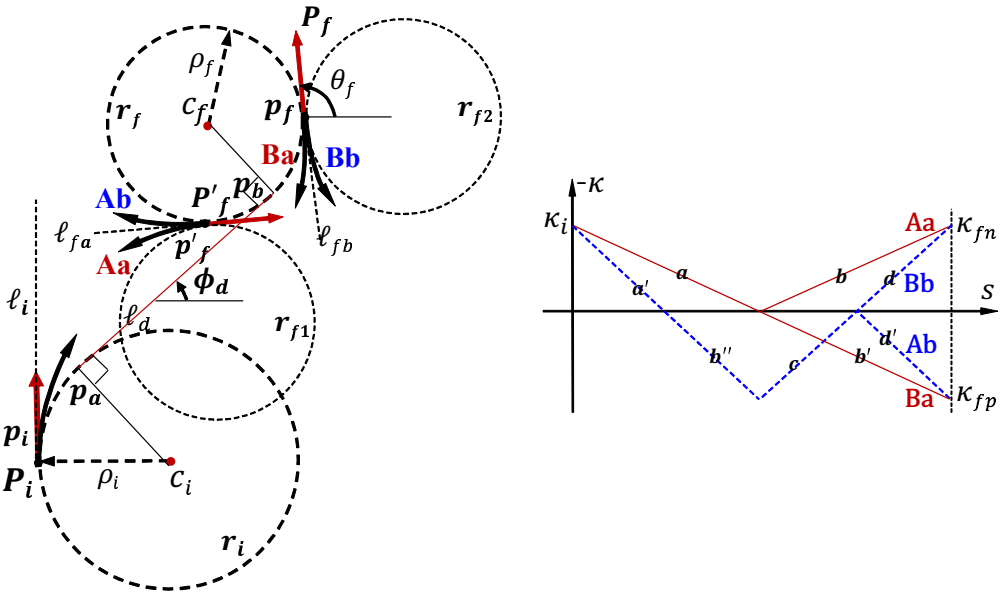
This path has a line ℓ_d, which connects two tangential points $\boldsymbol{p_a}$, $\boldsymbol{p_b}$ on the circles $\boldsymbol{r_{i,f}}$, and the angle $\boldsymbol{\phi_d}$ of a reference line ℓ_d is determined from geometric relations of the given configurations. The point $\boldsymbol{p_c}$ is also obtained according to the geometric relation $\overline{\boldsymbol{p_a p_b}} = \overline{\boldsymbol{p_c} c_f}$ (or $\overline{\boldsymbol{p_c} c_i}$), as shown in Fig. 2.6(a) (or (b)).

The length of $\overline{c_i c_f}$ and ρ_i, ρ_f determine $\overline{\boldsymbol{p_c} c_i}$ (or $\overline{\boldsymbol{p_c} c_f}$), which are used to identify the tangential points $\boldsymbol{p_a}$, $\boldsymbol{p_b}$. The resultant path is simply composed of one arc and two lines that connect $\boldsymbol{p_i}\widehat{\boldsymbol{p_a}}$, $\overline{\boldsymbol{p_a p_b}}$, and $\boldsymbol{p_b}\widehat{\boldsymbol{p_f}}$. Even if the *Dubins* path is the shortest between the two configurations, it is not continuous on the curvature, especially at the intersection point between the arc and/or line segment (e.g., $\boldsymbol{p_a}$ and $\boldsymbol{p_b}$ in Fig. 2.6).

Therefore, for **pCCP** generation in **Problem 3**, it is important to propose an additional sophisticated approach. One useful methodology is to analyze the clothoid composition cases in the curvature diagram. With the initial curvature fixed as $\kappa_i < 0$, two cases are possible with a finial configuration of $\kappa_f < 0$, $\kappa_f > 0$, where the curvature $\kappa(s)$ starts from the negative curvature region and end in the positive or negative region in the diagram.

Figure 2.7 describes four subcases dealing with a *arc-to-arc* problem. $\boldsymbol{P_i}$ is fixed, and only $\boldsymbol{P_f}$ changes for each subcase. Firstly, the problem focus is on a circle $\boldsymbol{r_f}$, as shown in Fig. 2.7(a), which has c_f as center point and ρ_f as its circular radius ($1/\kappa_f$). For each configuration, there are two cases according to whether κ_f is negative (as **Aa** or **Ba**) or positive (as **Ab** or **Bb**). Corresponding curvature diagrams are shown in Fig. 2.7(b). Each clothoid composition is denoted as **Aa** $(a-b)$, **Ab** $(a'-b''-c-d')$, **Ba** $(a-b')$, or **Bb** $(a'-b''-c-d)$, where every small alphabetical character presents an elementary clothoid. This diagram indicates that the feasible minimum number of clothoids depends on the final curvature's sign (positive/negative) as well as the final orientation.

The second to be noted is the relation of θ_f to the reference line ℓ_d, which is



(a) Geometric representation (b) Curvature diagram

Figure 2.7: Arc-to-arc problem subcases

derived in Fig. 2.6(a). The *Dubins* path-based reference line ℓ_d has an important role as a reference line to determine the minimal number of clothoids as two or four.

At first, focus on the circle $\boldsymbol{r_f}$ for $\kappa_f < 0$, where ℓ_d connects $\boldsymbol{p_a}$ to $\boldsymbol{p_b}$ and has the slope of $\tan \boldsymbol{\phi_d}$. Then, **Ba** and **Ab** are two subcases for final positions $\boldsymbol{p_f}$ and $\boldsymbol{p_f'}$, where the orientation is $\theta_f > \boldsymbol{\phi_d}$ and $\theta_f < \boldsymbol{\phi_d}$, respectively. The clothoid of **Ba** is generated on the left side of ℓ_{fb}, and **Ab** is located in the upper side of ℓ_{fa} which determines the feasible minimum number of clothoids as two and four, respectively, also being depicted in the curvature diagram of Fig. 2.7(b). Secondly, let us analyze the final configurations $\boldsymbol{P_f'}$ and $\boldsymbol{P_f}$. Each final configuration has two subcases **Aa**/**Ab** and **Ba**/**Bb** according to the final curvature's sign. As described in **Problem 1**, **Aa**/**Ab** and **Ba**/**Bb** have different directions to ℓ_{fa} and ℓ_{fb}, respectivly such that the feasible minimal number of clothoids are different and so must be analyzed separately.

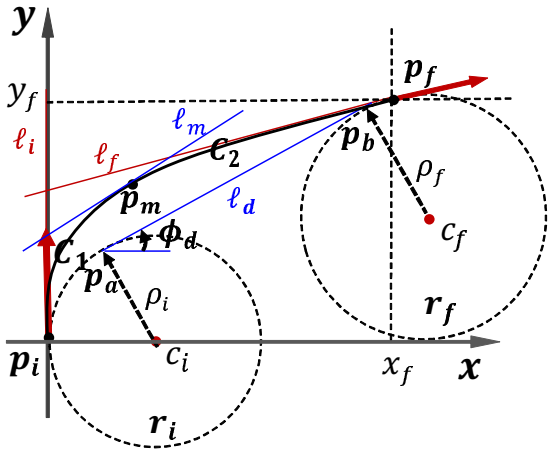
From the above analysis, one can observe that the *arc-to-arc* problem needs to be decomposed into four subcases according to the sign of κ_f and whether or not θ_f is larger than $\boldsymbol{\phi_d}$. In the next four subsections, **Problem 3** is analyzed with respect to each of the four subcases. According to the orientation condition, the problem is firstly decomposed into two cases, **A** and **B** based on the sign of κ_f with κ_i, and each case is separated into two subcases, **a** and **b**, based on the

relation of θ_f and $\boldsymbol{\phi_d}$.

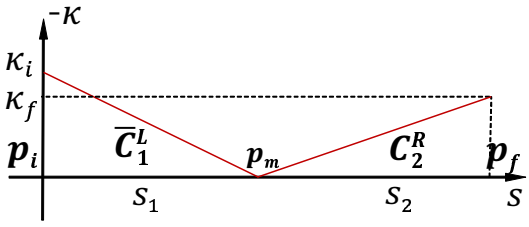
2.2.3.1 Case 3A ($\kappa_i \kappa_f > 0$)

i. **Case 3Aa** ($\theta_f < \boldsymbol{\phi_d}$)

The path starts with nonzero curvature $\kappa_i < 0$ and ends with $\kappa_f < 0$ (positive change of curvatures) and $\theta_f < \boldsymbol{\phi_d}$. Two clothoids can make this path as $\bar{\boldsymbol{C}}_{\boldsymbol{1}}^{\boldsymbol{L}} \boldsymbol{C}_{\boldsymbol{2}}^{\boldsymbol{R}}$, as shown in Fig. 2.8.



(a) Two clothoids composition



(b) Curvature diagram

Figure 2.8: **Case 3Aa** ($\bar{\boldsymbol{C}}_{\boldsymbol{1}}^{\boldsymbol{L}} \boldsymbol{C}_{\boldsymbol{2}}^{\boldsymbol{R}}$)

This subcase given in Fig. 2.8 is similar to *Dubins* the first path (cf. Fig. 2.6(a)) except that the circular arc segment is replaced by a clothoid. The $\boldsymbol{C_1}$ is generated from $\boldsymbol{p_i}$ to $\boldsymbol{p_m}$ with a decreasing curvature. The $\boldsymbol{C_2}$ is generated from $\boldsymbol{p_f}$ to connect to $\boldsymbol{C_1}$ at $\boldsymbol{p_m}$, and its curvature decreases. The generated path $\widehat{\boldsymbol{C}}$ circumvents *Dubins* path $\overline{\boldsymbol{p_i p_a p_b p_f}}$ (cf. Fig. 2.6(a)). For two clothoids to meet at a point $\boldsymbol{p_m}$, they should be inside the region $\boldsymbol{\Sigma_s}$, created by the three lines ℓ_i, ℓ_f, and $\overline{\boldsymbol{p_i p_a p_b p_f}}$. If a *Dubins* path exists in this subcase, then it is always possible

to create a **pCCP** using two clothoids, according to the following reasoning. If a *Dubins* path exists, then $\overline{\boldsymbol{p_i p_a p_b p_f}}$ is possible where $|\ell_d| \geq 0$ ($|\ell_d|$ is the length of the straight line). If ℓ_d exists, then ℓ_m can exist in the upper/outer side of the *Dubins* path, i.e., $\overline{\boldsymbol{p_i p_1 p_2 p_f}}$.

The tangential line ℓ_m at $\boldsymbol{p_m}$ constrains the orientation and curvature for both clothoids composition (cf. Eq. (2.1)). In this subcase, a simple composition method is two elementary clothoids without any arc or line. As depicted in Fig. 2.8(b), the curvature of the first clothoid $\bar{\boldsymbol{C}}_{\boldsymbol{1}}^{\boldsymbol{L}}$ decreases to zero, then the second clothoid $\boldsymbol{C}_{\boldsymbol{2}}^{\boldsymbol{R}}$ exhibits an increased curvature to κ_f.

ii. **Case 3Ab** ($\theta_f > \boldsymbol{\phi_d}$)

The path for this subcase starts with nonzero curvature $\kappa_i < 0$ and ends with $\kappa_f < 0$ and $\theta_f > \boldsymbol{\phi_d}$. Different to **Case 3Aa**, the final configuration is located on the left side of the circle boundary $\boldsymbol{r_f}$. After $\boldsymbol{C_1}$ generation, it is not possible to find any $\boldsymbol{C_2}$ to satisfy the final configuration since the end point of $\boldsymbol{C_2}$ is located on the right side of ℓ_f with $\boldsymbol{C^R}$ shape. In this configuration, an additional clothoids pair is required to create a feasible path, as shown in Fig. 2.9.

In Fig. 2.9(a), $\boldsymbol{C_1}$ and $\boldsymbol{C_2}$ are generated from $\boldsymbol{p_i}$ and $\boldsymbol{p_f}$, respectively. Since the position $\boldsymbol{p_{m2}}$ (or $\boldsymbol{C_2}(s_0)$) is located on the right side of ℓ_f, the end point of $\boldsymbol{C_1}$ cannot meet with $\boldsymbol{C_2}$ on the upper side of ℓ_d. Otherwise, if $\boldsymbol{C_1}$ meets $\boldsymbol{C_2}$ in the lower region of ℓ_d, the orientation is inconsistent (i.e., G^1 discontinuity) since $\theta_{m1} < \boldsymbol{\phi_d}$ at $\boldsymbol{p_{m1}}$, and $\theta_{m2} > \theta_f > \boldsymbol{\phi_d}$ at $\boldsymbol{p_{m2}}$; as a result, $\theta_{m1} < \theta_{m2}$. Here, note that $\kappa_{m1} = 0$ and $\kappa_{m2} = 0$ at positions $\boldsymbol{p_{m1}}$ and $\boldsymbol{p_{m2}}$ respectively.

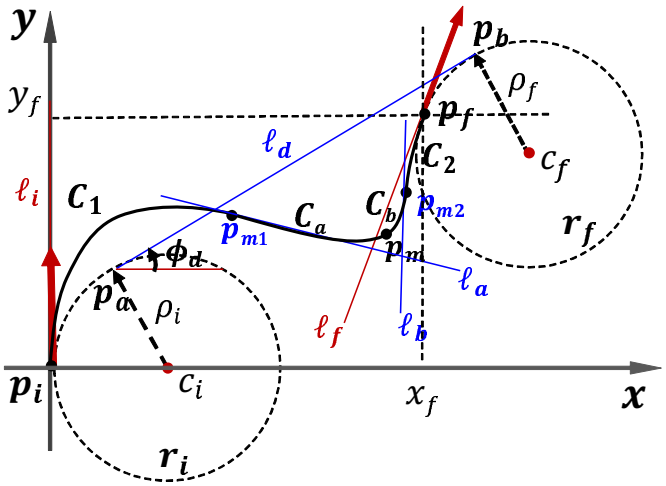
This creates another problem of configuration, i.e., *line-to-line* problem, where ℓ_a and ℓ_a act as ℓ_i and/or ℓ_f of **Problem 1**, respectively. Thus, a remaining issue is how to provide an appropriate geometric condition for the additional two clothoids composition. To this point, let us apply the determinant function λ from Eq.(1.9) between $\boldsymbol{p_{m1}}$ and $\boldsymbol{p_{m2}}$.

To obtain a path with four clothoids, $\boldsymbol{C_1}$ and $\boldsymbol{C_2}$ need to be generated to allow another clothoid pair problem **Case 1A** according to following three conditions,

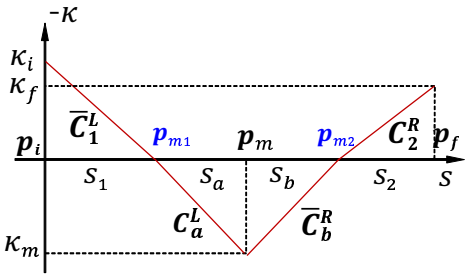
$$\lambda_{\boldsymbol{p_{m1}}}(\boldsymbol{p_{m2}}) > 0,\ \lambda_{\boldsymbol{p_{m2}}}(\boldsymbol{p_{m1}}) < 0, \text{and } \theta_{\overline{\boldsymbol{p_{m1}p_{m2}}}} > \theta_{\boldsymbol{p_{m1}}}, \tag{2.2}$$

where the determinant function $\lambda_{\boldsymbol{p_r}}(\boldsymbol{p_t})$ checks where a target point $\boldsymbol{p_t}$ is located from the line passing a reference point $\boldsymbol{p_r}$ (cf. Eq. (1.9)), and $\theta_{\overline{\boldsymbol{p_{m1}p_{m2}}}}$ is the orientation of the straight line $\overline{\boldsymbol{p_{m1}p_{m2}}}$.

Thus, the algorithmic procedure should identify a feasible point for $\boldsymbol{p_{m1}}$ and $\boldsymbol{p_{m2}}$ by attempting $\boldsymbol{C_1}$ and $\boldsymbol{C_2}$ generation while satisfying the conditions of Eq. (2.2). Since there are numerous feasible solutions for each provided configuration,



(a) Four clothoids composition



(b) Curvature diagram ($\kappa_i \neq \kappa_f$)

Figure 2.9: **Case 3Ab** ($\bar{C}_1^L C_a^L \bar{C}_b^R C_2^L$)

additional constraints or guidance must determine a unique solution. Similar to **Case 1B**, this issue is also addressed in the next section. Figure 2.9(b) shows the corresponding curvature diagram for **Case 3Ab**, which involves four clothoids. One can see that the same methodology used in **Case 1A** can be applied to the $\boldsymbol{C_a^L \bar{C}_b^R}$ in the curvature diagram.

2.2.3.2 Case 3B ($\kappa_i \kappa_f < 0$)

i. **Case 3Ba** ($\theta_f > \boldsymbol{\phi_d}$)

The path for this case starts with nonzero curvature $\kappa_i < 0$ and ends with $\kappa_f > 0$ (negative change of curvatures) and $\theta_f > \boldsymbol{\phi_d}$.

As shown in Fig. 2.10(a), two clothoids form this path as $\boldsymbol{\bar{C}_1^L C_2^L}$, and the corresponding curvature diagram of Fig. 2.10(b) is similar to the second *Dubins* path (cf. Fig. 2.6(b)) except that the circular arc segment is replaced by a clothoid.

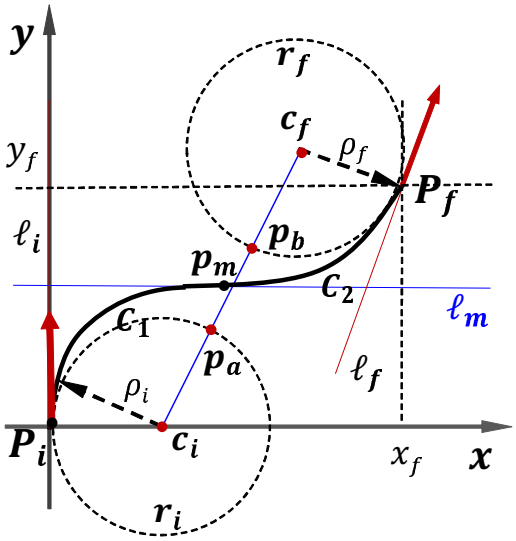
Two elementary clothoids, $\boldsymbol{C_1}$ and $\boldsymbol{C_2}$, can meet at $\boldsymbol{p_m}$ where the curvature is definitely zero and the orientation $\theta_m = \frac{\pi}{2} - \delta_1$ (or equals to $\theta_f - \delta_2$) according to the basic clothoid property (cf. **Property 4**).

If a *Dubins* path exists in this subcase, then it is always possible to make a **pCCP** using two clothoids, based on following reasoning. For a *Dubins* path, $\boldsymbol{\overline{p_i p_a p_b p_f}}$ should exist where $|\ell_d| \geq 0$ ($|\ell_d|$ is the length of the straight line). If ℓ_d exists, then ℓ_m also exists between the space of the two circles' outer boundaries, i.e., $\boldsymbol{\overline{p_1 p_2}}$. Therefore, it is inferred that $|\boldsymbol{\overline{p_1 p_2}}| \geq 0$ is a necessary condition for a *Dubins* path. If this subcase has the condition, $|\boldsymbol{\overline{p_1 p_2}}| < 0$, two clothoids composition is impossible. This subcase could be addressed by adding a clothoid $\boldsymbol{C_0^R}$ as depicted in its curvature diagram in Fig. 2.10(c). This subcase is similar to **Case 2B** (Fig. 2.4(b)) and not considered here to avoid redundancy.

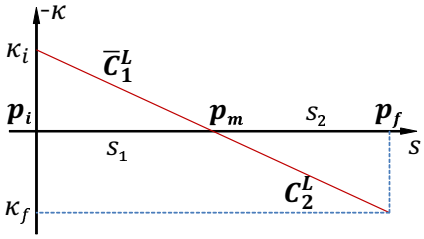
ii. **Case 3Bb** ($\theta_f < \boldsymbol{\phi_d}$)

In this subcase, $\boldsymbol{p_f}$ is positioned on the left side of the $\boldsymbol{r_f}$ boundary, and $\boldsymbol{C_2}$ is generated toward the upper region of ℓ_f. For the end of $\boldsymbol{C_1}$ ($\boldsymbol{p_{m1}}$) to meet $\boldsymbol{C_2}$ ($\boldsymbol{p_{m2}}$) with orientation continuity, it must be true that $\theta_{m1} = \theta_i - \delta_1 < \boldsymbol{\phi_d}$ since $\theta_{m2} = \theta_f - \delta_2 < \boldsymbol{\phi_d}$, i.e., $\theta_{m1} < \boldsymbol{\phi_d}$; thus $\delta_1 > \pi/2 - \boldsymbol{\phi_d}$. From **Property 1**, $\boldsymbol{p_{m1}}$ circumvents $\boldsymbol{p_f}$ as δ_1 increases; thus, it is impossible to meet $\boldsymbol{p_{m2}}$. The corresponding curvature diagram (cf. Fig. 2.11(b)) shows the fact that additional two clothoids, $\boldsymbol{C_a}$ and $\boldsymbol{C_b}$ are required to supplement the geometric insufficiency.

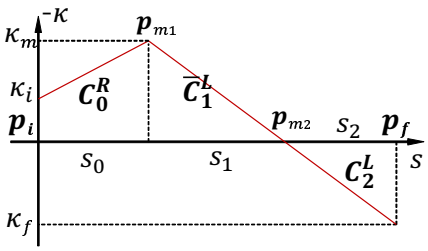
As described in **Case 3Ab**, this subcase includes an additional *line-to-line* problem due to $\kappa_{m1} = 0$ and $\kappa_{m2} = 0$ at positions of $\boldsymbol{p_{m1}}$ and $\boldsymbol{p_{m2}}$ respectively.



(a) Two clothoids composition

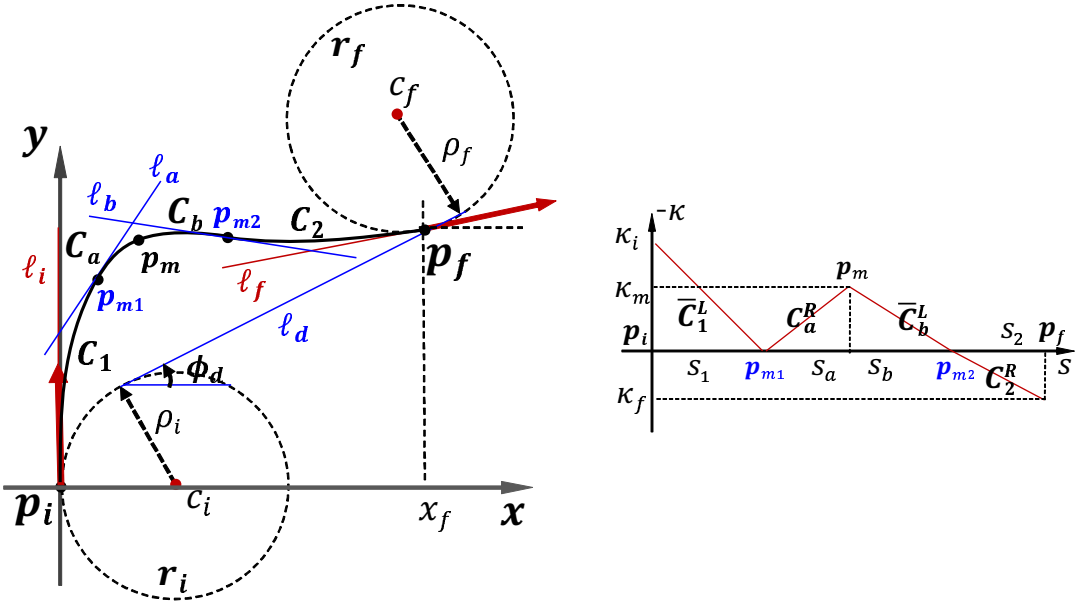


(b) Curvature (2 clothoids)



(c) Curvature (3 clothoids)

Figure 2.10: **Case 3Ba** ($\bar{C}_1^L C_2^L$, $C_0^R \bar{C}_1^L C_2^L$)



(a) Four clothoids composition (b) Curvature diagram

Figure 2.11: **Case 3Bb** ($\bar{C}_1^L C_a^R \bar{C}_b^L C_2^R$)

To obtain four clothoids path, $\boldsymbol{C_1}$ and $\boldsymbol{C_2}$ need to be generated according to the formulation given in the **Case 1A** condition (especially by ℓ_a and ℓ_b) using a similar methodology of Eq. (2.2) as follows,

$$\lambda_{p_{m1}}(\boldsymbol{p_{m2}}) < 0, \lambda_{p_{m2}}(\boldsymbol{p_{m1}}) < 0, \theta_{\overline{p_{m1}p_{m2}}} < \theta_{p_{m1}}. \tag{2.3}$$

In this subsection, the *arc-to-arc* problem is addressed. From the descriptions above, the following could be summarized.

Remark: *I*n the *arc-to-arc* problem,

a. If a *Dubins* path exists, a two-clothoids solution exists for **pCCP**.

b. If a *Dubins* path is not availabe, an additional **Case 1A** solution exists for **pCCP**.

In the algorithmic procedure, $\boldsymbol{C_1}$ and $\boldsymbol{C_2}$ are tried with an initial assumption for δ_1 and δ_2 (i.e., $\kappa_1 = \kappa_i$, $\kappa_2 = \kappa_f$), and it is iteratively varied until the condition of Eq. (2.3) is satisfied in order to provide the boundary configurations of **Case 1A**. The detailed procedures for **Problem 3** are described in **Algorithm 4** and **5**.

In **Algorithm 4**, both curvatures κ_i, κ_f are given, and the initial deflection δ_1 for $\boldsymbol{C_1}$ with variations $d\delta$ is fixed before entering the loop. In lines [15-19], the parameter is varied while taking into account **Property 4** and Eq. (1.7). Note that the varying parameter is δ_1 since κ_i and κ_f are all given. Other parts are all similar to **Algorithm 1** or **3**.

Algorithm 4 Arc-to-arc (Two clothoids generation)

Require: $\varepsilon, d\delta, \text{sol} = \text{FALSE}$ ▷ Pre-setting values for iterative convergence
Require: δ_1 ▷ Initial parameter assumption C_1 with given κ_i, κ_f
1: **procedure** CLOTHOID2AA($\boldsymbol{P_i}, \boldsymbol{P_f}$)
2: **while** sol == FALSE **do**
3: $\delta_1 \leftarrow \kappa_1, \alpha_1$ ▷ C_1 generation using Eq. (1.14)
4: $\delta_2 = \theta_i - \theta_f - \delta_1$ ▷ G^2 continuity using Eq. (2.1)
5: $\alpha_2 \leftarrow \kappa_2, \delta_2$ ▷ C_2 generation using Eq. (1.14)
6: $\widehat{C} \leftarrow C_1 \oplus C_2$ ▷ Two clothoids composition
7: $D_e \leftarrow \ell_m, \widehat{C}, \ D_e^{\perp} \leftarrow \ell_m^{\perp}, \widehat{C}$ ▷ Distance error parameters
8: **if** $|D_e| < \varepsilon$ **then**
9: **if** $D_e^{\perp} \geq 0$ **then**
10: sol = TRUE ▷ Convergence achieved
11: **return** $\boldsymbol{\alpha_{sol}, \delta_{sol}, \widehat{C}}$ ▷ Obtained solution
12: **end if**
13: **end if**
14: $\lambda = D_e, \lambda^{\perp} = D_e^{\perp}$
15: **if** $\lambda\lambda' < 0$ **then** ▷ Bisection method for δ
16: $d\delta = \frac{d\delta}{2}$
17: **end if**
18: $d\delta = -|d\delta| \cdot \text{sign}(\lambda)$ ▷ δ variation criteria
19: $\delta_1 = (\delta_1 + d\delta)$
20: $\lambda' = \lambda$ ▷ For next iteration
21: **end while**
22: **end procedure**

Algorithm 5 describes the procedures for **Case 3Ab** and **Case 3Bb**. Firstly, $\boldsymbol{C_1}$ and $\boldsymbol{C_2}$ are generated with given κ_i and κ_f by initial parameters δ_1, δ_2, respectively (lines [3-4]). Then the condition given in Eq. (2.2) or (2.3) for **Problem 1** is formulated to create $\widehat{\boldsymbol{C_{ab}}}$ in lines [7-9], which is comprised of $\boldsymbol{C_a}$ and $\boldsymbol{C_b}$, as shown in Fig. 2.8 or Fig. 2.10. Each iteration is continued until the convergence criteria is passed as lines [12-17]. Note that both conditions of line 5 or line 6 (by Eq. (2.2) or Eq. (2.3), respectively) are useful to determine the solvability of the additional **Case 1A** problem. These subcases have numerous solution paths according to $\boldsymbol{C_1}$ and $\boldsymbol{C_2}$ and might be difficult to converge iteratively. Thus, other constraints or guidance for parameter variation are required (like **Case 1B**). This issue is addressed in the next section.

Thus far, three problems with subcases were introduced and analyzed. Those solutions were derived based on clothoids composition with geometric representation and the curvature continuities for all the compositions were verified by

Algorithm 5 Arc-to-arc (Four clothoids generation)

Require: $\varepsilon, d\delta$ ▷ Pre-setting values for iterative convergence
Require: δ_1, δ_2 ▷ Initial parameters assumption C_1 and C_2 with given κ_i, κ_f

procedure CLOTHOID4AA($\boldsymbol{P_i}, \boldsymbol{P_f}$)
  while sol == FALSE **do**
    $\alpha_1 \leftarrow \kappa_i, \delta_1$ ▷ C_1 generation using Eq. (1.14)
    $\alpha_2 \leftarrow \kappa_f, \delta_2$ ▷ C_2 generation using Eq. (2.1)
    if $\lambda_{p_{m1}}(p_{m2}) > 0, \lambda_{p_{m2}}(p_{m1}) > 0, \theta_{\overline{p_{m1}p_{m2}}} > \theta_{p_{m1}}$ **then** ▷ Case 3Ab
      $(\lambda_{p_{m1}}(p_{m2}) < 0, \lambda_{p_{m2}}(p_{m1}) < 0, \theta_{\overline{p_{m1}p_{m2}}} < \theta_{p_{m1}})$ ▷ Case 3Bb
      procedure clothoid2LL($\boldsymbol{P_a}$, $\boldsymbol{P_b}$)
      $\widehat{C_{ab}} \leftarrow C_a \oplus C_b$
      return $\boldsymbol{\widehat{C_{ab}}}$
      $D_e \leftarrow \widehat{C_{ab}},\ \ell_{m2},\ D_e^{\perp} \leftarrow \widehat{C_{ab}},\ \ell_{m2}^{\perp}$ ▷ Distance error parameters
      $D_e \leftarrow \widehat{C_1}, \widehat{C_2}$
      if $|D_e| < \varepsilon$ **then**
        if $D_e^{\perp} \geq 0$ **then**
          sol = TRUE ▷ Convergence achieved
          return $\bar{\boldsymbol{\alpha}}_{\boldsymbol{sol}}$,$\bar{\boldsymbol{\delta}}_{\boldsymbol{sol}}$, $\widehat{\boldsymbol{C}}_{\boldsymbol{1,ab,2}}$ ▷ Obtained solution
        end if
      end if
      $\lambda = D_e, \lambda^{\perp} = D_e^{\perp}$
      if $\lambda\lambda' < 0$ **then** ▷ Bisection method for δ_1, δ_2
        $d\delta = \frac{d\delta}{2}$
      end if
      $d\delta = -|d\delta| \cdot \text{sign}(\lambda)$ ▷ $\delta_{1,2}$ variation criteria
      $\delta_{1,2} = (\delta_{1,2} + d\delta)$
      $\lambda' = \lambda$ ▷ For next iteration
    end if
  end while
end procedure

the curvature diagrams. Each problem was formulated to resolve all (as much as possible) the possible cases for vehicle local path planning from initial to final configurations. Table 2.2 summarizes the described problems with corresponding subcases.

Table 2.2: **pCCP** Problems with subcases

Problem	Type		Condition	No. of clothoids	Composition
line-line	Case 1A		$\theta_f < \phi$	2	$C_1^R \bar{C}_2^L$
	Case 1B		$\theta_f > \phi$	4	$C_{1a}^R \bar{C}_{1b}^L C_{2b}^L \bar{C}_{2a}^R$
line-arc	Case 2A		$\theta_f < \phi$	1	C^R
	Case 2B		$\theta_f > \phi$	3	$C_1^R \bar{C}_2^L C_3^L$
arc-arc	Case 3A	$\kappa_f > 0$	$\theta_f < \phi_d$	2	$C_1^L \bar{C}_2^R$
		$\kappa_f < 0$	$\theta_f > \phi_d$	4	$\bar{C}_1^L C_a^L \bar{C}_b^R C_2^L$
	Case 3B	$\kappa_f > 0$	$\theta_f > \phi_d$	2	$\bar{C}_1^L C_2^L$
		$\kappa_f < 0$	$\theta_f < \phi_d$	4	$\bar{C}_1^L C_a^R \bar{C}_b^L C_2^R$

2.2.4 Minimax sharpness constraint

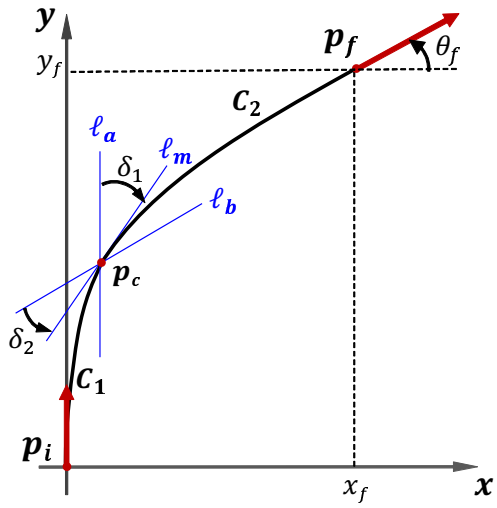
In the multiple clothoids problem, the proposed algorithms require additional constraint (or solution constraint) to obtain a unique solution. For example in the composition of using multiple clothoids such as **Case 1A**, **Case 1B** or **Case 3A**, **3B**, there can be numerous solutions which satisfies the given configurations. It is not general to impose any constraint on the curvature or sharpness from mechanical, electrical or control limit, in that the resultant path could be an extreme case compared to other normal paths feasible. It is not desirable to use a random from the obtained feasible paths. Composing multiple pairs of symmetric (two) clothoids is a good candidate for the constraint, however it is also an issue about how the symmetric clothoids are chosen. In this section, *sharpness* is adopted to implement as a key factor to impose constraint in obtaining solution. As discribed in the previous chapters, the sharpness has an important role in generating a clothoid with 1^{st} order form in Eq. (1.3) (p.28). In this respect, as the solution constraint, *minimax sharpness constraint* (**MSC**) for two clothoids is proposed as follows.

Lemma 1 (Minimax sharpness constraint for two clothoids) *When $\widehat{C}(\alpha_1, \alpha_2)$ is a feasible solution which satisfies given two boundary conditions, maximum sharpness is described as $max[\alpha_1, \alpha_2] = \hat{\alpha}$, then the solution by* **MSC** *is the one of which $\hat{\alpha}$ is the minimum among the feasible solutions. The* **MSC** *provides a feasible solution composed by two clothoids as,*

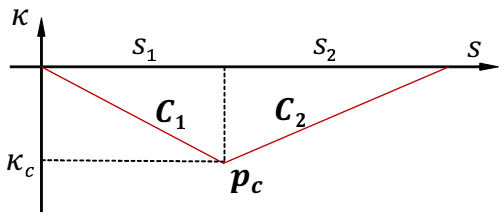
$$\alpha_1 = \alpha_2, \delta_1 = \delta_2 = \frac{\theta_1 - \theta_2}{2}. \tag{2.4}$$

Proof 1 *When two clothoids $\boldsymbol{C_1}$ and $\boldsymbol{C_2}$ are composed as Fig. 2.12(a), the both clothoids have a common peak curvature value (κ_c in Fig. 2.12(b)). For each clothoid in the composition, δ is varied according to α under the constraint of Eq. (1.14). In the graph given in Fig. 2.12(c), let assume that $\delta_m = \frac{\theta_1 - \theta_2}{2}$.*
If $\alpha_1 < \alpha_m$, then $\delta_1 > \delta_m$ by $2\delta_1\alpha_1 = C$ $(Const.)$.
From $\delta_1 + \delta_2 = \theta_1 - \theta_2 = 2\delta_m = C'$ $(Const.)$ by G^1 continuity at $\boldsymbol{p_c}$ of Fig. 2.12(a), it is derived as $\delta_2 = C' - \delta_1 = 2\delta_m - \delta_1$.
Since $\delta_1 > \delta_m$, thus $\delta_2 = 2\delta_m - \delta_1 < \delta_1$ which results in $\delta_2 < \delta_1$.
It is also true that $2\delta_2 < \delta_1 + \delta_2 = 2\delta_m$, so that $\delta_2 < \delta_m$.
*From $\kappa_c^2 = 2\delta_1\alpha_1 = 2\delta_2\alpha_2 = C$, it results that $\alpha_2 > \alpha_m$: $\max[\alpha_1, \alpha_2] > \alpha_m$ (**A**).*
*Vice versa, if $\alpha_1 > \alpha_m$, it also results that $\alpha_2 < \alpha_m$: $\max[\alpha_1, \alpha_2] > \alpha_m$ (**B**).*
*From above two results (**A**), (**B**), α_m is the minimax sharpness with the condition of $\delta_1 = \delta_2 = \frac{\theta_1 - \theta_2}{2}$ and $\alpha_1 = \alpha_2$: a symmetric pair of clothoids.*

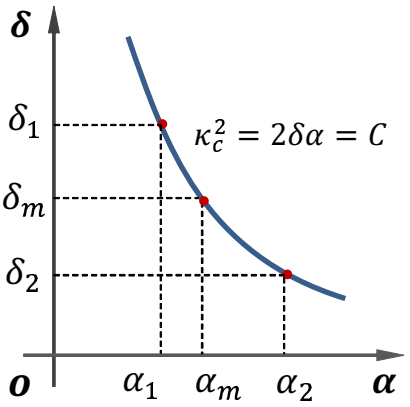
To accomodate the understanding about this concept, an example is described as follows.



(a) Two clothoids composition



(b) Curvature diagram of (a)



(c) $\alpha - \delta$ relation

Figure 2.12: Two clothoids composition scheme

In Fig. 2.12(a) and (b), a pair of clothoids satisfies the curvature continuity at $\boldsymbol{p_c}$ with common κ_c and θ_c, thus it constrains two deflections of δ_1 and δ_2 to satisfy $\delta_1 + \delta_2 = \theta_1 - \theta_2$.

It is also assumed from curvature diagram that the sharpness is constant through each clothoid and the magnitude is proportional to the slope of the curvature (cf. Eq. (2.4)). Each solution for the clothoids pair has two different sharpness values for right turning ($\kappa < 0$) and left turning ($\kappa > 0$) (cf. Fig.2.12(b)). Under the definition of **pCCP** problem (**Section 2.1**), since line segment(s) is included to satisfy the configurations, there are numerous solutions even in the two clothoids composition. To obtain a unique **pCCP** solution for the given problem, **MSC** is applied to the multiple number of clothoids composition. To clarify the proposed conceptof **MSC**, **Case 1A** is addressed first. For the two clothoids composition, there can be an infinite number of clothoid pairs that satisfy the algorithmic convergence criteria of Eq. (1.8).

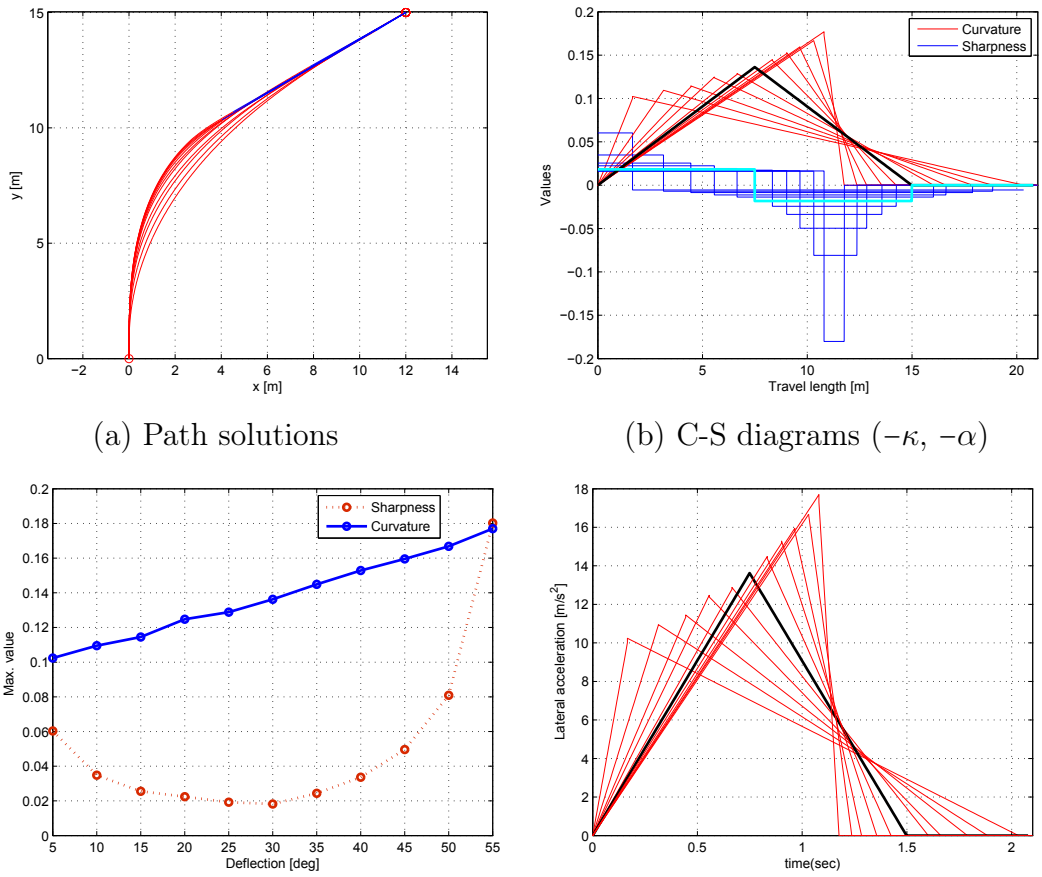
Figure 2.13 shows the analysis of feasible **pCCP** solutions for both configurations $\boldsymbol{P_i}\,(0, 0, 90°, 0)$ and $\boldsymbol{P_f}\,(12, 15, 30°, 0)$.

In Fig. 2.13(a), all composition pairs are candidates of **pCCP** for both configurations. Except the solution that exactly coordinated with the arrival position, other solutions include additional straight line segment with slope $\tan 20°$. Figure 2.13(b) plots curvatures and sharpness along the length (*C-S diagram*) for all path solutions. Each solution pair has two different sharpness values. The sharpness is constant through each clothoid, and the magnitude is proportional to the slope of the curve.

Here, note the bold block lines for curvature and bold blue dotted lines for sharpness. This marked solution has the same sharpness magnitude, and the two curvature/sharpness segments are symmetric to each other along the length. Compared to this marked solution, the other solutions have one higher sharpness value but lower values for the other. Namely, the marked solution has the *minimax sharpness*, i.e., *the maximum sharpness for each pair is the minimum* among all solutions in sharpness magnitude.

Furthermore, this solution contains another important property about deflection. Figure 2.13(c) plots the maximum sharpness value (in magnitude) among the two different sharpness in the solution as the deflection δ_1 varies from 5° to 55°. It is observed that the minimum sharpness is obtained when the deflection is 30°, which is exactly half of the orientation difference between the two configurations, i.e., $\delta_1 = (\theta_i - \theta_f)/2$.

Figure 2.13(d) plots lateral acceleration for travel time when the vehicle follows this path with constant velocity 10 m/s. Since the lateral acceleration is estimated using $\kappa(t)v(t)^2$ (cf. **Section 2.1**), it follows the shape of the curvature



(a) Path solutions

(b) C-S diagrams ($-\kappa$, $-\alpha$)

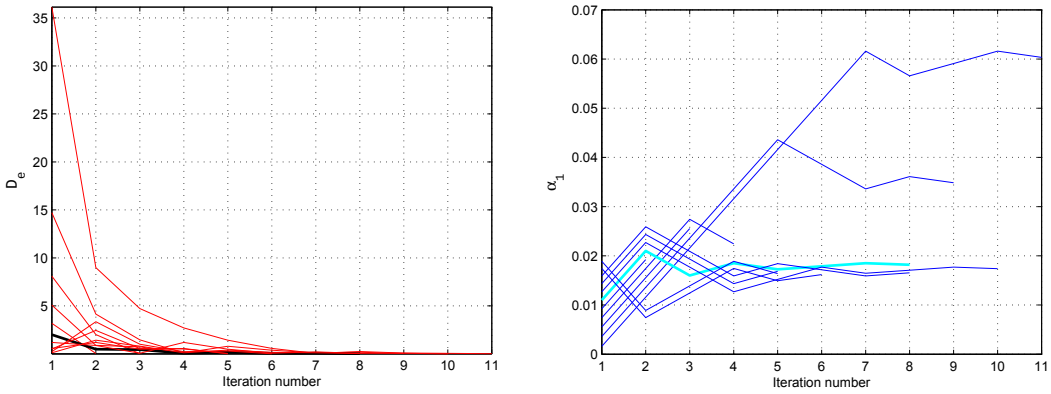
(c) Max. value for deflection

(d) Acceleration for travel time

Figure 2.13: Solution analysis for the two clothoids compositions (**Case 1A**)

diagram of Fig. 2.12(b). This result implies that the **MSC** provides moderate lateral acceleration (peak value) and moderate steering action time (period $\kappa > 0$) compared to the other feasible solutions. Thus, the **MSC** can be applied to obtain **pCCP** for vehicle path following (cf. **Section 4.2.1**).

Considering algorithmic efficiency and convergence of the proposed algorithm with the **MSC**, D_e and α_1 variation according to iteration number are depicted in Fig. 2.14.



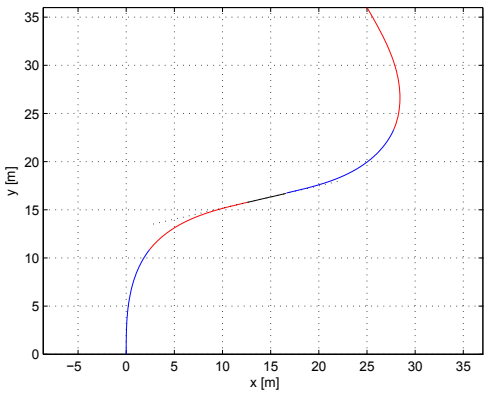
(a) D_e variation ($[m]$) (b) α_1 variation

Figure 2.14: Iterative convergence for the two clothoids composition (**Case 1A**)

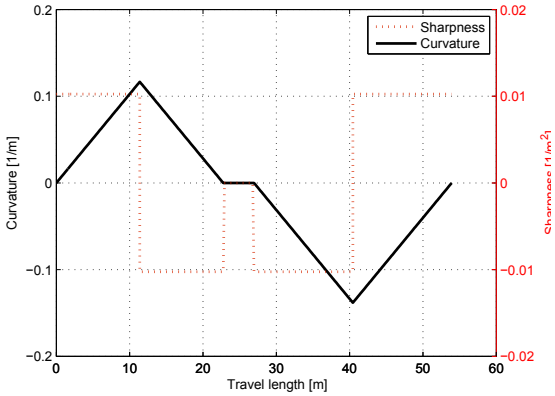
In Fig. 2.14, iterative convergence for D_e and α_1 (where each bold line is the **MSC** are completed in at most 10 loops, requiring less than about 50 ms in MATLAB (Each loop spends less than 5 ms, cf. **Chapter 4**).

Now, let this condition be expanded to further complex problems such as **Case 1B**. In the four clothoids composition, the **MSC** imposes an additional constraint to address the given parameter variations. As denoted in **Algorithm 2**, two paired clothoids require a common reference line ℓ_m with slope $\tan\theta_m$; thus, θ_m also varies in an iterative procedure. The **MSC** can confine θ_m in line 2 to be half of $(\theta_i - \theta_f)$ by Eq. (2.4). Furthermore, the same property also produces a *deflection* constraint in both procedures of line 4 and line 6, according to $\delta_{1a} = \delta_{1b} = \frac{\theta_i - \theta_m}{2}$ and $\delta_{2a} = \delta_{2b} = \frac{\theta_m - \theta_f}{2}$. By the **MSC**, the amount of iteration is remarkably reduced and enables the four clothoids composition to obtain a unique solution. The following results are shown for the **Case 1B** problem with $\boldsymbol{P_i}(0,\ 0,\ 90°,\ 0)$ and $\boldsymbol{P_f}(25,\ 36,\ 120°, 0)$.

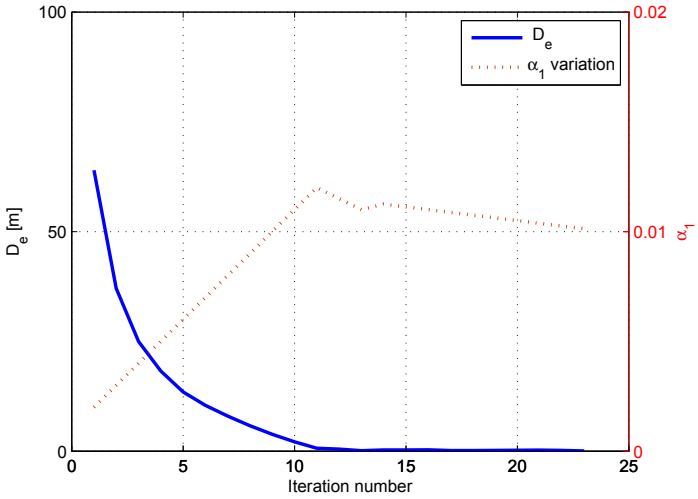
Figure 2.15 depicts the algorithmic solution for the four clothoids composition of **Case 1B**. In Figs. 2.15(a) and (b), the obtained path and its corresponding



(a) Four clothoids path



(b) C-S diagram ($-\kappa$, $-\alpha$)



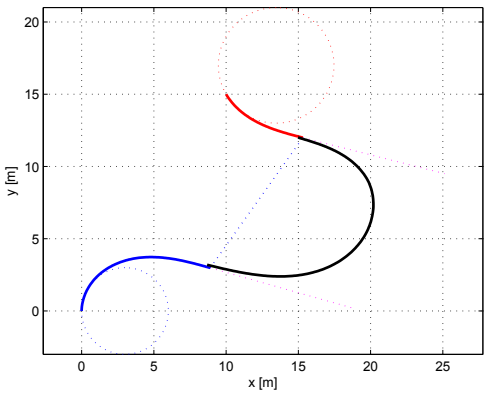
(c) Iterative convergence

Figure 2.15: Four clothoids solutions (**Case 1B**) by **MSC**

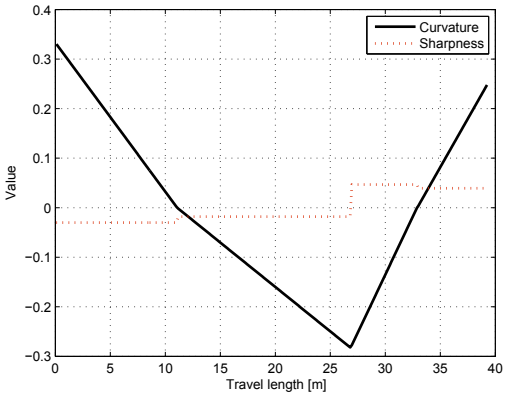
curvature/sharpness diagram (C-S diagram) are plotted. The two pairs have the same sharpness magnitude and the same deflection value according to the **MSC**. In the iterative procedure of **Algorithm 2**, convergence was attained in 20 loops (less than 100 ms) by α_1 variation, as shown in Fig. 2.15(c).

Other four clothoids composition subcases are **Case 3Ab** and **Case 3Bb** in the *arc-to-arc* problem. Different to **Case 1B**, these two subcases are resolved after the clothoids at both ends are generated; thus, there could be numerous solutions of four clothoids composition according to the set of generated two clothoids. Therefore, under the compliance with conditions for **Case 1A** by Eq. (2.2) (or Eq. 2.3), the **MSC** is adopted for the subcases to have a unique solution. To be more specific, the first two clothoids $\bar{\boldsymbol{C}}_{\boldsymbol{1}}^{\boldsymbol{L}}$ and $\boldsymbol{C}_{\boldsymbol{2}}^{\boldsymbol{R}}$ for **Case 3Ab** or $\bar{\boldsymbol{C}}_{\boldsymbol{1}}^{\boldsymbol{L}}$ and $\boldsymbol{C}_{\boldsymbol{2}}^{\boldsymbol{L}}$ for **Case 3Bb** (cf. Figs. 2.9 and 2.11) are adjusted for deflection in order to have the same value. This guidance was also attempted to produces corresponding sharpness values, in agreement with **Lemma 1**.

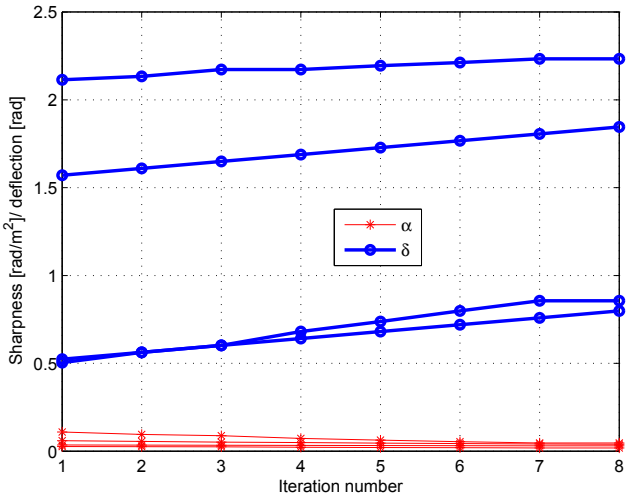
Figures 2.16 and 2.17 illustrate the four-clothoid composition by adopting the proposed minimax sharpness constraint for **Case 3Ab** with $\boldsymbol{P_f}(10,\ 15,\ 120^\circ, \frac{1}{4})$ and for **Case 3Bb** with $\boldsymbol{P_f}(10,\ 12,\ 20^\circ, \frac{1}{4})$, both with $\boldsymbol{P_i}(0,\ 0,\ 90^\circ,\ \frac{1}{3})$. In each figure, subfigure (a) displays the obtained path, (b) is the corresponding curvature diagram, and (c) shows the convergence by iteration number for the two parameters α, δ of all the composed clothoids. In all results of Figures 2.16 and 2.17, algorithmic convergence is achieved in 10 loops, requiring less than 100 ms (also in MATLAB) for the additional loops compared to **Case 1A**.



(a) Four clothoids path

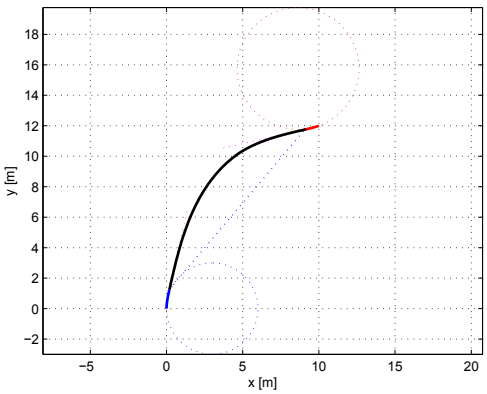


(b) C-S diagram($-\kappa$, $-\alpha$)

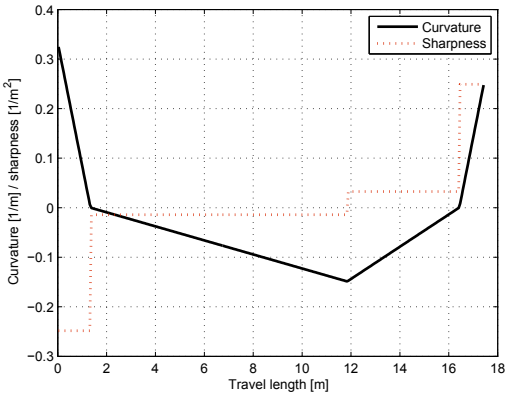


(c) Iterative convergence

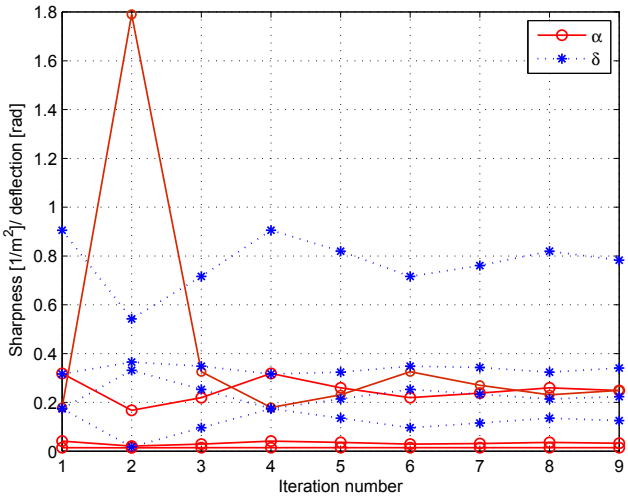
Figure 2.16: Four clothoids solutions (**Case 3Ab**) by **MSC**



(a) Four clothoids path



(b) C-S diagram ($-\kappa$, $-\alpha$)



(c) Iterative convergence

Figure 2.17: Four clothoids solutions (**Case 3Bb**) by **MSC**

2.3 Conclusion

In this chapter, a general solution for clothoid-based **pCCP** problem is proposed and analyzed. To cover all the feasible cases as much as possible, three problems from **Problem 1** to **Problem 3** are defined and each problem is divided into several subcases according to the geometric condition in the configuration variables. Each case is analyzed by elementary clothoid(s) and the algorithmic procedure by iterarive convergence is constructed/formulated using the clothoid properties founded by explicit empirical studies. To obtain a unique solution in the problem, additional constraints such as Minimax sharpness constraint (**MSC**) and elementary clothoid (or 1^{st} order of sharpness) are adopted.

Even if the proposed problems with subcases extend the coverage of **pCCP** problem to the larger scope, it still has a room to elaborate more on how to obtain the unique solution for all the possible cases and its algorithmic procedures are also required to be optimized in the point of practical applicability such as convergence time, parallel processing and algorithmic rules. Those issues could be more critical when the final boundary configuration (or target configuration) is varying, and especially when the given path should be replanned. To compensate those shortcomings and cope with such a dynamic environment, additional problem **Problem 4** is introduced and tackled with algorithmic consideration in the following chapters.

Chapter 3

Local pCCP generation for dynamic target

3.1 Introduction

In **Chapter 2**, a general methodology to generate **pCCP** is addressed with three problems and subsequent cases. All the defined problems have full set of configuration information, i.e., position, orientation and curvature in the initial and final configurations. However, in real environment or dynamic one where other vehicles or obstacles are moving around the considered vehicle and their velocity (or speed) could change, an adaptive capability and optimized functionality to the final configuration are required to the proposed **pCCP**. Thus, in this chapter, parametric Continuous Curvature Path for dynamic target or in dynamic environment (**d-pCCP**) is addressed.

The **d-pCCP** is especially relevant for the determination of the clothoids' parameters under the limitation of incomplete set of final configuration., thus it could be more desirable to construct sequential and multiple **pCCP**s which are connected with all geometric continuities in a global path. Furthermore, it can be applied to the urgent status that requires a faster and short path generation within an extremely short period.

Dynamic obstacle avoidance is tackled in this chapter while using planning and re-planning methods based on **d-pCCP** and an appropriate velocity planning. The methods present specialized frameworks to deal with checking a collision between obstacle and the considered vehicle and thus, the considered path should recognize the boundary or covering range on the 3D volumetric information of the vehicle and obstacle. In the corresponding section, a simple but efficient model using spheres or ellipsoids are adopted to analyze the collision status and

the modelling method, is integrated with 4D configuration space. Under those conceptual frameworks, some demonstrative examples are applied to realize the performance of **d-pCCP** in obstacle avoidance maneuvers by pose and velocity replanning.

3.2 Sequential pCCP generation for dynamic target: Problem 4

In this section, a problem definition and its solution to generate **d-pCCP** is described.

Problem 4: $\boldsymbol{P_i}\left(x_i, y_i, \frac{\pi}{2}, \boldsymbol{\kappa_i \geq 0}\right)$ to $\boldsymbol{P_f}\left(x_f, y_f, \widetilde{\boldsymbol{\theta_f}}, \widetilde{\boldsymbol{\kappa_f}}\right)$

Until **Problem 3**, the boundary condition for final configuration is given. However, there exists a case which has not a full boundary condition at the final configuration. Such case could be found in generating a slalom-like global path by connecting sequential local path or path replanning in dynamic environment. For slalom-like global path, it is difficult to distinguish between line and arc segments where the arcs have complex variation of curvatures. Replanning a path for dynamic obstacle is another case that the final configuration could not be specified. For example, when a vehicle needs to modify its path to avoid a moving obstacle in the middle of turning (not a zero steering handle position), the final desired vehicle configuration could be specified in terms mainly of position and no specific final orientation or curvature. Indeed, among the important functionality for obstacle avoidance is to guarantee always a certain minimum distance with the obstacle (cf. **Chapter 2**), which is guaranteed mainly in terms of distance between the vehicle and the obstacle. These cases occur occasionally in an unstructured road environment and possess *S-shaped* irregular pattern of curvature variation. To cope with described cases, **d-pCCP** problem should be formulated differently to **Problems 1** to **3** with following assumptions.

Assumption 1 *Path representation by a non-elementary clothoid*

i. Each local path can be presented by elementary or non-elementary one clothoid.

ii. A line or arc segment can be included at both sides of each clothoid.

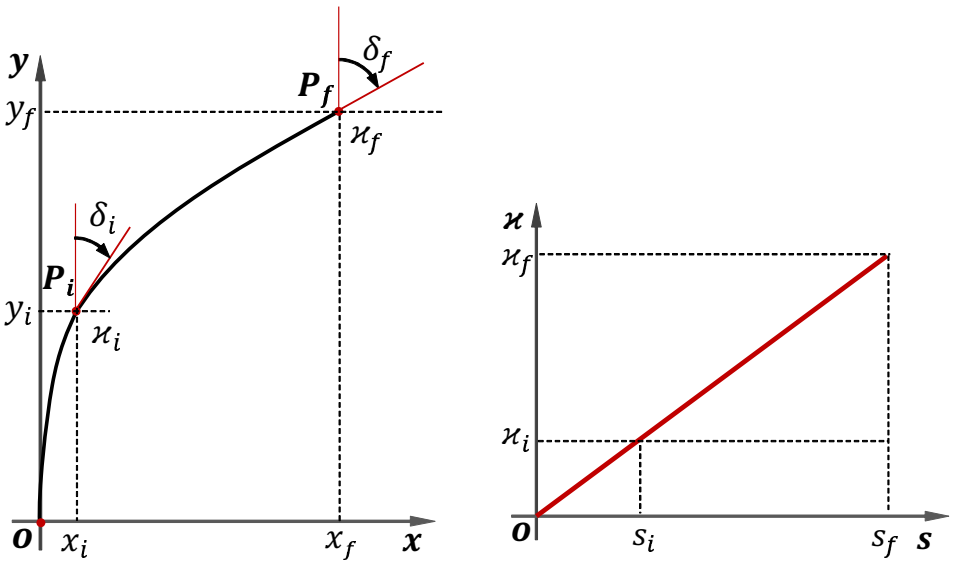
iii. Each way point is selected/acquired as sufficiently dense as it can represent the original curve.

iv. Each way point is selected/acquired periodically (by constant speed) or with equal distance.

v. Desired curvature or orientation in each boundary condition can be defined from geometric consideration of the original global path.

The assumption is based on the preposition that the original or reference path is represented most closely by several local paths of sequentially connected **pCCP**s, where the local path is not limited to elementary clothoid.

Under the assumption described above, two kinds of strategy in the problem definition and solution are performed where one is so called orientation following and the other is curvature following. For the defined **Problem 4**, the orientation or curvature at final configuration is denoted as $\widetilde{\theta_f}$ or $\widetilde{\kappa_f}$ respectively, where the symbol tilde $\widetilde{(\cdot)}$ signifies *indefinite* before obtaining a solution. Between two configurations, it is required to obtain a **d-pCCP** solution where the solution is a non-elementary clothoid including an arc segment.



(a) Path in the $x-y$ coord. (b) Curvature diagram ($-\kappa$)

Figure 3.1: Geometric representation of one clothoid formulation

Fig. 3.1 shows a geometric representation for a non-elementary clothoid [1] in the *Cartesian* coordinate frame (a) and curvature of the *Frenet-Serret* frame (b). In Fig. 3.1(a), a clothoid is generated from its origin **o** with zero curvature and deflection to the end point, thus it makes nonzero curvature and deflection at the end. When two configurations are defined as $\boldsymbol{P_i}$ and $\boldsymbol{P_f}$, each configuration has nonzero deflection (or curvature) as δ_i and δ_f (or κ_i and κ_f in Fig. 3.1(b)). To formulate the problem, a segment is extracted from $\boldsymbol{P_i}$ to $\boldsymbol{P_f}$ (i.e. non-elementary clothoid) and rotated by *counter-clockwise* δ_i to align with the vehicle frame.

The clothoid for these cases is formulated as follows.

[1] the *non-elementary* clothoid has its initial curvature of non-zero with the deflection below $\frac{\pi}{2}$ cf. Fig. 1.3 in **Chapter 1** (p.31).

$$\begin{cases} x(s) = \sqrt{\pi/\alpha} \cdot (\mathcal{FC}\left(\sqrt{\alpha/\pi}\ s\right), \\ y(s) = \sqrt{\pi/\alpha} \cdot (\mathcal{FS}\left(\sqrt{\alpha/\pi}\ s\right) \end{cases} \tag{3.1}$$

where $s \in [s_i, s_f]$ with $s_i = \frac{\kappa_i}{\alpha}$ and $s_f = \frac{\kappa_f}{\alpha}$. Then, the path is rotated counter-clockwise by the amount of $\delta_i = \frac{\kappa_i^2}{2\alpha}$ as,

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos(-\delta_i) & -\sin(-\delta_i) \\ \sin(-\delta_i) & \cos(-\delta_i) \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} \tag{3.2}$$

The resultant clothoid is used to find solution by its parametric regulation.

The **Problem 4** can be defined from $\boldsymbol{P_i}$-$\boldsymbol{P_f}$ to $\boldsymbol{O}$-$\boldsymbol{P_{tg}}$ by the coordinate transformation in the *Cartesian* coordinate as depicted in Fig. 3.2.

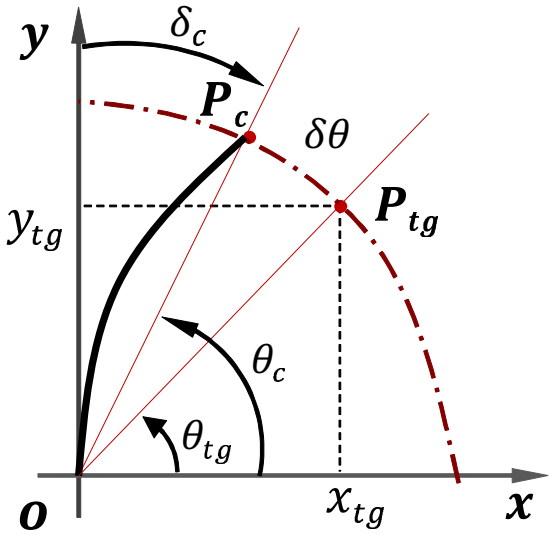


Figure 3.2: **Problem 4** formulation

In Fig. 3.2, a target position $\boldsymbol{P_{tg}}$ is given and a circular line of equal radius ρ_f by the distance between **O** and $\boldsymbol{P_f}$, i.e. $\rho_f = \sqrt{x_f^2 + y_f^2}$. A clothoid is formulated and generated to have its end point $\boldsymbol{P_c}$.

For the end point of the clothoid to reach at the target point $\boldsymbol{P_{tg}}$, two geometric conditions should be satisfied where at first, the radial distance is satisfied by regulating δ_f then secondly, lateral distance by regulating the arc portion $\Delta\theta_c$[2].

[2]Two parameters are treated independently as the *Polar* coordinate axes.

Since only a clothoid segment has a limited covering range for its end point to reach at a given target position, the non-elementary clothoid can include additional arc segment to satisfy the target boundary configuration. Two kinds of arc addition method are provided where one is *deflection-keeping* and the other is *length-keeping* for the combined curve. Each method can be useful for different purpose according to the problem condition. Under the given κ_i, κ_f, the *deflection-keeping* method retains the deflection of the resultant clothoid and the *length-keeping* retains its curve length. The arc length in the both methods makes change on the end position of the generated curve, thus the arc length can be also a parameter in the algorithmic procedure. Two methods are named as arc-A and arc-B for *deflection-keeping* and *length-keeping* purpose and described as **Algorithm 6** and **7** respectively.

Algorithm 6 arc-A

$\delta_c = \delta_f - \Delta\theta$
$[x_c, y_c] = \text{clothoid\_gen}(\delta_c, \kappa_f)$ ▷ $\widehat{C_c}$
$\alpha_1 = \frac{\kappa_f^2}{2\delta_c}$ ▷ Eq.(1.14)
$\delta_2 = \frac{\kappa_i^2}{2\alpha_1}$ ▷ Eq.(1.12)
$[x_2, y_2] = \text{clothoid\_gen}(\delta_2, \kappa_i)$ ▷ $\widehat{C_2}$
$\widehat{C_c} \ominus \widehat{C_2}$

Algorithm 7 arc-B

$\delta_c = \frac{\kappa_f^2}{2\alpha}$ ($s_0 = \frac{2\delta_f}{\kappa_f}$, $s_\ell = \kappa_f \Delta\theta$, $\alpha = \frac{\kappa_f - \kappa_i}{s_0 - s_\ell}$)
$[x_c, y_c] = \text{clothoid\_gen}(\delta_c, \kappa_2)$ ▷ $\widehat{C_c}$
$\alpha_1 = \frac{\kappa_f^2}{2\delta_c}$ ▷ Eq.(1.14)
$\delta_2 = \frac{\kappa_i^2}{2\alpha_1}$ ▷ Eq.(1.12)
$[x_2, y_2] = \text{clothoid\_gen}(\delta_2, \kappa_i)$ ▷ $\widehat{C_2}$
$\widehat{C_c} \ominus \widehat{C_2}$

Under the assumptions and description above, two subcases for **Problem 4** are provided as Tab. 4.1.

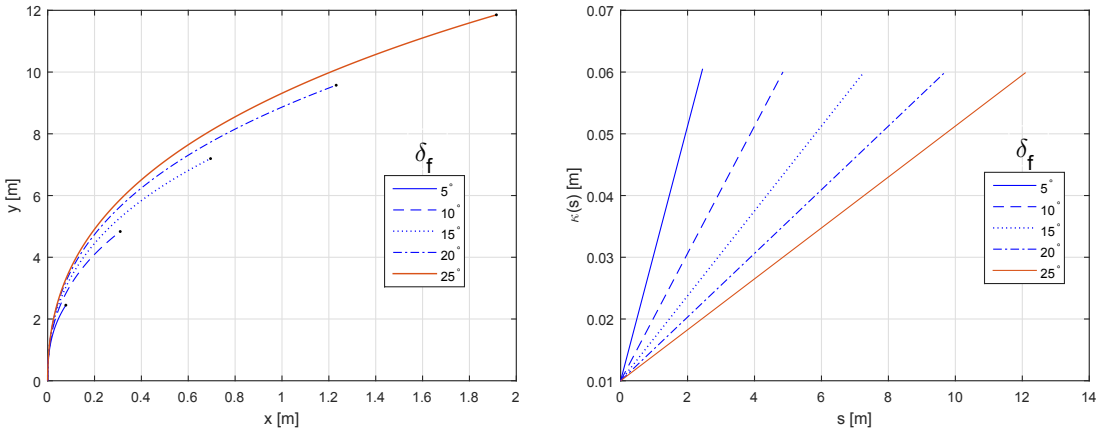
Table 3.1: Subcases of **Problem 4**

Problem case	given	regulating param.	arc addition	property
Case 4A	κ_f	$\delta_f,\ \Delta\theta$	arc-A	const. δ
Case 4B	δ_f	$\kappa_f,\ \Delta\theta$	arc-B	const. s

3.2.0.1 Case 4A ($\widetilde{\theta_f}$)

In this subcase, the θ_f is not fixed while all the information in $\mathbf{P_i}$ and x_f, y_f, κ_f are given, thus the θ_f should be obtained or determined from algorithmic solution. To satisfy these boundary configurations, clothoid parameters are regulated. Since one clothoid is used, there is little room for regulating the clothoid, especially δ and line/arc segment are only available.

To move the end position of the clothoid which is the result of Eq.(3.2), δ_f could be a regulating parameter since δ_i is fixed due to the relation that $\alpha = \frac{\kappa_f^2}{2\delta_f}$ makes $\delta_i = \frac{\kappa_i^2}{2\alpha}$, i.e., If δ_f is determined, then all the parameters become fixed and the clothoid is frozen.



(a) Path in the $x-y$ coord. (b) Curvature diagram

Figure 3.3: Clothoid representation by δ_f variation

Fig. 3.3 depicts the clothoid variation results for the fixed $\kappa_i = 0.01$ and $\kappa_f = 0.06$ while δ_f varies from 5° to 25°. As shown in Fig. 3.3, the path expands for its end position to move outward and curvature has increasing slope as δ_f increases.

For the formulated clothoid to meet the target $\boldsymbol{P_{tg}}$, two iterative procedures are performed until the convergence criteria is satisfied. Specifically, δ_f is varied to make $\boldsymbol{P_c}$ converge to the target radius ρ_f, and then the following convergence criteria between $\boldsymbol{P_c}$ and $\boldsymbol{P_f}$ are checked to make the second iterative procedure.

Criteria 1 (iteration #1) *find a feasible δ_f which makes $\boldsymbol{P_c}$ be close to the circular boundary ρ_{tg} within a fixed threshold by the following rule,*

$$\delta_f = \delta_f + |\partial\delta| \cdot sgn(\mathbf{D}_\delta), \tag{3.3a}$$

$$with\ \mathbf{D}_\delta = \rho_c - \rho_{tg}, \tag{3.3b}$$

$$\partial\delta = \begin{cases} \frac{\partial\delta}{2}, & if\ \mathbf{D}_\delta \cdot \mathbf{D}_\delta^0 < 0, \\ \partial\delta, & otherwise, \end{cases} \tag{3.3c}$$

where $sgn(\cdot)$ is the sign function for positive and negative value, $\mathbf{D}_\delta$ is the distance error to determine δ_f adjustment as well as algorithmic convergence (like $\boldsymbol{D_e}$ in **Algorithm 1** (p.43)) and $\mathbf{D}_\delta^0$ is the $\mathbf{D}_\delta$ at the previous iteration.

Criteria 2 (iteration #2) *find a feasible* $\Delta\theta$ *which makes* θ_c *be close to the circular boundary* θ_{tg} *within a fixed threshold by the following rule,*

$$\Delta\theta = \Delta\theta + |\partial\theta| \cdot sgn(\mathbf{D}_\theta), \tag{3.4a}$$

$$with\ \mathbf{D}_\theta = \theta_c - \theta_{tg}, \tag{3.4b}$$

$$\partial\theta = \begin{cases} \frac{\partial\theta}{2}, & if\ \mathbf{D}_\theta \cdot \mathbf{D}_\theta^0 < 0, \\ \partial\theta, & otherwise \end{cases} \tag{3.4c}$$

where $\mathbf{D}_\theta$ *is the circular distance error to determine* $\Delta\theta$ *adjustment as well as algorithmic convergence (also like* $\boldsymbol{D_e}$ *in* **Algorithm 1** (p.43)) and $\mathbf{D}_\theta^0$ is the $\mathbf{D}_\theta$ at the previous iteration.

The convergence criteria for the iteration procedures are defined by Eq.(3.5).

$$\mathbf{D}_\delta < \varepsilon_\delta, \tag{3.5a}$$

$$\mathbf{D}_\theta < \varepsilon_\theta, \tag{3.5b}$$

where 0.05 m and 1° could be applied to ε_δ and ε_θ respectively for faster convergence with allowable accuracy. After the condition related to δ_f is satisfied, then the lateral distance should be minimized by adding arc using the method of *arc-A*. The geometric effect of *arc-A* method depends on the relation of the position between $\boldsymbol{P_f}$ and $\boldsymbol{P_c}$. For increasing curvature clothoid, if $\boldsymbol{P_f}$ is located in the upper side of $\boldsymbol{P_c}$, then arc should attach to the larger curvature side, and vice versa. For decreasing curvature clothoid, if $\boldsymbol{P_f}$ is located in the upper side of $\boldsymbol{P_c}$, then arc should be attached to the smaller curvature side, and vice versa.

Algorithm 8 arc-Ah

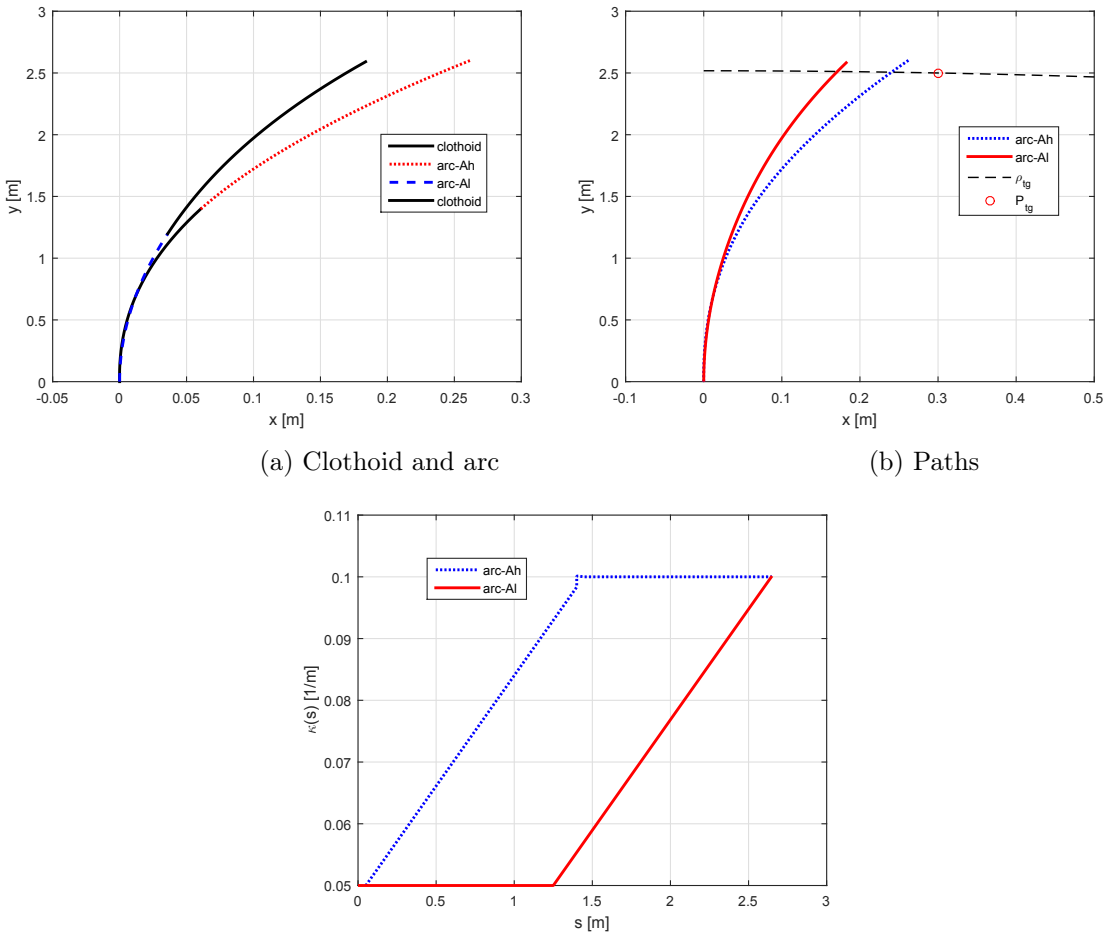
$\rho_i = \frac{1}{\kappa_i},\ \rho_f = \frac{1}{\kappa_f}$
$\delta_c = \delta_f - \Delta\theta$
`[`x_c`,` y_c`] = clothoid_gen(`δ_c`,` κ_f`)` ▷ $\widehat{C_c}$
$\alpha_1 = \frac{\kappa_f^2}{2\delta_c}$ ▷ Eq.(1.14)
$\delta_2 = \frac{\kappa_i^2}{2\alpha_1}$ ▷ Eq.(1.12)
`[`x_2`,` y_2`] = clothoid_gen(`δ_2`,` κ_i`)` ▷ $\widehat{C_2}$
$\widehat{C_c} \ominus \widehat{C_2}$ ▷ $\widehat{C_a}$
`[`x_r`,` y_r`] = arc_gen(`ρ_f`,` $\Delta\theta$`)` ▷ **Appx. B.3**
$\widehat{C_a} \oplus \widehat{C_r}$ ▷ Obtained curve

Algorithm 9 arc-Aℓ

$\rho_i = \frac{1}{\kappa_i},\ \rho_f = \frac{1}{\kappa_f}$
$\delta_c = \delta_f - \Delta\theta$
`[`x_c`,` y_c`] = clothoid_gen(`δ_c`,` κ_f`)` ▷ $\widehat{C_c}$
$\alpha_1 = \frac{\kappa_f^2}{2\delta_c}$ ▷ Eq.(1.14)
$\delta_2 = \frac{\kappa_i^2}{2\alpha_1}$ ▷ Eq.(1.12)
`[`x_2`,` y_2`] = clothoid_gen(`δ_2`,` κ_i`)` ▷ $\widehat{C_2}$
$\widehat{C_c} \ominus \widehat{C_2}$ ▷ $\widehat{C_a}$
`[`x_r`,` y_r`] = arc_gen(`ρ_i`,` $\Delta\theta$`)` ▷ **Appx. B.3**
$\widehat{C_r} \oplus \widehat{C_c}$ ▷ Obtained curve

Algorithm 8 is for the arc addition at the high curvature (arc-Ah) and the **Algorithm 9** is at the low curvature (arc-Aℓ). In **Algorithm 8** and **9**, the

function [x_r, y_r] = arc_gen(ρ, θ) generates an arc segment (x_r, y_r) of which radius is ρ with θ as the angle of the circumference. By the operator $\oplus$, two curve segments are connected with orientation conformity.



(a) Clothoid and arc

(b) Paths

(c) Curvature diagram

Figure 3.4: Combined curves with a clothoid and an arc

Fig. 3.4 illustrates the combination of a non-elementary clothoid and an arc by the method *arc-A* where the boundary condition is given as $\boldsymbol{P_i}(0,\ 0,\ \theta_i = \frac{\pi}{2}, \kappa_i = 0.05)$ and $\boldsymbol{P_f}(0.3,\ 2.5,\ \kappa_f = 0.1)$. In this example, it is set as $\boldsymbol{P_{tg}}(0.3, 2.5)$ and thus $\rho_{tg} = 2.518$ with $\delta_f = 15°$ and $\Delta\theta = 7°$. In Fig. 3.4(a) and (b), two different results for *arc-A* as curvature-high arc-Ah and curvature-low arc-Aℓ are compared under the same other conditions. It is shown that the effect of arc-Ah makes the end point of the combined curve to move positively in *x-axis* while arc-Ah pushes

the end point to move negatively in *x-axis*. Fig. 3.4(c) plots the corresponding curvature diagram which discriminates between arc-Ah and arc-Aℓ.

Thus, the iterative procedure include the following rule for $\Delta\theta$ variation as,

$$\text{arc-A} = \begin{cases} \text{arc-A}h \text{ with } \partial\theta, & \text{if } \mathbf{D}_\theta > 0, \\ \text{arc-A}\ell \text{ with } \partial\theta, & \text{else if } \mathbf{D}_\theta < 0, \end{cases} \tag{3.6}$$

Using the properties of clothoid parameters and criteria for iterative convergence, the solution for the defined problem is obtained by the following **Algorithm 10**.

Algorithm 10 Case 4A

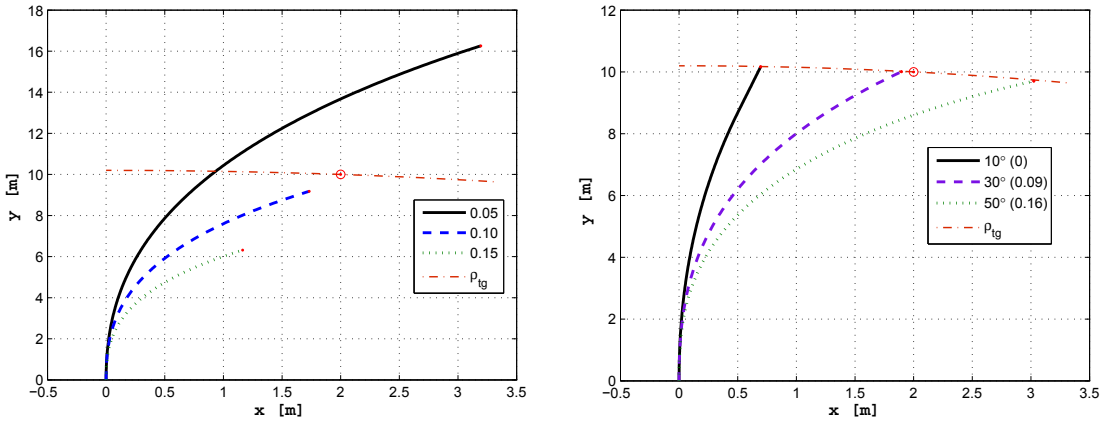
Require: δ_f ▷ initial assumption

```
procedure CLOTHOID1A(P_tg, κ_i, κ_f)
    α = κ_f^2 / (2δ_f)                          ▷ Eq. (1.14)
    δ_i = κ_i^2 / (2α)
    while D_θ < ε_θ do                          ▷ Criteria 2
        while D_δ < ε_δ do                      ▷ Criteria 1
            iteration # 1
        end while
        iteration # 2
    end while
    return δ_f, θ_f                             ▷ Solution obtained
end procedure
```

Fig. 3.5 shows a test result for problems of having different target positions with the same circular boundary. By regulating δ_f as well as $\Delta\theta$, each path arrives at the target circular boundary ρ_{tg} from **Criteria 1** as 10°, 30° and 50°, but different δ_f makes the shape to be more inclined to the right side by $\Delta\theta$ variation as 0°, 5.157° and 9.167°.

3.2.0.2 Case 4B ($\widetilde{\kappa_f}$)

In this subcase, the κ_f is not fixed while other information of $\mathbf{P_i}$ (x_f, y_f, θ_f) are given, thus the κ_f should be obtained from algorithmic solution. Under the same formulation and methodology of **Case 4A**, two conditions of ρ_{tg} and θ_{tg} should be satisfied. At first, the condition of ρ_{tg} could be resolved as following criteria.



(a) δ_f variation (b) $\Delta\theta$ variation

Figure 3.5: δ_f and $\Delta\theta$ variation in the iterative procedures

Criteria 3 (iteration #3) *find a feasible* κ_f *which makes* $\boldsymbol{P_c}$ *be close to the circular boundary* ρ_{tg} *within a fixed threshold by the following rule,*

$$\kappa_f = \kappa_f + |\partial\kappa| \cdot sgn(\mathbf{D}_\kappa), \tag{3.7a}$$

$$\mathbf{D}_\kappa = \rho_c - \rho_{tg}, \tag{3.7b}$$

$$\partial\kappa = \begin{cases} \frac{\partial\kappa}{2}, & if\ \mathbf{D}_\kappa \cdot \mathbf{D}_\kappa^0 < 0, \\ \partial\kappa, & otherwise \end{cases} \tag{3.7c}$$

where $\mathbf{D}_\kappa^0$ *is the* $\mathbf{D}_\kappa$ *at the previous iteration.*

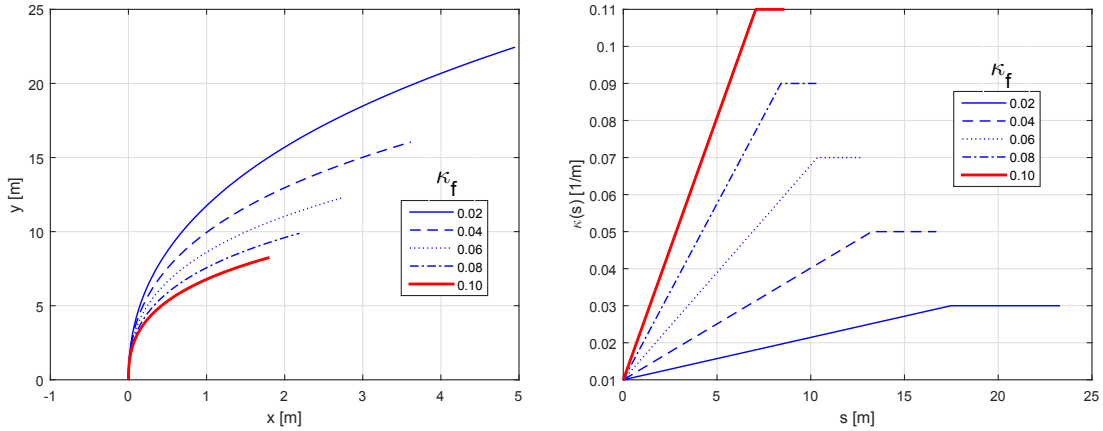
Fig. 3.6 depicts the clothoid variation results for the fixed $\kappa_i = 0.01$ and $\delta_f = 30°$ with $\Delta\theta = 10°$ while κ_f varies from 0.02 to 0.1 by 0.02 interval. As shown in Fig. 3.6, the path expands for its end position to move outward and curvature has increasing slope as κ_f increases.

The convergence criteria for the iteration procedures are defined by Eq. (3.8).

$$\mathbf{D}_\kappa < \varepsilon_\kappa, \tag{3.8a}$$

$$\mathbf{D}_\theta < \varepsilon_\theta, \tag{3.8b}$$

After the condition regarding κ_f is satisfied, then the lateral distance should be minimized by adding arc using the method of *arc-B*.



(a) Path in the $x-y$ coord. (b) Curvature diagram

Figure 3.6: Clothoid representation according to κ_f variation

The geometric effect of *arc-B* method depends on the relation of the position between $\boldsymbol{P_{tg}}$ and $\boldsymbol{P_c}$. For increasing curvature clothoid, if $\boldsymbol{P_{tg}}$ is located in the upper side of $\boldsymbol{P_c}$, then arc should attach to the larger curvature side, and vice versa. For decreasing curvature clothoid, if $\boldsymbol{P_{tg}}$ is located in the upper side of $\boldsymbol{P_c}$, then the arc should be attached to the smaller curvature side, and vice versa.

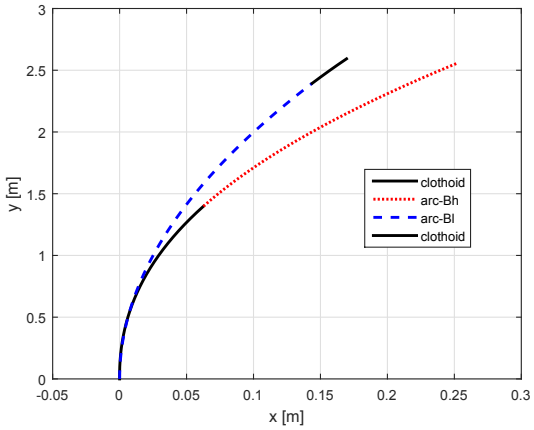
Algorithm 11 arc-Bh

$\rho_i = \frac{1}{\kappa_i},\ \rho_f = \frac{1}{\kappa_f}$
$s_f = \frac{2\delta_f}{\kappa_f}$
$\alpha = \frac{\kappa_f^2}{2\delta_f}$ ▷ Eq.(1.14)
$s_i = \frac{\kappa_i}{\alpha}$
$s_c = s_f - s_i$
$s_r = \rho_f \Delta\theta$ ▷ Arc length
$\alpha_c = \frac{\kappa_f - \kappa_i}{s_c - s_r}$
$\delta_c = \frac{\kappa_f^2}{2\alpha_c}$
$[x_c,\ y_c]$ = `clothoid_gen`$(\delta_c,\ \kappa_f)$ ▷ $\widehat{C_c}$
$\alpha_1 = \frac{\kappa_f^2}{2\delta_c}$ ▷ Eq.(1.14)
$\delta_2 = \frac{\kappa_i^2}{2\alpha_1}$ ▷ Eq.(1.12)
$[x_2,\ y_2]$ = `clothoid_gen`$(\delta_2,\ \kappa_i)$ ▷ $\widehat{C_2}$
$\widehat{C_c} \ominus \widehat{C_2}$ ▷ $\widehat{C_a}$
$[x_r,\ y_r]$ = `arc_gen`$(\rho_f,\ \Delta\theta)$ ▷ **Appx. B.3**
$\widehat{C_b} \oplus \widehat{C_r}$ ▷ Obtained curve

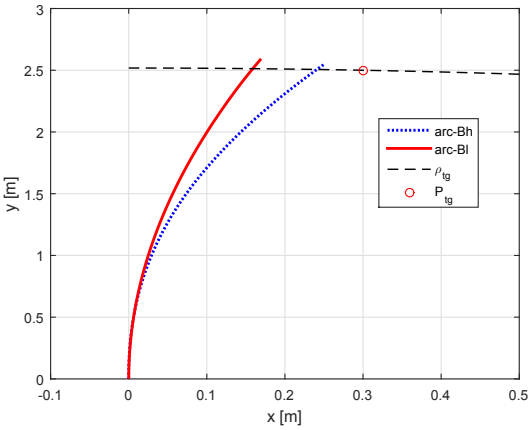
Algorithm 12 arc-Bℓ

$\rho_i = \frac{1}{\kappa_i},\ \rho_f = \frac{1}{\kappa_f}$
$s_f = \frac{2\delta_f}{\kappa_f}$
$\alpha = \frac{\kappa_f^2}{2\delta_f}$ ▷ Eq.(1.14)
$s_i = \frac{\kappa_i}{\alpha}$
$s_c = s_f - s_i$
$s_r = \rho_i \Delta\theta$ ▷ Arc length
$\alpha_c = \frac{\kappa_f - \kappa_i}{s_c - s_r}$
$\delta_c = \frac{\kappa_f^2}{2\alpha_c}$
$[x_c,\ y_c]$ = `clothoid_gen`$(\delta_c,\ \kappa_f)$ ▷ $\widehat{C_c}$
$\alpha_1 = \frac{\kappa_f^2}{2\delta_c}$ ▷ Eq.(1.14)
$\delta_2 = \frac{\kappa_i^2}{2\alpha_1}$ ▷ Eq.(1.12)
$[x_2,\ y_2]$ = `clothoid_gen`$(\delta_2,\ \kappa_i)$ ▷ $\widehat{C_2}$
$\widehat{C_c} \ominus \widehat{C_2}$ ▷ $\widehat{C_a}$
$[x_r,\ y_r]$ = `arc_gen`$(\rho_i,\ \Delta\theta)$ ▷ **Appx. B.3**
$\widehat{C_r} \oplus \widehat{C_b}$ ▷ Obtained curve

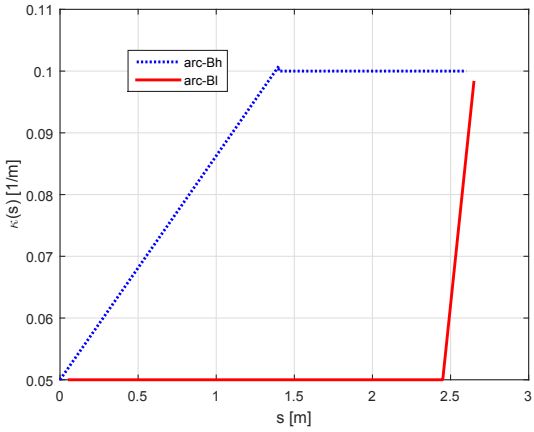
Algorithm 11 is for the arc addition at the high curvature (arc-Bh) and **Algorithm 12** is at the low curvature (arc-Bℓ).



(a) Clothoid and arc



(b) Paths



(c) Curvature diagram

Figure 3.7: Combined curves with a clothoid and an arc

Fig. 3.7 illustrates the combination of a non-elementary clothoid and an arc by the method *arc-B* where the boundary condition is given as $\boldsymbol{P_i}(0,\, 0,\, \theta_i = \frac{\pi}{2}, \kappa_i = 0.05)$ and $\boldsymbol{P_f}(0.3,\, 2.5,\, \delta_f = 15^\circ)$. In this example, it is set as $\boldsymbol{P_{tg}}(0.3, 2.5)$ and thus $\rho_{tg} = 2.518$ with $\kappa_f = 0.1$ and $\Delta\theta = 7^\circ$. In Fig. 3.7(a) and (b), two different results for *arc-B* as curvature-high arc-Bh and curvature-low arc-Bℓ are compared under the same other conditions. It is shown that the effect of arc-Bh makes the end point of the combined curve to move positively in *x-axis* while arc-Bh pushes the end point to move negatively in *x-axis*. Fig. 3.7(c) plots the corresponding curvature diagram which discriminates between arc-Bh and arc-Bℓ.

Thus, the iterative procedure include the following rule for $\Delta\theta$ variation as,

$$\text{arc-B} = \begin{cases} \text{arc-B}h \;\; \text{with} \;\; \partial\theta, & \text{if } \mathbf{D}_\theta > 0, \\ \text{arc-B}\ell \;\; \text{with} \;\; \partial\theta, & \text{else if } \mathbf{D}_\theta < 0. \end{cases} \tag{3.9}$$

Different to arc-A method, arc-B method makes the total length of the generated curve to be constant by $\Delta\theta$ variation (cf. Table. 3.2). This property means that $\boldsymbol{D_\theta}$ iteration can be independently performed after $\boldsymbol{D_\kappa}$ iteration procedure. From the relation between curvature and deflection along the length, following function can be derived to apply in the solution procedure.

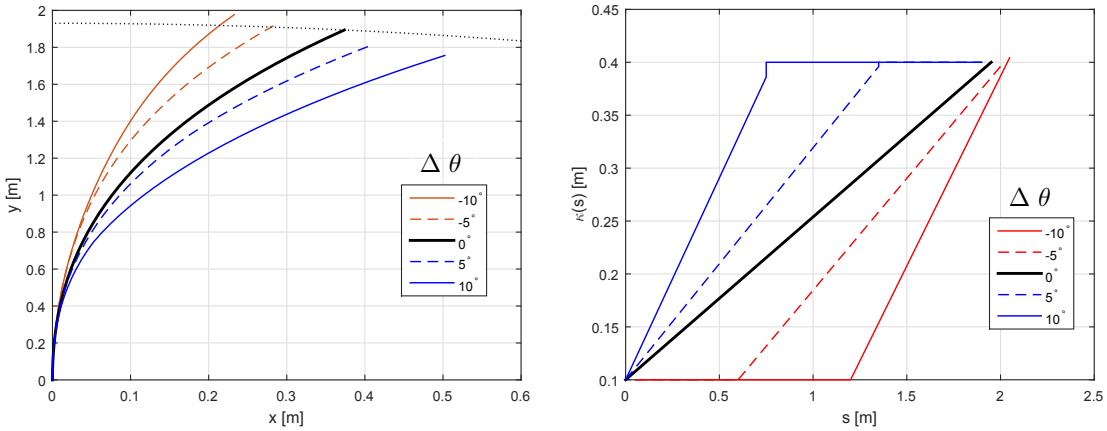
$$\Delta\theta = \begin{cases} \frac{D_\theta 2\kappa_f}{\kappa_f - \kappa_i} & \text{for} \quad \kappa_i < \kappa_f, \\ \frac{D_\theta 2\kappa_i}{\kappa_i - \kappa_f} & \text{for} \quad \kappa_i > \kappa_f. \end{cases} \tag{3.10}$$

By using Eq. (3.10), $\Delta\theta$ can be definitely obtained from the given information. Using the properties of clothoid parameters and criteria for iterative convergence, the solution for the defined problem is obtained by the following **Algorithm 13**.

Algorithm 13 case4B

Require: κ_f ▷ Initial assumption

1: **procedure** CLOTHOID1B($\boldsymbol{P_{tg}}, \kappa_i, \delta_f$)
2: $\quad \alpha = \frac{\kappa_f^2}{2\delta_f}$ ▷ Eq.(1.12)
3: $\quad \delta_i = \frac{\kappa_i^2}{2\alpha}$ ▷ Eq.(1.14)
4: $\quad$ **while** $\boldsymbol{D}_\kappa > \varepsilon_\kappa$ **do** ▷ **Criteria 3**
5: $\quad\quad$ **Iteration # 3**
6: $\quad$ **end while**
7: $\quad \Delta\theta$ ▷ Eq. (3.10)
8: $\quad$ **return** κ_f, δ_f ▷ Solution obtained
9: **end procedure**



(a) Path in the $x-y$ coord. (b) Curvature diagram

Figure 3.8: Clothoid representation by κ_f variation

Fig. 3.8 illustrates results of $\Delta\theta$ variation, when $\kappa_i = 0.1$, $\kappa_f = 0.4$ and $\delta_f = 30°$. If it is assumed that κ_f satisfies $\boldsymbol{D_\kappa}$ condition of Eq. (3.7a), as $\Delta\theta$ varies from $-10°$ to $10°$ by $5°$ difference, the end point of the combined curve moves righward with almost same length, where this property could be more clearly seen in the correnponding curvature diagram of Fig. 3.8(b).

3.3 Path replanning for dynamic obstacle

When a path is given, the vehicle could follow it by steering control while observing the vehicle pose and target pose along the given referenced path. However, when a dynamic obstacle crosses the initial reference path, a collision between the vehicle and the obstacle could be expected. In that case, (if the vehicle speed is constant) the path is required to be replanned to avoid a collision with this dynamic obstacle.[3] In this section, path replanning method using the proposed **d-pCCP** is described.

3.3.1 Introduction

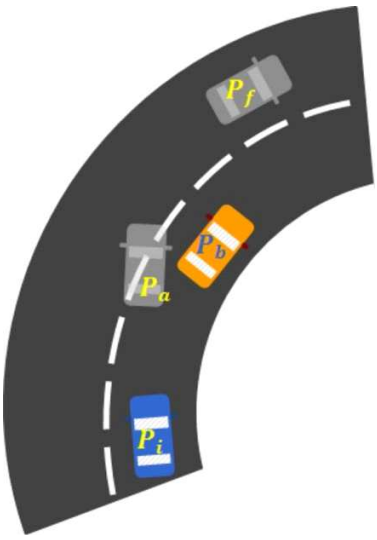
Two examples in traditional roads are considered here, where one has curved and the other has straight lanes. In those roads, a vehicle would take over an obstacle (vehicle) ahead of it. However, when the obstacle changes its direction or position by changing its lane or speed, the planned path could be dangerous for the risk of collision. To cope with such condition or environment, replanning the previous path while maintaining the curvature continuity (or steering smoothness) is required.

Fig. 3.9 depicts two representative cases that require the path replanning due to dynamic obstacle. In Fig. 3.9(a), a vehicle has a generated path which passes through $\mathbf{P_i}$, $\mathbf{P_a}$ and $\mathbf{P_f}$, where the path is to change the lane for a static (or steady) obstacle[4] of the pose $\mathbf{P_b}$ by overtaking maneuver. However, as shown in Fig. 3.9(b), when the obstacle changes its speed and/or orientation, thus the position of the obstacle moves to $\mathbf{P_b'}$ at the vehicle position $\mathbf{P_i'}$, then a collision between two vehicles occurs by $\mathbf{P_a}$ and $\mathbf{P_b'}$. Similar case can be found when a vehicle changes its lane in the straight road as shown in Fig. 3.9(c), where the vehicle follows a path passing through $\mathbf{P_i}$, $\mathbf{P_a}$ to $\mathbf{P_f}$. However, when the vehicle arrives at the pose $\mathbf{P_i'}$ as shown in Fig. 3.9(b), the position of the obstacle has changed from $\mathbf{P_b}$ to $\mathbf{P_b'}$, thus it is expected that both vehicles are collided by $\mathbf{P_a}$ and $\mathbf{P_b'}$. To avoid this collision, it needs to replan a new path from the pose $\mathbf{P_i'}$.

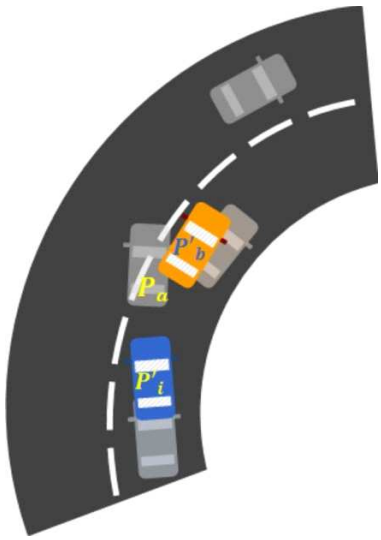
In the described two cases, it is important to consider non-zero curvature in boundary condition, (or non-zero steering wheel angle) during the path replanning procedure. The new path should satisfy the curvature continuity constraint as well as collision safety at the target pose. Here, the path replanning is performed

[3] If the collision is unavoidable with current speed and environment, the only way to reduce the risk of collision is to stop the vehicle [Fraichard 03].

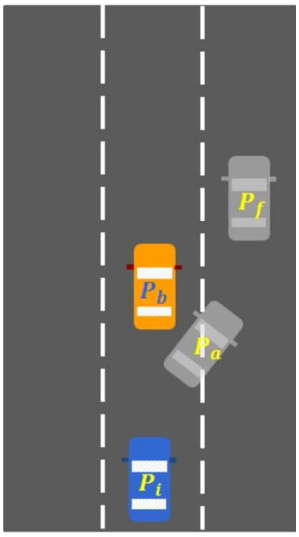
[4] With the constant speed of the both vehicles, the condition of the environment can be assumed to be static.



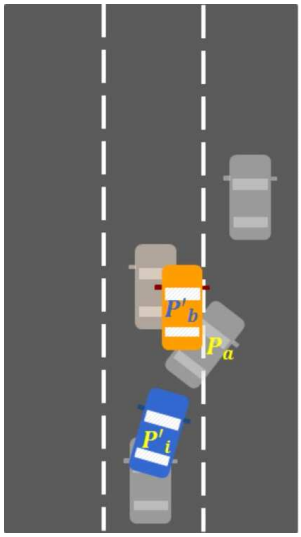
(a) Path for curved road



(b) Collision during curved lane change



(c) Path for straight road



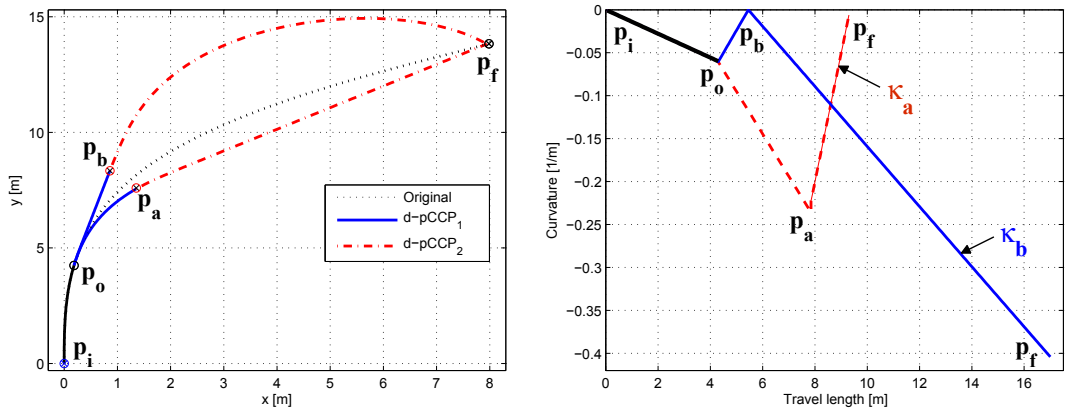
(d) Collision during straight lane change

Figure 3.9: Path replanning examples of dynamic obstacle avoidance

by finding clothoid parameters which satisfies initial and final pose with curvature continuity constraint which was described in **Section 3.1.1**.

3.3.2 Demonstrative simulations for path replanning

The proposed **d-pCCP** for path replanning is applied for two demonstrative examples such as lane change maneuver in the curved and straight road (cf. Fig. 3.9).



(a) Replanned paths (b) Curvature diagrams

Figure 3.10: Path replanning in the curved road

Fig. 3.10 presents the first example in the curved road (cf. Fig. 3.9(a)). In this example, two different routes are tried to apply and validate the effectiveness of the proposed **d-pCCP**. The original path is assumed to be given from the previously defined problem where $\mathbf{P_i}(0, 0, \frac{\pi}{2}, 0°)$ and $\mathbf{P_f}(8, 14, \frac{\pi}{6}, 0°)$ and **d-pCCP** problem is constructed by its initial $\mathbf{P_0}(0.18, 4.24, 0.06, 82.46°)$ to the final position $\mathbf{p_f}(8, 14)$. (Note that in the **d-pCCP** problem, the final configuration is not fully defined to obtain the feasible path.)

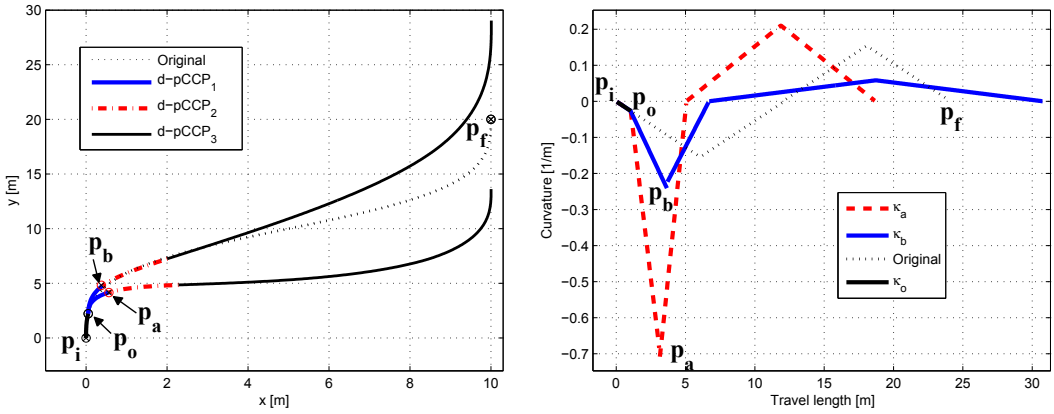
From this vehicle pose $\mathbf{p_0}$, path replanning is performed to avoid the expected collision. Thus, a new way point is given manually[5] It is considered that the obstacle vehicle maintains the constant speed and the vehicle should be faster than the obstacle to perform the overtaking maneuver.[6]

[5]The method to determine the way point to avoid the collision with the obstacle is described in **Section 3.3**.

[6]The velocity planning is further addressed in **Section 3.3**.

The proposed algorithm of the **d-pCCP** provides a solution path (Replanned $\mathbf{d-pCCP_1}$) to reach safely the new target position $\mathbf{p_a}$ (or $\mathbf{p_b}$), and then the second path (Replanned $\mathbf{d-pCCP_2}$) which connects the final position of the original path is generated in the same way.

Fig. 3.10(b) shows the corresponding curvature diagrams of the obtained paths for two subcases. One can observe that the path for way point $\mathbf{p_b}$, i.e., $\kappa_\mathbf{b}$ requires the longer travel distance than the path for way point $\mathbf{p_a}$ and $\kappa_\mathbf{a}$.



(a) Replanned paths (b) Curvature diagrams (κ)

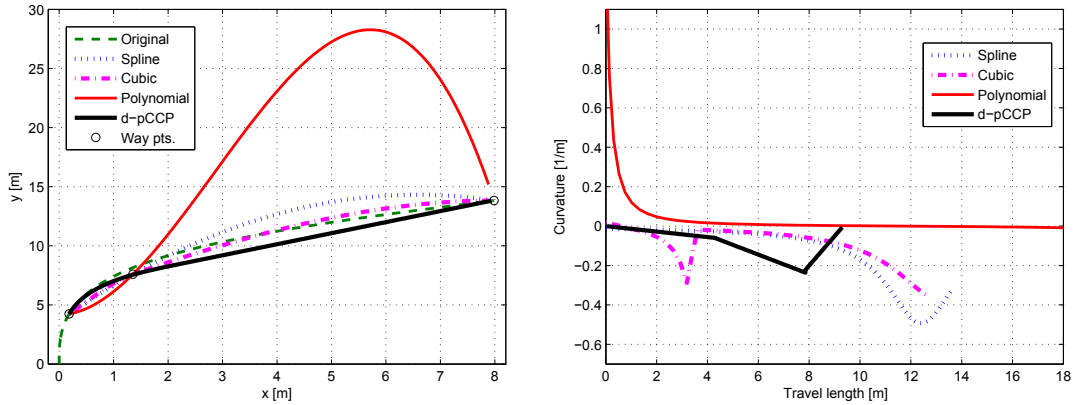
Figure 3.11: Path replanning for the straight road

Fig. 3.11(a) presents the replanned paths for the straight road. Two subcases are also carried out by two different way points. Fig. 3.11(b) plots the corresponding curvature diagrams for the obtained paths for two subcases. It could be also observed that the path for way point $\mathbf{p_b}$ (or $\kappa_\mathbf{b}$) requires the longer travel distance. Note that the path for the straight lane change requires four clothoids to change to another lane, and 3^{rd} path is generated by the same method of the curved lane change maneuver case. (Its detailed description is skipped to avoid further complexity.) As shown in Fig. 3.10(b) and Fig. 3.11(b), all the obtained paths strictly keep the curvature continuity with positional accuracy at the way point or target.

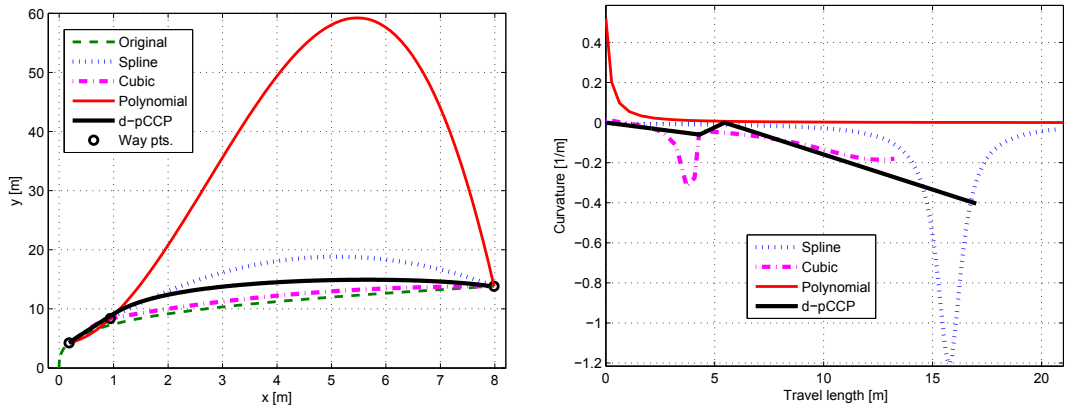
After obtaining the solution paths, it is compared with other methods such as cubic, spline and polynomial interpolation based paths to verify the efficiency and advantages of the proposed **d-pCCP** for dynamic path replanning.

To analyze and compare the obtained **d-pCCP** with other smooth path generation methods, two examples that having the way points of $\mathbf{p_a}$ and $\mathbf{p_b}$ for the

curved road are applied to those methods with the same initial and final configurations.



(a) Obtained $\mathbf{d-pCCP_a}$ and $\boldsymbol{\kappa_a}$

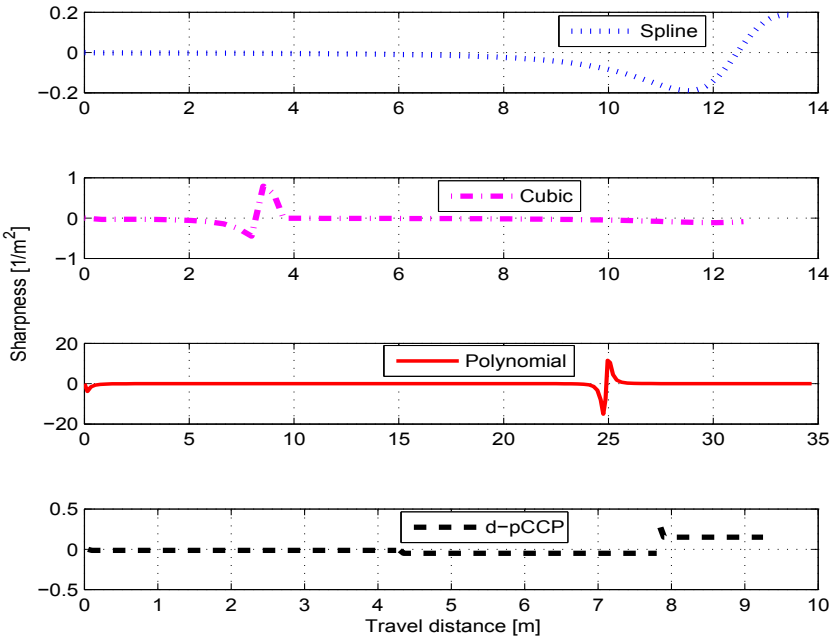


(b) Obtained $\mathbf{d-pCCP_b}$ and $\boldsymbol{\kappa_b}$

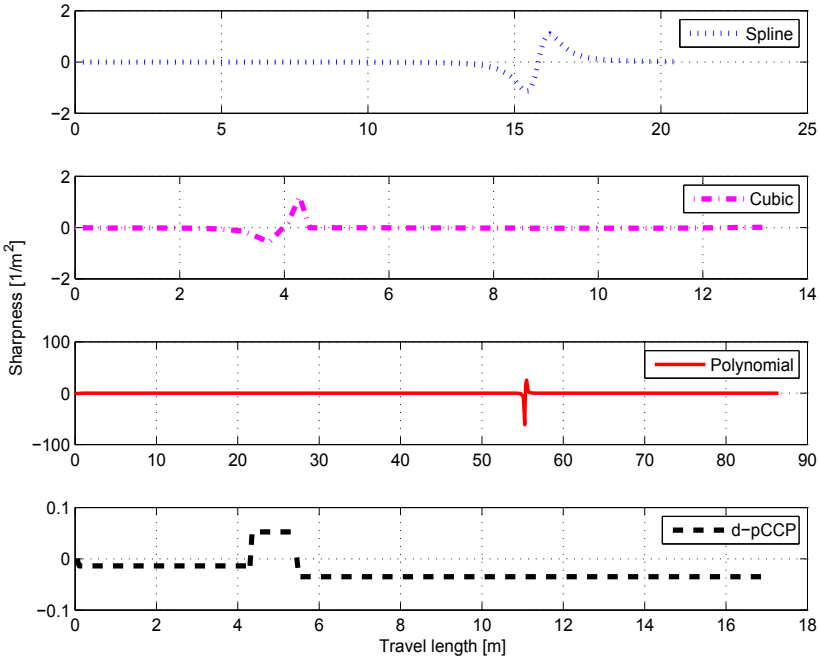
Figure 3.12: Comparisons with other smooth path generation methods

Fig. 3.12 includes two paired results of the paths and the corresponding curvature diagrams for $\mathbf{p_a}$ and $\mathbf{p_a}$ paths. Under the same method with same conditions as the input, three other methods such as *Spline* (*Piecewise Cubic Spline*), *Cubic* (*Shape-preserving Piecewise Cubic Spline*) and *Polynomials* (5^{th} *order polynomials*) with **d-pCCP**. As one can see, even if obtained paths of the other methods are smooth in the *Cartesian* coordinate, but their curvatures are not fully continuous.

In Fig. 3.13, the sharpness diagrams of the proposed **d-pCCP** and others are compared, where Fig. 3.13(a) is for the $\mathbf{d-pCCP_a}$ and Fig. 3.13(b) is for the $\mathbf{d-pCCP_b}$. In each figure, note that all sharpness are different in scale and fluc-



(a) Obtained $\mathbf{d-pCCP_a}$ with others



(b) Obtained $\mathbf{d-pCCP_b}$ with others

Figure 3.13: Sharpness comparison with other methods

tuation. As expected from the known curvature diagrams, other methods bring about remarkable jump or drastic change compared to the proposed **d-pCCP**, which verifies that the proposed solution could be more effective or convenient to follow while accomplishing the path replanning task.[7] In the following, qualitative measures for the obtained results are analyzed.

Table 3.2: Qualitative comparison for $\mathbf{p_a}$ path ($[rad], [m]$)

Method	κ_{min}	κ_{max}	α_{min}	α_{max}	**s**	$\mathbf{d_{err}}$
Spline	-0.4345	0.0150	-0.1477	**0.1286**	13.3509	1.1104
Cubic	-0.3798	0.0169	-0.3611	0.9408	12.7276	**0.4037**
Poly.	-3.7128	1.5038	-11.3844	9.9302	32.1035	11.8864
d-pCCP	**-0.2584**	**-0.0007**	**-0.0495**	0.2943	**8.9000**	0.8523

Table 3.3: Qualitative comparison for $\mathbf{p_b}$ path ($[rad], [m]$)

Method	κ_{min}	κ_{max}	α_{min}	α_{max}	**s**	$\mathbf{d_{err}}$
Spline	-1.2019	-0.0057	-61.5252	25.2804	19.5563	5.0498
Cubic	**-0.3193**	0.0162	-0.5709	1.2043	**13.2962**	**0.7767**
Poly.	-8.5245	0.5201	-1.1329	1.0943	78.2928	37.2688
d-pCCP	-0.4041	**0**	**-0.0350**	**0.0526**	18.00	2.3190

Table. 3.2 and 3.3 summarize the qualitative measures about the obtained results for $\mathbf{d-pCCP_a}$ and $\mathbf{d-pCCP_b}$ respectively, where $\boldsymbol{s}$ is the travel distance and $\mathbf{d_{err}}$ is the summation of the deviated distances between the considered path and the original path. As the maximum (or minimum) curvature κ_{max} (κ_{min}) for a path is higher, the more lateral acceleration is affected to driver/passenger by the relation of $\kappa \cdot v^2$ (v is the vehicle speed), thus it is unfavorable. Also for α, the higher α_{max} (or lower α_{min}) worsen the comfort by sudden changes of the steering angle or the vehicle orientation with higher lateral acceleration and jerk.[8] It is notable that even if the **Cubic** path looks efficient in the travel length

[7] The sharpness diagram for the given path could provide a useful information about its path following performance and passenger comfort.

[8] The effect of the path and human comfort is addressed in **Chapter 5**.

or deviations from the origianl path by the smallest $\mathbf{d_{err}}$ (for $\mathbf{p_a}$ path) or $\mathbf{s}$ (for $\mathbf{p_b}$ path), its drastic change in the curvature or larger sharpness could be unrealistic to be followed by nonholonomic vehicles. As validated in both Table. 3.2 and 3.3, the proposed **d-pCCP** shows the least values in overall measures, thus it is the most efficient or maximally comfortable path.

3.4 Velocity planning based on d-pCCP

3.4.1 Velocity planning in 4D configuration space

To manage safe navigation in a cluttered environment or complex road condition of interacting with dynamic obstacles, the velocity planning on the generated path is necessary. So far, **pCCP** (and **d-pCCP**) for car-like vehicles having inherent nonholonomic constraint has been addressed. In this section, the velocity planning of the generated **pCCP** is considered which generate a *trajectory*, i.e., the path defined through the specified time. Thus, the trajectory can be named as *parametric Continuous Curvature Trajectory* (**pCCT**).

Two methods can be applied in generating the **pCCT** for obstacle avoidance. The first one is replanning the (forward) path while checking the state of the surrounded obstacle like artifical potential field method [Khatib 86b, Giesbrecht 04] or velocity obstacle model approach [Fiorini 98, DenBerg 08, Wilkie 09]. However, such methods lack the curvature continuity in the generated path, for car-like vehicles it can be performed by **d-pCCP** under some optimized formulations of safety criteria.

As the second method, velocity planning along the given (or generated) path can be a solution for the obstacle avoidance. In this method, the original path is conserved and only velocity profile is modified to avoid collision with obstacle, thus an essential task for this method is to analyze future trajectory. This method is efficient in the sense that a dynamic path planning can be accomplished like the static environment.

According to the status of the environment and obstacle motion, the two methods described above (**d-pCCP** and velocity planning) can be fused or sequentially performed, i.e., path replanning and then velocity planning. Here, an important guidance of choosing the method is geometric feasibility to generate a new path, whether it is in collision/obstacle-free space or drivable road conditions[Gim 12] under current velocities of the vehicle and obstacle(s). By checking the feasibility to generate a new **d-pCCP**, one of the methods is selected or fused. The guidance or criteria about how to determine the fused method or how to fuse the

two methods are required to be dealt with in the future research topic. In this section, the velocity planning is focused on.

The procedure to select the method is described as shown in **Algorithm 14**.

Algorithm 14 Procedure to dynamic obstacle avoidance

1: An original path is given and an obstacle interrupts the given path at a certain time.
2: Under the current velocity conditions, two objects (vehicle and obstacle) are expected to collide with each other and obstacle avoidance maneuver for the vehicle should be necessary. ▷ Two circles model
3: Firstly, the final pose (or target pose) is determined to avoid the collision with the obstacle and **d-pCCP** is generated.
4: When the obtained path is not available by mechanical constraints (cf. **Appendix B.5**), **d-pCCP** is modified using the maximal mechanical constraints (α_{limit}, κ_{limit} in Eq.(5.3) of **Chapter 5**, p.166) and the final pose is also modified by replanned path.
5: Analyze the obtained path with the current velocities (obtained trajectory) and check the collision with the obstacle in time domain. ▷ 4D space analysis
6: If the collision is expected for both trajectories, the velocity planning for the velocity is performed to avoid the obstacle. ▷ Velocity planning

Specifically, between the vehicle and obstacle[9], after the **d-pCCP** is generated by considering an obstacle as stationary, then the velocity for safe maneuvering is planned while checking obstacle's velocity in every sampling time. This method separates path and velocity generation apart, and it can reduce the burden to compute all required procedures to obtain the solution in a limited sampling interval. Another possible benifit can be found in the aspect of the actual vehicle control system, that is, the steering actuation takes part in the path whereas the acceleration/deceleration takes charge the velocity planned. Both actuations are independently or parallely controlled to reduce the computational burden. However, this method has also a limitation that it can not cope with obstacle's orientation change. In this case, the given path should be replanned as described in **Section 3.2**.

The work presented in what follows focuses on velocity planning under given **d-pCCP**(or **pCCP**) with the consideration of dynamic obstacle avoidance. It is

[9]In the analysis of collision, only two objects are considered, where the one is the vehicle considered to generate a path and the other is an obstacle which occupies the path. The obstacle could be another car-like vehicle, bicycle, cart or person that have also the nonholonomic motion with velocity.

assumed that the vehicle has a predefined path and the local planner generates a flexible and smooth trajectory in the presence of any obstructing dynamic obstacle. The path should be sufficiently accurate and smooth to be followed by the car-like vehicle.

In the proposed velocity planning, a contributing point is introducing the concept of 4D configuration [Wu 11] into the **d-pCCP** to analyze the future vehicle's trajectory and determine the final avoidance pose with velocity set-points. Compared to the velocity obstacle model approach, the proposed model has the capability of analyzing all the expected collision cases for varying velocity of obstacles as well as non-straight trajectory. Resulting velocity planning with the given **d-pCCP** in 4D configuration (cf. **Algorithm 14** step 5) provides thus smooth trajectory even when dealing with dynamic obstacles.

In what follows, velocity planning on the given **pCCP** is addressed on 4D configuration, i.e., $x(t)$, $y(t)$, $z(t)$ and t ($x \in \mathbb{R}^3$, y $\in \mathbb{R}^3$, z $\in \mathbb{R}^3$ and t $\in \mathbb{R}$) for obstacle avoidance, where plane motion is considered to avoid conceptual complexity thus $z(t) = 0$ in what is given.

3.4.2 Two-circles vehicle representation for collision checking

For the object modelling, there are many kinds of geometry to effectively bound the whole object. Among them, circles (or spheres), polygonal or complex shapes are used. Nevertheless, ellipsoid are the most popular shape to surround the objects because the number of parameter to represent it is still small and cover many kinds of obstacle such as pedestrian, stroller, bicycle and various type of vehicles in uniform way [Premebida 07], [Chakravarthy 12], [Vilca 13]. In this work, in order to check easily the future possible collisions of the vehicle with others objects in the environment, we propose to surround the vehicle with two circles (cf. Fig. 3.17) where one circle has same center with frontal wheels base axis and the other circle with rear wheels base axis. These two circles modeling permits to predict collision with other obstacle by checking the distance between the vehicle (represented by this two circles) and the obstacle.

Figure 3.14 depicts the representation for the considered car-like vehicle model on x-y coordinate plane. First, the rectangular shape and its rounded shape is shown (red bold line) where major components are also depicted as skeletal frame with front, rear axis and four wheels. Secondly, the ellipse model is shown with dotted closed line where the geometric parameters of long axis, short axis are determined to cover the rectangular shape by minimal area. The last one is two circles which are proposed for collision checking. The rectangular model is the

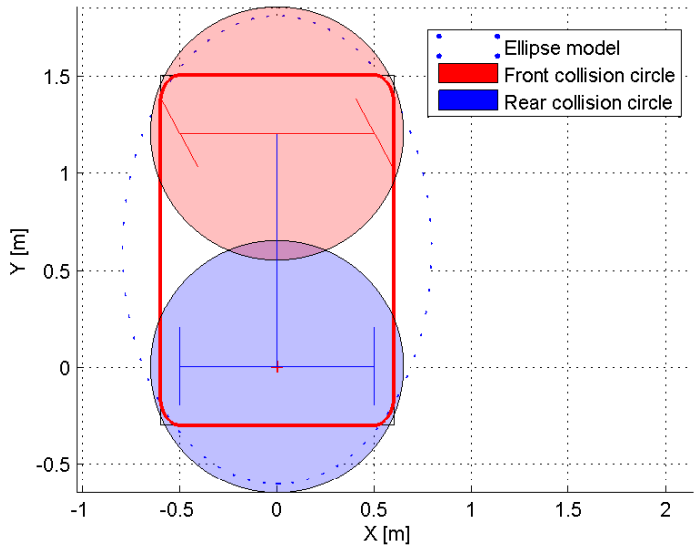


Figure 3.14: Two circles representation for vehicle

most closed one to real shape of the vehicle, however it requires many geometrical information to describe the collision model and it needs larger computations to check the shortest distance to other objects. The ellipsoidal representation covers the vehicle shape with five parameters in total where the parameters are the long, short axis distances, center positions coordinates and its heading (orientation). Even if the model encompasses the vehicle's shape well, it includes large vacant spaces at both sides than other models. Furthermode, the distance parameter to check collision with the other obstacle is not simple due to the fact that the boundary is ellipsoid.

Thus, in this work, the vehicle model having two identical circles is considered for its geometrical simplicity and adaptability to 4D configuration analysis. The advantages of the two-circles representation for vehicle could be detailed as follows. At first, the circles hold kinematic constraint for the vehicle and they permit to simply check both circular distances between the circle in the vehicle side and the circle in the obstacle side. Considering the collision avoidance maneuver, there are two kinds of maneuvers as *front* and *rear* avoidances where the *front* avoidance (**FA**) stands for the vehicle's collision avoidance by passing over the obstacle's front and the *rear* avoidance (**RA**) presents the vehicle's avoidance by passing behind the obstacle's rear part. The front avoidance maneuver presents the case that the vehicle goes ahead before the obstacle arrives, on the other hand, the vehicle in the rear avoidance can avoid the collision by passing

behind the obstacle or reducing its speed for the rear circle and obstacle circle not to intrude each other. For both avoidances, the two-cicles model[10] can check the collision simply by measuring the distance between the front circle and the obstacle for the **FA**, and the diatance between the rear circle and the obstacle for the **RA**.

As shown in Fig. 3.14, the centers of the vehicle's frontal and rear axle match with the centers of the defined front and rear circles respectively while each diameter is set to cover the vehicle's rectangular vertices inside, i.e., when the radius of the front circle is $\boldsymbol{r_f}$, then it should be $\boldsymbol{r_f} \geq (\boldsymbol{R} + \epsilon)$ where $\boldsymbol{R}$ corresponds to the radius and ϵ could be additional safety margin such as bumper part, tyre thickness and its steering boundary. In the following subsection, the described vehicle model is applied in the 4D configuration analysis.

3.4.3 4D space analysis for obstacle avoidance

In this subsection, 4D space framework (3D *Cartesian* space and time) is considered to analyze collision detection and to plan the obstacle avoidance maneuver for a vehicle. The *Cartesian* coordinated 3D configuration space (ℂ-space) with additional time axis are constructed to analyze future status for dynamic objects (cf. Fig. 3.15).

A representative description is shown in Fig. 3.15 where two vehicles move in each direction by $\boldsymbol{V_a}$ (obstacle) and $\boldsymbol{V_b}$ (subject vehicle) respectively during the time from $\boldsymbol{t_i}$ to $\boldsymbol{t_f}$ and it is expected to make a collision at $\boldsymbol{t_c}$ by meeting of the two future trajectories L_a and L_b under the condition that each vehicle maintains each velocity (speed with heading). Note that each trajectory constructed by path and velocity locates on the trajectory plane such as S_a for $\boldsymbol{V_a}$ vehicle. It is shown here that the obstacle vehicle moving in velocity $\boldsymbol{V_a}$ and a vehicle moving with $\boldsymbol{V_b}$ collide at the state S_c in the 4D space of Fig. 3.15. In this space, it is easy to check collision among dynamic objects by measuring minimum distance between each future trajectory whether the measured distance is less than a collision threshold (where the threshold is bounded by object modeling scheme).

In the depicted 4D space, the vehicle trajectories L_a and L_b can be transformed by replanning path or velocity profile. Assuming that the obstacle vehicle maintains current trajectory, there are two strategies to avoid collision. One possible avoidance plan is to modify the velocity profile by encompassing the collision boundaries with minimal changes of the original velocity profile and the other one is to modify the arrival pose of the path.

[10]When the considered vehicle has a long axle distance like bus or truck, it requires to include additional circle(s) between the two axles to check collision on the edge sides.

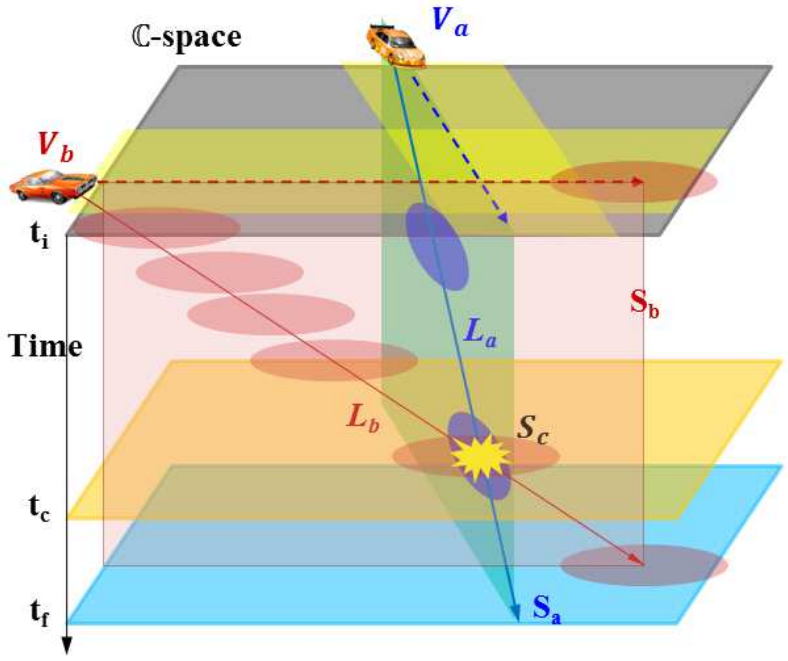
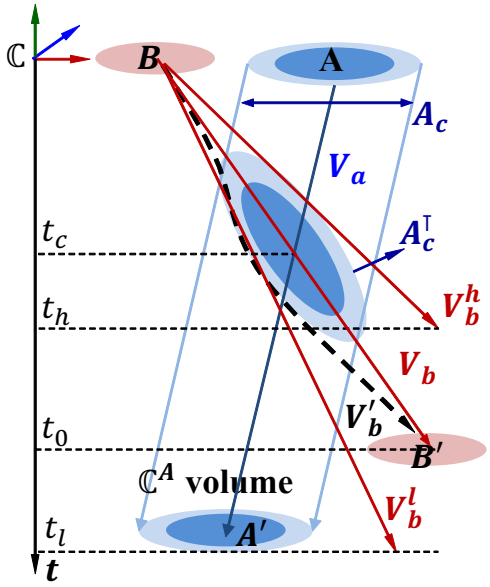
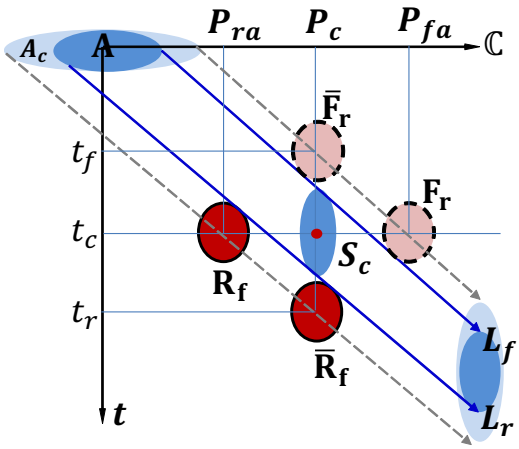


Figure 3.15: 4D configuration analysis for collision avoidance



(a) Velocity profile planning on S_b



(b) Avoidance pose planning on S_a

Figure 3.16: Two avoidance strategies in collision plane

In Fig. 3.16(a), planning the velocity profile is considered to avoid collision in the space, where two vehicles are modelled by an ellipsoid in 3D $\mathbb{C}$ space and several cases for velocity profiles of $\boldsymbol{B}$ vehicle (V_b) are planned on S_b plane which is denoted in Fig. 3.15. Note that the considered ellipsoid has the dimension summing the both sides boundary as A^c to check collision distance between the two vehicles. On S_b plane, $\mathbb{C}^A$ (space volume for $\boldsymbol{A}$ vehicle) is projected to generate an ellipse $A_c^{\intercal}$ and this ellipse is used to generate velocity profile in the plane. When V_b is an original velocity profile which results in a collision with $\boldsymbol{B}$ at $\boldsymbol{t_c}$, two feasible constant velocity profiles can be planned as V_b^h and V_b^l for higher and lower velocity than V_b respectively. Planning with variable velocity is also possible as V_b' by encompassing $A_c^{\intercal}$ with the shortest length. This velocity profile can be treated like smooth curve generation problem as *limit cycle model* which is addressed in [Adouane 13, Adouane 16][11]. In this work, the constant velocity planning is considered to make a safe trajectory with the given **pCCP**.

Fig. 3.16(b) is a diagram on S_a for the described strategy of constant velocity planning while considering two circles model for the subject vehicle $\boldsymbol{B}$. In Fig. 3.16(b), the velocity or path for the vehicle $\boldsymbol{B}$ can be replanned to avoid collision by allocating the vehicle circle outside the boundary of $\boldsymbol{A}$ space trajectory[12]. While keeping the path, the velocity is raised to precede the collision case as $\bar{\boldsymbol{F}}_{\boldsymbol{r}}$ or reduced to go behind the obstacle as $\bar{\boldsymbol{R}}_{\boldsymbol{r}}$. To maintain the time, the path should be modified as $\boldsymbol{F_r}$ and $\boldsymbol{R_f}$ for **FA** and **RA** respectively, which are all located on t_c[13].

In Fig. 3.16(b), when the subject vehicle $\boldsymbol{B}$ has the position $\boldsymbol{P_c}$ at t_c (or S_c), a collision with $\boldsymbol{A}$ is occured. In the plane, there are two possible avoidance poses according to their locations around the obstacle boundary ($A_c^{\intercal}$). Specifically, in the upper region of the boundary, noted as L_f, it is possible to make front avoidance, whereas in the lower part of the boundary L_r, rear avoidance is possible by $\boldsymbol{B}$. Each circle in the two circles model is used to check collision such that the front circle is checked for rear avoidance as $\boldsymbol{R_f}$ (lined circle) and rear circle is checked for front avoidance as $\boldsymbol{F_r}$ (dotted circle) in Fig. 3.16(b).

Note that $\bar{\boldsymbol{F}}_{\boldsymbol{r}}$, $\bar{\boldsymbol{R}}_{\boldsymbol{f}}$ locate at the same vertical line as $\boldsymbol{P_c}$ where both circles pass by the expected collision position $\boldsymbol{P_c}$ but with different arrival time t_f and t_r respectively. For avoiding by poses, it constrains the arrival time but differentiates the arrival poses as rear avoidance $\boldsymbol{P_{ra}}$ and front avoidance $\boldsymbol{P_{fa}}$. In what follows,

[11] Different to the referenced works, the proposed velocity planning addresses no collaborative avoidances.

[12] For the dynamic obstacle avoidance behavior, there are three kinds of arrival subsets such as front, rear and diverging avoidance [Fiorini 98] and here, two avoidance poses (front and rear) are used with two circles model.

[13] Note that the front avoidance should check on the rear circle, and the rear avoidance should check on the front circle.

the strategy to avoid collision is to find the time for $\boldsymbol{F_r}$ or $\boldsymbol{R_f}$, at which both poses are at the same level of t_c while being closest to the obstacle boundary. In the following subsection, the described velocity planning is simulated on an example case.

3.4.4 A demonstrative example of dynamic obstacle avoidance

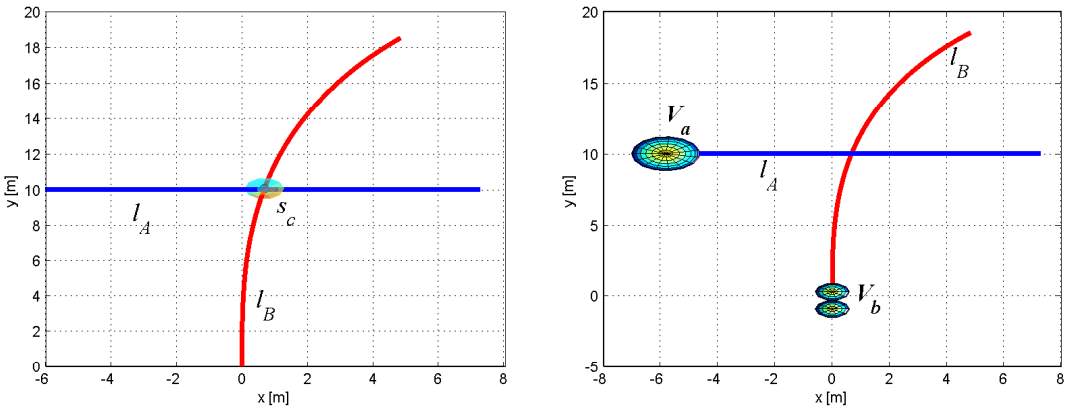
A simple example is considered to apply the proposed velocity planning and to validate its usefuleness of obstacle avoidance in the 4D configuration space. With a given **pCCP** for the car-like vehicles (cf. **Section 2.3**), the velocity planning is proposed to apply the path for dynamic obstacle avoidance. Two objects are considered in the space, where one is obstacle vehicle and the other is subject vehicle and it is assumed that the paths for both vehicles satisfy the nonholonomic constraint[14]. In the space, the obstacle is modeled as a circle and the vehicle is modelled by two identical circles [Chakravarthy 11] which having constant velocity, where the vehicle has the dimension given in Fig. 3.14. i.e., the radius of the two circles is 0.6 m and the obstacle is modelled as an ellipsoid having the radius of $\boldsymbol{r}_{ob} = [1.2, 1.2, 0.6]$ where $\boldsymbol{r} \in \mathbb{R}^3$ with $[r_x, r_y, r_z]$ in $[m]$.

The obstacle moves from the position $(-6, 10)$ to the positive x direction while having 0° orientation along the given path $\boldsymbol{l_A}$ (cf. Fig. 3.17(a)) and the vehicle drives on the curved **CCP** $\boldsymbol{l_B}$ from $(0, 0)$ to the positive y direction, and the both paths are intersected at the position s_c. Each vehicle maintains each own speed during the overall travel time.

Fig. 3.17(a) illustrates the collision case between the obstacle and the vehicle on the top view (2D plane view), where two paths are crossed at s_c. As shown in Fig. 3.17(b), the obstacle and vehicle are modelled as ellipsoid and two identical spheres respectively. With the given velocities 0.68 $[m/s]$ for the obstacle and 1 $[m/s]$ for the vehicle, $\boldsymbol{V_a}$ makes a trajectory by a straight line from $[0, 0, 0]$ to $[13.62, 10.00, 19.62]$ and $\boldsymbol{V_b}$ draws another trajectory from $[-6, 10, 0]$ to $[4.82, 18.49, 19.62]$ by a clothoid ($\alpha = 0.004$ and $\delta = \frac{\pi}{4}$). Two trajectories are intersected at $s_c = [0.68, 10, 10.09]$. 4D configuration analysis is performed as follows.

In Fig. 3.18, totally three scenes are captured in the simulation where each subfigure ((a) to (c)) shows two captures: the first one is 2D top view of the second one in 4D view. Through the scene #1 to #3, both vehicles get closer

[14]Actually, this representation is based on the assumption that the road surface is idealistically flat. For the real road surface with different elevation(*z-axis*), *manifold* [Wikipedia 17] projection plane could be required to represent all components in the 4D space representation.



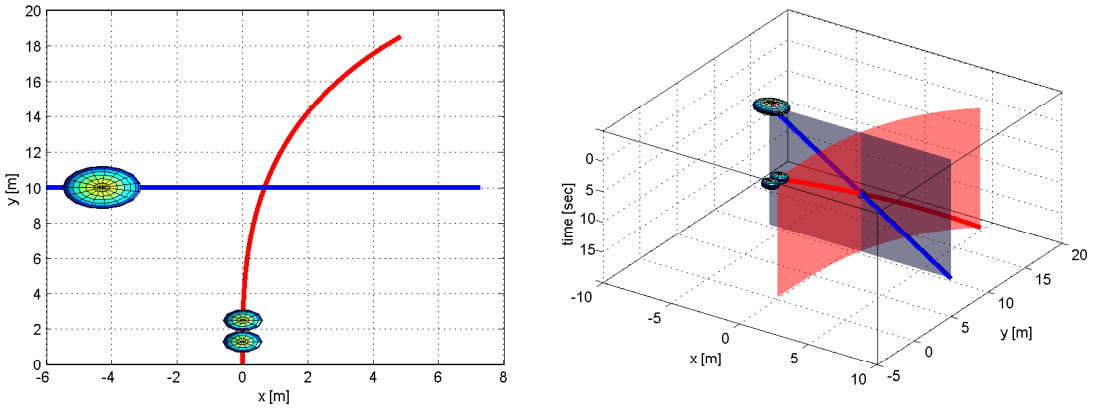
(a) Intersected two paths (b) Paths with vehicle/obstacle models

Figure 3.17: A demonstrative collision case

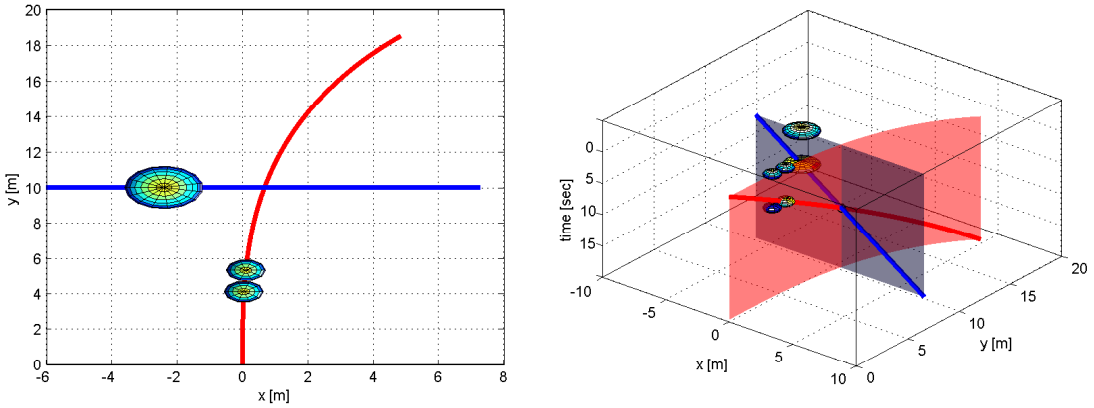
each other and collide in result, where at the position $(-0.13, 10.00)$ in ($x \in \mathbb{R}$, $y \in \mathbb{R}$)[15] for the obstacle and $(0.44, 8.67)$ for the vehicle with the same elapsed time 8.67 *sec* as captured in Fig. 3.18(c).

To avoid the collision between the obstacle and the vehicle, two methods described in **Section 3.4.3** are applied. In Fig. 3.19(a), velocity profiles to avoid collision are plotted with the collision path of Fig. 3.18 in 4D space, where $\boldsymbol{V_F}$ is $\boldsymbol{V_b^h}$ and $\boldsymbol{V_R}$ is $\boldsymbol{V_b^l}$ of Fig. 3.16(a) respectively. In Fig. 3.19(b), the second method of avoidance pose planning is applied to the same configuration space which is shown as $\boldsymbol{V_p}$. The planning results are presented on the path plane of the vehicle-**B** as Fig. 3.19(c), where the closed (red) circles are the rear avoidances which check for the front circle of the model, while the dotted (blue) circles correspond to the front avoidance and check for its rear circle. In this figure, the collision is occured at $(6.82, 10.09)$ where the collision pose for the vehicle corresponds to the travel length of 6.82 m and the elapsed time of 10.09 *sec*. Under the same configuration plane, it can be observed that the two avoidances by speed change maintain the same arrival pose 6.82 m but have different arrival times as 7.24 *sec* for higher speed and 12.94 *sec* for late speed respectively, whereas the other two avoidances using poses planning are same in the arrival time 10.09 *sec* but different arrival poses by 4.89 m and 8.74 m in the travel length. Those values are easily calculated using simple geometrical relation between the lines $\boldsymbol{L_f}$ and $\boldsymbol{L_R}$ (the size of the ellipsoid) and the radius of the circle (two circles model) as described in **Section 3.4.3**.

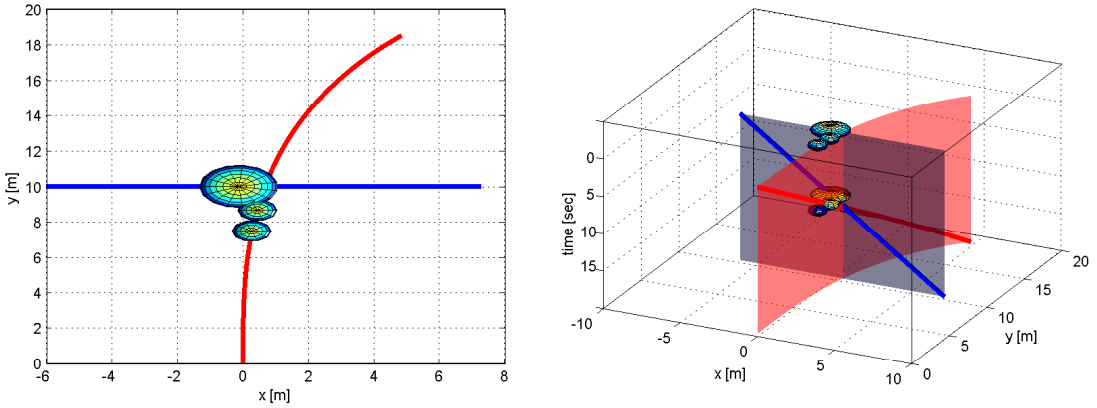
[15] $z \in \mathbb{R}$ is ommitted since $z = 0$.



(a) Scene #1

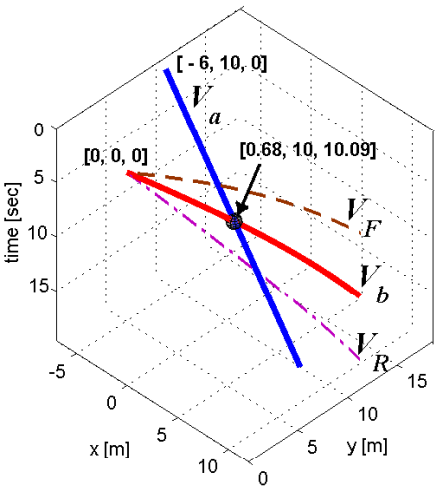


(b) Scene #2

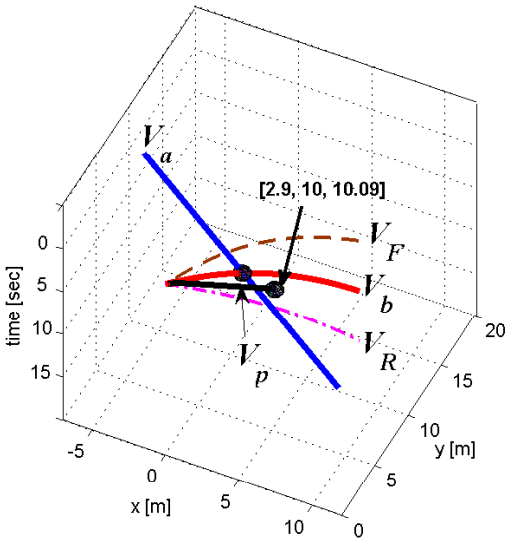


(c) Scene #3

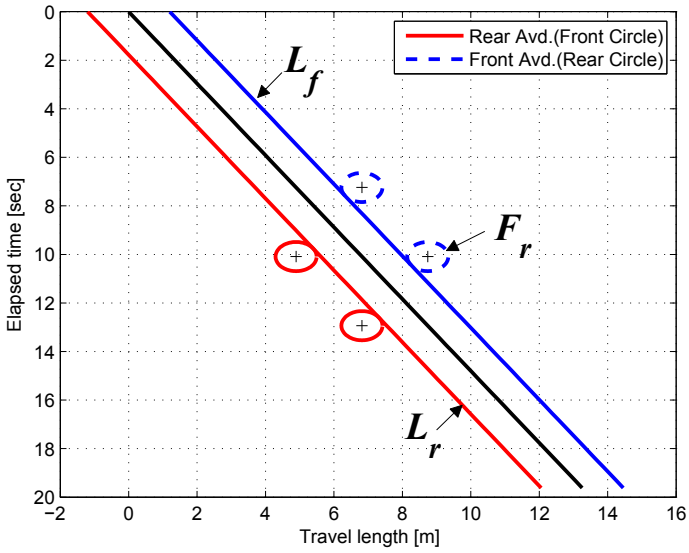
Figure 3.18: 4D configuration analysis for the collision case



(a) Velocity adaptation $\boldsymbol{V_F}$ and $\boldsymbol{V_R}$



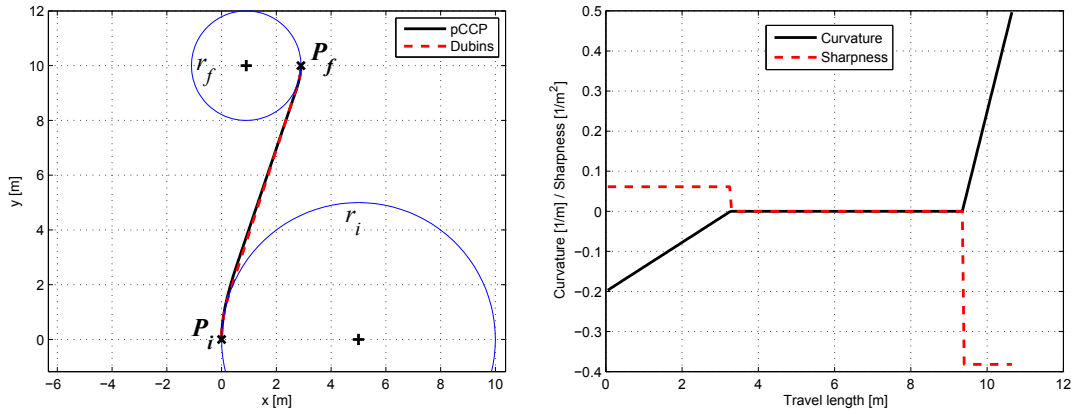
(b) Path/velocity adaptation $\boldsymbol{V_p}$



(c) Avoidance poses on collision plane

Figure 3.19: Velocity planning for dynamic obstacle avoidance

For the front avoidance path which depicted as $\boldsymbol{V_p}$ in Fig. 3.19(b) and also the dotted circle with denoted as $\boldsymbol{F_r}$ is generated using **pCCP** as follows.

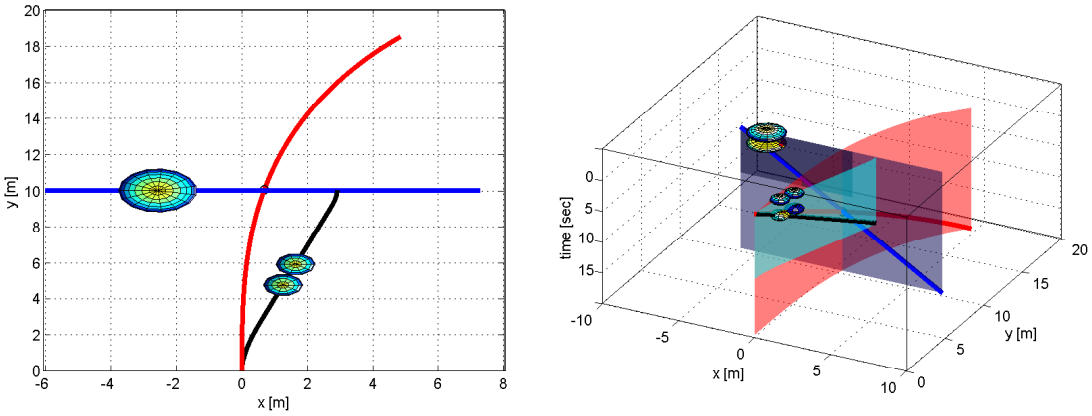


(a) Avoidance path (b) C-S diagram

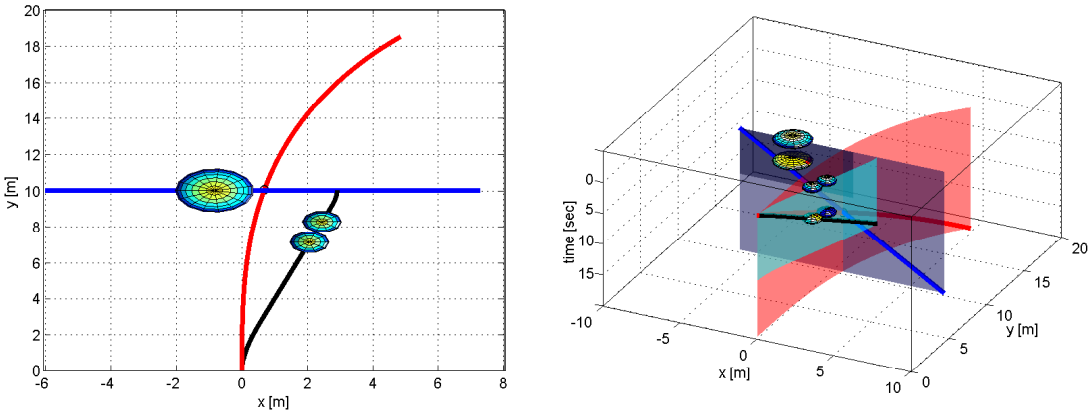
Figure 3.20: **pCCP** for obstacle avoidance

In Fig. 3.20, the solution path for the front avoidance is generated under the *arc-to-arc* boundary condition **Case 3Ba** (p.58) with **Algorithm 4**(p.61). In this problem, the initial and the final configurations are given by $\boldsymbol{P_i}(0, 0, \frac{\pi}{2}, \frac{1}{5})$ and $\boldsymbol{P_f}(2.9, 10, \frac{\pi}{2}, \frac{1}{2})$. As shown in Fig. 3.20(a), the obtained **pCCP** is similar to *Dubins* path but, preserves its curvature continuity that is proved by the corresponding C-S diagram of Fig. 3.20(b).

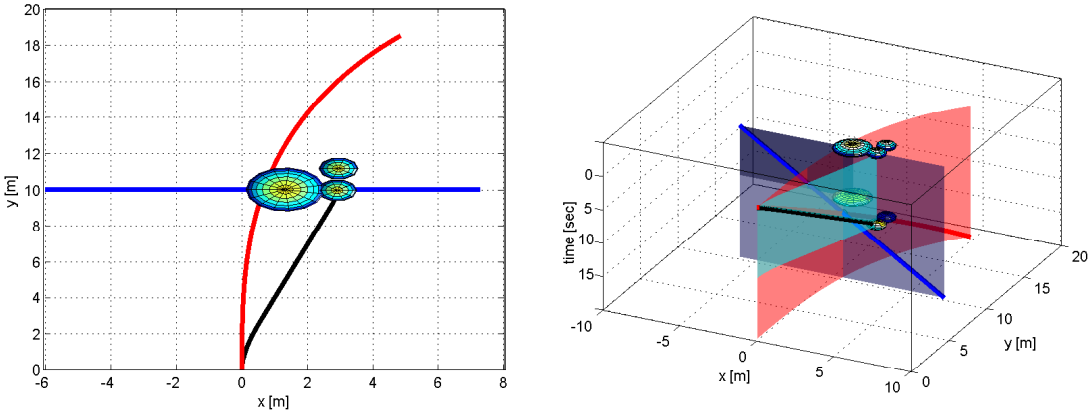
Figure 3.21 presents a demonstrative 4D space simulation for dynamic obstacle avoidance by three captured scenes similarly to Fig. 3.18. In Fig. 3.21(a), a new trajectory of the vehicle generated from Fig. 3.20 is plotted in the space under the same conditions as in Fig. 3.21. To avoid the collision $\boldsymbol{V_b}$ = 1 m/s is changed to $\boldsymbol{V_p}$ = 0.87 m/s, cf. Fig. 3.19 (by the travel distance 8.74 m and the arrival time 10.09 *sec* at $\boldsymbol{P_f}$). Since the obtained velocity is the least value to avoid the collision by front avoidance where the collision boundary is checked on the rear circle in the vehicle, thus the safe velocity should be greater than 0.87 m/s.



(a) Scene #1



(b) Scene #2



(c) Scene #3

Figure 3.21: 4D configuration analysis for the obtained **pCCP**

3.5 Conclusion

This chapter presented a smooth trajectory generation for dynamic obstacle avoidance. At first, as the subsequent problems of **Chapter 2**, the **Problem 4** is defined and solved to generate **d-pCCP**. Using the proposed **d-pCCP**, some practical points are considered in the obstacle avoidance maneuver for the vehicle, and two-circles representation for the vehicle and ellipsoide for an obstacle are integrated in the 4D configuration framework.

With the 4D configuration framework, efficient avoidance poses are determined with the safe velocity planning. Under the assumption of the constant velocity for the considered obstacle and vehicle, the future status on the risk of collision is checked on the space. Two methods of collision avoidance are also proposed, where the first modifies only the vehicle velocity profile and the second modifies only the path geometry, based on **pCCP** (cf. **Section 2.1.3** p.51), while considering the velocity as a constant and two circular shapes as vehicle envelope. For the second method, front and rear avoidances are also described under the assumption of constant velocity of the obstacle. The proposed velocity planning and replanning method could be efficiently integrated or parallelly computed/performed within the **d-pCCP** (or **pCCP**) algorithmic procedures, since the path and the velocity are independent to each other. For instance, when a path shape is modified, the velocity could be guided by the situation encountered by the vehicle. This will be subject to future development.

Extending **d-pCCP** to obstacle avoidance maneuver example permits the generated path to be reactive for dynamic obstacle. By collision checking in 4D space analysis with two circles vehicle representation for the vehicle, the avoidance poses are determined to avoid any risk of collision on the future. The resultant trajectory enables smooth steering on the path as well as smooth velocity changes along the path. Demonstrative examples on a dynamic obstacle show the effectiveness of the proposed methods and expected to be implemented for more complicated dynamic environments such as multiple obstacles or cluttered areas with real time performance. The more detailed cases and solution methodologies are not further considered since it is out of the scope of this work and could be investigated in future work.

Chapter 4

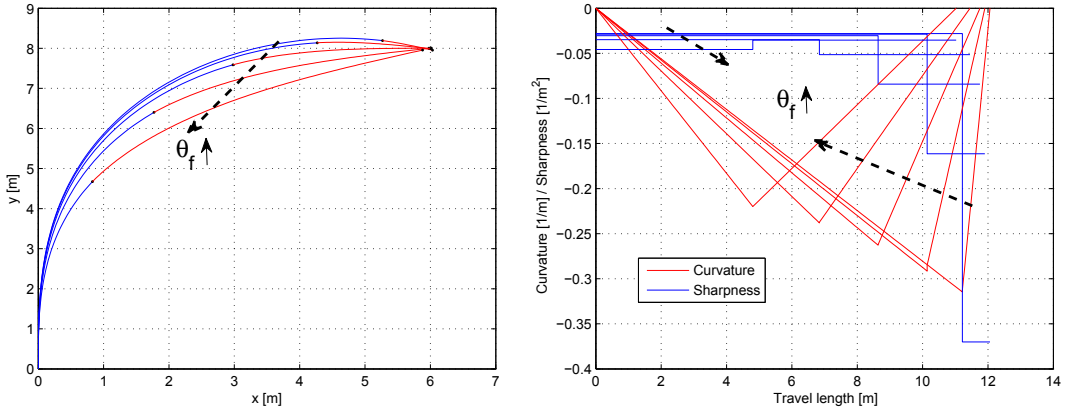
Application of pCCP and its validations

4.1 Introduction

Using the solutions and algorithmic procedures from the previous chapters, various practical examples are resolved with demonstrative simulations. Local path planning solutions are extended to solve the global path generation problems. To validate the effectiveness of the proposed **pCCP**, *Lyapunov*-based controller [Canudas 96, DeLuca 98, Adouane 08] was tested and evaluated on the obtained paths.

4.2 Application to local path planning

The proposed algorithm is applied to obtain **pCCP** for local path generation examples. In the first example, path generation for cornering motion and lane change maneuver are carried out, where each path planning corresponds to the problem of **Case 1A**, **1B**, respectively. For the cornering-like motion, the final configuration is given by $\boldsymbol{P_f}(6, 8, \theta_f, 0)$, and θ_f varies from $-10°$ to $30°$ per $10°$ as depicted in Fig. 4.1 with increasing $\theta_f \uparrow$ of a dotted arrow direction.



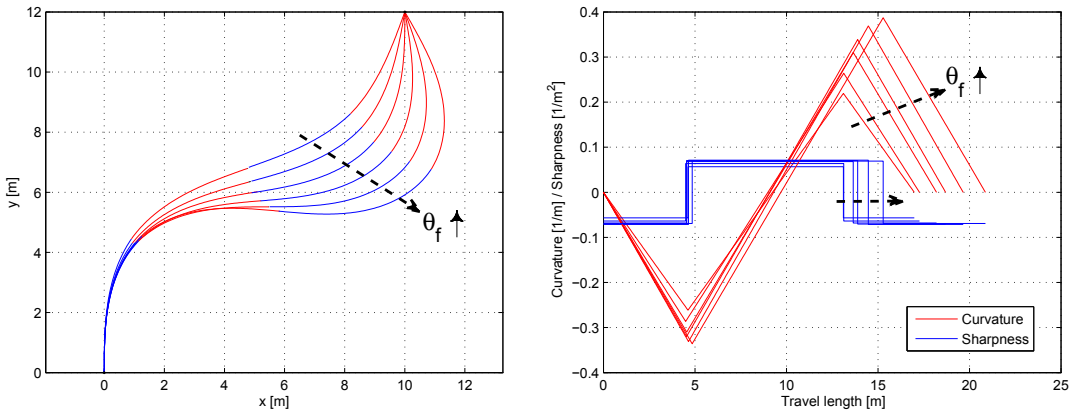
(a) Two clothoids composition (b) C-S diagram

Figure 4.1: **pCCP** for **Case 1A**

Figure 4.1 shows two clothoids **pCCP** for **Case 1A**, where the obtained path and corresponding curvature-sharpness diagram (C-S diagram) with negative y *axis* are depicted. In Fig. 4.1(b),it can be observed that, as the final orientation grows, the minimum curvature decreases from -0.22 to -0.32 as a dotted arrow direction by increasing θ_f ($\uparrow$). For the sharpness, the peak point of curvature moves right, and the sharpness value increases as θ_f decreases. This diagram illustrates that, in vehicle driving, steering behavior becomes more difficult as the orientation difference $(\theta_i - \theta_f)$ increases.

Secondly, lane change motions are performed with a lane change of 10 m width (up to three lanes) with different final orientations. The final configuration is given as $\boldsymbol{P_f}(10, 12, \theta_f, 0)$, where θ_f increases from $70°$ to $130°$ per $10°$ for each path generation.

In Fig. 4.2, the four clothoids **pCCP** for **Case 1B** is solved under the constraint of **Lemma 1** using the **MSC**, constraining θ_m by $\theta_i - \theta_f$ (cf. line 2 in **Algorithm 2**). The corresponding curvature diagrams show different path and curvature



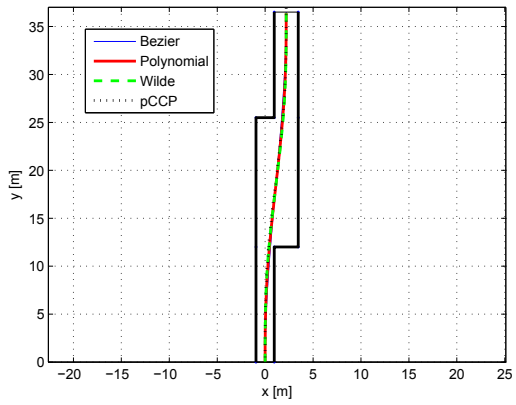
(a) Four clothoids composition (b) C-S diagram

Figure 4.2: **pCCP** for **Case 1B**

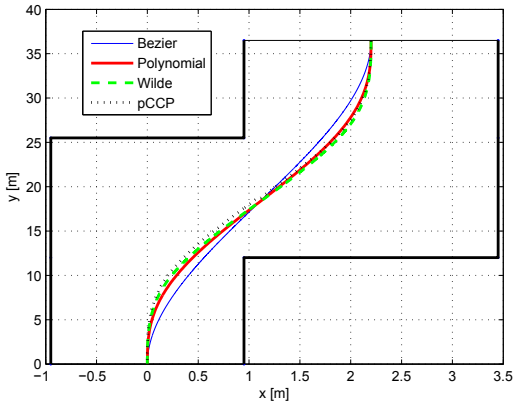
without loss of continuity, and the obtained four clothoids produce the minimal sharpness in overall length as shown in Fig. 4.2(b).

To validate the minimal late of turning or minimal curvature variation of the proposed path (cf. **Case 1B**, p.(44)), a severe lane change maneuver example (*ISO 3882-2*, lane width: 2.2 m, travel distance: 36.5 m) is applied with other methods in literatures of [Chen 13, Wang 98, Wilde 09]. The path of [Chen 13] uses piecewise quadratic Bèzier curve, and the method of [Wang 98] employs quintic polynomial function.

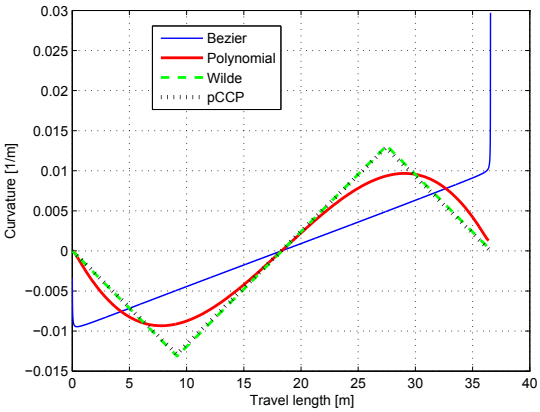
Fig. 4.3 shows obtained paths, cuvature and sharpness compared with other paths, where (b) is zoom-in plot with normal axes scale to give more clear view for their differences in the position domain. In Fig.4.3, all the paths are geomatrically smooth and satisfy the boundary condition in (a), (b), but have different curvature/sharpness diagrams in (c), (d). At first, the *Bèzier* path have the steepest changes at both boundary for its curvature and sharpness. The polynomial path shows smooth and small curvature, but high sharpness at both boundary. The proposed **pCCP** and *Wilde*'s path record almost same results, where *Wilde*'s path employs approximation function for the minimal amount of steering and the least maximum curvature [Wilde 09], thus the obtained path has also the same characteristics as *Wilde*'s path. Here it is noted that the proposed **pCCP** is motivated to provide a human-natural and easily drivable path with minimal amount of steering as *Wilde*'s path, but it overcomes the limitation of *Wilde*'s path which is only applicable to the straight lane case. Thus, **pCCP** has the advantage of efficient steering as well as the generality of cases.



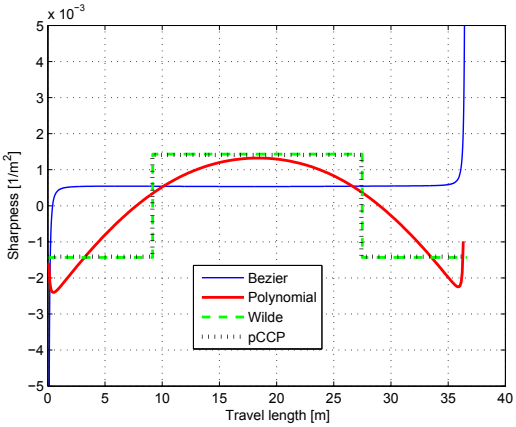
(a) Paths of equal x-y scale



(b) Zoom-in plot of (a)



(c) Curvature diagram



(d) Sharpness diagram

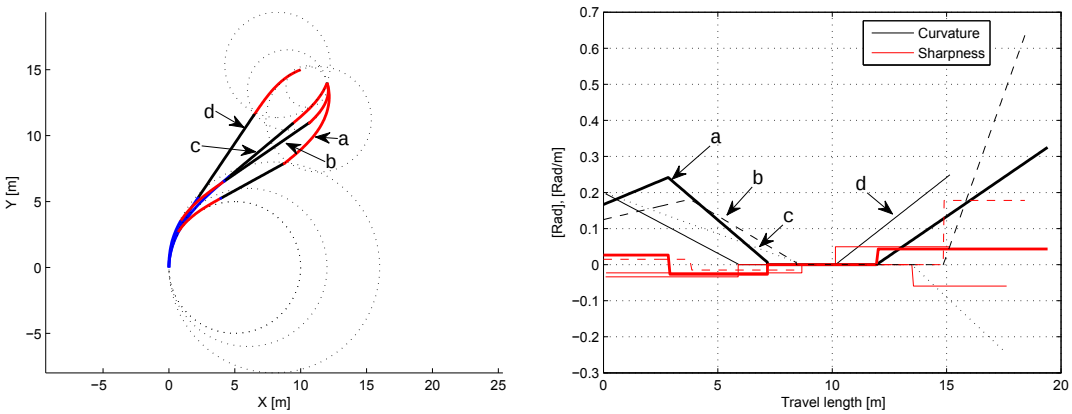
Figure 4.3: ISO 3882-2 lane change paths

Table 4.1: Paths Comparison

Method	κ_{max}	κ_{min}	α_{max}	α_{min}
Bèzier	0.0095	- 0.0297	0.2638	- 1.3210
Polynomial	0.0093	- 0.0097	0.0024	- 0.0013
Wilde's	0.0131	- 0.0131	0.0014	- 0.0014
pCCP	**0.0129**	**- 0.0129**	**0.0014**	**- 0.0014**

Table 4.1 summarizes the comparison results for the lane change methods. For curvatures maximal or minimal values, the polynomial path shows the better performance, however the *Wilde*'s and **pCCP** record the least α_{max} value which reduces the rate of turning and exerts minimal jerk to the vehicle passenger. Even if the proposed path does not hold superior performance in minimizing the κ_{max} (or maximizing the κ_{min}), the curvatature diagram shows simple and linear shape, thus easily drivable and efficient for the autonomous vehicle's path following.

In the last example, local path planning for obstacle avoidance is fulfilled with a *arc-to-arc* problem (cf. Table. 2.2 p.63) with **Algorithm 4**(p.61) where the vehicle has a nonzero steering angle and arrives at the final position with a nonzero steering angle. Actually, such a configuration of $\kappa_f \neq 0$ is usual for obstacle avoidance where the obstacle is safely modeled as a bounded circle or ellipsoid [DenBerg 08, Chakravarthy 12].



(a) for four different configurations (b) C-S diagrams

Figure 4.4: **pCCP** for **Case3Aa**, **3Ba**

In Fig. 4.4, four different configurations for **Case3Aa** and **3Ba** are applied for **pCCP** generation. From $\boldsymbol{P_i}(0, 0, \theta_i, \kappa_i)$ to $\boldsymbol{P_f}(x_f, y_f, \theta_f, \kappa_f)$, only the initial position is same and all other variables are different, i.e., different final position, both orientations, and both curvatures, where four cases from `a` to `d` are denoted in Fig. 4.4(a) for the paths and Fig. 4.4(b) for the corresponding C-S diagram. Each generated path with its curvature/sharpness diagram satisfies both configurations proposed by **Algorithm 4** and **5**.

4.3 Application to global path planning using sequential pCCPs

This chapter addresses the global path planning by expanding the local path planning by **pCCP**. In the previous section, local path planning was performed using **pCCP** where the problem gave only two boundary conditions without any additional way point between two boundaries.

For such problems as providing a number of way points between two boundary conditions, a local path planning can not satisfy the conditions of passing all the way points. As described in the **Introduction**, a global path planning problem can be tackled by a number of local path planning problems; first, divide the problem into sub-problems and solve each local path problem, then combine all the obtained solutions, i.e. *divide-and-conquer*.

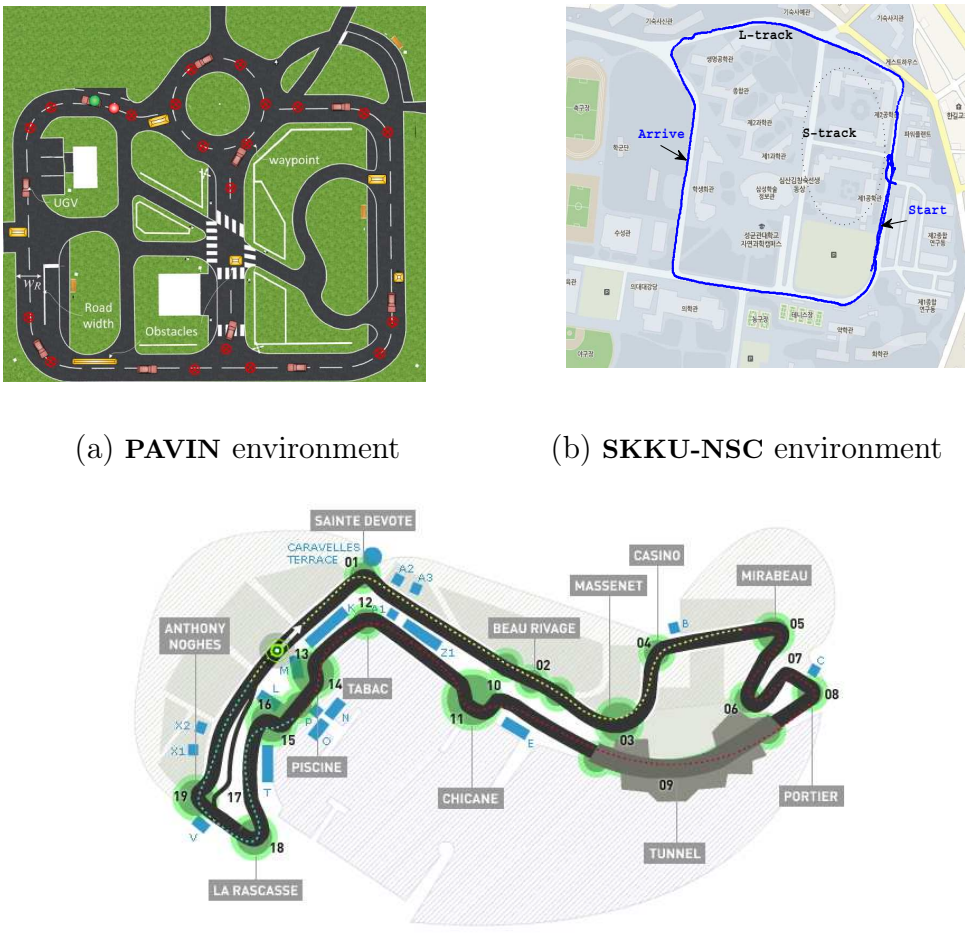
By the way, in **Chapter 2**, each local path problem is categorized by boundary condition whether line or arc. Thus, for the *divide-and-conquer* strategy, the way points disposition or allocation determines the difficulty of constructing sub local path problems; if each problem is easy to be divided by explicit line or arc segment, the global path problem can be constructed by sequential local path problems, however it is difficult to separate into sequential local problems when there is little line segment or complex arc segment like slalom shape.

In this section, three environments are applied to the problems of global path planning. The first one is **PAVIN** (Platforme d'Auvergne pour Véhicules INtelligents, Fig. 4.5(a)) located at Campus Cézeaux of Blaise Pascal University in Clermont-Ferrand and required data are acquired from [IPDS 16]. The **PAVIN** is an artifical environment of a total area of $5,000\ m^2$ where the urban sector has a path of 317 m length containing a scaled street with several traffic junctions and roundabouts with traffic sing boards with lights [Adouane 16].

As a second example, Natural Sciences Campus in SungKyunKwan University (**SKKU-NSC**, Fig. 4.5(b)) which is located at Suwon, Gyeonggi province in Korea was employed. In the environment, two tracks are used to apply proposed methods for different purposes as a long track (about 1.56 km) and a short track (around 450 m), **L-track** and **S-track** respectively. Specifically the **L-track** is extended along open area, thus has the benifits of receiving clerer data from GPS, whereas the **S-track** has GPS-denied region in tall trees and buildings. For two different tracks, different sequential **pCCP** methods are implemented to correspond to the environment. The **S-track** requires a method to overcome the inaccuracy of the position in the GPS-denied region, and sequential local paths generation based on graphical guidance in the given digital map is proposed.

The third environment is taken from a popular racing vehicle track, the Monte

Carlo Circuit (**MCC**, Fig. 4.5(c)) at Monaco [GP 16]. Similar to the **PAVIN**, the **L-track** is well structured road, i.e. straight region and curved region are well separated, and thus the local path problem of **pCCP** is easily constituted from explicit boundary conditions. On the other hand, the **MCC** has many sections of sudden changes in turning, and explicit boundary conditions are hard to extract. A dataset of way points from the *veDYNA-entry* program [Tesis-DYNAware 16] is utilized to apply sequential **pCCP** method to this irregular environment.



(a) **PAVIN** environment

(b) **SKKU-NSC** environment

(c) Monte Carlo Formular one track

Figure 4.5: Environments for global path simulation

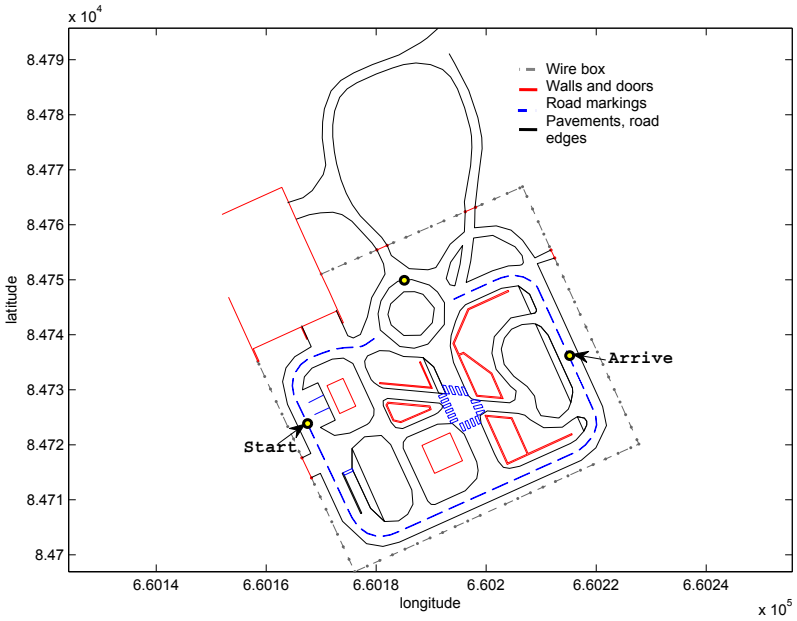
4.3.1 Global path smoothing and path following in PAVIN

In **PAVIN** platform, there has been numerous experiments for autonomous navigation, where a dedicated platform VIPALAB (length 1.96 m, width 1.30 m and height 2.11 m) is equipped with several cameras, range sensors, GPS and a ground-truth providing RTK-GPS (Real Time Kinematic-GPS) and elaborated softwares are implemented to control the steering/driving actuator systems [Vilca 13, IPDS 16]. In this subsection, a raw recorded data defining a possible vehicle path, has been smoothed using the concatenations of several **pCCP**s. In this example, a part of dataset acquired from driving vehicle is utilized for generating a global path. The dataset is collected from RTK-GPS (DGPS), thus having positional accuracy less than 2 cm. Each data point is sampled at 10 Hz, and the vehicle tries to maintain a constant velocity of 1 m/s. Figure 4.5(a) depicts the road network of **PAVIN** by geographic coordinates (longitude, latitude) and a section between two marks as **Start** and **Arrive** is drived by manual control and the data acquisition is carried out. After the procedure, the raw data which have the position information is analyzed in its smoothness especially in the point of curvature.

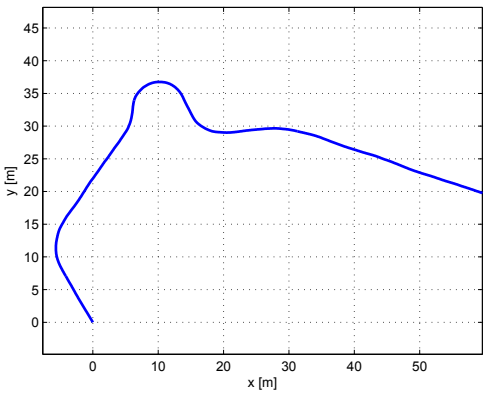
This global path has remarkably fine set of way points than other positional sensors, but include noise and discontinuity in the curvature due to the inherent characteristics of exterior position sensor. The sequential **pCCP**s can be a promising solution of providing smoothness in the overall path, and the smoothness or continuity of the curvature can be evaluated by path following performance of nonholonomic car-like vehicle. The raw data is displayed and analyzed by point-to-point as shown in Fig. 4.6(b) and (c) respectively.

Even if the real data in Fig. 4.6(b) looks geometrically smooth in real scale, its analysis in terms of orientation and curvature involves some noisy fluctuations due to sensor errors. Focusing on this raw data, smoothed path generation is carried out using the proposed solution, which should have continuous curvature while reconstructing, as close as possible, the original path. From the raw data acquired, the boundary conditions(initial and final configurations) are set to formulate several local path generation problems.

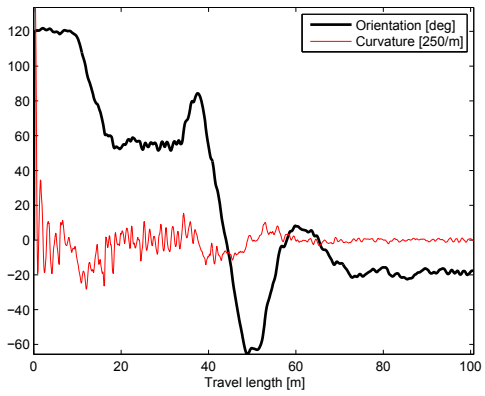
Focusing on these raw data, the smoothed trajectory generation is carried out by proposed solution where it should have continuous curvature while reconstructing as close as possible to the original experimental conditions. From the acquired raw data, it needs to set the boundary conditions for solving the clothoid segments problem as successive local path planning. For the general and automatic procedure that could be applied in the global framework, some pre-procedures are required to find the geometric boundary points to obtain the more appropriate clothoids for the corresponding conditions with constraints.



(a) Georeperenced 2D map of PAVIN

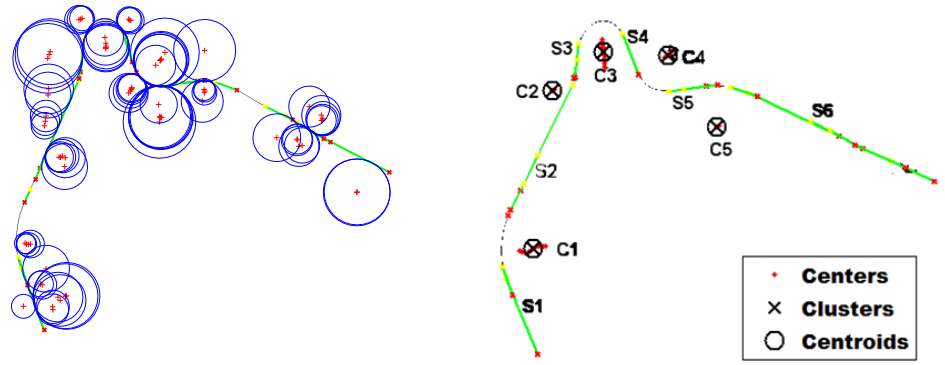


(b) Raw data path



(c) Orientation$[deg]$/curvature$[1/m]$

Figure 4.6: Path analysis for raw data in PAVIN



(a) Detection of linear/curved sections (b) Determination of circular centers

Figure 4.7: Procedure for extracting the boundary conditions

In the pre-procedures, the geometric data is mapped to the image plane and then the *Hough Transform* (**HT**) which is especially useful to extract the regional information for line (*Hough Line Transform*: **HLT**) or arc (*Hough Circle Transform*: **HCT**) [Duda 72] is performed on that image. These procedures, named as *Global path planner 1*, are described by **Algorithm 15** as follows.

In **Algorithm 15**, the overlapped lines are integrated into one segment and the circle are chosen to make boundary conditions to determine maximal curvature point $\boldsymbol{p_c}$. For circular redundancy, firstly, the minimal radius is taken from the vehicle's kinematical constraints as $\kappa \in [\kappa_{min}, \kappa_{max}]$ and then the circles made from the straight regions (detected from **HLT**) are extracted out. Finally, the circular objects outside the turning curve are deleted, where for instance the left side circles on the right turning region are redundant and vice versa.

The application of **HT** [Duda 72] for line and arc segments are performed as shown in Fig. 4.7(a). There are unnecessary information and noises in the arc results, thus clustering by *k-means* nearest neighbor method are proceeded [Duda 01] and the cluster centers are sorted by minimal value of the distance summation as defined by,

$$\min_{\hat{\mu}} \sum_{i \neq j}^{(i,j=r)} \|\sigma_i - \sigma_j\|, \tag{4.1}$$

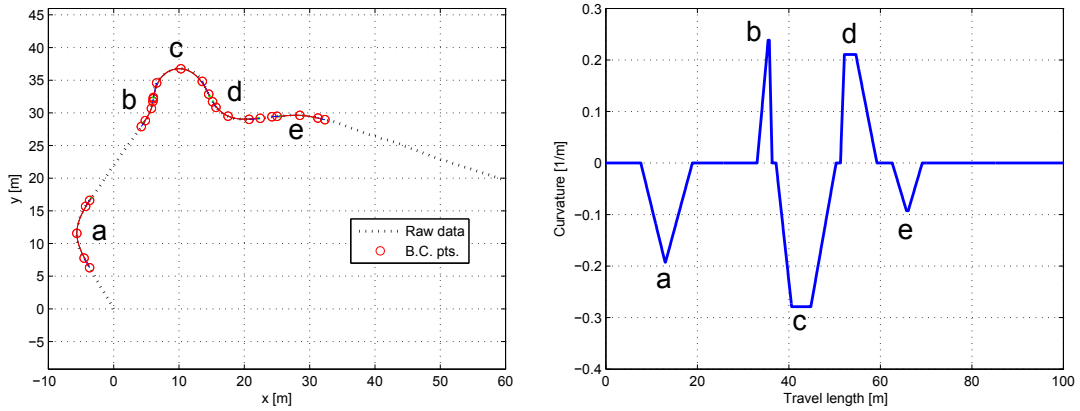
where σ_i is the radius of the i^{th} circle in a cluster $\hat{\mu}$ with r number of circles and $\mathcal{L}^2$(Euclidean) distance between the path and the circle center of minimal radius is selected as $\boldsymbol{p_c}$ point of the path.

Algorithm 15 **pCCPs** global path planner 1

Require: $P(\bar{\mathbf{x}}, \bar{\mathbf{y}})$ ▷ Raw data set

procedure **pCCPs** GLOBAL PATH PLANNER 1$(\bar{\mathbf{x}}, \bar{\mathbf{y}})$
  Pre-processing ▷ Data smoothing by noise filtering
  procedure MAP ANALYSIS
    $P(\bar{\mathbf{x}}, \bar{\mathbf{y}}) \longrightarrow I(\bar{\mathbf{u}}, \bar{\mathbf{v}})$ ▷ Mapping on Image map
    HLT : $\mathbf{S_m}$, $(m = 1, \ldots, p)$ ▷ n Line boundary points
    HCT : $\mathbf{C_n}$, $(n = 1, \ldots, q)$ ▷ m Arc boundary points
    Elimination of redundant points
    Boundary points determination
    $I(\bar{\mathbf{u}}, \bar{\mathbf{v}}) \longrightarrow P(\bar{\mathbf{x}}, \bar{\mathbf{y}})$ ▷ Inverse mapping on Cartesian map
  end procedure
  for [all $\boldsymbol{S_m}$ and $\boldsymbol{C_n}$] **do**
    procedure LOCAL PATH PLANNER **pCCP**$(\boldsymbol{p_i}, \boldsymbol{p_f})$
      if $\mathbf{C_n}$ **then**
        Configurations definition ▷ $\boldsymbol{p_c}$ from $\mathbf{C_n}$
        Local path by **Algorithm 3** ▷ **Case 2A**
      else ▷ Between $\boldsymbol{S_m}$ and $\boldsymbol{S_{m+1}}$
        Local path by **Algorithm 1** or **2** ▷ **Case 1A** or **Case 1B**
      end if
    end procedure
  end for
  Post processing ▷ Connecting the obtained solutions
end procedure

The integrated result in this procedure is shown in Fig. 4.7(b), where the line and clothoid/arc segments are noted respectively by $\mathbf{S_m}$ for $m = 1, 2, ..., p$ and $\mathbf{C_n}$ for $n = 1, 2, ..., q$, where p and q are the number of the isolated line and arc segments respectively. From the results, isolation of straight and curved sections gives each problem for local planner **pCCP** where the straight section provides **Problem 1** and the curved section makes **Problem 2**, then the global continuous curvature path is generated by connecting each solution result. Each clothoid solution has its own sharpness which is found from the geometric boundary configuration. Connected clothoids solution by **Algorithm 3** (p.43) and **3** (p.50) makes the continuous curvature diagram along the whole travel length. The diagram gives fully defined global reference trajectory in steering control. In every segment of the diagram, the constant sharpness is found to meet the configuration where the curvature grows up to the maximal curvature with given boundary pose.



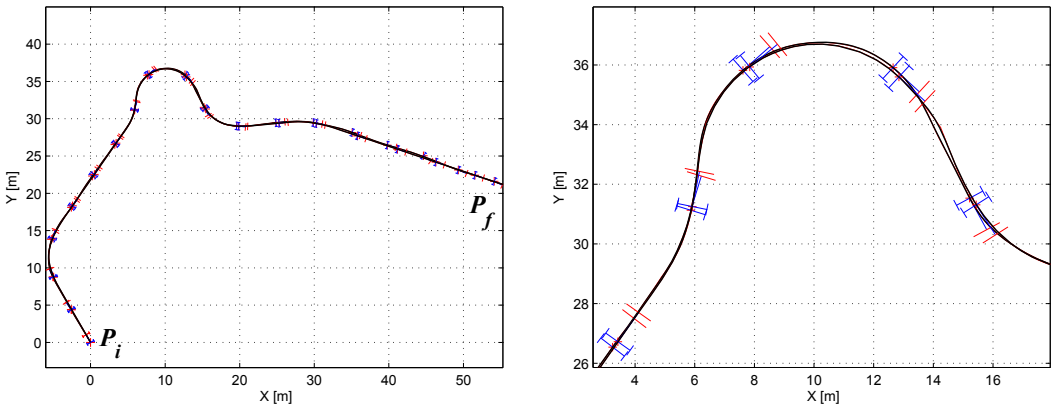
(a) Boundary conditions (b) Curvature diagram of **pCCP**

Figure 4.8: Global path generation using **pCCP** in **PAVIN**

Fig. 4.8(a) depicts the results of the procedure which formulate several boundary conditions and Fig. 4.8(b) shows the resultant curvature diagram for the obtained **pCCP** where each **pCCP** is solved for each boundary condition set as from **a** to **e** sections marked in Fig. 4.8(a). The corresponding curvature diagram validates its continuity along the travel length from **a** to **e** sections with intermediate linear sections as well as simple shape compared to the noisy shape of the raw data in Fig. 4.8(b).

The obtained **pCCP** was applied for autonomous vehicle navigation using an appropriate *Lyapunov*-based controller [Canudas 96] in the global framework.

The proposed algorithmic solution has tested using the *Lyapunov*-based controller for comparing its performance with the control using raw data generated by DGPS.



(a) Path following on simulation (b) Vehicle on path following

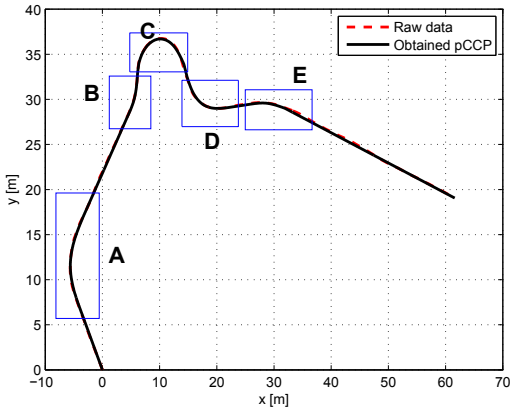
Figure 4.9: Path following simulation with a standard controller

Fig. 4.9 is the simulated resut on the path following. The car-like vehicle starts from its initial pose $\boldsymbol{P_i}$ with errors, d_e = 10 cm and θ_e = 10° with constant velocity of 1 m/s and stops at the end position of the given path as $\boldsymbol{P_f}$ shown in Fig. 4.9(a). Two simulations are performed on raw data path and obtained **pCCP** and the resultant trajectory (followed path along the time) of a car-like vehicle using *Lyapunov controller* as being zoom-in in Fig. 4.9(b).

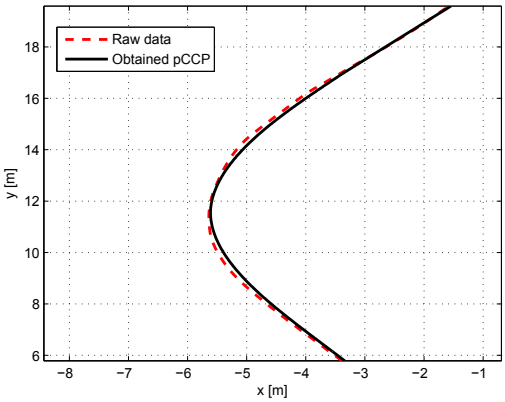
Fig. 4.11 depicts the comparisonal results of the steering set-points and atual steering angle for the two paths, where the actual steering angle is controlled to follow each path by the given controller. Fig. 4.11(a) and (b) explicitly show that the actual vehicle steering has better performances to follow the set-point defined according to the proposed **pCCP**. In fact, less steering fluctuations are observed over the travel time which implies better smoothness of the actual vehicle steering.

Fig. 4.12(a) and (b) compares the control results on the curvature and sharpness respectively where both results discriminate on the diminuation of fluctions on behalf of **pCCP**. The qualitative items for these preformance are listed by $|\kappa_{max,avg}|$, $\int |\kappa|\, ds$ and $|\alpha_{max,avg}|$ as compared in Table. 4.2.

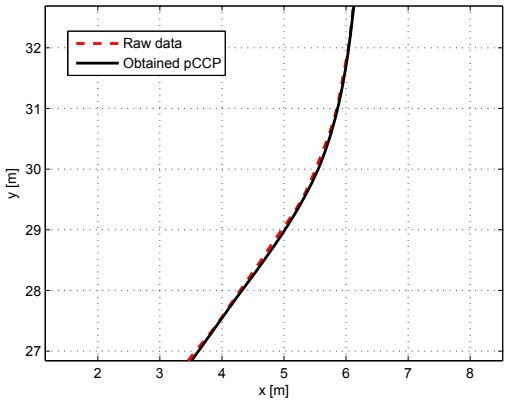
Figure 4.13 show another comparisonal result for path following performance by deviation errors of lateral distance and orientation. It is clear that according to the results shown in Fig. 4.13, the used **pCCP** shows better performances, highlighted with reduced lateral and orientation errors w.r.t. to the followed



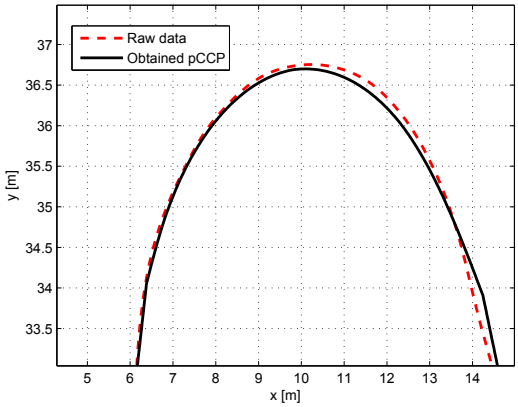
(a) Path following results



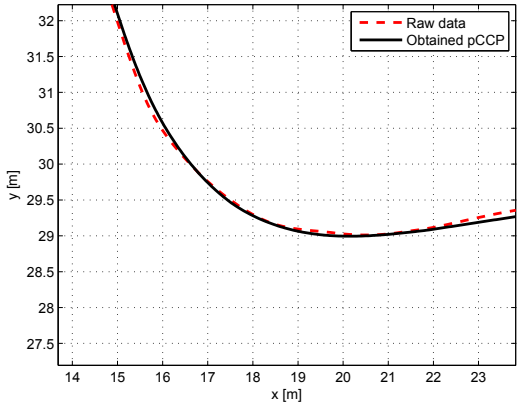
(b) Area-**A**



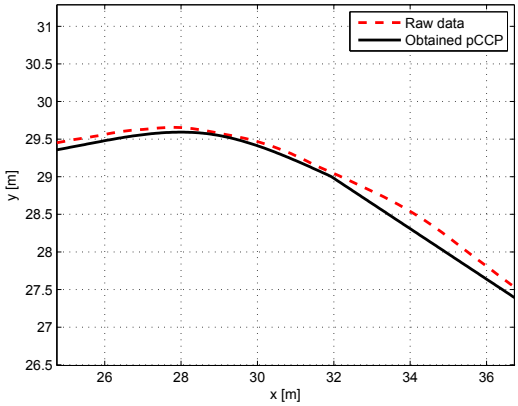
(a) Area-**B**



(b) Area-**C**

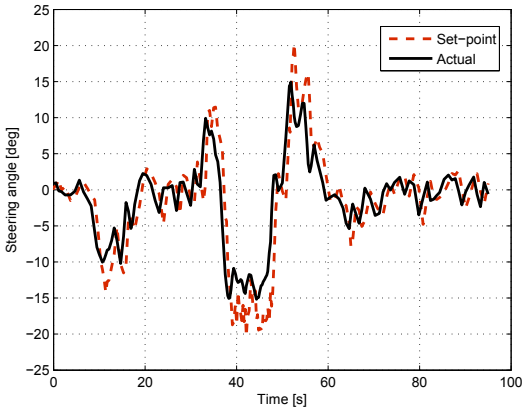


(d) Area-**D**

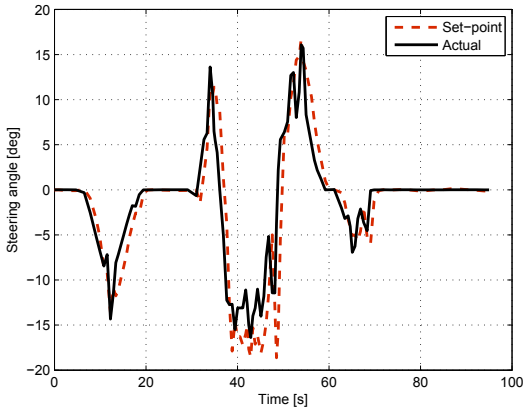


(e) Area-**E**

Figure 4.10: Path following results for raw data and **pCCP**

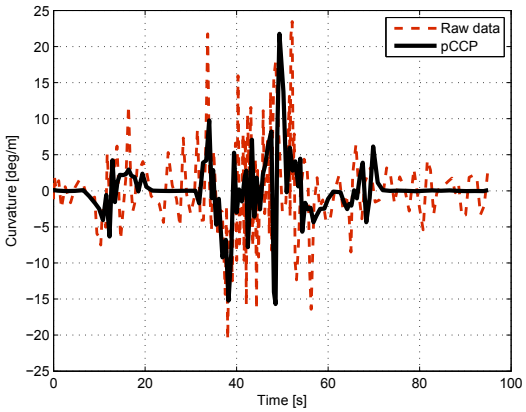


(a) Steering control for raw path

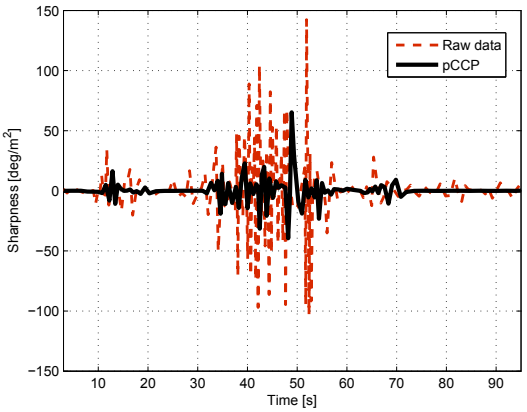


(b) Steering control for obtained path

Figure 4.11: Comparison of steering control with/without the proposed **pCCP**

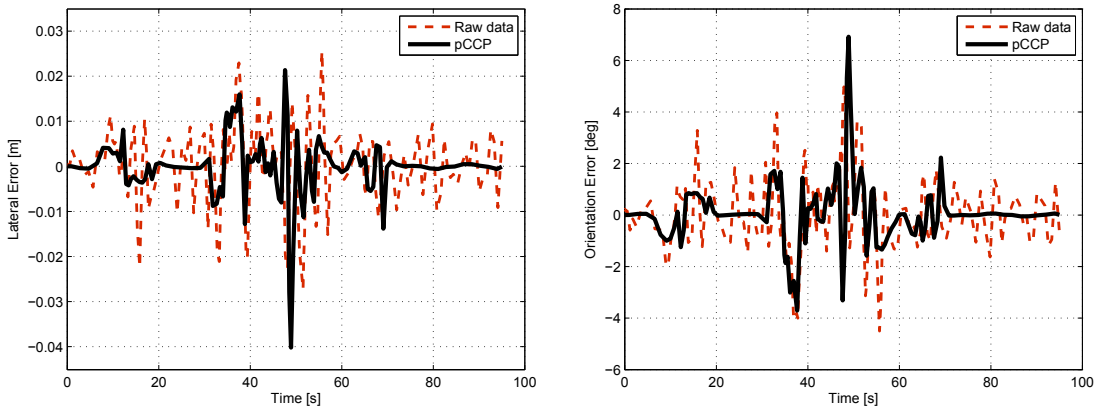


(a) Curvatures comparison



(b) Sharpnesses comparison

Figure 4.12: Comparison of path following with/without **pCCP**



(a) Lateral pose error (d_e) [m] (b) Orientation error (θ_e) [°]

Figure 4.13: Path following performances with/without the proposed **pCCP**

path. More details about the features of the obtained **pCCP** are summarized in Table 4.2.

Table 4.2: Performance Comparison

Performance	p^e_{max}	p^e_{avg}	θ^e_{max}	θ^e_{avg}	$\lvert\kappa_{max}\rvert$	$\int \lvert\kappa\rvert ds$	$\lvert\alpha\rvert_{max}$	$\lvert\alpha\rvert_{avg}$
Raw (a)	0.027	0.007	4.9847	1.1459	15.184	974.606	23.461	4.652
pCCP (b)	0.028	0.003	3.8961	0.6875	17.349	574.852	16.317	2.491
$\frac{a-b}{a}$ [%]	-2.6	54.4	21.7	42.6	-14.3	41.0	30.5	46.5

In Table 4.2, the performances are evaluated using quantitative and qualitative measures such that the peak values (max) evaluates the quality of each path, and integrating or average (avg) values illustrate the quantitative level to analyze the performance in a synthetical manner. The maximum and average lateral position errors are denoted by p^e_{max} and p^e_{avg}, respectively, in $[m]$. It is also noted for orientation error θ^e_{max}, θ^e_{avg} in $[°]$ and absolute sharpness by α_{max} and α_{avg} in $[\frac{deg}{m^2}]$ and for the maximal absolute steering κ_{max} in $[\frac{1}{m}]$ respectively.

The sharpness terms α_{max} and α_{avg} result in 30.5 % and 46.5 % improvement, respectively, with respect to the raw data, and these verify that the obtained steering behavior using the global **pCCP** is much smoother than the case using raw data. The remarkable difference is also shown in the total steering work

through the path $\int |\kappa| \, ds$. The steering work is reduced by 41.0 % compared to the driving based on the raw data. This shows that the **pCCP** algorithm generates an energy-saving trajectory for the vehicle using smooth and continuous steering. Only deteriorated results on the lateral and orientation errors are found at the time around 47 *sec* in Fig. 4.15 which corresponds to the section **c** in Fig. 4.10. This deviation is caused from the long circular section of **c** in the original path where the proposed path is not optimized to fit for the overall circular segment, and the proposed algorithm is required to be improved for regenerating closeness to those sections.

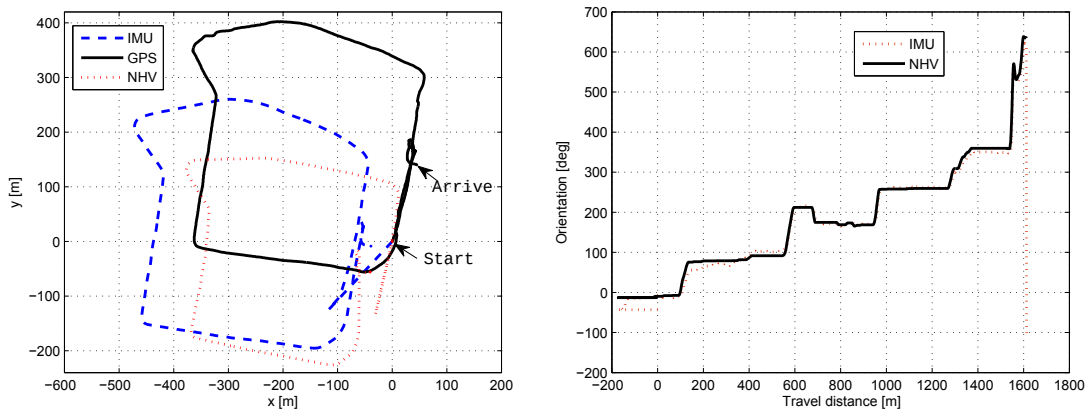
Results given in from Fig. 4.11 to Fig. 4.13 and Table 4.2 demonstrate that the smoothed path determined by clothoid segments provides an enhanced performance for the path driven by a nonholonomic car-like vehicle. The reduced steering work and sharpness decrease enhance passenger comfort by reducing the lateral acceleration and vehicle jerk [Labakhua 08].

With regard to the algorithmic performance, every clothoid generation takes less than 5 ms when using an Intel Celeron CPU with 1.50 GHz in a MATLAB® environment. The number of iterations for algorithmic convergence is $8 \sim 10$ loops for achieving the solution within the convergence threshold $D_e = 10^{-2}$ m.

4.3.2 Global path generation in SKKU-NSC

4.3.2.1 Long track navigation and paths analysis

For the **L-track** driving and data aquisition in the **SKKU-NSC**, an electric prototype vehicle; **AURORA** (width 1.5 m, length 3.1 m, height 2.3 m) was deployed [Gim 12], where the vehicle is equipped with 4-layer LiDAR (Ibeo LUX 2010™), stereo camera system (Point Grey Bumblebee2™) and integrated IMU/GPS module (XSENS™MTi-G). Besides these exterior sensors, the vehicle has odometry sensors at rear wheels and steering encoders to sense or record vehicle motion. From basic experiments of vehicle motion, the scale factor to convert the variation of odometry and steering encoder data into actual dimenstion scale (m, deg) are obtained, and applied to present the resultant paths on the Cartesian coordinate frame. The vehicle was manually driven along the **L-track** and all data from installed sensors were recorded in 10 Hz. Using the obtained sensor data set, the vehicle path can be presented by three different methods; One is to use orientation of **IMU** according to wheel travel distance, another one is to obtain the position data directly from **GPS**, and the last one is to apply differential relation of nonholonomic vehicle by steering encoder data (**NHV**), cf. Eq.(1.1).



(a) Paths results (b) Orientation results

Figure 4.14: Paths results from different sensors in **SKKU-NSC**

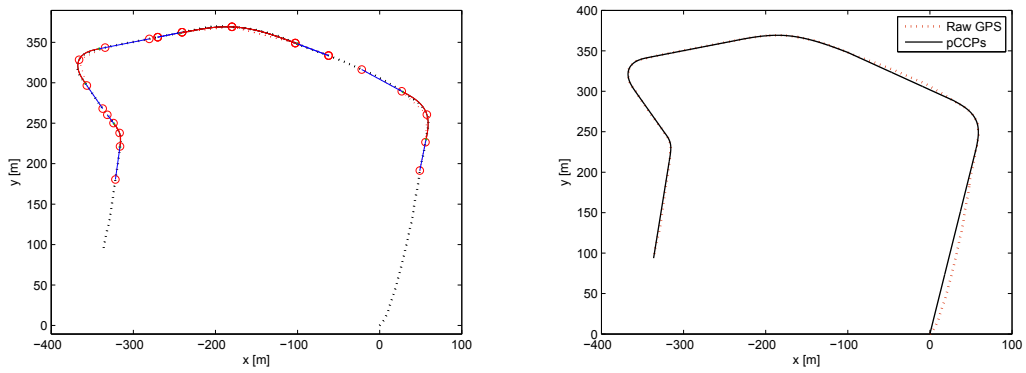
Fig. 4.14(a) shows the three paths results while driving from the marked point-**Start** to the marked point-**Arrive** by **IMU**, **GPS** and **NHV** respectively, and (b) is the orientation records for **IMU** and **NHV** data. Results in two figures inform that it is not easy to acquire clear as well as same path results. At first, for **IMU** data, it can be seen that there exists erroneous section at both the **Start**

and **Arrive** position. These error also cause the discrepancy in the **NHV** paths since initial orientation value is adopted from IMU sensor data. Compared with other two methods, **GPS** method shows the closest record to the actual path of the digital map; Fig. 4.5(b), however the acquired points are not so smooth along the path even if the vehicle is smoothly driven.

A major reason for these orientation error is from an initial orientation setting from magnetometer inside **IMU** sensor, where the earth magnetic is vulnerable to surrounding electrical devices, thus initially it is not acceptable to use as a reference.

Fig. 4.14(b) displays that the orientation calculated from steering angle, **NHV** can lead to a difference with the orientation from the **IMU**, such that the resultant path builds up a distorted path loop, compared to other two paths.

From those experiments, it is found that **GPS** data is reliable to use it as a referenced path in the **L-track**, however the data has noisy discontinuities, thus the path of **IMU** is required to supplement the procedure to extract boundary conditions as described **Algorithm 15** in the previous **Section 4.2.1**. The detailed procedure to formulate local **pCCP** problems is almost same as that of the **PAVIN** and thus it is skipped.

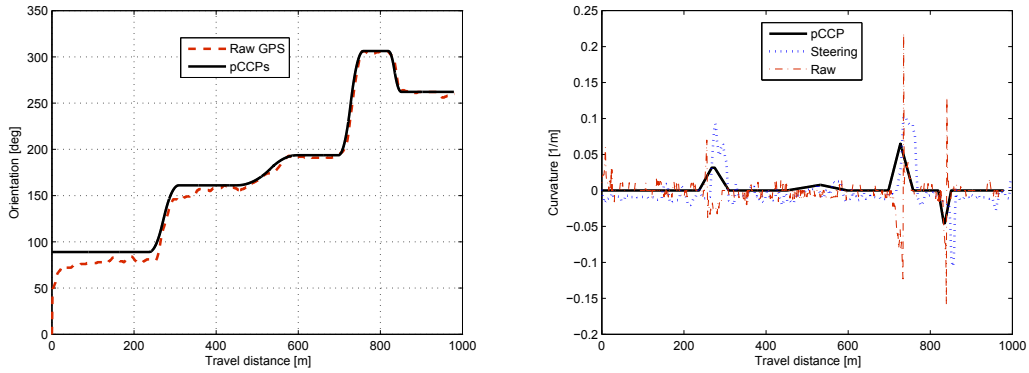


(a) Boundary conditions (b) Comparison of paths

Figure 4.15: Global path generation using **pCCP** in **SKKU-NSC**

In Fig. 4.15(a), the boundary conditions and multiple local path problems are formulated, and then each problem is solved by **pCCP**. Between two sequential **pCCPs**, a line segment is involved to smoothly connect two obtained paths, which results in a global path Fig. 4.15(b).

Fig. 4.16 includes two results about orientation and curvature for the obtained global path, where (a) compares the orientation of raw position data from GPS



(a) Orientations (b) Curvatures

Figure 4.16: Comparison of orientation and curvature with/without **pCCP**

and (b) compares curvature results calculated from GPS and steering data respectively. The obtained path, orientation and curvature results for the **pCCP** verify smoothness and continuity through the travel length, even if the actual data from exterior position and interior odometry sensors record frequent noises and fluctuations.

4.3.2.2 Short track navigation including GPS-denied region

As a second track in the **SKKU-NSC**, the **S-track** is located inside the **L-track** (cf. Fig. 4.5(b)), where building structures and tall trees hinder clear GPS signal from satellite, thus build up GPS-denied region. In such a GPS-denied region (**GD-region**), it is difficult to recognize self-location in the map during navigation, thus it requires to be supported by other landmarks or path generation method.

Fig. 4.17(a) presents a focused view for the **S-track** with targetted way points to be passed from **Start** to **Arrive** position. To attain the goal, the prototype vehicle is driven to loop the track three times with same conditions as the driving in the **L-track** and its data are recorded to display on the given digital map. Fig. 4.17(b) displays the resultant path from raw GPS data on the map, where the georeferenced positions are scaled and shifted to match the actual path with the given map. Distorted data are found inside the yellow-dotted circle of (b), where the position data is not accurate and erroneous due to GPS-denied region marked as **GD-region**.

To investigate these position errors, two methods are tried and compared, where one is from odometry and steering data (**odometry** method) and the other

(a) Way points along the **S-track**

(b) Recorded GPS data

Figure 4.17: Path of the **L-track**

is a fusion with GPS and odometry (**fusion** method).

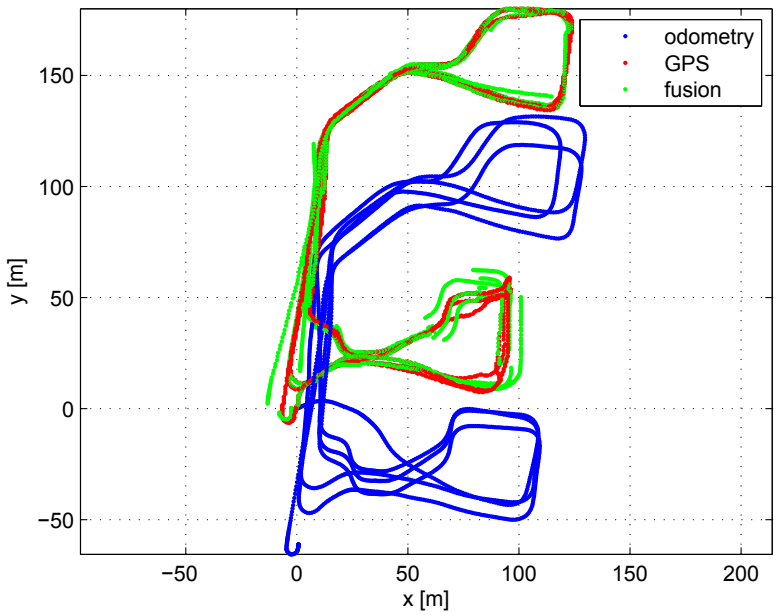


Figure 4.18: Path results with GPS and odometry data in the **S-track**

In Fig. 4.18, the path of the **odometry** method is constructed from rear wheel encoders and steering data as used in **NHV** of the **L-track**, and the path of the **fusion** method is generated from raw GPS data and conditional replacement into the **odometry** method by checking the change of orientation over threshold angle, $|30|$ *deg*. Similar to **IMU** or **NHV** of the **L-track**, the path of the **odometry** method has a large positional shift to the path of the **GPS** method due to initial orientation error. When it is focused on **GD-region**, the path of the **odometry** method is closer to actual driving path, however its every loop path is different each other due to the accumulated orientation error calculated from the steering data. Even if the path of the **fusion** method integrates the correct part of the **odometry** method in the **GD-region**, the resultant path has discontinuities and it is also difficult to determine what condition is the errorneous status for the received GPS data or when to replace it with odometry data.

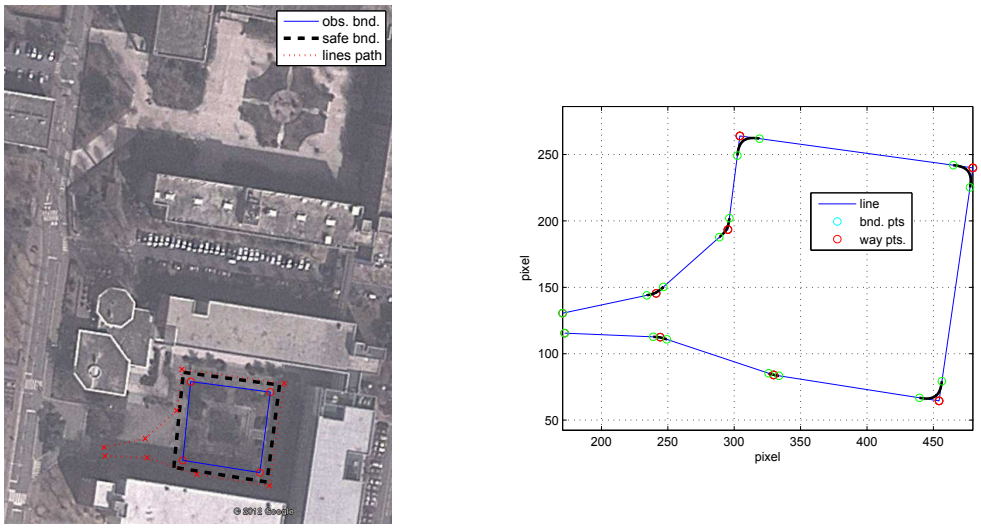
To ovecome these difficulties in the **GD-region**, **pCCP** is implemented with graphical guidance on the map. Several way points are manually marked on the image map of Fig. 4.17(b), then boundary conditions are constructed and solved to generate **pCCP**. This process is adopted from mapping in **GIS** (Geographic Information System) [ESRI 16, Kim 16] and can provide accurate position to

generate a nonholonomic path without dependence on the exterior position sensor in the **GD-region**.

The process is described as following.

Algorithm 16 **pCCPs** global path planner 2 (Map based **pCCPs**)

1: Obstacle is bounded with polygonal shape by marking vertices.
2: Mark way points from the start position to the arrive position while two sequential way points are connected by line segment.
3: If any line intersects the obstacle boundary, relocate the corresponding way point(s).
4: Extracts boundary condition to formulate **pCCP** problem from the generated lines path.
5: Solve and Generate **pCCP** for each problem.

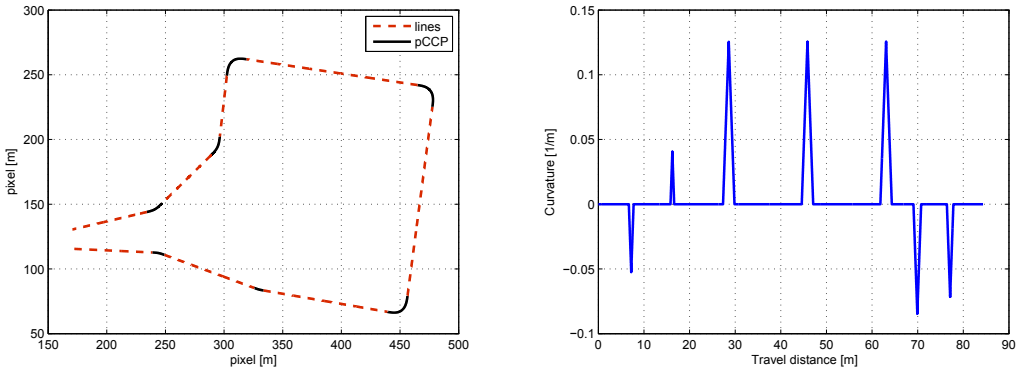


(a) Obstacle polygon and way points (b) BCs and **pCCP** solutions

Figure 4.19: Map based procedure for the **S-track**

In the map of the **S-track** as shown in Fig. 4.19(a), an obstacle boundary (obs. bnd.) is constructed by four way points. Then, a safe boundary (safe bnd.) is generated to avoid collision with the vehicle and makes a distance outside the obstacle. Considering the safe boundary, way points from the start point to the arrive point are marked sequentially and connect to each other by line segment.

All way points are manually picked. Fig. 4.19(b) depicts a result for marked way points, where firstly, lines path are generated and then boundary points are determined to formulate **pCCP** problems.



(a) Obtained path (b) Curvature diagram

Figure 4.20: Obtained solution path of the **S-track**

Fig. 4.20 depicts the obtained **pCCP**s and its curvature diagram. The global path is comprised of lines segment and symmetric **pCCP**s, where each **pCCP** smooths the sharp corner in the lines path. The corresponding curvature diagram provides a reference for steering control model that guides the vehicle to navigate in the **GD-region**.

4.3.3 Global path generation in Slalom-like track

This section addresses a global path generation or smoothing (in the curvature) the slalom-like track. A slalom-like track is the course that a vehicle navigates frequent zigzag or *S-shaped* motions. The track in a flat 2D plane is considered to focus on the application of proposed **pCCP** solution without velocity change by elevation differences.

In the slalom-like track, the vehicle has drastic curvature variation along the course and thus it is diffcult to extract line segment or boundary condition for **pCCP**. Specifically *line-to-line* or *line-to-arc* problems are difficult to apply to such tracks, and even the *arc-to-arc* problem can not provide appropriate solution to the part of nonuniform curvature variation; how to determine boundary condition can be an another big problem.

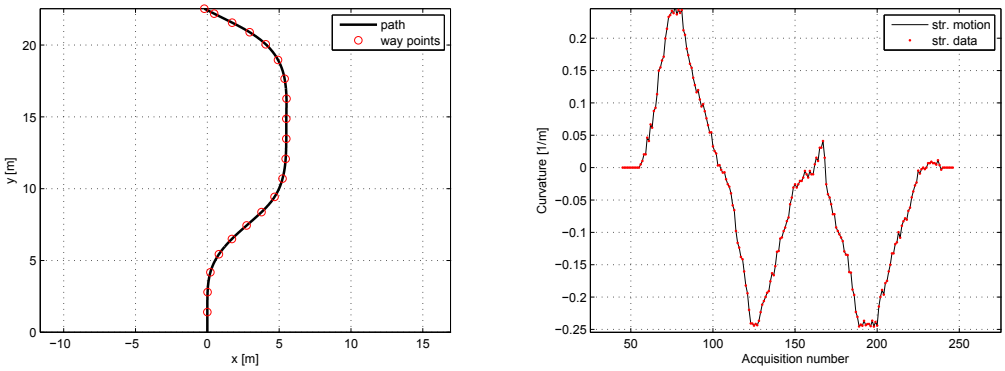
Thus, in this section, equally distanced boundary condition (or way point) is attributed to the whole path and *one* clothoid generation of **Problem 4** (cf. **Section 2.1.4**) is implemented. Two kinds of tracks are adopted, where one is relatively short and simple, and the other is long and more complicated. Before tackling into the racing track of *Monte Carlo* which can be obtained from the official webside [GP 16] (Fig. 4.5), a simple and short *S-shape* track is firstly examined and analyzed, then the *Monte Carlo* track is solved to generate global path by sequential **pCCPs**.

4.3.3.1 Short slalom-like track: SS-track

A short and simple path dataset is obtained from an experimental vehicle (cf. **Section 3.2**) by one cycle of zig-zag turning of the steering wheel.

Fig. 4.21(a) depicts the obtained path with equally distanced (1.39 m) way points and (b) shows the curvature diagram transitioned from the obtained steering encoders dataset. The purpose of the sequential **pCCPs** is to regenerate or reproduce the path as close as possible by the nonholonomic vehicle from a simple curvature diagram. The simple curvature is obtained by the sequential clothoids where each boundary condition is defined by the given way points. The curvature according to travel distance generates the vehicle orientation (cf. Eq. (1.6)), and the nonholonomic path is differentially determined by Eq. (1.1) and (1.2). Thus, the evaluation of the proposed path can be performed by comparing the original path and the generated path.

For that purpose, two kinds of *one-clothoid* **pCCP** method are tried to overcome the slalom-like track, where the first one is that each clothoid is generated under the given (or known) curvatures at both ends, thus *curvature-following*



(a) Obtained path (b) Curvature diagram

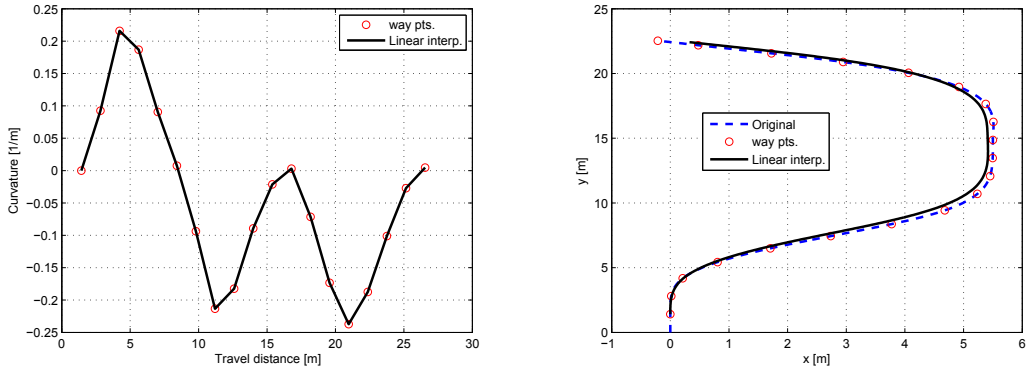
Figure 4.21: Obtained dataset of the **SS-track**

method (**Case 4A**) and the latter is that each clothoid is generated under the given orientation constraint, *orientation-following* method (**Case 4B**). One clothoid (or added arc) is generated between two sequential way points, where each way point has its own pose (position and orientation) and curvature. Here, how to select the way point is actually a major issue, since the distance between each pair of way points could determine the boundary condition and thus **pCCP** problem type as well as algorithmic resolution such as iterative variation, convergence threshold and initial assumption. A guideline to determine way point is to make the **Assumption 1** (cf. **Section 2.1.4**) be followed.

In this subsection, the way point is equally distanced (or acquired with equal sampling time) by 1.39 m in average and **pCCP** problem is sequentially solved to reconstruct the original path. The situation can be described by a leader-following example. A vehicle moves in advance while a following vehicle traces the position of the former vehicle periodically. The following vehicle reproduces the path of the leader vehicle by reconstructing the curvature profile as well as controlling the steering. As described in **Section 2.1.4.1** and **2.1.4.2** with the summary in Tab. 2.3, *curvature-following* method requires information about the curvatures at all the way points, whereas the *orientation-following* method needs desired orientations at all the way points.

Curvature following method: using Case 4A From the given way points, curvature is calculated by the nearest two points, forward and backward ones [Nunez 07, Zhu 14, Seo 14]. In this problem, orientation change δ for a generated clothoid is dependent on the radial distance (cf. D_δ in Eq. (3.3b) (p.81)) , thus

θ_f is not fixed and it is not gauranteed that any desired orientation at each way point is satisfied.



(a) Linearly interpolated curvature (b) Reconstructed path

Figure 4.22: Path reconstruction from curvature of the way points in **SS-track**

Fig. 4.22 illustrates the curvature diagram for the given way points and its reconstructed path by nonholonomic vehicle motion. The curvature diagram (a) is constructed from 20 way points and then connected with linear interpolation. The corresponding nonholonomic path is generated, however it has difference to the original path (or way points) where its statistical values are listed in Table 4.3.

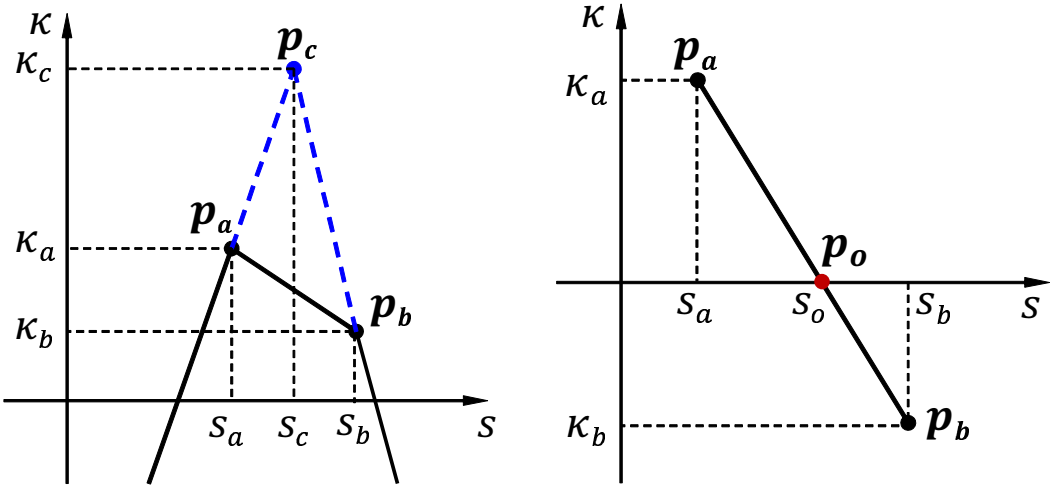
Table 4.3: Path reconstruction result ($[m]$)

Performance	Min.	Max.	Avg.	Std.	Sum
Curvature following method #1	0.009	0.221	0.109	0.060	1.9707

Table 4.3 records 5 representative results of minimum, maximum, average, standard deviation and summation values on the reconstructed path that is deviated from the given way points. Even though the reconstructed path is generated from the linearly interpolated curvature diagram, thus it assures to be continuous curvatured; however the path has more than 0.1 m of distance error in average which is not negligible.

One of the major causes of those errors is the difference between the original curvature and the interpolated curvature which can be seen from Fig. 4.21(b) and Fig. 4.22(a). Since the curvature estimated at each way point is different to the

original curvature, it is inevitable to avoid truncation error. Specifically, notable difference for the positive or negative peak values between two curvature diagrams can be seen at the original 0.25/-0.24 and 0.22/-0.23 at the interpolated one. Therefore, from the above analysis, the curvature following method requires more accurate curvature estimation to obtain less erroneous path. To begin with the path generation of the curvature following by one clothoid generation of **Case 4A**, zero crossing point in the curvature diagram is required to formulate boundary condition, i.e. zero curvature value at one end. Two curvature modifications which find peak and zero-crossing points are applied using linear interpolated curvature diagram as follows.



(a) Interpolated peak $\boldsymbol{p}_c$ (b) Zero-crossing point $\boldsymbol{p}_o$

Figure 4.23: Curvature estimation for reshaping procedure

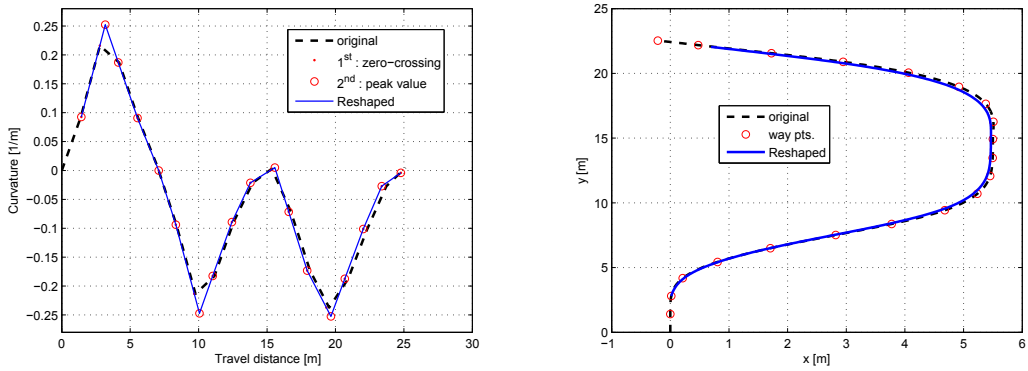
In Fig. 4.23(a), estimating method for the peak curvature value $\boldsymbol{p}_c$ from $\boldsymbol{p}_c$ and $\boldsymbol{p}_c$ is depicted; the peak is found at s_c using Eq.(4.2a) and Eq.(4.2b) respectively.

$$\kappa_c = m_a(s_c - s_a) + \kappa_a = m_b(s_c - s_b) + \kappa_b, \tag{4.2a}$$

$$s_c = \frac{m_a s_a - m_b x_b + \kappa_b - \kappa_a}{m_a - m_b}, \tag{4.2b}$$

To find the zero-crossing point s_o from $\boldsymbol{p}_a$ and $\boldsymbol{p}_b$, Eq.(4.3) is used as follows.

$$s_o = s_a + \frac{(s_b - s_a)|\kappa_a|}{|\kappa_a| + |\kappa_b|}. \tag{4.3}$$



(a) Reshaped curvature diagram (b) Reconstructed path

Figure 4.24: Path reconstruction from curvature reshape in **SS-track**

By reshaping the curvature at given way points through zero-crossing (1^{st}) and peak finding (2^{nd}), the reshaped curvatures are provided to generate the sequential **pCCPs** under the curvature-following method as shown in Fig. 4.24. Compared to the path drawn in Fig. 4.22(b), the obtained path by reshaping curvatures shows less deviation to the given reference path. The deviation of the obtained path from all the way points are measured and listed in Table 4.4.

Table 4.4: Path reconstruction result by curvature reshaping $[m]$

Performance	Min.	Max.	Avg.	Std.	Sum
Curvature following method #2	0.005	0.2584	0.0645	0.0577	1.1605

The representative values in Table 4.4 records enhanced performance especially in the average and summation of distance errors as 0.065 m and 1.1605 m respectively, which achieves 40.4 % and 41.1 % diminution respectively.

The procedure is given in **Algorithm 17**.

Orientation following method: using Case 4B For the given **SS-track**, the sequential **pCCPs** generation of one-clothoid generation using orientation-following method is applied. The orientation-following method is based on the algorithm in **Case 4B** and the boundary condition at way point is different to the curvature-following method, i.e. the target position and (desired) orientation is

Algorithm 17 **pCCPs** global path planner 3 (Curvature following method)

Require: $P_n, \kappa_n, n \in [1, ..., N]$ ▷ Given way points, N : number of points

for $n \in [1, ..., N-1]$ **do**
  $\kappa_i \leftarrow \kappa_n, \kappa_f \leftarrow \kappa_{n+1}$
  $P_{tg} = P_{n+1} - P_n$
  $P_{tg} = \texttt{rotate\_th}(P_{tg}, \frac{\pi}{2} - \theta_n)$ ▷ **Appendix B.2**
  procedure CLOTHOID1A($P_{tg}, \kappa_i, \kappa_f$)
    $\alpha = \frac{\kappa_f^2}{2\delta_f}$ ▷ Eq. (1.14)
    $\delta_i = \frac{\kappa_i^2}{2\alpha}$
    while $D_\theta < \varepsilon_\theta$ **do** ▷ **Criteria 2**
      while $D_\delta < \varepsilon_\delta$ **do** ▷ **Criteria 1**
        Iteration # 1
      end while
      Iteration # 2
    end while
    $d\theta = \delta_f + \Delta\theta$ ▷ **Iteration 2**
    return $C_{sol}, d\theta$ ▷ $C_{sol}(x_{sol}, y_{sol})$ ▷ Solution obtained
  end procedure
  $C_n = \texttt{rotate\_th}(C_{sol}, -(\frac{\pi}{2} - \theta_n))$
  $C_n = P_n + C_n$ ▷ Shift based on the way pt.
  $C_{path} = [C_{path}, C_n]$ ▷ Addition for the obtained path
  $\theta_{n+1} = \theta_n + d\theta$ ▷ for next iteration
end for
return C_{path} ▷ Obtained sequential **pCCPs**

given. The orientation at each way point can be estimated by tangential direction at the point, thus, the orientation is calculated by nearest two points at both sides as averaging the both orientation angles.

The procedure for the orientation-following method is given in Algorithm 18.

Algorithm 18 **pCCPs** global path planner 4 (Orientation following method)

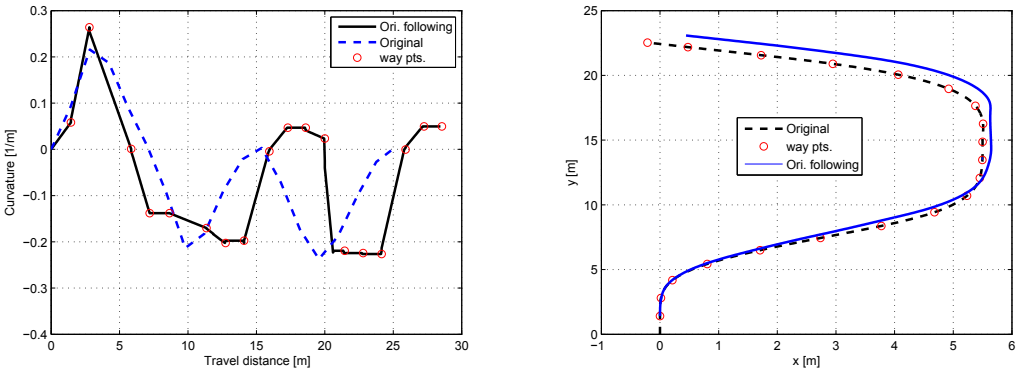
Require: $P_n, \delta_n, n \in [1, ..., N]$ ▷ Given way points, N : number of points

for $n \in [1, ..., N-1]$ **do**
  $\delta_f \leftarrow \delta_n$
  $P_{tg} = P_{n+1} - P_n$
  $P_{tg} = \texttt{rotate\_th}(P_{tg}, \frac{\pi}{2} - \theta_n)$ ▷ **Appendix B.2**
  $\kappa_i \leftarrow \kappa_n$
  $\kappa_f \leftarrow \kappa_f^*$ ▷ Assumption by δ rate
  procedure CLOTHOID1B($P_{tg}, \kappa_i, \delta_f$)
    $\alpha = \frac{\kappa_f^2}{2\delta_f}$ ▷ Eq. (1.14)
    $\delta_i = \frac{\kappa_i^2}{2\alpha}$
    while $D_\kappa < \varepsilon_\kappa$ **do** ▷ **Criteria 3**
      Iteration # 3
    end while
    $d\theta = \delta_f + \Delta\theta$ ▷ **Iteration 2**
    return $C_{sol}, d\theta$ ▷ $C_{sol}(x_{sol}, y_{sol})$ ▷ Solution obtained
  end procedure
  $C_n = \texttt{rotate\_th}(C_{sol}, -(\frac{\pi}{2} - \theta_n))$
  $C_n = P_n + C_n$ ▷ Shift based on the way pt.
  $C_{path} = [C_{path}, C_n]$ ▷ Addition for the obtained path
  $\theta_{n+1} = \theta_n + d\theta$ ▷ For next iteration
  $\kappa_{n+1} \leftarrow \kappa_f$ ▷ For next iteration κ_i
end for
return C_{path} ▷ Obtained sequential **pCCPs**

where, κ_f^* is assumed by following rule.

$$\kappa_f^* = \begin{cases} 5\kappa_i, & \text{if } \delta_{n+1} \geq \delta_n, \\ \frac{1}{5}\kappa_f, & \text{else } \delta_{n+1} < \delta_n. \end{cases} \tag{4.4}$$

Fig. 4.25(a) depicts the obtained path and curvature diagram. Observing the curvature diagram shown in Fig. 4.25(b), the obtained curvature has nontrivial difference to the original curvature shown in Fig. 4.22. Such a difference also



(a) Curvature diagram (b) Reconstructed path

Figure 4.25: Results of orientation following method in **SS-track**

causes the resultant path to deviate from the original path as Fig. 4.25(b). However, the obtained curvature diagram shows the curvature continuity along the whole travel length and peak curvature values are not so different to the original or reshaped curvatures in Fig. 4.22(a) and Fig. 4.24(a) respectively (cf. Table 4.5).

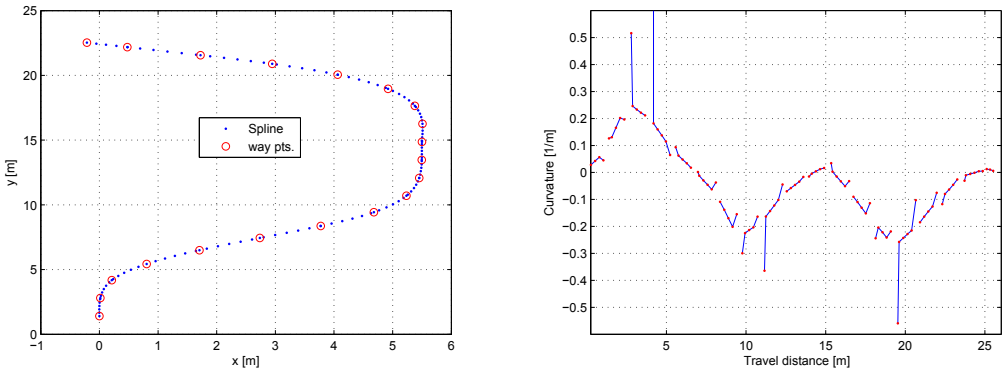
Table 4.5: Path reconstruction result by orientation following method $[m]$

Performance	Min.	Max.	Avg.	Std.	Sum
Orientation following method	0.0067	0.7863	0.2844	0.2789	5.1194

All data in Table 4.5 indicate that the result of the orientation-following method is inferior to the curvature-following method. One of the major reasons can be a difference between the obtained curvature and the original curvature. To overcome this difference, additional consideration for the way point selection or optimization of the way point distance with the algorithmic convergence threshold; this issue requires to be transfered as a future work.

To validate the obtained curvature and path result, another smooth path generation method, *Periodic Interpolating Cubic Spline Curve*[1] (*Cubic spline method*) is applied in the same track.

[1]The MATLAB® functions, *cscvn* and *fnplt* are used to generate and evaluate at the given way points respectively.



(a) Reconstructed path (b) Curvature diagram

Figure 4.26: Results of cubic spline method in **SS-track**

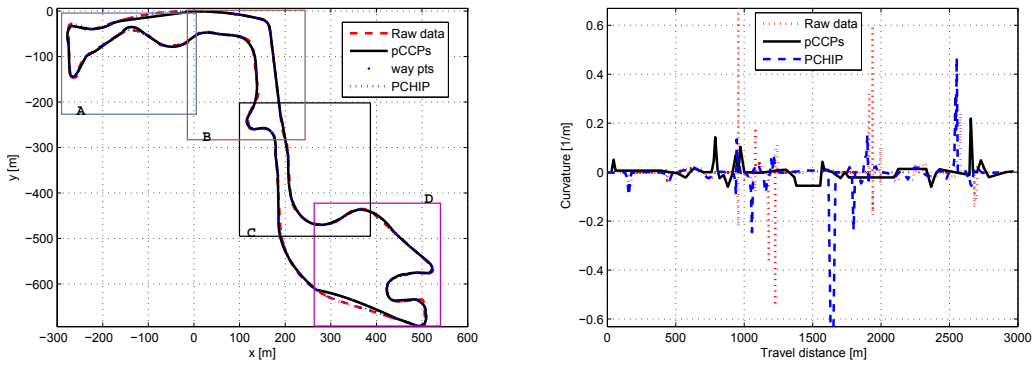
Fig. 4.26 illustrates the resultant path and curvature for cubic spline method, where the path is almost close to the original path, however the curvature has discontinuities and sudden jumps (which is not directly followable by smooth steering for the vehicle) at every way point as shown in Fig. 4.26(b).

4.3.3.2 Long racing track: LR-track

One of the most slalom-like track is the F1 formula racing route. To evaluate the proposed sequential **pCCPs** in a extreme environment, the *Monte Carlo* track [GP 16] (Fig. 4.5) was tried. For the long racing track (**LR-track**), a raw experimental data about track following was taken from a vehicle dynamics simulation software, ve-DYNA Entry® [Tesis-DYNAware 16]. In the software, the GPS position data along the course is provided by the inherited database and it is converted into the target path for vehicle's following by using several inherited algorithms. Under some constraints and setting in the program, a smooth path using 50 way points are obtained by *Piecewise Cubic Hermite Interpolating Polynomial* (**PCHIP**) algorithm[2]. Specific procedures about the data acquisition and conversion into the target path are briefly described in **Appendix B**.

Fig. 4.27(a) illustrates the obtained paths for raw data, **pCCPs** and **PCHIP** for the **LR-track**, where initial position for the path is set as $(0, 0)$. In Fig. 4.27(b), the obtained curvature diagrams are shown. The results verifies that the proposed path maintains smaller curvature for its maximum and minimum peaks through the travel length compared to the other paths, which is also listed in Table. 4.6.

[2]The same function can be applied by *pchip* function in MATLAB.



(a) Path results (b) Curvature diagram

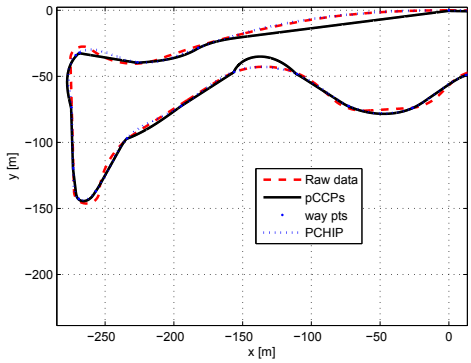
Figure 4.27: Obtained paths and curvatures in **LR-track**

Analyzing curvature diagram in Fig. 4.27(b) with Table. 4.6, the raw data and **PCHIP** have discontinuities as well as jumped paeks over the length, but the proposed path has much lower peak. The maximum curvature value (2.6576) can be found at around 900 m for the raw data path, and the minimum peak (-1.0861) is recorded at around 1650 m. For **pCCPs** the maximum peak(0.2188) is rated at the length of 2700 m, however the value is less than the half of the peak in **PCHIP** and it is proved that the proposed **pCCPs** is the most superior performance to generate low curvatured path, thus provides the least lateral acceleration to the vehicle under the same speed.

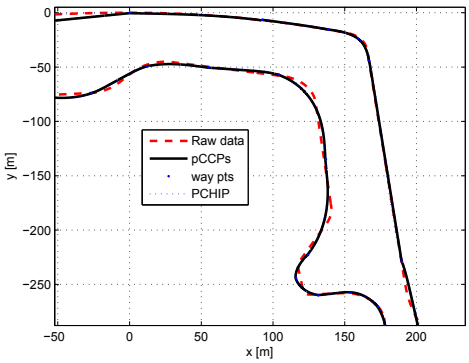
Table 4.6: Comparison results for the obtained curvature diagrams [$1/m$]

Curvature	Raw data	PCHIP	pCCPs
Max	2.6576	0.4680	0.2188
Min	-0.5320	-1.0861	-0.0601

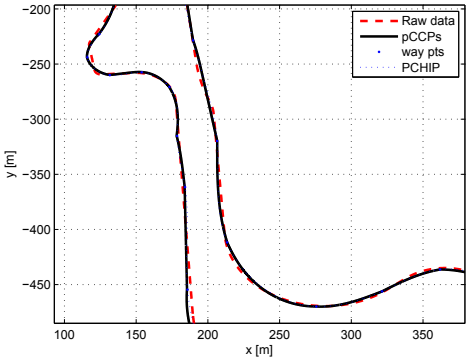
In Fig. 4.28, **LR-track** is divided into four regions (A, B, C and D) to show each path more clearly and to compare the paths. From the region-A to region-D, it can be seen that the **pCCPs** has larger discrepancy to the raw data path than **PCHIP**, thus it has weakness in the closeness to the original path. One of the major drawback could be from using the way points and desired orientation of **PCHIP** into the problem of sequential **pCCPs** generation. From the results analysis, it can be also derived that the way points selection is the important



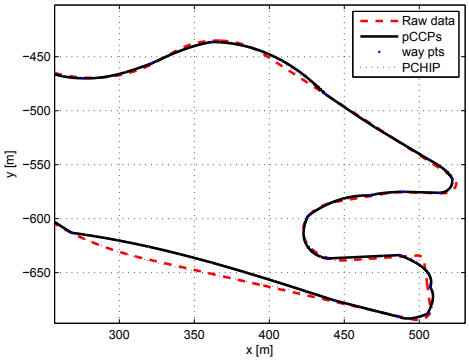
(a) Region - A



(b) Region - B



(a) Region - C



(b) Region - D

Figure 4.28: Enlarged views on the obtained paths in **LR-track**

problem to define the problem of sequential **pCCPs**, and it is also required to assign the boundary conditions to optimize the number of **pCCP** segment with minimal distance deviation from the reference path as well as minimal curvature variation. For the given **LR-track**, the curvature following method is not tried here since the exact curvature can not be extracted in this path and the efficient boundary conditions are also difficult to define. Thus, these issues needs to be transfered to be a future problem.

4.4 Conclusion

In this chapter, global path generation/reconstruction for several path examples that are difficult to define boundary conditions for **pCCP**, has been addressed. Such cases could be found when a vehicle follows a leader-vehicle with a distance where the follower-vehicle traces the leader-vehicle path by keeping the curvature continuity (or steering wheel smoothness). For this purpose, two specific methods, curvature-following and orientation-following method by non-elementary one clothoid generation was introduced and applied to two examples of short and long tracks. In curvature-following method, the extraction of curvature at the way point (or boundary condition) is important to increase the closeness to the original path, whereas the orientation-following method is easy to deviate from the original path without way points optimization.

The sequential **pCCPs** gets to be more complicated according to the shape of the given target path, thus the problem requires more refined constraints and optimization guidance to obtain more closer path to the original. This issue remains as a future work. To overcome this current limitation for the proposed one-clothoid generation algorithm and obtain a perfect solution, human aspect into the solution methodology could be a useful guidance in formulating optimization rule in clothoids composition. These points of consideration are tackled by introducing human driving pattern in the next chapter.

Chapter 5

Bio-inspired Obstacle Avoidance Strategy based on Human Driving Behaviors

5.1 Introduction

To cope with the complexity in the algorithmic procedure and to enable generated paths to have a human favorable quality and comfort, some works have sought for enhanced path generation methods by investigating human driving patterns. For example, in [Sharp 00] and [Macadam 10], desired paths were made by cubic splines in lane change maneuvers. In both works, the human steering pattern was divided into *avoidance* and *stabilization*, and each steering control rule was then applied by minimizing the lateral deviated distance of the vehicle to the desired path. In [Markkula 14], the steering control was executed by tuning a few gain parameters and comparing the resultant steering control with human driver models. The above studies focused on the design of the steering control from human driving data. In [Edelmann 07], a human driver model for the steering control to follow a curved lane was formulated to enhance the simulation reality. It found that the human driver used preview information ahead of the recognized curvature on the road and controlled steering of the vehicle with a reaction time delay. In [Petrov 13], an adaptive steering controller was applied to an autonomous lane change maneuver to avoid a static obstacle, where the desired trajectory was generated by a *cycloid*. When the desired trajectory was made to avoid static obstacle, a circular path was considered with a maximum steering angle such that the path kept a minimum distance from the obstacle.

Some studies have also performed to implement human driver intentions or

moods into a steering control strategy. Human driving patterns are addressed in [Yuhara 01] and a vehicle steering control strategy is discussed in [Bevan 07]. In the work performed in [Yuhara 01], the controller adjusted the steering control gain according to the driver mood from *aggressive* to *leisurely*. The authors in [Bevan 07] made a steering controller by differentiating the degree of emergency in the lateral collision avoidance control, from the emergent steering for avoiding collision to the gentle steering to stabilize for an approaching lane. The proposed schemes were tested on a severe lane change track of *ISO 3882-2*. The authors in [Yamakado 08] extracted control parameters from experimental investigations in which an expert human driver combines the longitudinal and lateral controls using jerk information during a cornering maneuver. In [Hayashi 12], steering and braking control was implemented for collision avoidance, where the reference path is generated by combination of line and arc segments. In this methodology, the reference path is non-continuous (especially in the point between line and arc) so that a controlled result could not be assured to be exactly same as the reference path. The authors in [Rothhamel 14] investigated drivers' patterns in heavy vehicle's steering and found that experienced drivers perform with more margin of safety and low lateral acceleration during cornering motions.

In [Wilde 09], a **CCP** was investigated with human factors. It presented a smooth and easily drivable path (*Wilde*'s path) which is composed by four identical clothoids using rational approximation functions. It also imposed a minimum *sharpness* (or rate of curvature) constraint which closely resembles natural and safe human driving. The algorithmic procedure was fast, but the path was only useful for the lane change maneuvers in obstacle free environments. Related to the clothoids composition in the path generation problem, in [Gim 14a] and the previous chapter (cf. **Chapter 2**), the parametric **CCP** algorithmic solution was proposed by iterative regulating on the clothoid parameters. That solution path has a weakness in its algorithmic efficiency, i.e., the number of iterations could increase in order to obtain a solution according to given boundary conditions. Even if the proposed solutions in [Wilde 09] or [Gim 14a] assure a smooth path for no1nholonomic vehicle motion, they are not optimized for the human (passenger) comfort. Furthermore, it is better to consider the influence of environmental conditions on human comfort and steering patterns from urgent, imminent (quick/sharp turning) to relaxed, leisurely responses (slow/gentle turning).

This chapter is linked to the authors' previous works given in [Gim 14a, Gim 14b, Gim 17], but this work is different to the previous work from the viewpoint of problem statement and its optimization guidance. First of all, the major performance objective is the application of the resulting solution to the lane change maneuver and also that the problem is defined with constraints for human factors while enhancing algorithmic efficiency and ride comfort. To be more

specific, to reduce the number of iteration loops in the algorithmic procedure, parameter constraints are derived from: the data analysis of human driving patterns such as human response; comfort and reactions to environmental conditions during the vehicle motion. For the constraints, human driving pattern data are formulated and deduced from the designed experimental set-up. Eventually, a smooth path solution is obtained by imposing parametric constraints to solve online the lane change maneuver problem.

In this chapter, a new solution inspired by human driving patterns is proposed, it is named as *human inspired-Continuous Curvature Path* (**h-CCP**). This study first addresses the algorithmic complexity in solving the clothoid path generation problem and develops a compact and fast procedure to obtain the **h-CCP**. At first, the main ideas used to resolve the issue are obtained from the experimental data of human driving. A lane change maneuver for obstacle avoidance is tested with human drivers and a traditional shape/pattern of the curvature diagram is extracted to emulate the proposed path solution. After the path solution is obtained, the velocity planning is integrated with vehicle dynamics simulation to evaluate its effectiveness on passenger comfort. The velocity planning is applied from a normal human driving data of accelerating/decelerating patterns.

5.2 Experimental database construction of human driving expertise

This section proposes a solution inspired by human driving patterns, which is named as *human inspired continuous curvature path* (**h-CCP**). The algorithmic complexity in solving the clothoid path generation problem is addressed and a compact and fast procedure to obtain the **h-CCP** is developed. Firstly, some ideas used to resolve this issue are obtained from the experimental data of human driving and then, the lane change maneuver for obstacle avoidance is tested with several human drivers and some traditional shapes/patterns of the obtained curvature diagrams are extracted in order to be emulated in the proposed algorithms.

The resulting solution constructs a smooth and safe path for nonholonomic car-like vehicles which could also contribute to passengers' ride comfort with human favorable driving. This work aims to propose a complementary solution methodology to enhance algorithmic efficiency and reliability for the proposed **pCCP**. An important performance objective of the **h-CCP** is ensuring that it should provide human favorable driving and ride comfort.

In this section the evolution of the curvature will depends on actual human behavior to avoid any hindered obstacles. This knowledge will be caught from the human expertise. To this respect, an experimental investigation on human driving patterns provides guidance or constraints to obtain the desired solution.

To produce a human favorable path solution for obstacle avoidance and find inspiration from human driving patterns, three experiments were done. The first experiment was planned for obstacle avoidance and was performed in a vacant parking lot. The second experiment was undertaken to examine lane changing with obstacle avoidance on a city-road environment and the third experiment was performed in a road without obstacles in order to observe.

Note that these experiments were not for making any generalization for all humans, but just to find normal patterns for sampled human applicants. Thus, for this purpose; before starting each test, each human driver was instructed on how to perform the test. All the guidelines and constraints from each experiment were designed to determine useful human driving patterns which are then implemented into the **h-CCP** solution (cf. **Section 5.3**).

The used experimental vehicle is shown in Fig. 5.1. The dimension of this vehicle is 1.5×1.4×3.1 (width × height × length in $[m]$) with a wheelbase of 2.18 m. It has sensors for recording the steering/wheel angular, position and orientation by integrating the data from an IMU (Inertial Measurement Unit)/GPS and odometers.

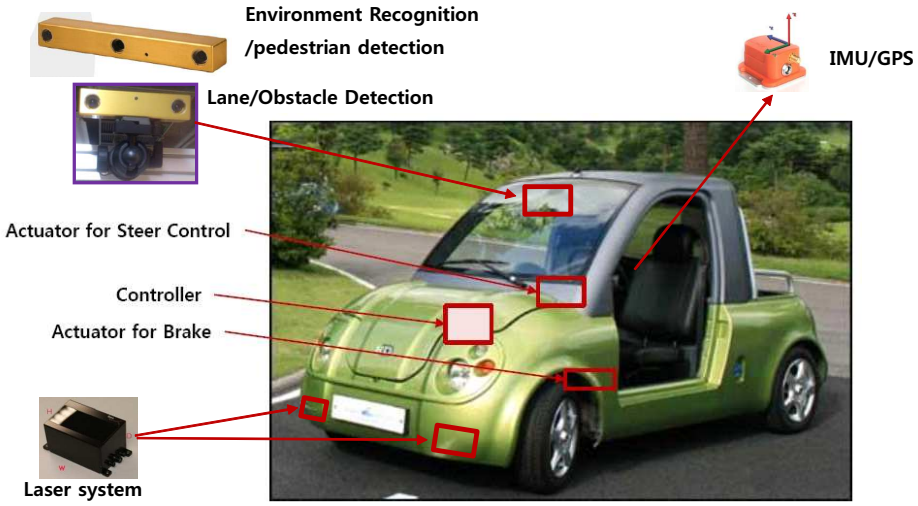


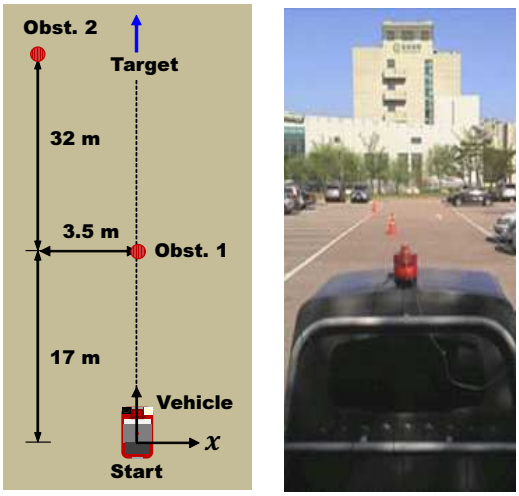
Figure 5.1: Experimental vehicle

5.2.1 Experiment 1: Obstacle avoidance by human drivers

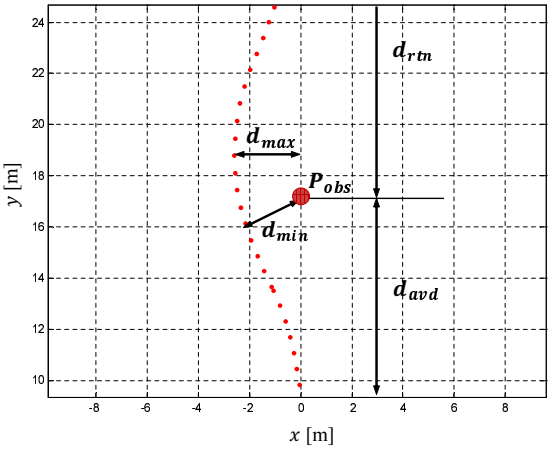
An obstacle avoidance test was performed by human drivers. The obstacle is placed on the front side of the vehicle at its initial position and the driver was told to avoid the obstacle on the left-side and then return to the center (initial) line until reaching the target position. The obstacle is a cone-type structure with $0.5 \times 0.5 \times 1.0$ $(width \times length \times height)$ in $[m]$ so that the drivers could recognize its size and entire shape as well as its position. At first, twenty persons were applied to this experiment but five drivers were chosen who were showing coherent patterns in the several repetitive trials [1], where applicants' ages range from 22 to 45 years and pure driving experience from 6 months to 10 years. All drivers were instructed to drive the vehicle as fast and close as possible with minimum discomfort. Each person tried the test several times and then chose the most satisfying result by his or her judgement.

In Fig. 5.2(a) and (b), the geometric setup for the given **Exp.1** with its picture shot is shown, where the initial position of the vehicle is its origin in the coordinate frame; and the first obstacle (**Obst.1**) is located at 17 m from the front of the vehicle and the other obstacle (**Obst.2**) is located at the left side of the target to instruct to the driver the distance to the target position of 32 m

[1] Actually, the human sampling or filtering from massive driving data is not an easy task and could be addressed in more details in future works, while interacting with researcher in the domain of human factor area.



(a) Experimental setup and its picture



(b) Path evaluation parameters

Figure 5.2: Obstacle avoidance by human drivers: **Exp.1**

behind the **Obst.1**. In Fig. 5.2(c), four geometrical parameters are defined to evaluate the performance of the obtained path data for each human driver, where the *avoidance distance* ($\boldsymbol{d_{avd}}$) refers to the distance between the obstacle and the position at which the vehicle starts to turn, and the *return distance* ($\boldsymbol{d_{rtn}}$) is the distance between the obstacle and the return position which is the same as the initial vehicle direction. The *minimum distance*, $\boldsymbol{d_{min}}$ is found at the position where the vehicle's right side is the closest to the obstacle and the *lateral distance* $\boldsymbol{d_{max}}$ is the maximum lateral position from the line of $x = 0$ in the local coordinate frame.

Under the experimental set-up described above, the **Exp.1** is performed as follows. From the **Start** position, the human driver starts to move and accelerates as much as possible until they need to turn. To avoid collision, the driver turns the steering handle to the left and then return to its initial direction to approach the given target. When turning to avoid as well as returning, each driver try to his or her best to reduce discomfort.

Table. 5.1 records the obtained path data by defined four parameters Fig. 5.2(b) and its evaluation is performed by summing all the parameter values for each human.

In Table. 5.1, five applicants are listed up by evaluated points, **Total**, as the point of **Total** is low, the performance of driving is high. Thus, drivers are ranked from the applicants ⓐ to ⓔ for the **Exp.1**. Even if all parameters are different to all applicants, a pattern could be found that the most experienced driver (ⓐ)

Table 5.1: Results analysis of **Exp.1**

Applicant	(1) d_{avd}	(2) d_{min}	(3) d_{max}	(4) d_{rtn}	**Total** *	Period (years)
ⓐ	6.8	1.2	1.3	5.8	15.1	10
ⓑ	7.8	1.8	1.9	7.1	18.6	3
ⓒ	8.0	1.7	1.8	7.9	19.4	5
ⓓ	10.3	2.1	2.4	9.0	23.8	1.5
ⓔ	12.8	3.7	3.8	12.2	32.5	0.5

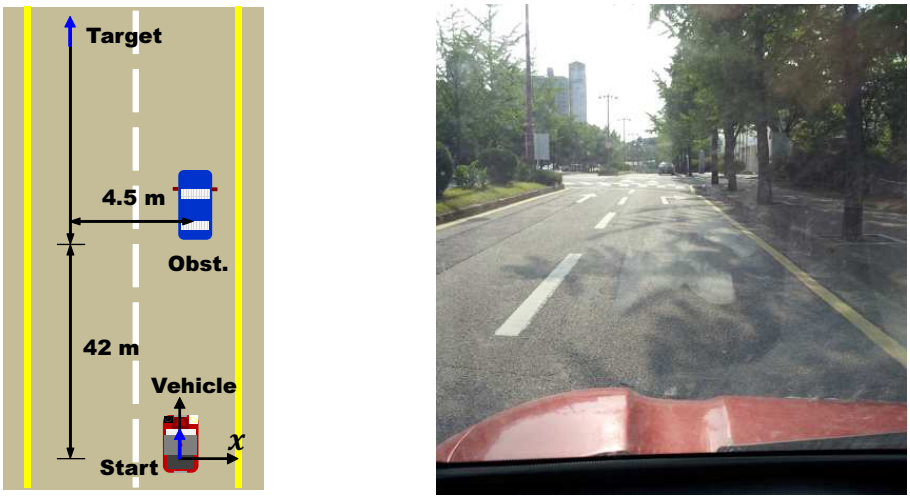
Total : ((1)+(2)+(3)+(4))

records the lowest point of **Total**, thus the highest performance, whereas the least experienced driver (ⓔ) records the worst performance from the point **Total**. From Table. 5.1, it is also evident that experienced drivers (more than 3 years driving) follow paths with shorter distances to the obstacle and generate inside turning path than the less-experienced drivers. Actually the four parameters has a close relation to the total length of the path, since as all parameter values increase, the total length for the maneuver also increase. From the obtained evaluation result, a human driving pattern could be derived such that the experience driver makes the obstacle avoidance path be shorter by the closest turning around the obstacle than the unexperienced drivers. However since the speed or travel time for the **Exp.1** is not included in the maneuver, the speed or velocity for the maneuver requires to be considered for more reliable knowledge.

5.2.2 Experiment 2: Lane change for obstacle avoidance

In the experiment 2 (**Exp.2**), lanes on city roads are used for a lane change maneuver test and an obstacle (parked vehicle) was placed on the right side road with 42 m far front of the vehicle start position. The obstacle is so large that human driver could not oversee behind the obstacle (e.g. the obstacle blocks the driver's *view*), where one could not recognize whole shape of the obstacle and it is also difficult to expect what other obstacle remains behind. The drivers include five drivers who are selected from **Exp.1**. In this experiment, each driver was ordered to increase the vehicle speed sufficiently and then maintain the speed (the avoidance speed $\boldsymbol{v_{avd}}$) until the vehicle starts to turn at the steering angle. The driver is also told to approach the obstacle as close as possible and then keep going without pushing the accelerator or brake pedal during lane change maneuver. These constraints are used to investigate the desirable relative distance between

the vehicle and the obstacle and to analyze the geometric pattern of the resultant path. Each driver is also instructed to test the same lane change maneuver with five steps of $\boldsymbol{v_{avd}}$ as *slowest*, *slow*, *middle*, *fast* and *fastest*. Even if each step for the $\boldsymbol{v_{avd}}$ is determined by each human driver, thus it is different to other drivers, the pattern between $\boldsymbol{v_{avd}}$ and the $\boldsymbol{d_{avd}}$ could be derived.



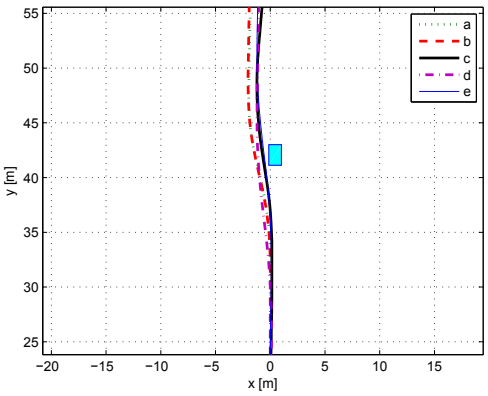
(a) Experimental setup

(b) View of environment

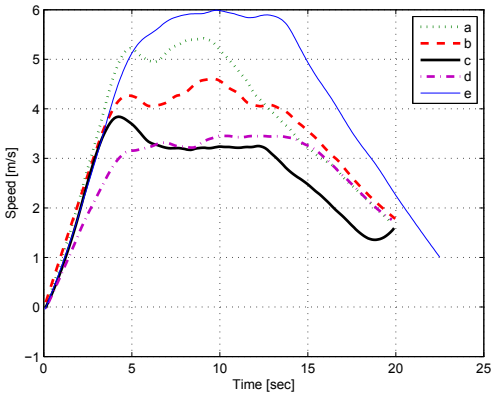
Figure 5.3: Lane change maneuver for obstacle avoidance: **Exp.2**

With the experimental setup depicted in Fig. 5.3, results for human lane change maneuver are shown in Fig. 5.4 for the path (a), speed (b), steering (c), and captured scenes from driver side (d). All the drivers started to turn from 7 seconds and maintained the maneuver for 8 to 10 seconds. From the driver's view (cf. Fig. 5.4(d)), the sight behind the obstacle could be recognized (5^{th} scene) after some seconds from avoidance starting, and this seems to make the whole path be non-symmetric along the travel time or distance. A steering pattern could be found in Fig. 5.4(c), where two peak values are recorded from the start position and the first peak is sharper than the following peak (reversed shape). Here, the sign (or direction) of steering angle is reverse to that of curvature due to the wheel encoder sensor data in the experimental vehicle, i.e. steering is positive for clockwise rotation.

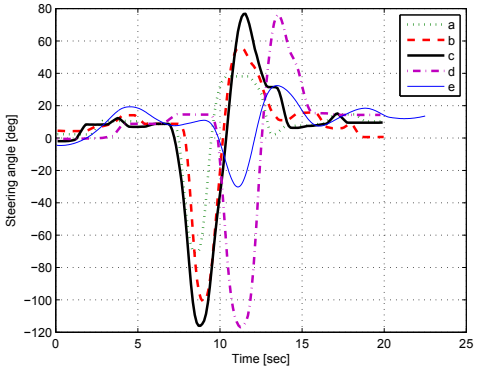
Fig. 5.6 include five results of the steering angle along the travel length for five drivers, where each subfigure has in itself five datasets of steering angle record for different $\boldsymbol{v_{avd}}$ $[m/s]$. In Fig. 5.6, the avoidance distance $\boldsymbol{d_{avd}}$ according to the vehicle's maximum speed $\boldsymbol{v_{avd}}$, increases for all the drivers from ⓐ to ⓔ. The steering records of each driver (ⓐ to ⓔ) are displayed in five subfigures of



(a) Paths for five drivers



(b) Speeds for five drivers



(c) Steering angles for five drivers

(d) View of driver

Figure 5.4: Human driving data results for lane change maneuver

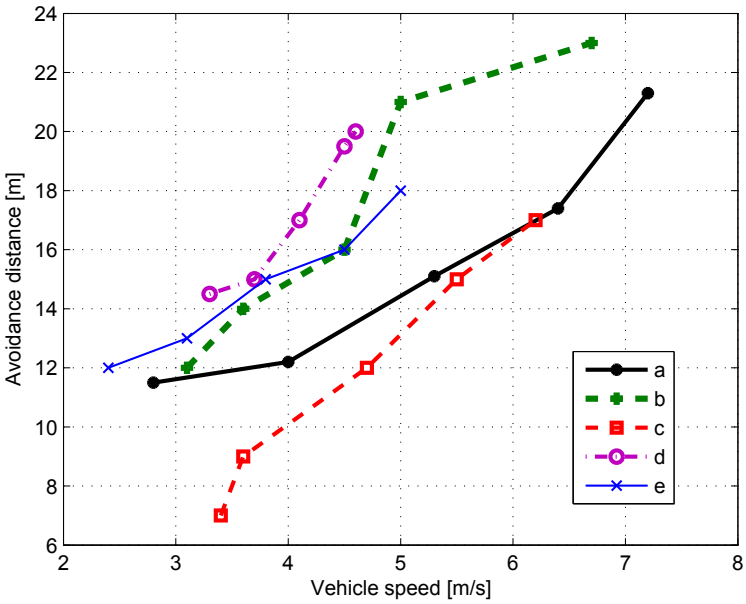
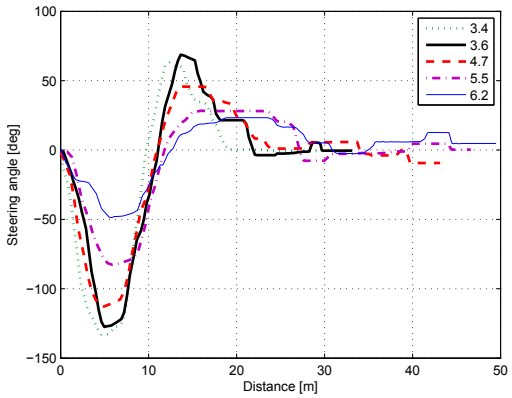


Figure 5.5: Avoidance distances according to vehicle speeds for five drivers

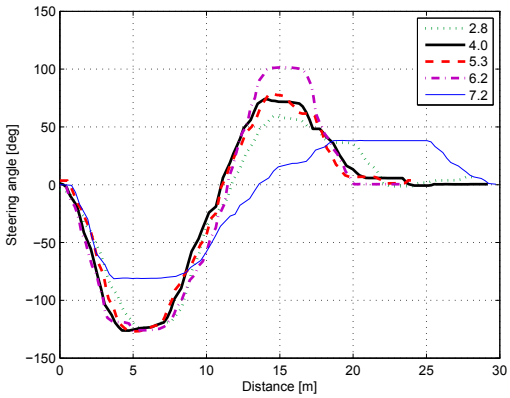
Fig. 5.6 respectively, and each subfigure contains several test results indexed by different $\boldsymbol{v_{avd}}$ values. To extract the steering pattern effectively, only the steering data during turning motion are plotted after aligning on the same origin position of *x-axis*. Using the data set of driver-ⓐ shown in Fig. 5.5 (black lines with cross markers), the relation between $\boldsymbol{v_{avd}}$ ($[m/s]$) and $\boldsymbol{d_{avd}}$ ($[m]$) can be modelled by line fitting as follows.

$$\boldsymbol{d_{avd}} = 2.67\ \boldsymbol{v_{avd}} + 1.31. \tag{5.1}$$

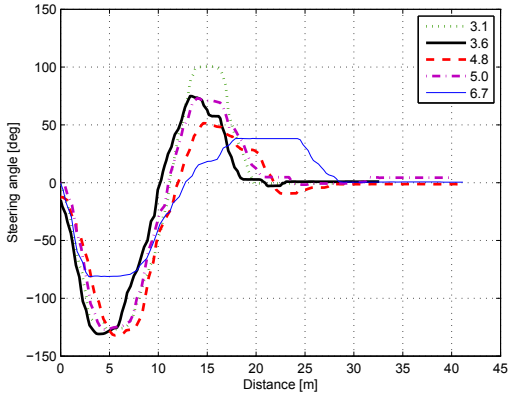
Note that even if the pattern shown in Eq. (5.1) is not exactly same to other drivers, it could be generalized that $\boldsymbol{d_{avd}}$ increases proportional to $\boldsymbol{v_{avd}}$ for all human drivers. It could be also found From the data analysis of Fig. 5.6 that two different turning motions (or peaks) exists during lane change maneuver and this could be classified as two different modes, which could be named as *avoidance* and *recover* mode. This result is important in that human lane change maneuver which has a non-symmetric shape, which was symmetrically treated in previous works [Hayashi 12, Chen 13, Markkula 14] as well as in [Wilde 09]. Although these patterns of **Exp.2** are restricted in the strictly planned experiments, it could be a good guidance or constraints to construct an optimization problem for path generation scheme.



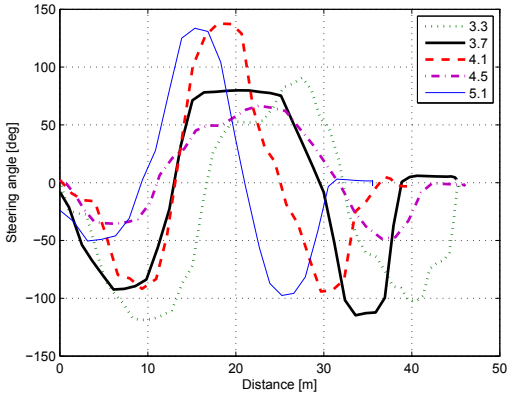
Driver ⓐ



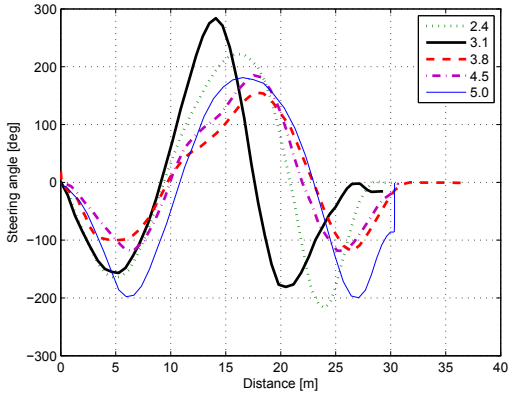
Driver ⓑ



driver-ⓒ



Driver ⓓ



Driver ⓔ

Figure 5.6: Steering patterns for drivers with different speeds $[m/s]$

5.2.3 Experiment 3: Driving on obstacle-free road

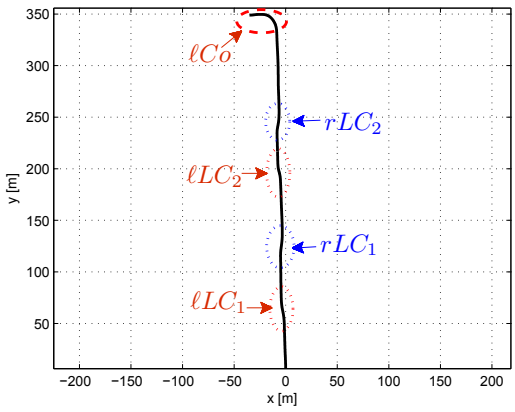
For the third experiment, free road driving was executed in the SungKyunKwan University campus of Korea. The road track has one lane per path and all experimental data are gathered under controlled obstacle free conditions. Even though the constraints which were used for **Exp.1** and **Exp.2**, no constraint was set for **Exp.3** for the driving and the driver recorded path, steering, and velocity of the vehicle before stopping.

This experiment was designed to understand human driving pattern in terms of velocity changes (acceleration/deceleration) under the various cornering or turning conditions. Three experienced drivers were selected and driver performed pretest more than three times to become accustomed to the complete road geometry and ground condition. Three tracks, **Track #1**, **Track #2** and **Track #3**, were applied to collect free driving data where the **Track #1** includes straight double lanes and the **Track #2** has a cornering road geometry of one steep and one less steep curve, and the **Track #3** has a square shape, thus many 90° angles were involved in the cornering road geometry. When the analysis was focused on the velocity changes while turning, little difference was observed between each driver's data. Thus, one representative record for each track was chosen to be analyzed.

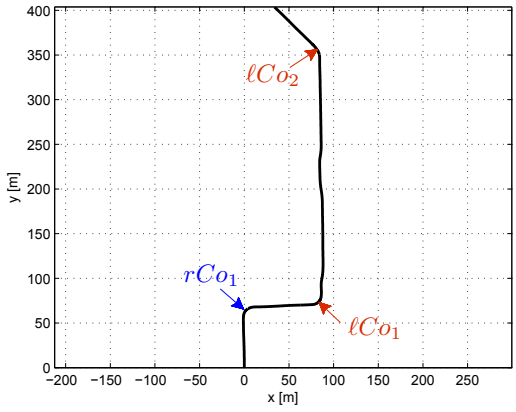
Fig. 5.7 and Fig. 5.8 present representative data for each track respectively. For the acquired path records, left/right lane change or cornering movements are denoted as ℓLC, rLC, or ℓCo, rCo respectively. Matched with the turning movements between Fig. 5.7 and Fig. 5.8, it could be found that the lateral acceleration affected to the human driver is proportional to the amount of steering change (or sharpness) and each driver exerts its steering angle as low as possible for smaller lateral acceleration.

5.2.4 Main behavioral observations with results

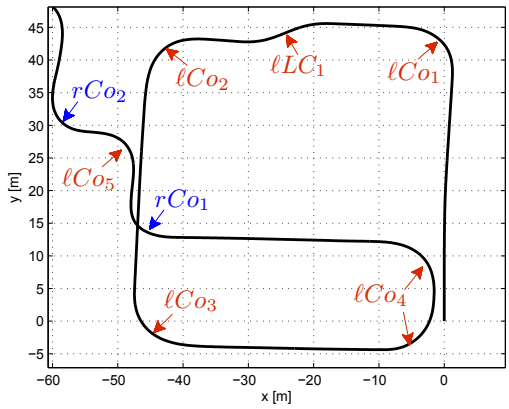
From the results analysis of **Exp.1** to **Exp.3**, human driving patterns are summarized, and solution guidances for the path generation procedure are provided. It is also important that the obtained solution retains the characteristics of which an experienced human driver has in controlling the vehicle.. The following



(a) **Track #1**

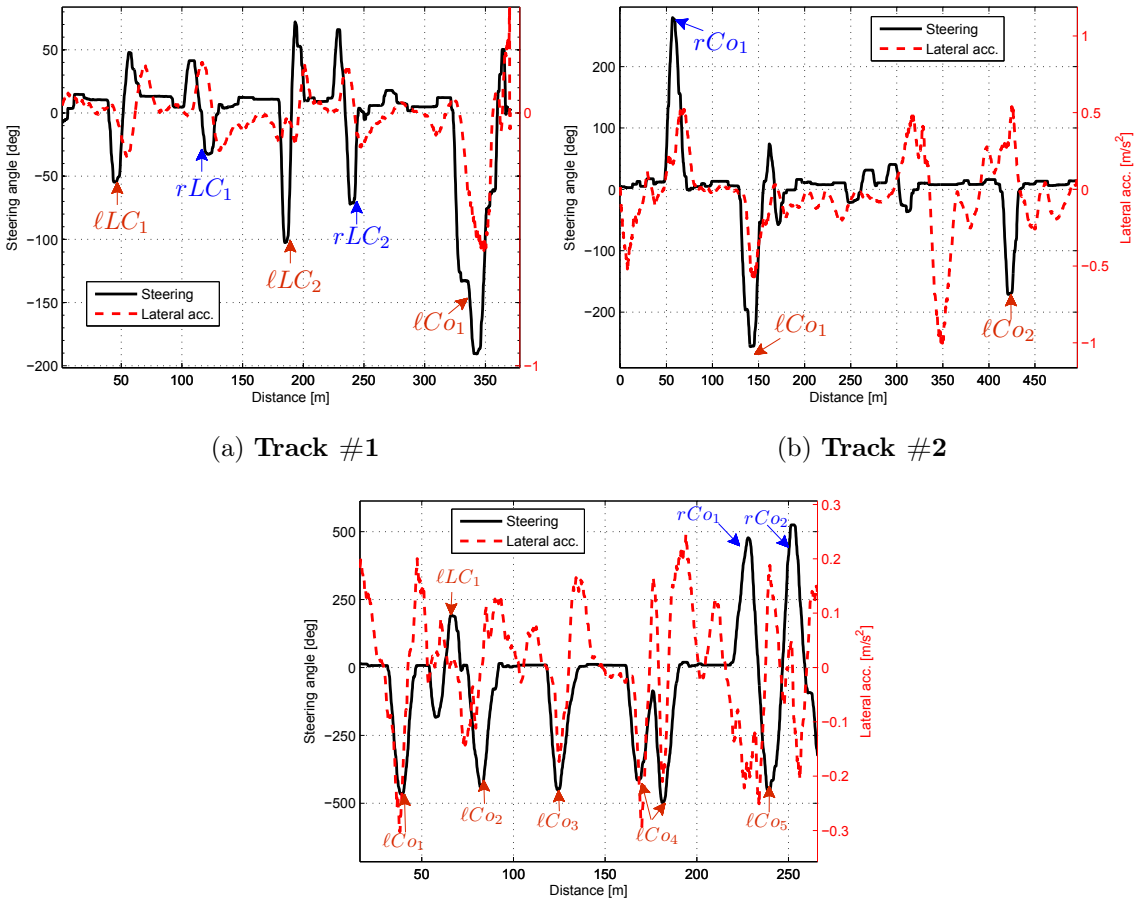


(b) **Track #2**



(c) **Track #3**

Figure 5.7: Path results for **Exp.3** tracks



(a) **Track #1**

(b) **Track #2**

(c) **Track #3**

Figure 5.8: Steering and lateral acceleration results for **Exp.3**

remarks can be derived from the performed experiments, in terms of lane change maneuver for obstacle avoidance.

Property 5 *Human driving patterns during lane changing for obstacle avoidance*

i. Efficient path by experienced driver has a shorter travel length by closer turning than other paths during the obstacle avoidance.

ii. Avoidance distance increases as the vehicle speed increases.

iii. Steering rate or sharpness is proportional to the relative velocity of the vehicle and the obstacle, i.e. $\alpha \propto v_{rel}$

iv. Steering rate changes between before and after avoidance.

From these findings in **Property 5**, lane change path for obstacle avoidance needs to be divided into two different paths as *avoidance* and *recover* path in respect of steering rate.

In particular, it is noticed that the steering patterns of *avoidance* and *recover* mode are different to each other where, the steering in the *avoidance* mode has a sharper turning shape and shorter travel length than *recover* mode (cf. Fig. 5.6) and the resultant path is non-symmetric. These facts act as the major motivation for implementing the human driving pattern in the **CCP** solution procedure, **h-CCP**.

From the analysis for human driving patterns, it is known that a path with its curvature is closely related to human motion of steering wheel. To formulate the problem of **h-CCP** with performance objective to be optimized, it is efficient to summarize the relations between clothoid parameters, steering angle and human motion, and this relations are listed with performance measures in Table. 5.2.

Table 5.2: Relation between path, steering and human driving

path		steering		human		performance measure	
curvature	$\int \alpha ds$	angle	γ	work	$\int \tau d\theta$	rotation	$\wp_r$
sharpness	α	angular vel.	$\dot{\gamma}$	power	$\tau d\theta$	speed	$\wp_s$
torsion	$\frac{d\alpha}{ds}$	angular acc.	$\ddot{\gamma}$	torque	τ	torque	$\wp_t$

In Table. 5.2, each item for the path is expressed through α and the corresponding item for steering γ and performance measures are matched, where $\wp_r$, $\wp_s$ and $\wp_t$ imply the *amount of rotation*, *speed* and *torque* for steering respectively. Note that the *torsion* (rate of sharpness) corresponds to the angular acceleration of the steering or the steering torque that is exerted by human arms. Thus, the

torsion of the given path affects the amount of the steering torque. With the given notations, a steering work for human $\mathcal{W}_{st}$ is defined as follows.

$$\mathcal{W}_{st} = \int_{t_1}^{t_2} \tau\gamma dt = \int_{s_1}^{s_2} |\tau\alpha| ds, \tag{5.2}$$

where τ is the torque exerted by human arms and $\mathcal{W}_{st}$ can be obtained by torque integration between two states between time t_1 and t_2, or s_1 and s_2. Eq. (5.2) implies the steering work is determined by sharpness variation and steering rotation. To reduce steering work, sharpness variation or steering rotation should be minimized. To attain this goal, human patterns are applied to formulate optimization rule and constraints in solving the given problem with algorithmic procedure in **Section 5.3**.

5.3 Problem definition and solution derivation

A **h-CCP** problem is solved by iterative algorithm under the optimization rule with constraints. The optimization rule can be maximizing performance measure such as shortest travel length or minimum traveling time, and the constraints can be curvature limit (or steering angle limit), sharpness limit (or steering rate limit). In the lane change maneuver problem, *S-shaped* curve was generated by composing symmetric *C-shaped* curves under the constraint such as the *golden section rule* [Chen 13], the minimum sharpness [Wilde 09] or vehicle's dynamics stability [Attia 14]. In this section, the **h-CCP** problem and its clothoid based iterative solution are integrated with human driving patterns. The human driving pattern enables the formulation of the **h-CCP** problem by providing optimization rule while taking into account the passenger comfort.

In contrast to the general **pCCP** problem, this problem does not provide all the variables of the initial and final poses, such that only the obstacle position and lane width are given. This is a practical consideration since in normal straight road, the lane change task can be accomplished only by being re-aligned from the previous lane to the next lane. Therefore, in this problem, the distance between the vehicle and the obstacle when the vehicle starts avoidance turning ($\boldsymbol{d_{avd}}$), and a final position to be arrived should be resolved while avoiding the obstacle under the constraint of lateral road width. Furthermore, the obstacle and vehicle are modeled by a circle [Chakravarthy 98, Wilkie 09] in which each circle encompasses the obstacle's geometric shape and vehicle respectively. By using the circular model, it is efficient to represent the safe path for collision avoidance with an equivalent obstacle boundary which integrates the radius of the obstacle

and vehicle circles. In this respect, the **h-CCP** problem is defined as follows.

Problem definition: h-CCP generation for lane change maneuvers
In a straight lanes road, when a static obstacle is detected in front of the vehicle, a lane change path should be generated with minimum lateral acceleration (or maximum comfort), where only the lateral distance and the obstacle boundary are given as geometric constraint. Here, it is assumed that the obstacle size is not over the lane width and the obstacle is far enough from the vehicle to be avoided. Under the geometric constraints described above, find a feasible **CCP** *with the minimal number of clothoid while satisfying given boundary configurations.*

To solve the **h-CCP** problem, it needs to be divided into two complementary problems, **Problem 5A** and **Problem 5B**, where each sub-problem is for *avoidance* and *recover* mode respectively. Here, the terminologies, *avoidance* and *recover* are given for the driver's intention to vehicle motion, where *avoidance* refers to the vehicle avoiding an obstacle and *recover* refers to a vehicle's orientation(heading) recovering to its initial orientation(heading). The two sub-problems share a common boundary condition, meeting pose $\boldsymbol{p_m}$) at the transition between *avoidance* and *recover* mode, where the position, orientation and curvature in the two paths are all the same.

5.3.1 h-CCP for avoidance mode: Problem 5A

This subsection deals with avoidance mode (**Problem 5A**). It is assumed that an obstacle is static and located in front of the vehicle such that if the vehicle does not change its lane, a collision a collision will be thus unavoidable. As described in **Exp.2**, the avoidance situation is urgent and the driver/passenger experiences a threatened status. The most prior objective of the avoidance path is that it should be safe from any risk of collision. And the path should also be the shortest by small turning radius as known from **Property 5-i**. When a vehicle travels along a straight road and a static obstacle is initially located far from the vehicle by avoidance distance (cf. Fig. 5.3(a)), then, **Problem 5A** is defined as follows.

Problem 5A: h-CCP for avoidance mode
When the initial pose is $\mathbf{p_i}(x_i, y_i, \theta_i^\star, \kappa_i^\star)$ *and the meeting pose is* $\mathbf{p_m}(x_m, y_m, \theta_m, \kappa_m^\star)$*, find a short and continuous curvature path with minimum lateral acceleration (or maximum comfort), where* $\theta_i^\star = \frac{\pi}{2}$ *and* $\kappa_i^\star = \kappa_m^\star = 0$ *(the superscript* $\star$ *refers to known information from geometric constraints).*

Since the given boundary condition is incomplete for x_i, y_i, x_m, y_m and θ_m, and moreover there exists too many parameters to be varied, thus additional constraints are required to obtain the solution.

Here, two kinds of contraints are implemented, of which one is the geometric safety constraint and the other is the solution constraint. As the geometric safety constraint, the boundary condition at the meeting pose $\boldsymbol{p_m}$ should be specified. Since the obstacle boundary is modeled by a circle, the $\boldsymbol{p_m}$ is located at the point around the circle to assign the path to be efficient by the closest turning which results in the shortest travel length while assuring safety. At this point, the orientation θ_m at $\boldsymbol{p_m}$ becomes a design parameter such that the point around the obstacle circle (i.e. meeting position x_m, y_m) can be determined from the tangential line on the circular boundary as its slope $\tan(\theta_m)$.

Under the geometric safety constraint, this problem can be solved by two clothoids composition and its parametric iterations [Gim 14a, Gim 14b]. Two clothoids are composed as follows. The first clothoid C_1 is generated from its origin to end with the deflection δ_1 (> 0), κ_1 and θ_1 ($= \frac{\pi}{2} - \delta_1$) by a constant α_1. The second clothoid C_2 is also generated in the same way as C_1, but it is rotated and translated for its end to meet the end of C_1 while satisfying G^1 and G^2 continuity. A solution could be obtained such that the composed clothoids satisfy both boundary conditions as given in **Problem 5A**. However, since numerous pair of clothoids could satisfy the boundary conditions, additional constraint is required to reduce the variable parameters or number of iterations. In this respect, as the solution constraint, *minimax sharpness constraint* (**MSC**, cf. **Section 2.2.4** p.64) for two clothoids is applied.

The remained objective of the solution is to minimize lateral acceleration. Since the maximum lateral acceleration which affects the passenger ride comfort is proportional to the maximum curvature in the path ($a_{lateral} = \kappa v^2$), the following optimization rule is imposed to formulate the problem into algorithmic procedure.

$$\begin{aligned} &\underset{\hat{\alpha}}{\text{minimize}} \quad \max_{\alpha_i \in C_i} \widehat{C}(\alpha_1, \alpha_2) && : \wp_{\mathbf{s}} \\ &\text{subject to} \quad \kappa \le \bar{\kappa}_{limit},\ \alpha \le \bar{\alpha}_{limit},\ (s \in [s_i, s_m]), \end{aligned} \tag{5.3}$$

where $\widehat{C}$ is a feasible **h-CCP** solution using two clothoids, and each clothoid has a sharpness of α_1 and α_2 respectively. The performance measure $\wp_s$ corresponds to the speed for the human arms, thus the optimized solution minimizes the maximum speed of the human arms during the avoidance. The rate of steering angle turning is limited by mechanical or actuator limits as $\bar{\kappa}_{limit}$ and $\bar{\alpha}_{limit}$.

It is assumed that the vehicle speed is constant through lane change maneuver, and **MSC** constraint is adopted to solve the **Problem 5A**.

In Fig. 5.9(a), two clothoids composition and its parametric variation are depicted with geometric constraint. In the figure, when θ_m is initially assumed, then $\boldsymbol{p_m}$ is determined by geometric safety constraint as described above. Two clothoids C_1 and C_2 are composed to have both end points as $\boldsymbol{p_o}$ and $\boldsymbol{p_o'}$, and the resultant segment is shifted for $\boldsymbol{p_o'}$ to be equal to $\boldsymbol{p_m}$ with the same orientation of the line ℓ_m. As α increases, the end point $\boldsymbol{p_o}$ varies as the dotted arrow ($\nearrow$) in Fig. 5.9(a) directs toward. A convergence criteria for this variation is $\boldsymbol{D_\alpha}$ which measures the horizontal distance between $\boldsymbol{p_o}$ and ℓ_i. For $\boldsymbol{p_o}$ to be closer enough to $\boldsymbol{p_i}$ within a given threshold, θ_m also needs to be varied. Thus, $\boldsymbol{D_\alpha}$ of measuring the vertical distance between $\boldsymbol{p_o}$ and $\boldsymbol{p_i}$ is used for regulating θ_m.

Eq. (5.4) presents the rules for parametric iteration and its convergence (**Iteration #1**), where the parameter α is iteratively varied until the corresponding determinant function D_α gets smaller than a threshold. The parameter α is halved when the sign of D_α changes, i.e. the iteration for convergence rule is same as the bisection method [Kiusalaas 07].

$$\alpha = \alpha + |\partial\alpha| \cdot sgn(D_\alpha), \tag{5.4a}$$

$$D_\alpha = x_o - x_i, \tag{5.4b}$$

$$\partial\alpha = \begin{cases} \frac{\partial\alpha}{2}, & \text{if } D_\alpha \cdot D_\alpha' < 0, \\ \partial\alpha, & \text{otherwise,} \end{cases} \tag{5.4c}$$

where x_o is the x-coordinate value of $\boldsymbol{p_o}$ and $sgn(\cdot)$ is the sign function of having positive or negative value and D_α' is D_α at the previous iteration.

$$\theta_m = \theta_m + |\partial\theta| \cdot sgn(D_\theta), \tag{5.5a}$$

$$D_\theta = y_o - y_i, \tag{5.5b}$$

$$\partial\theta = \begin{cases} \frac{\partial\theta}{2}, & \text{if } D_\theta \cdot D_\theta' < 0, \\ \partial\theta, & \text{otherwise.} \end{cases} \tag{5.5c}$$

where y_o is the y-coordinate value of $\boldsymbol{p_o}$.

Eq. (5.6) presents **Iteration #2** for θ_m with D_θ which is similar to Eq. (5.4). The convergence criteria for both iteration procedures are defined as follows.

$$D_\alpha < \varepsilon_\alpha, \tag{5.6a}$$

$$D_\theta < \varepsilon_\theta, \tag{5.6b}$$

where 10^{-3} m could be values of ε_α and ε_θ for fast convergence with allowable accuracy.

The algorithmic procedure for the avoidance mode (**Algorithm 19**) is described as follows. At first, the parameter θ_m is assumed (initially $\frac{\pi}{4}$), then $\boldsymbol{p_m}$ on the obstacle boundary is determined by the tangential line. In the next, the composed clothoid $\widehat{C}$ (cf. Fig. 5.9(a)) is translated for its end to $\boldsymbol{p_m}$, then a solution using **MSC** is obtained with $\boldsymbol{d_{avd}}$ using **Iteration #1**. If the current position of the vehicle differs to $\boldsymbol{d_{avd}}$, the parameter θ_m is iteratively varied until the distance between the obstacle and the vehicle is close enough to $\boldsymbol{d_{avd}}$ within a threshold $\varepsilon = 10^{-3}$ $[m]$ by **Iteration #2**. Note that for each iteration loop, **Iteration #2** is performed after **Iteration #1** passed the convergence check.

If the solution has its maximum (or minimum) curvature or sharpness over $\bar{\kappa}_{limit}$ or $\bar{\alpha}_{limit}$ and when the condition of too close obstacle position as $\theta_m < 0$ is reached, then the obtained path is not feasible. In that case, it remains the only way to reduce the vehicle speed to stop (cf. **Exception** in line 9 of **Algorithm 19**). **Algorithm 19** describes the procedure to obtain the proposed solution.

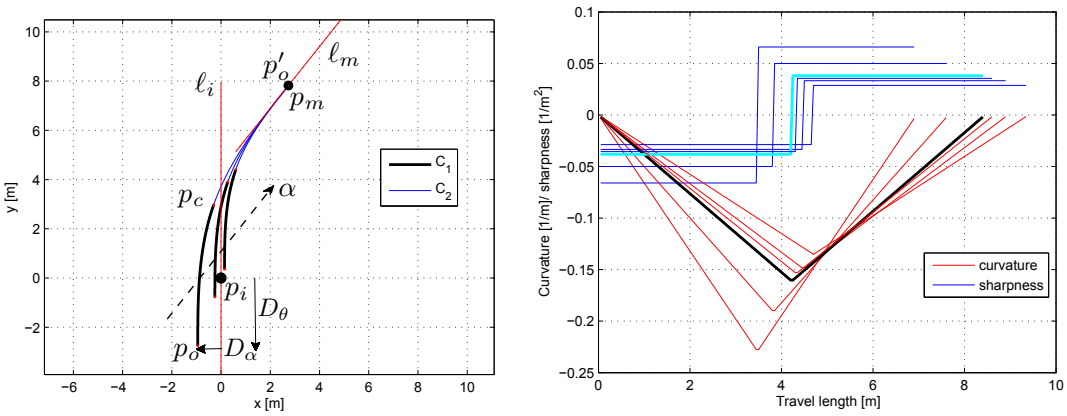
Algorithm 19 Avoidance mode path

```
Require: θ_m, ∂θ, ∂α, ε_{θ,α}                          ▷ Initial assumption
1: procedure AVDPATH(p_i, p_m)
2:     Ĉ ← [2]α_1 = α_2, δ_1 = δ_2 = (θ_i − θ_m)/2      ▷ MSC
3:     while (D_θ > ε_θ and D_α > ε_α) do               ▷ Convergence for α and θ_m
4:         Iteration # 1
5:         Iteration # 2
6:     end while
7:     Exception                                        ▷ Check solution feasibility
8:     return Ĉ                                         ▷ Solution obtained
9: end procedure
```

The obtained solution assumes vehicle speed constant. From the pattern of avoidance distance acquired from **Exp.2**, i.e. **Property 5**-ii and **Property 5**-iii, the desirable vehicle speed is taken from the relation as Eq. (5.1), such that the obtained $\boldsymbol{d_{avd}}$ could be matched with the desirable vehicle speed. Thus, if the current vehicle speed is over the calculated desirable vehicle speed, it should be reduced to the desirable speed before the avoidance mode.

When a representative example for **Problem 5A** is given as shown in Fig. 5.9(a), the curvature/sharpness diagram (*C-S diagram*) for the obtained solution is depicted in Fig. 5.9(b).

[2]← indicates that two clothoids are composed by given parameters.



(a) Convergence by parameter variation (b) C-S diagram

Figure 5.9: **Problem 5A** solution result

In Fig. 5.9(a), two clothoids C_1 and C_2 are generated with given parameters $\alpha_{1,2}$, $\kappa_{1,2}$ and $\delta_{1,2}$ where the subscripts $(1,2)$ present the parameters of both clothoids are same (cf. **Algorithm 19**: line 2), from $\boldsymbol{p_o}$ and $\boldsymbol{p'_o}$ to the center point $\boldsymbol{p_c}$ respectively, and composed to be $\widehat{C}$. The determinant function D_α is defined as the distance between $\boldsymbol{p_o}$ and $\boldsymbol{p_i}$ (or ℓ_i) along *x-axis*, and D_θ is the distance between $\boldsymbol{p_o}$ and $\boldsymbol{p_i}$ along *y-axis*.

By varying θ_m and $\alpha_{1,2}$ independently, the algorithm finds the feasible clothoids pair C_1, C_2 that satisfies both boundary conditions of $\boldsymbol{p_i}$ and $\boldsymbol{p_m}$, while being tangential to ℓ_i and ℓ_m, respectively (cf. Fig. 5.9(a)). The solution is obtained after convergence is passed both the conditions of D_α and then, D_θ. It is also clear that the solution path is generated under ℓ_m line and the obstacle boundary is located in the upper side of ℓ_m, thus the vehicle motion along the path is safe from the risk of collision.

The corresponding C-S diagrams are depicted in Fig. 5.9(b). Among the feasible solutions that has passed the D_α and the D_θ condition, a solution satisfying **MSC** is chosen as being marked by *black-bold* line in curvature and *cyan-bold* line in sharpness respectively in Fig. 5.9(b).

5.3.2 h-CCP for recover mode: Problem 5B

This subsection solves **Problem 5B** for recover mode. After the obstacle is avoided, for instance, it is behind the vehicle, the driver steers the vehicle to make the transition to other free lane (lane change) while recovering to its initial

orientation with *zero* curvature or *zero* steering angle). The initial pose $\boldsymbol{p_m}$ (the final pose of **Problem 5A**) is given but, the final pose is not fully known where only the lateral distance to be recovered is given by the lane width constraints. Thus, the **Problem 5B** is defined as follows.

Problem 5B: h-CCP for recover mode

When the initial pose is $\mathbf{p_m}(x_m^\star, y_m^\star, \theta_m^\star, \kappa_m^\star)$ *and final pose is* $\mathbf{p_f}(x_f, y_f, \theta_f^\star, \kappa_f^\star)$*, find a* **h-CCP** *solution which minimizes steering work under given maximal curvature and sharpness, where* $\kappa_m^* = \kappa_f^* = 0$*,* $\theta_f^* = \frac{\pi}{2}$ *(the superscript* $\star$ *refers to known information from the solution of problem A and boundary constraint).*

As described in **Section 5.2**, the steering pattern for the *recover* mode displays a less sharp, looser and flatter shape than that of the *avoidance* mode. This steering pattern implies that human driver operates to minimize his/her steering work under the allowable lateral acceleration during *recover* mode, thus it could be emulated by a similar shaped curvature diagram that has lower peak and longer travel length than that of *avoidance* mode. For that curvature diagram design, solution constraint for **Problem 5B** could be formulated to have two performance measures to be optimized; steering rotation ($\wp_{\mathbf{r}}$) and steering speed (or **MSC**) ($\wp_{\mathbf{s}}$) under constraints as follows.

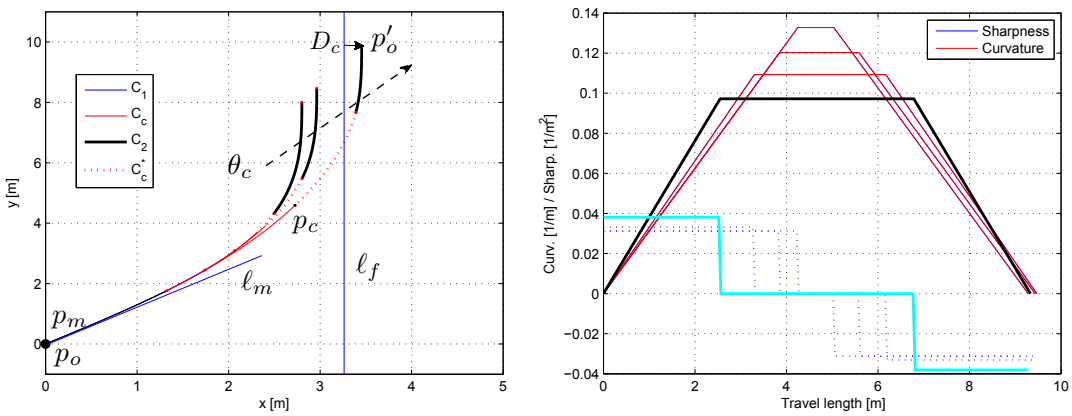
$$\begin{aligned} &\underset{\kappa(s)}{\text{minimize}} && \int_{s_m}^{s_f} |\alpha(s)|\, ds && : \wp_{\mathbf{r}} \\ &\underset{\hat{\alpha}}{\text{minimize}} && \max_{\alpha_i \in C_i} \widehat{C}(\alpha_3, \alpha_4) && : \wp_{\mathbf{s}} \\ &\text{subject to} && \kappa \le \bar{\kappa}_{max},\ \alpha \le \bar{\alpha}_{max},\ (s \in [s_m, s_f]) \end{aligned} \tag{5.7}$$

where, κ_{max} is given by the allowable lateral acceleration and vehicle speed as $a_{max} = \kappa_{max} v^2$ (v: vehicle speed $[m/s]$) and *ISO 2631-1* comfort measure, e.g. for *Not uncomfortable* $a_{max} \le 0.315\ m/s^2$.

Eq. (5.7) describes solution constraint of minimizing the amount of steering rotation and the maximum steering speed which are matching to the amount of curvature variation and maximum sharpness respectively. The solution could be obtained by minimizing $\wp_{\mathbf{r}}$ and $\wp_{\mathbf{s}}$ simultaneously by adjusting clothoids parameters. By the way, since the vehicle speed is constant through the overall lane change maneuver, the constraints κ_{max} and α_{max} of **Problem 5B** are always below the values of the obtained solution in **Problem A**. Thus, it is derived that the minimax sharpness of this problem is constrained to the sharpness of **Problem 5A** solution, i.e. $\alpha_{3,4} \le \alpha_{1,2}$.

The clothoids composition is similar to **Problem 5A**, except the condition that the lateral distance to be arrived is longer than the lateral distance avoided as

$(x_m - x_i) \leq (x_f - x_m)$, thus the composed path requires additional arc segment. To explain this geometric compromise, following example is described with Fig. 5.8.



(a) Convergence by parameter variation (b) C-S diagram

Figure 5.10: **Problem 5B** solution results

With the given boundary condition $\boldsymbol{p_m}$, the end position of the composed segment $\boldsymbol{p'_o}$ should be close enough to the line ℓ_f within a threshold. The symmetric clothoids $\boldsymbol{C_1}$, $\boldsymbol{C_2}$ and the symmetric arcs $\boldsymbol{C_c C_c^*}$ are composed with each other under the orientation continuity constraint. The parameter θ_c is the angle of the correnponding circumference for $\boldsymbol{C_c}$ or $\boldsymbol{C_c^*}$ and, as θ_c increases, the end point $\boldsymbol{p'_0}$ expands outward by the dotted arrow ($\nearrow$) as shown in Fig. 5.10(a).

In Fig. 5.10, the solution path and its C-S diagram for **Problem 5B** example are shown where $\boldsymbol{p_m}(0, 0, \frac{\pi}{6}, 0)$, $\boldsymbol{p_f}(3, y_f, \frac{\pi}{2}, 0)$, and the solution includes two symmetric clothoids with additional arc segment. Each solution path in Fig. 5.10(b) satisfies the boundary conditions at both ends, but the sharpness and maximum curvature value differs to other solutions. Among the obtained candidate solutions, first, the optimized solution is to select the path of which the maximum curvature value κ_{max} is the smallest. One can observe in Fig. 5.10(b) that the steering rotation ($\wp_\mathbf{r}$) decreases as the length of the arc segment gets longer since $\int_{s_a}^{s_b} |\alpha(s)|\ ds = 0$ for the arc segment $s \in [s_a, s_b]$. At the same time, it is also noticed that the maximum sharpness ($\wp_\mathbf{s}$) of the composed path increases as the arc segment expands. Thus, both performance measures are minimized until the sharpness of **Problem 5B** increases to the sharpness of **Problem 5A**, which is denoted in Fig. 5.10(b) by the *black-bold* line (curvature) and the *cyan-bold* line (sharpness).

In **Algorithm 20**, the sharpness $\alpha_{3,4}$ (for two clothoids C_3 and C_4 in the

Algorithm 20 Recover mode path

Require: $d\theta_c, \varepsilon$ ▷ Initial assumption

procedure RCVPATH(p_m, p_f)

2: $\delta_0 = \frac{\theta_m - \theta_f}{2}, \delta_{3,4} = \delta_0 - \theta_c$

$\widehat{C} \leftarrow \alpha_{3,4} = \alpha_1, \delta_{3,4}, \theta_c$

4: ▷ $\widehat{C}$: Composition by symmetric pair of $C_{3,4}$ and $C_c C_c^*$

$D_c \leftarrow \widehat{C}, \ell_f$ ▷ Distance error between $\widehat{C}$ and ℓ_f

6: **while** $D_c \geq \varepsilon$ **do** ▷ Convergence for θ_c

Iteration # 3

8: **end while**

Exception ▷ Check solution feasibility

10: **return** $\widehat{C}$ ▷ Solution obtained

end procedure

recover mode path) are all the same as the sharpness $\alpha_{1,2}$ obtained in **Problem A** and the deflections $\delta_{3,4}$ are initially set to be δ_0 and varied by δ_c summation (line 2). $\widehat{C}$ is composed of two clothoids, C_3 and C_4 and symmetric two circular arcs $C_c C_c^*$ (line 4). The only variable for iterative convergence is the internal angle θ_c for the circular arc, which determines the deflections of $\delta_{3,4}$ (line 3).

$$\theta_c = \theta_c + |\partial\theta| \cdot sgn(D_c), \tag{5.8a}$$

$$D_c = x_{tg} - x_o^{'}, (x_{tg} = x_f - x_m) \tag{5.8b}$$

$$\partial\theta = \begin{cases} \frac{\partial\theta}{2}, & \text{if } D_c \cdot D_c' < 0, \\ \partial\theta, & \text{otherwise} \end{cases} \tag{5.8c}$$

Eq. (5.8) describes **Iteration #3**, where θ_c varies to expand the arc portion in C_3 and C_4. By adjusting the arc portion, the end point ($p_o^{'}$) of the composed clothoid ($\widehat{C}$) could reach the target line ℓ_f within a threshold (cf. Fig. 5.8(a)). The convergence criteria is similar to **Iteration #1**, **#2** as below.

$$D_c < \varepsilon_c, \tag{5.9}$$

where ε_c could be 10^{-3} m. Note that for the path resulting in $\theta_c \leq 0$, **Iteration #3** in this algorithm is not effective for given boundary condition and it needs to be solved by **Iteration #1** to find a new $\alpha_{3,4}$. [3] The feasibility checking for the obtained solution is performed in **Exception** stage whether the obtained solution

[3]This motion corresponds to narrow lane change or urgent recover, and it is not considered in the problem formulation.

exceeds given limits such as $\bar{\kappa}_{max}$, $\bar{\alpha}_{max}$ in Eq. (5.8). The proposed algorithm ensures its convergence with finite number of iterations.

5.3.3 Lane change maneuver based on h-CCP

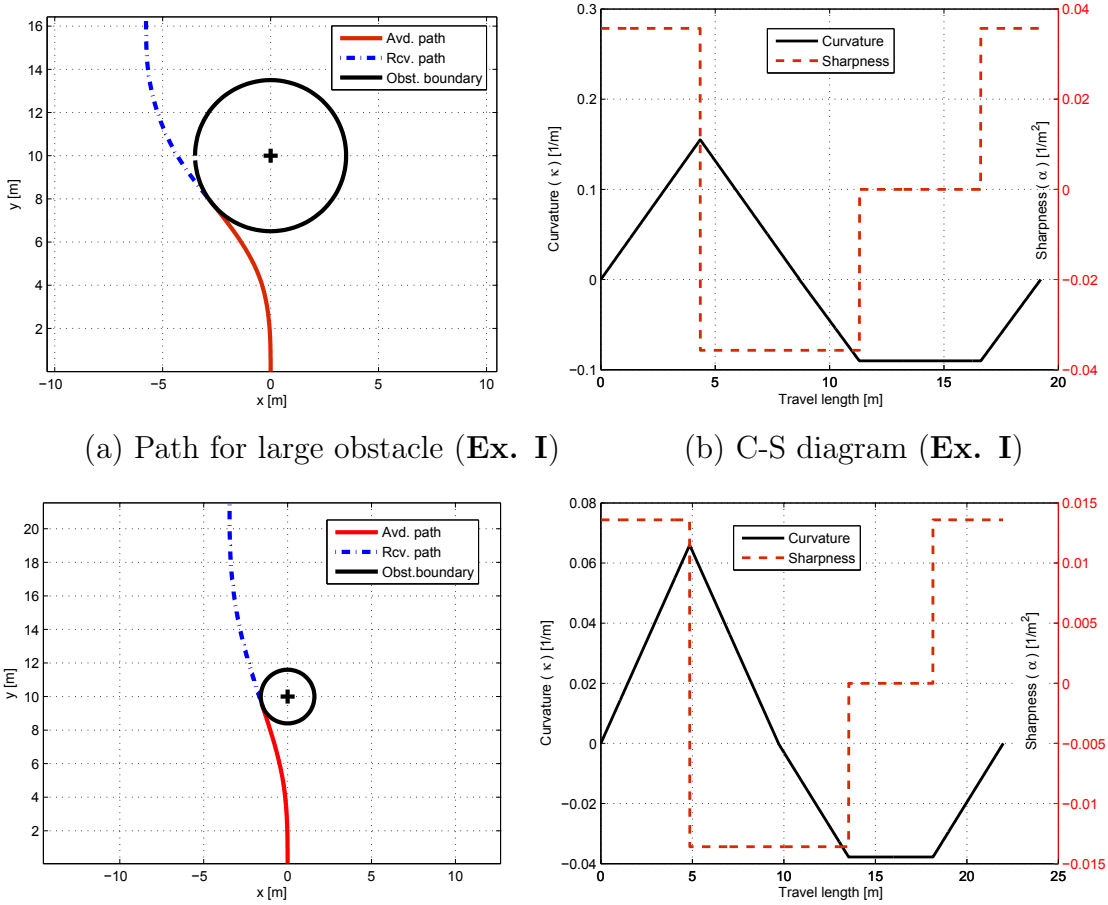
From the defined sub-problems (**5A** and **5B** given in previous sections) ant the obtained solutions, the **h-CCP** is generated with the corresponding C-S diagram. The **h-CCP** has a total of four clothoids and its curvature diagram has two different convex curves of the *avoidance* and *recover* modes. The proposed solution is applied to two examples of the lane change maneuver where the first example is for a large obstacle of radius r_d = 4 m with lateral distance of 6 m and the second example is the strict lane change for a small obstacle (passenger vehicle) of r_d = 1.6 m with lateral distance of 3.5 m.

In the results shown in Fig. 5.11, two examples, **Ex. I** and **Ex. II** are resolved by tackling two different types of paths (avoidance/recover modes) and both obtained paths successfully avoid the obstacle while taking continuous curvature along the travel length as shown in Fig. 5.11(b) and Fig. 5.11(d). The proposed algorithms implement the human driving patterns and benefit from the reduced number of iterations by restricting the variable parameters while assuring the convergence. Furthermore, the algorithms cope with the different obstacle sizes/positions and lane width in spite of the incomplete boundary conditions. The algorithmic efficiency is analyzed from a representative graph as shown in Fig. 5.12.

Fig. 5.12(a) shows the variation of travel length s (scaled by $\frac{1}{50}$), the maximum curvature (κ_{max}), and the maximum sharpness (α_{max}) according to the θ_m variation (cf. **Problem A** and **Algorithm 19**), where the solution converges to α_{max} and κ_{max} by minimizing $\wp_s$ in Eq.(5.3) and satisfies the $\boldsymbol{d_{avd}}$ constraint by increasing s (the converged θ_m is 51.45 $[deg]$). Fig. 5.12(b) shows an example of the algorithmic convergence $\boldsymbol{D_\theta}$ to the given error bound of 0.01 m within 10 iterations, which estimates about 150 ms (with **MATLAB** and *Pentium Dual Core* 1.90 GHz) and the proposed algorithmic solution has a prospect to be implemented in real experimental vehicle.

5.4 Evaluation of the proposed h-CCP

In this section, the obtained path is compared with other lane change methods. There are two groups in lane change solution methodology, where one is based on smooth interpolation by multiple points such as polynomials and *Bézier*



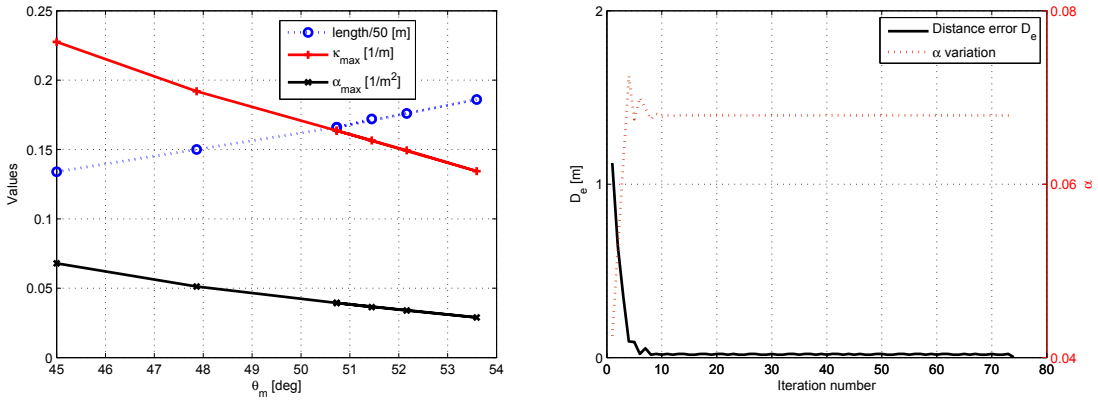
(a) Path for large obstacle (**Ex. I**)

(b) C-S diagram (**Ex. I**)

(c) Path for small obstacle (**Ex. II**)

(d) C-S diagram (**Ex. II**)

Figure 5.11: **h-CCP** of obstacle avoidance using lane change behavior



(a) Avoidance path variation by θ_m (b) Iterative convergence

Figure 5.12: Algorithmic convergence for recover path

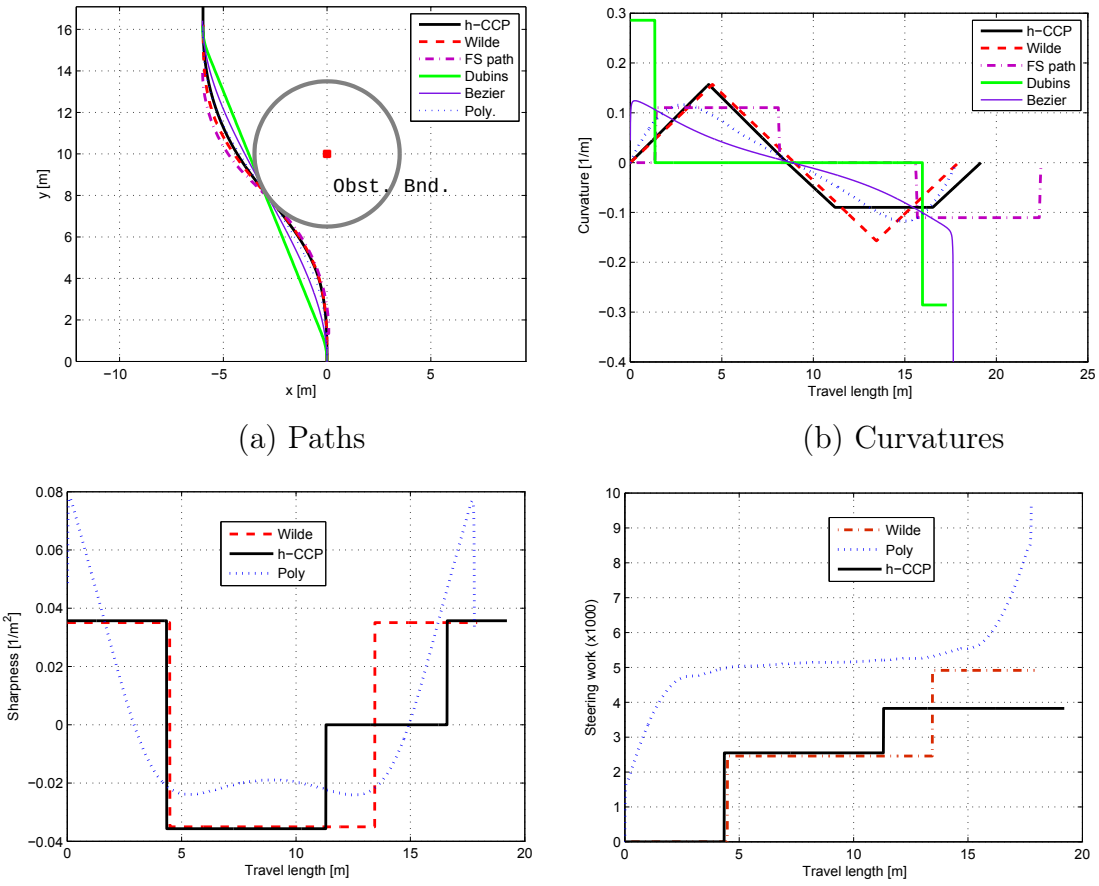
spline (*interpolation-based method*), and the other is made by composition of basic segments such as line, arc and clothoid (*segment-based method*).

The interpolation-based method makes a path by connecting consecutive points on the *Cartesian* coordinate with interpolating function, thus it is easy to show the geometrical smoothness of the generated path. However, the curvature diagram could not be obtained directly from the interpolation function, but it is acquired only after calculating all the curvature values by differential equation between consecutive three points in the path.

On the contrary, the segment-based method is difficult to get an analytic solution of satisfying the boundary conditions, but gives easily the curvature diagram from the segments information of curvature and sharpness. The proposed path belongs to the second group, and there are standard methods such as *Dubins* path [Dubins 57], **FS** path [Fraichard 04] and *Wilde*'s path [Wilde 09] which could be applied to the same lane change maneuver for the given problem.

To evaluate the performance of the proposed solution, two representative interpolation-based methods (quintic polynomials [Nelson 89] and *Bézier* spline [Chen 13, Gonzalez 14, Poussot-Vassal 11]) and three segment-based methods (*Dubins* path, **FS** path and *Wilde*'s path) are obtained and compared. All the paths should keep the lateral width for lane change, and avoid the given obstacle within an error bound (e.g. 10^{-3} m). Thus, each path is iteratively obtained under those constraints. Figure 5.11 depicts for the comparative analysis of the obtained path with other methods for the example I.

In Fig. 5.13(a), all the obtained paths avoid the given obstacle and arrive at



(a) Paths (b) Curvatures

(c) Sharpnesses (d) Steering works

Figure 5.13: Comparison of the proposed **h-CCP** with other paths

other lane, and even if all the paths are geometrically smooth in the position and orientation domain (G^0 and G^1), they are largely different in the curvature domain (G^2) which are shown in Fig. 5.13(b). Especially, the curvature of **FS** path [4], *Dubins* and *Bézier* paths have much higher peaks or drastic changes in the curvatures, thus these three paths are considered to be not appropriate for smooth steering. Then, except those three paths, sharpness diagrams are shown in Fig. 5.13(c). All the paths can not be shown in the same scale due to much high peaks in those three paths.

In the sharpness diagrams, the results of the obtained **h-CCP**, *Wilde*'s and *Poly* paths are compared, where the segment based methods (**h-CCP**, *Wilde*'s path) have simple and steady curvature, but the *Poly* path has greater fluctuations at boundary conditions. As mentioned in [Wilde 09], the *Wilde*'s path has minimum sharpness and curvature so as to be human natural and easily drivable.

The **h-CCP** also shows almost the same κ_{max} but smaller κ_{min}, $\mathcal{W}_{st}$ (cf. Eq. (5.2)) than the *Wilde*'s path. Fig. 5.11(d) highlights the steering work comparison for the three paths. As shown in the figure, *Poly* path increases the steering work than other methods up to the double at the end. For the **h-CCP** and the *Wilde*'s path, the proposed **h-CCP** shows smaller steering work than the *Wilde*'s path by −20 %. Qualitative measures for above results are listed in Table. 5.3.

Table 5.3: Performance evaluation of the proposed path ([m],[rad])

path	length	$\mathcal{W}_{st}$	κ_{max}	κ_{min}	α_{max}	α_{min}	τ_{max}	$\wp_r$
Poly	17.79	0.0097	0.1175	-0.1163	0.0241	-0.078	**2.6871**	**0.4607**
Bézier	17.65	12.6924	0.1243	-0.3999	3.5488	-17.8001	1422.7	0.6128
Dubins	**17.30**	1632.7	0.2857	-0.2857	∞	-∞	2857.14	0.5753
FS	22.43	5.6430	**0.1104**	-0.1104	1.227	-1.227	122.67	0.4893
Wilde's	17.92	0.0049	0.1570	-0.1570	0.0351	**-0.0351**	7.0101	0.5753
h-CCP	19.16	**0.0039**	0.1568	**-0.0898**	0.0366	-0.0366	7.3266	0.4893

In Table. 5.3, the items of length and τ_{max} signify the travel length, and the maximum difference for α change respectively. The better performance in each item is marked in bold. The *Poly* path is superior to other methods by the smallest τ_{max} and $\wp_{\mathbf{r}}$, however it has larger α_{min} and κ_{min} and $\mathcal{W}_{st}$ than the proposed path. Although **Dubins**, **FS** or *Bézier* paths get higher performance in specific items, the curvature or sharpness diagrams indicate that they are not

[4]The mechanical limit $\bar{\kappa}_{limit}$ = 0.489 and $\bar{\alpha}_{limit}$ = 1.227 are used. (cf. Eq. (5.3))

appropriate for steering control due to the drastic change of the sharpness (α_{min}) or big steering torque required (τ_{max}).

The proposed path has superior performance in the sharpness and curvature as *Wilde*'s path and spends small $\wp_s$ close to the *Poly* path. Furthermore, the proposed path records the smallest value of $|\kappa_{min}|$ which is proportional to lateral acceleration or deterioration of ride comfort. Consequently, it is obvious that the proposed path is smooth and simple in curvature and sharpness such that the steering control follows the path without complexity, and it also ensures the safety for obstacle avoidance with minimal magnitude of steering rate (or sharpness) as well as minimal steering work for recover mode, thus it provides natural and human favorable maneuver. Note that if the vehicle speed is constant through the lane change maneuver, the comfort measure depends on κ_{max} and $\boldsymbol{v_{avd}}$ (cf. Eq.(5.1)) thus, under the constraint of *Not uncomfortable*, the desirable $\boldsymbol{v_{avd}}$ should be below 1.42 m/s for the proposed **h-CCP**.

5.5 Velocity planning on the proposed h-CCP

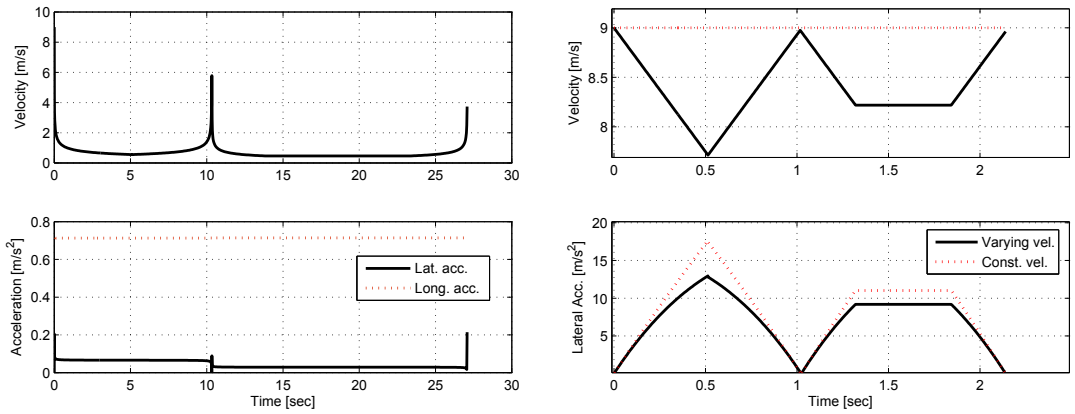
A trajectory is generated from both path and velocity, where the path and the velocity are independent of each other since the path is only geometric position data and the velocity is only the travel length data in time. However, if a ride comfort measure like lateral acceleration is involved to generate the path, two planning, path and velocity are linked each other. For example, as the vehicle speed increases, the avoidance distance should increases as shown in **Exp.2**, and as the sharpness of the path increases, the vehicle speed should decrease as shown in **Exp.3**. From the human drivng pattern of velocity changes due to accelerating and braking (e.g., Fig. 5.6(d), Fig. 5.7(d)), the velocity along the obtained path is planned in this section. From the acquired data, the human driver makes the vehicle velocity change while performing the steering motions. Specifically, the velocity increases as the steering increases and vice versa. In this respect, the velocity could be planned from a given curvature diagram on the path. A reasonable scheme is to consider the lateral acceleration κv^2 to be steady as,

$$\Delta v = -\frac{v}{2\kappa}|_t \Delta\kappa, \tag{5.10}$$

where Δv and $\Delta\kappa$ indicate the variation for velocity v and curvature κ at time t respectively. Eq.(5.10) signifies that the lateral acceleration value could be maintained by velocity variation which is proportional to steering variation. Thus, the velocity planning could be designed according to the Eq. (5.10) guidance rule. Meanwhile, even if the guidance is ideal for keeping lateral acceleration, the resultant velocity profile is not continuous such that the real vehicle could not generate such tractive or brake force, thus it is not feasible.

Another practical scheme is to refer to the human driving pattern data or related research such as [Glauz 80, Akcelik 01, Wang 04]. In [Glauz 80], the mean acceleration/deceleration data for a light passenger vehicle is used to implement velocity changes such as 2.5 m/s^2 with the same acceleration value.

Fig. 5.14 depicts the velocity profiles from an initial velocity of 9 m/s with its lateral acceleration during travel time. In the case of dramatic velocity change from Eq. (5.10), Fig. 5.14(a) records the low lateral acceleration of under 0.25 m/s^2 and the longitudinal acceleration and braking reach 0.71 m/s^2, which is estimated as 1.0 for the comfort measure of *ISO 2631-1* as *fairy-uncomfortable* and more than 27 seconds travel time is taken, which is thus, impractical. On the contrary, a constant velocity or proposed velocity profile for the 2.5 m/s^2 acceleration case shows realistic results for the passenger vehicle. The proposed velocity profile compared to the constant velocity shows enhanced performance with a



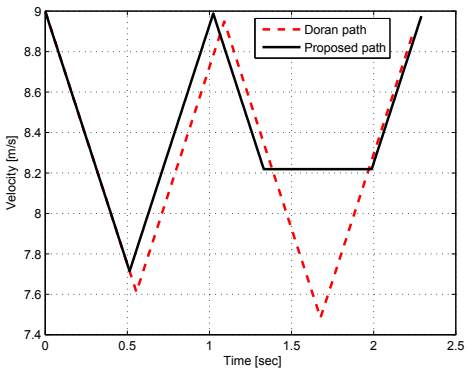
(a) Acc. by Eq. (5.10) (b) Acc. $2.5m/s^2$ vs. const vel.

Figure 5.14: Velocity planning cases and lateral acceleration profiles

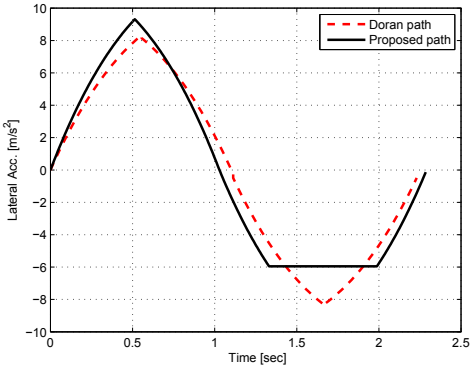
reduced lateral acceleration from 12.595 to 9.255 in $[m/s^2]$ by 26.5% without sacrificing the traveling time of 2.14 *seconds*.

Lastly, comparisonal results of the proposed path with *Wilde*'s path about the velocity planning, lateral acceleration and ride comfort measure (*ISO 2631-1*) are shown in Fig. 5.15. Two small peaks in Fig. 5.15(c) are from the acceleration changes, 2.5 to 0 m/s^2 and 0 to -2.5 m/s^2. With the velocity planned in Fig. 4.15(b), the proposed path records better performance in the lateral acceleration as well as the ride comfort measure, especially in the period of return mode; this indicates that more favorable driving for human passenger is performed by the proposed path than the *Wilde*'s path.

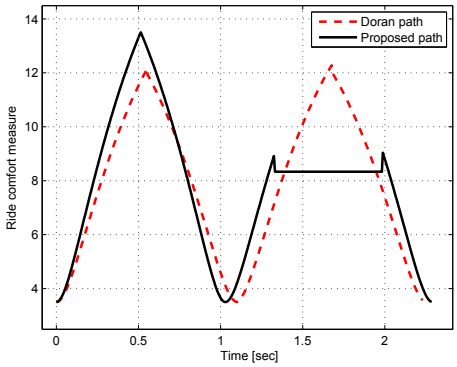
In the next section, results with vehicle dynamics simulation for constant and variable velocity cases are performed.



(a) Velocity planning



(b) Lateral acc.



(c) Ride comfort measure

Figure 5.15: Comparison with the *Wilde*'s path and the proposed **h-CCP**

5.6 Vehicle Dynamics Simulation

The path solution for lane change maneuver and a desirable velocity has been proposed. For the application of the proposed solution to light passenger vehicle, vehicle dynamics and path following control are required to consider for practical implementation. The proposed solution provide curvature diagram for steering control thus, the controllability on the path following by a steering control rule could be examined in the vehicle dynamic simulation framework.

5.6.1 Performance evaluation with constant velocity profile

One useful aspect for implementing vehicle dynamics simulation is that it identifies realistic results except other unwanted effects or conditions, e.g. ground surface nonflatness, wind, etc., and the results could visualize only the experimental factors focused on. Moreover, another importance is the difference between curvature and steering angle for the vehicle to follow the path which is largely affected by tyre dynamics on the ground. For lateral force and acceleration analysis, the effects of load carried, steer or slip angle, and camber angle should be found [Bastow 04]. However, it is too complicated and inefficient for analyzing only the steering control with curvature along the path. Thus, for this vehicle dynamics simulation, a well known bicycle model [Rajamani 06, Ulsoy 12] and *Magic Formula* tyre data [Pacejka 92] are used and it is assumed that ground surface friction is constant with the coefficient 0.85 and the camber angle, aerodynamics force are ignored.

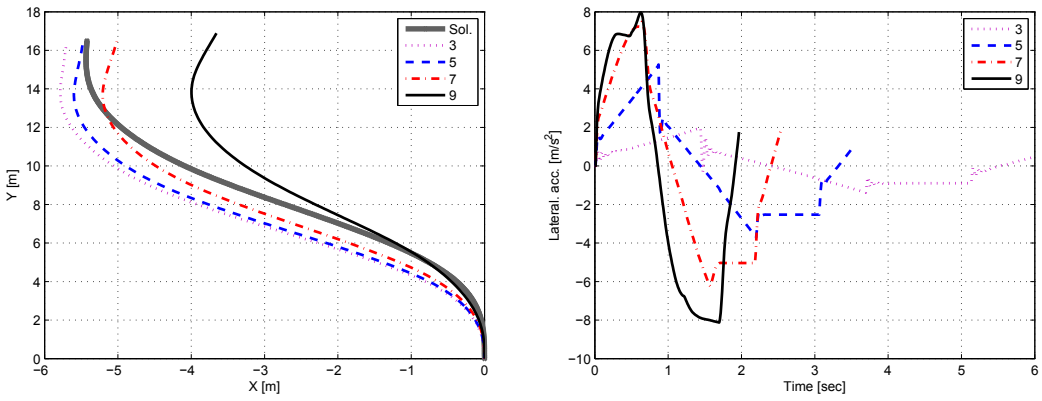
Table 5.4: Vehicle parameters

m	J	L	L_f	L_r	$C_{\alpha f}$	$C_{\alpha r}$
1500	2420	2.43	1.07	1.36	44000	47000

Table. 5.4 shows vehicle parameters to be applied in the simulation, where m is vehicle mass in $[kg]$, J is yaw moment of inertia in $[kg \cdot m^2]$, L is the wheel base and L_f/L_r is the distance between front/rear wheel and center of mass respectively. $C_{\alpha f}/C_{\alpha r}$ are lateral (cornering) stiffness for front/rear wheels respectively. Note that the considered bicycle model is drived by rear wheel traction and braking force and thus the longitudinal force is determined by normal human acceleration/deceleration value 2.5 m/s^2 for entering or leaving curved course.

In the first simulation test, the constant velocity case is investigated with the different initial values as 3, 5, 7 and 9 m/s where the velocity ranges are chosen

for the vehicle model to maintain steady state turning with given tyre-ground slip condition.



(a) Resultant paths (b) Steering and lateral acc. results

Figure 5.16: Feedforward path following results according to velocity

Fig. 5.16 depicts the resultant path in (a) by given curvature of proposed solution and its lateral acceleration profiles on travel time in (b). As shown in the figure, the resultant path being steered only by given curvature reference results in large error in x lateral positions and its error gets larger as the initial velocity increases from 3 to 9 m/s. As for the lateral acceleration, that is closely related to passenger ride comfort, also becomes higher as the initial velocity increases. From the resultant path in Fig. 5.16(a), even for a simple bicycle model, the tyre slip during path following should be considered to steering control. Theoretically, exact steeing angle for that consideration is described as follows.

$$\delta_f(s) = \kappa(s)L + \alpha_f - \alpha_r, \tag{5.11}$$

where δ_f is steering angle for front wheel and $\alpha_{f,r}$ is the slip angle in front and rear wheels respectively and it is assumed for the vehicle to be in stready state turning or *neutral steering* which are also referenced in the literatures of [Gillespie 92, Rill 11, Rajamani 06, Ulsoy 12] in more details. From Eq. (5.11), it is evident that if slip angle of front and rear wheels are fully known all times, accurate path following is possible only by regulating steering angle to the given curvature along the travel length.[5]

[5]However, in a real vehicle system, it is hard to measure or estimate lateral slip angle exactly since ground condition and tyre model is also varying [Doumiati 13].

5.6.2 Path following control for the proposed solution

As described in **Section 5.2**, the proposed **h-CCP** solution provides curvature reference for the vehicle to follow, thus its steering angle reference could be easily derived by manipulating vehicle and tyre dynamics considering combined slip conditions of both longitudinal and lateral sides.

In the following simulations, the planned velocity is implemented in vehicle dynamics framework and a simple steering control by orientation or heading angle feedback is used for integrating the proposed **h-CCP** solution and its path following control. The procedure in the simulation is descrived as follows.

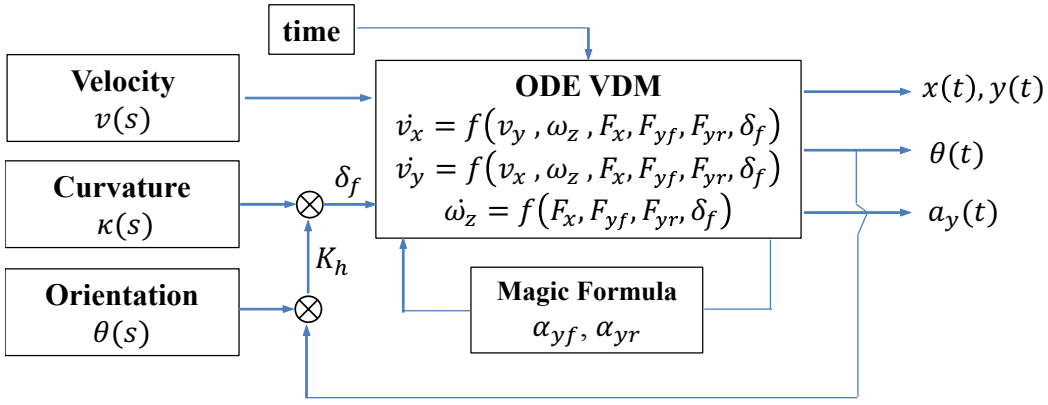
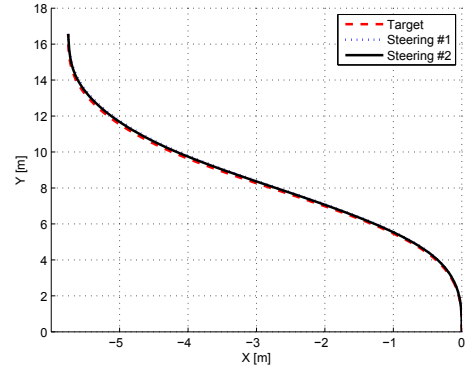


Figure 5.17: Block diagram of vehicle dynamics simulation

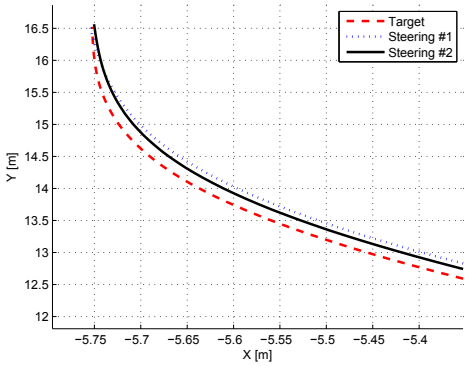
In Fig. 5.17 Vehicle Dynamics Model (**VDM**) by Ordinary Differential Equation (**ODE**) is presented with 0.01 *sec* sampling time iteration. The **ODE VDM** model has three state variables v_x, v_y and ω_z which means longitudinal, lateral velocity and yaw rate respectively. Per every sampling time, longitudinal force F_x and lateral force $F_{f,r}$ are required and F_x is applied by human acceleration/ deceleration pattern of 2.5 m/s^2 and $F_{f,r}$ are calculated from $\alpha_{f,r}$ by *Magic Formula* [Pacejka 92] data table. To **VDM**, proposed velocity $v(s)$ and curvature $\kappa(s)$ profiles are applied with time scheduling, where the steering input δ_f is calculated from the given curvature and orientation error feedback as,

$$\delta_f(t) = \kappa(t)L + K_h(\theta(t) - \theta_r(t)), \tag{5.12}$$

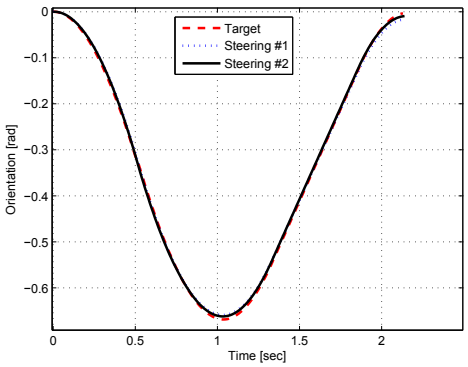
where K_h is the proportional gain for heading (or orientation) error feedback and the error is assumed to be measured by visual information on the lane or road geometry by *LiDAR*.



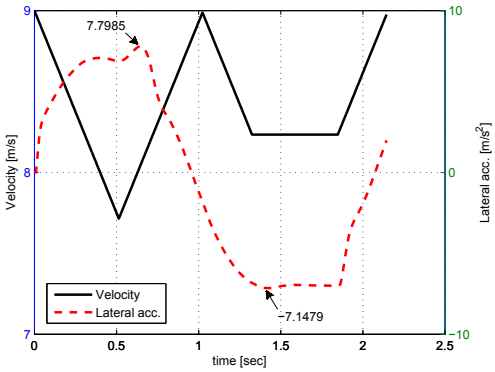
(a) Resultant paths



(b) Resultant path zoomed in



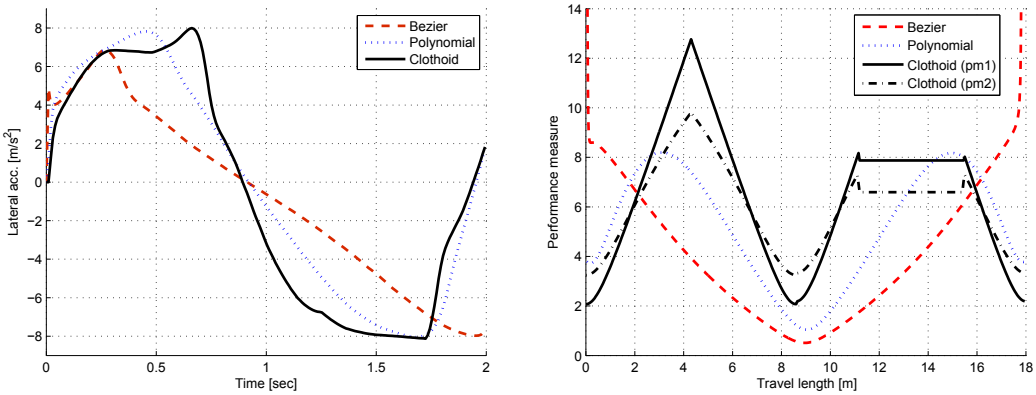
(c) Orientation results



(d) Velocity and Lateral acc.

Figure 5.18: Feedback path following results according to velocity profile

Fig. 5.19 depicts the feedback control results by using orientation information for planned velocity profile. Results in Figs. 5.19(a) to (c) are denoted by three cases of reference target, constant gain L (Steering #1) and orientation error gain K_h = 5.6 (Steering #2) denoted in Eq. (5.12). As shown, the case of Steering #2 has better path following performance within 1 *cm*. Fig. 4.18(d) illustrates the given velocity and curvature profiles and the corresponding lateral acceleration results. For lateral acceleration result, other methods are compared with performance measure, $\sqrt{a_{long}^2 + a_{lat}^2}$ which could estimate the passenger ride comfort (cf. *ISO 2631-1*) except vertical vibrations.



(a) Lateral acc. by **VDM** (b) Performance measure

Figure 5.19: Comparison with other methods

Fig. 5.19(a) compares the lateral acceleration results with other methods. In the result, *Bézier* path records higher jerking which could shock on the passenger and other two results show similariy in its peak except the symmetry. Compared to the result of polynomial path, the proposed clothoid path records lower rate change at both ends but higher rate of change in the middle (at 1 *sec*). Fig. 5.19(b) shows the performance measure **pm** results of three paths where the constant velocity and variable velocity cases for the proposed clothoid path are denoted as **pm 1** and **pm 2** respectively. In Fig. 5.19(b), it is obvious that the variable velocity has low peaks than the constant velocity case, thus enhanced performance. However, it is not so clear that it is efficient than the polynomial path in its peak values. On this point, this is from the unsymmetry of the path inspired from human patterns, more specifically, the proposed result **pm 2** has two different modes with different priorities such as higher priority for safety in the first half and then priority for low steering work with low sharpness in the second half of the path.

5.7 Conclusion

To the best of the author's knowledge, a **CCP** solution integrating human driving pattern has not been proposed in any previous research. As found in the planned experiments, the sharpness could be an important parameter that reflects the environmental status for the driving vehicle and it also represents human feeling and ride comfort while following the given path by the driver. In this respect, the path generation by separating two modes is efficient to consider both human aspect and algorithmic efficiency. Furthermore, the proposed methodology could be expanded by representing environmental status with risk measures to provide the pasengers in the vehicle with human natural and favorable maneuver.

This chapter presents a continuous curvature path generation method inspired by human driving patterns (**h-CCP**) in lane change maneuver for obstacle avoidance. From the analysis of human driving data, two different driving modes for static obstacle avoidance are found, and those human patterns are implemented to formulate the path generation problem of lane change maneuver. The resulting solution constructs a smooth and safe path for nonholonomic car-like vehicles which could also contribute to passengers' ride comfort. The problem is then defined specifically with additional constraints to be solved using efficient iterative algorithms. Compared to other smooth paths, the proposed **h-CCP** shows noticeable enhancement in passenger ride comfort and minimized steering work, similar to the way that it is performed by experienced human drivers.

This chapter proposes thus a methodology to integrate human driving patterns into algorithmic solution procedure, however it depends closely on quantity (*how much are acquired*) and quality (*how to acquire*) of the used human data and its sampling (*what to acquire*). It also has a room for expanding the obtained path solution to the velocity planning in dynamic environments for the purpose of its practical application into real experiments. In order to implement the proposed solution in a real vehicle system, the algorithmic performance needs to be more intensively examined and enriched with even more amount of driving data that will be done in our future work.

Conclusion and Prospects

This PhD thesis proposes solutions of continuous curvature path (or inverse clothoid problem) by constructing algorithmic methodology to generate **pCCP**. The obtained path allows the vehicle not only to guarantee smooth motion but also to enhance passenger comfort. For autonomous navigation of car-like vehicle, the proposed **pCCP** can be utilized to local path as well as global path generation.

To provide a capability of general solution of this inverse clothoid problem, the problem is classified into three sub-problems and each sub-problem is also divided into independent cases according to the geometric/kinetic relation between two boundary configurations. For each problem, algorithmic procedure to obtain **pCCP** was designed and several constraints for ensuring iterative convergence such as minimax sharpness constraint, mechanical limits, minimal number of clothoids and lateral acceleration were imposed for algorithmic efficiency. In all the algorithmic procedures, the sharpness (or rate of curvature) plays a major role to change the shape/pose of the clothoid in order to converge to the solution. The sharpness determines the rate of change of the vehicle steering, thus it implies that higher sharpness causes the rapid change for vehicle direction with high lateral acceleration. It consequently gives rise to high inertial force to the vehicle which is not desirable for passenger comfort. In that point, it was addressed a condition to find a minimax sharpness solution for the multiple clothoids composition to obtain a unique solution.

Flexibility of the proposed path was addressed for dynamic target, where the **pCCP** is adapted to satisfy the constraint and boundary condition of the problem, thus it makes **d-pCCP**. To accomplish safe navigation for dynamic target, obstacle avoidance maneuver should be required, and thus velocity planning on the generated **pCCP** is considered. Future status is analyzed in 4D configuration space and a strategy to avoid the obstacle is proposed by avoidance poses and velocity adaptation. The obtained velocity profile with the given path completes a trajectory to be followed by vehicle control system.

This thesis addressed an additional kind of **pCCP**, i.e., **h-CCP** that is inspired from human driving pattern for lane change maneuver in a straight road. From a

pre-defined experimental tests, avoidance and recover modes were extracted, and constraints for the problem were imposed to obtain a unique solution of **h-CCP**. Additional important solution constraint was obtained from human driving data. From the experiments of human expertise, a solution constraint was derived into an algorthmic procedure and the sharpness is regulated to optimize the human comfort measures while minimizing the steering work. In the local path planning, **h-CCP** was validated for effectiveness regarding passengers' comfort in a vehicle dynamics simulation.

The usefulness of the proposed solutions are enforced by some demonstrative examples applied to both local path planning and global path generation. The demonstrations highlighted the efficiency of the proposed algorithmic solutions and enhanced performance compared with other previous smooth path methods, and thus prospected its practical applicability into real autonomous navigated vehicle system. The generated path could give smooth steering with lower lateral acceleration in work-saving way. Its effectiveness to the practical implementation to real vehicle was validated with several simulated demonstrations. In the simulation using a standard controller, the vehicle follows the smoothed path with better performance than the reference path obtained by raw data which is acquired from precise position sensors. This **pCCP** has the advantage that the vehicle follows the fully defined curvature diagram along the travel distance for the steering control which is constrained to its predefined track and thus it could exert the role of *virtual rail* to the nonholonomic vehicle. It has a noticeable significance that the continuous curvature transmits the steering command during the travel distance directly to the low level control system and it could reduce any error on the resultant trajectory of the real vehicle. Under the conditions of using exact kinematics/dynamics vehicle model without ground slip, sensor/actuator errors and control uncertainties, the vehicle could follow the curvature/sharpness diagram for accurate path following which looks like *traveling along a virtual rail* and could overcome GPS or sensor outages.

To be more useful for the application into real autonomous navigated vehicle system, this thesis still requires for additional proofs and further investigative works as follows. In generating **pCCP** for static target and its implementation into a global path construction, how to benefit an optimality on selecting boundary configurations and its performance evaluation with computational burden could enforce the applicability and praticality of the proposed solutions to commercial vehicle system. Also in **d-pCCP** for dynamic target, the method to decide between avoidance pose replanning and velocity adaptation requres to be more specified in its optimization rule and performance criteria for various cases of dynamic environments. For the algorithmic point of view, the proposed algorithm also needs to be enhanced. Especially, the algorithmic time consumption increases as

the number of clothoids increase, thus the calculation time should be remarkably reduced by optimizing the initial parameters setting. Furthermore, more various cases for the problem need to be analyzed and solved for the efficiency of the proposed algorithmic procedures (to be robust regardless of range and scale in the boundary configuration). The solution constraint should be more investigated from various experiments data by human expert drivers with various type of vehicles and roads environment. As the data gets bigger for those environments, the optimization guidance could be defined better in order to guarantee its real-time applicability in all vehicle's dynamics and/or maneuvers.

The proposed solution desires for a *virtual rail* that includes nonholonomic path and guarantees accurate path following for car-like vehicles. So that the proposed trajectories become more practical, dynamics effect between steering and tyres on the ground should also be investigated in depth by both theoretical and experimental considerations. Considering all numerous environmental cases to clarify actual dynamics is difficult, thus applying the path solution to vehicle dynamics simulation program could be an efficient strategy before entering into road test. Furthermore, since only a few optimization criteria (such as mini-max sharpness, maximal comfort, minimal travel time/length, minimal velocity changes and so on) to construct the problem is not sufficient to satisfy human passenger in the vehicle, human driving data could be much more interesting in further researches for vehicle autonomy.

Appendix

Appendix A

Classification of autonomous navigated vehicle

A formal classification system for autonomous navigated vehicles has been proposed by the National Highway Traffic Safety Administration (NHTSA) in USA.

- Level 0: Driver has complete control of vehicle at all times.
- Level 1: Some vehicle controls are automated, e.g. automatic braking.
- Level 2: Two or more controls can be automated at the same time, e.g. cruise control and lane keeping.
- Level 3: The driver can cede control in certain circumstances.
- Level 4: Driver not expected to play any part in the driving process at all.

Appendix B

Elementary clothoid generation

B.1 Clothoid property for parameter variation

From Eq. (1.14) (p.29), following is derived.
When $s = (2\ \delta)^{\frac{1}{2}} \cdot \alpha^{\frac{-1}{2}}$, then $\frac{ds}{d\alpha} = \alpha^{\frac{-3}{2}} \cdot [-\delta^{\frac{1}{2}}]$. Thus, if a constant $\delta > 0$ with $\alpha > 0$, then $\frac{ds}{d\alpha} < 0$, and it is true that for an increasing α, the length s shrinks while the clothoid bends upward (cf. Fig. 1.2, p.30).
Other patterns of $\frac{ds}{d\kappa}$ or $\frac{ds}{d\delta}$ could be obtained by the same analogy.

B.2 Clothoid generation function

To compromise mathematical coordinate with the vehicle coordinate frame, following transformation procedure is performed. When a clothoid is defined as Eq.(1.7) and Eq.(1.8) (p.28), a rotation transformation around *z-axis* (counter-clockwise as positive) is defined from (x_1, y_1) to (x_2, y_2) as

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}, \tag{B.1}$$

where this transformation is realized by MATLAB function `[x2, y2] = rotate_th(x1, y1, theta)` and `theta` is the amount of rotation angle. To align the vehicle path in the forward direction, a original path (x_0, y_0) is transformed to (x, y) as follows.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos(\frac{\pi}{2}) & -\sin(\frac{\pi}{2}) \\ \sin(\frac{\pi}{2}) & \cos(\frac{\pi}{2}) \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ -y_0 \end{bmatrix} \tag{B.2}$$

where it is presentedy by `[x, y] = rotate_th(x0, -y0, pi/2)` in MATLAB.

Thus an original path of Eqs.(1.7) and (1.8) (p.28) is reformulated into Eq.(B.2) and it is realized by MATLAB as `[x, y, α, s] = clothoid_gen(δ, κ)`

Algorithm 21 clothoid generation function

Require: δ, κ, ds ▷

1: **function** `[x, y, α, s] = clothoid_gen(δ, κ)`
2: $\alpha = \frac{\kappa^2}{2\delta}$
3: $s = \sqrt{\frac{2\delta}{\alpha}}$ ▷ total travel length
4: dt $= \sqrt{\frac{\alpha}{\pi}}$ ds
5: $\mathbf{t}$ = dt:dt:$\sqrt{\frac{2\delta}{\pi}}$
6: $\mathbf{x} = \frac{\sqrt{2\pi\delta}}{\kappa}$ `cumtrapz`$(\sin(\frac{\pi}{2}\mathbf{t}^2))dt$
7: $\mathbf{y} = \frac{\sqrt{2\pi\delta}}{\kappa}$ `cumtrapz`$(\cos(\frac{\pi}{2}\mathbf{t}^2))dt$
8: `[x, y] = rotate_th(x, -y,` $\frac{\pi}{2}$`)`
9: **return** $\mathbf{x}$, $\mathbf{y}$, α, s
10: **end function**

where $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{t}$ are all vectors of the same size which can be obtained by `length(x)` in MATLAB. The `cumtrapz`*Cumulative trapezoidal numerical integration* function computes anapproximation of the cumulative integral via trapezoidal method with unit spacing iincrement and can be referenced also in MAT-

LAB. Through all demonstrative problems, $ds = 0.05\ m$ is applied to compromise between the algorithmic loop time and convergence error.

B.3 Arc generation function

The function `[`x_r`, `y_r`] = arc_gen(`ρ`, `θ`)` generates an arc segment (x_r, y_r) of which radius is ρ with θ as the angle of the circumference. By the operator $\oplus$, this arc is connected to a clothoid segment with orientation continuity.

Algorithm 22 Arc generation function

Require: $\rho, \Delta\theta$, ds ▷

function `[`$\mathbf{x_r}$`, `$\mathbf{y_r}$`] = arc_gen(`ρ`, `$\Delta\theta$`)`

- $c_x = \rho$, $c_y = 0$ ▷ Center of circular arc
- $s_r = \rho \cdot \Delta\theta$ ▷ Arc length
- $\partial\theta = \frac{ds}{\rho}$ ▷ Points interval along the arc
- $\theta_{\mathbf{c}} = \partial\theta : \partial\theta : \Delta\theta$
- $\mathbf{x_r} = c_x + \rho \cos(\pi - \theta_{\mathbf{c}})$
- $\mathbf{y_r} = c_y + \rho \sin(\pi - \theta_{\mathbf{c}})$
- **return** $\mathbf{x_r}$, $\mathbf{y_r}$

end function

B.4 Convergence criteria for an elementary clothoid

This part is to provide fundamental algorithm for solvability or convergence criteria with multiple clothoids problem. It is also important to describe coverage with given number of clothoids to guarantee the convergence for the problem. For that purpose, a simple problem which finds an elementary clothoid while reaching only a given target position is considered. The problem has a following boundary condition.

Problem: $\boldsymbol{P_i}\left(x_i, y_i, \frac{\pi}{2}, \boldsymbol{\kappa_i = 0}\right)$ to $\boldsymbol{P_f}\,(x_f, y_f)$

where an elementary clothoid can be generated by regulating two parameters and there is no limitation for θ_f (i.e. δ) or κ_f.

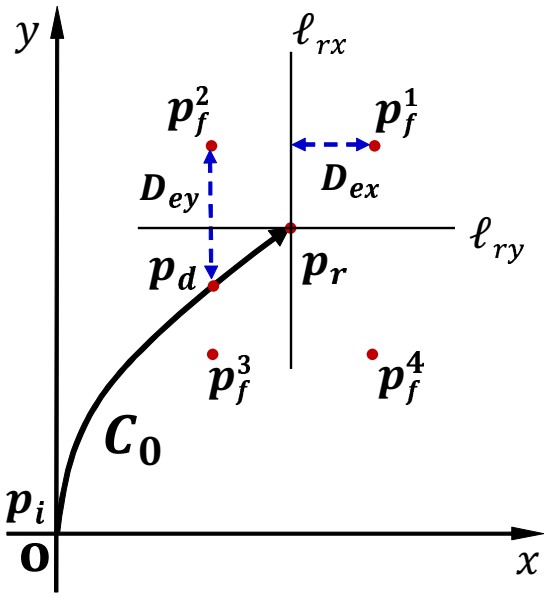


Figure B.1: Geometric representation for convergence criteria

An elementary clothoid $\boldsymbol{C_0}$ is generated to have its end point, (a reference point) $\boldsymbol{p_r}$. The objective of the algorithmic loop is to determine the parameter variation rule for $\boldsymbol{p_r}$ to reach the target position, $\boldsymbol{p_f}$. For that purpose, two reference lines are defined as ℓ_{rx}, ℓ_{ry} which are drawn from $\boldsymbol{p_r}$ with horizontal (parallel to *x-axis*) and vertical (parallel to *y-axis*) direction respectively. Then, the convergence criteria is defined by,

$$|\boldsymbol{D_{ey}}| \leq \varepsilon,\ \boldsymbol{D_{ex}} \leq 0 \tag{B.3}$$

where $\boldsymbol{D_{ex}}$ is the horizontal (minimum) distance from $\boldsymbol{p_f}$ to ℓ_{rx} and and $\boldsymbol{D_{ey}}$ is the vertical (minimum) distance from $\boldsymbol{p_f}$ to $\boldsymbol{p_d}$ respectively. The point $\boldsymbol{p_d}$ is determined by the closest one in $\boldsymbol{C_0}$ of satisfying $x \geq x_f$ when $\boldsymbol{D_{ex}} \leq 0$, where $[x, y] \in C_0$.

The criteria in Eq. (5.3) indicates that the convergence is attained if the absolute distance error $|\boldsymbol{D_{ey}}|$ is less than a designed threshold ε while holding a positive value for $\boldsymbol{D_{ex}}$. Here, two conditions affect the parameter variation rule in each iteration loop; $|\boldsymbol{D_{ey}}|$ makes a formulation for α variation, and $\boldsymbol{D_{ex}}$ guides δ variation by following determinant functions.

$$\begin{cases} \lambda_{rx}(\boldsymbol{p_f}) = D_{ex} > 0 \ (< 0) & : \boldsymbol{p_f} \text{ is on the right (left) side of } \ell_{p_r}. \\ \lambda_{ry}(\boldsymbol{p_f}) = D_{ey} > 0 \ (< 0) & : \boldsymbol{p_f} \text{ is on the upper (lower) side of } \boldsymbol{C_0}. \end{cases} \tag{B.4}$$

The determinant function $\lambda_{ry}(\boldsymbol{p_f})$, which checks the geometric relation between $\boldsymbol{p_f}$ and $\boldsymbol{p_d}$ by $y_d - y_f$, can be used to change clothoid parameter by using one of the following rule.

Property 6 *Parameter variation rule for elementary clothoid*

a. If $\lambda_{ry}(\boldsymbol{p_f}) > 0 \ (< 0)$, *then* $\boldsymbol{\alpha}$ *should increase (decrease) with constant* δ,

b. If $\lambda_{rx}(\boldsymbol{p_f}) > 0 \ (< 0)$, *then* $\boldsymbol{\delta}$ *should increase (decrease) with constant* α.

In Fig. B.1, target positions in the four quadrants are denoted from $\boldsymbol{p_f^1}$ to $\boldsymbol{p_f^4}$ in counter-clockwise. Based on **Property 6**, both $\boldsymbol{p_f^1}$ and $\boldsymbol{p_f^4}$ hold the condition of $D_{ex} > 0$, thus δ should increase, whereas $\boldsymbol{p_f^2}$ and $\boldsymbol{p_f^3}$ are under the conditions as $D_{ey} < 0$ and $D_{ey} > 0$ respectively, thus α should decrease for $\boldsymbol{p_f^2}$ and increase for $\boldsymbol{p_f^3}$. Using above properties, the algorithm finds a clothoid solution which reaches the target position $\boldsymbol{p}_f$.

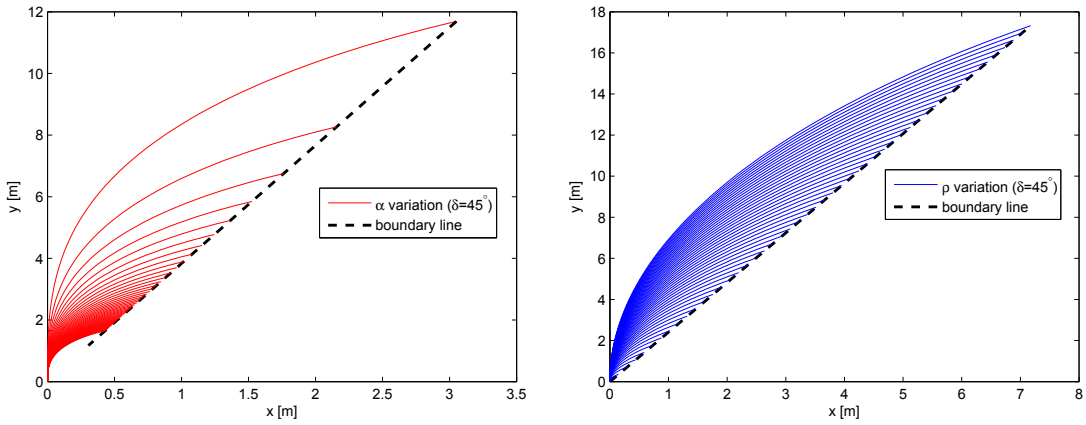
Algorithm 23 enables the convergence into a solution for the given target position, however there is limitation under the elementary clothoid constraint (cf. Fig. 1.3, p.31) so that it can not guarantee the convergence.

Algorithm 23 Target reaching by one clothoid

Require: x_f, y_f ▷ Given target
Require: $\varepsilon, \delta, \alpha, d\delta, ds, \text{sol} = \text{FALSE}$ ▷ Initial assumptions
procedure CLOTHOID1(x_f, y_f)
  while sol = FALSE **do**
    $\kappa = \sqrt{2\delta\alpha}$
    `[x, y, α, s] = clothoid_gen`(δ, κ)
    index $\leftarrow$ find$(\mathbf{x} > x_f)$
    if $\sim$ isempty(index) **then**
      $\lambda = \mathbf{y}(\text{index}(1)) - y_f$ ▷ $\boldsymbol{p_d} \leftarrow \boldsymbol{C_o}(\text{index}(1))$
      $D_{ey} = |\lambda|$
      if $|D_{ey}| \leq \varepsilon$ **then**
        sol = TRUE
        $\mathbf{x}' = \mathbf{x}(1 : \text{index}(1)), \mathbf{y}' = \mathbf{y}(1 : \text{index}(1))$
        return $C_0(\mathbf{x}', \mathbf{y}')$ ▷ Obtained solution
      end if
      if $\lambda\lambda' < 0$ **then** ▷ Bisection method for α
        $d\alpha = \frac{d\alpha}{2}$
      end if
      $d\alpha = |d\alpha| \cdot \text{sgn}(\lambda)$
      $\alpha = \alpha + d\alpha$
      $\lambda' = \lambda$ ▷ For next iteration
    else
      $\delta = \delta + d\delta$
    end if
  end while
end procedure

B.5 Covering range and convergence limitation

One of the most important constraint for convergence is the limitation on the reacheability of the elementary clothoid. To determine the boundary of coverage by clothoid, following plottings by parameter variations are performed in Fig. B.2.



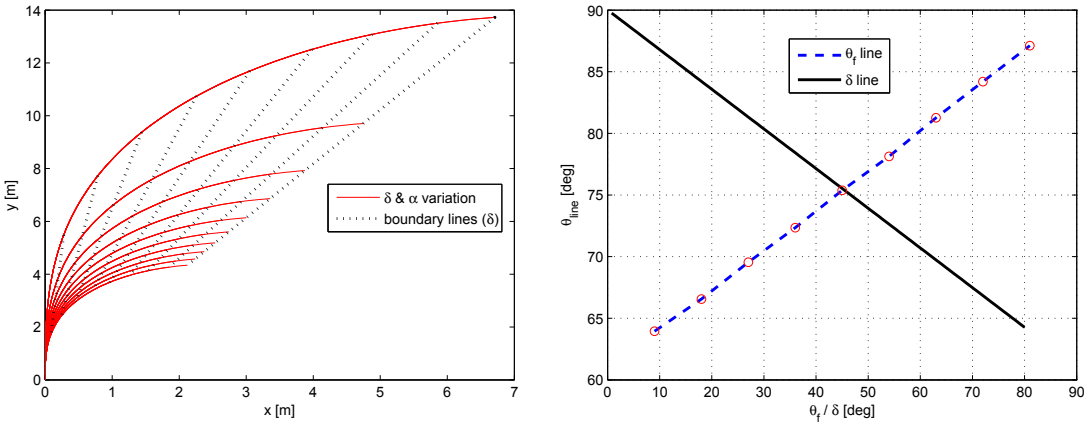
(a) Clothoid boundary (b) Arc boundary

Figure B.2: Boundary lines for fixed $\delta = 45°$ $[deg]$

With the fixed $\delta = 45°$ $[deg]$, α was varied for the elementary clothoid to be shown in Fig. B.2(a) and ρ (radius of the arc) was varied for circular arc to be depicted in (b). By the constraint on the other parameters, each segment has regular pattern (cf. Fig. 1.2 and **Property 1**, p.30), where end points for the given δ could be connected to make a boundary line. The boundary lines shown in Fig. B.2(a) and (b) signifiy the coverable region by clothoid and circular arc respective; therefore a target point outside the boundary line is not accessible by the segment or it is not able to make a convergence by parametric variation. To make a formulation for convergence range on a clothoid, α and δ are varied to formulate a function for its end points by linear fitting.

Fig. B.3(a) illustrates the boundary lines for different δ variations, where each line is fitted within a negligible error bound. Fig. B.3(b) shows the corresponding line function between θ_f (δ) and θ_{line} where θ_{line} is the slope of the boundary line for the clothoids ends. Obtained linear fitting result provides an linear equation for determining a general function of checking the coverage range of a clothoid as well as a circular arc about the given target.

$$\theta_{line} = -0.3218(\delta - 81) + 63.9444 \tag{B.5}$$



(a) Clothoid boundary (b) Arc boundary

Figure B.3: Boundary lines for δ variation

Eq.(B.5) is a function which determines the slope of the boundary line according to δ variation. From the above analysis, an algorithm to check the convergence for an elementary clothoid could be formulated as follows.

Algorithm 24 Checking for convergence

Require: x_f, y_f, δ, κ

function `result = check_range`(x_f, y_f, δ)
  $\theta_f = (\frac{\pi}{2} - \delta)\frac{180}{\pi}$ ▷ radian to degree
  $\theta_{line} = 0.3218(\theta_f - 9) + 63.9444$ ▷ slope for boundary line, Eq.(B.5)
  `[x, y,` α`, s] = clothoid_gen(`δ`,` κ`)`
  if $(y_f \geq \tan(\theta_{line})(x_f - \mathbf{x}(\text{end})) + \mathbf{y}(\text{end}))$ **then**
    result = TRUE ▷ Within boundary; reachable by a clothoid
  else
    result = FALSE ▷ Out of boundary; not reachable by a clothoid
  end if
end function

The result for **Algorithm 24** provides the possibility of convergence for the given target by an elementary clothoid, and this method could be applied in multiple clothoids problem to check whether the given target is accessible by the composition of elementary clothoids.

B.6 Enhancement of iterative convergence

As the number of composed clothoids increases, the number of parameters increases, and convergence time is delayed on an exponential scale. Thus, efficient constraints in the proposed algorithmic procedures are required; the following subsections address this point based on initial parameter assumptions and a solution constraint. Before entering into iteration, both end configurations are given. Thus, α_{1i} should be tried first. However, if the generated $\boldsymbol{C_1}$ is far from the target position ($\boldsymbol{p_f}$), the procedure requires a lot of convergence time. In this respect, the method to assume an initial parameter is proposed as follows.

In the initial step, the only information from given configurations is the configuration distance (d in Fig. 1.4(a), p.32) between the ends named by d_{if}. This means that there might be an efficient formulation linking α_{1i} to d_{if}. While being interested in the relation among the distance and clothoid parameters, the process starts from the basic property for a clothoid, as presented in **Section 1.2**. From Fig. 1.2(a) (p.30), one can observe that the distance d from the origin to the clothoidal end decreases as the sharpness increases with fixed deflection.

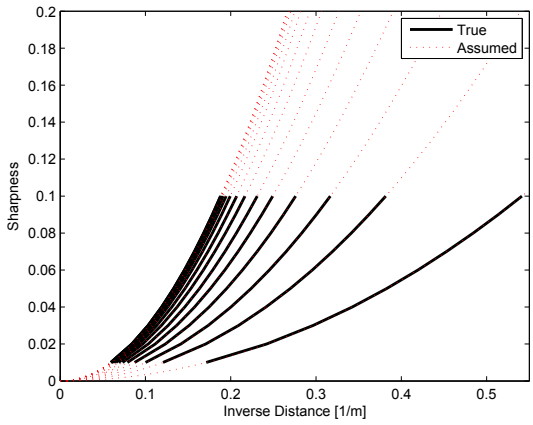
This result reversely says the distance d_{if} increases as the sharpness decreases. Using this relation, a function that determines sharpness from the distance parameter could be approximated using a simple second-order polynomial equation[1] of $d_{\text{if}} = \mathcal{G}\alpha^2$, where $\mathcal{G}$ is a coefficient for the relation. It is also found that $\mathcal{G}$ changes with deflection δ in the clothoid; thus, δ needs to be included as a variable into the coefficient $\mathcal{G}$ of the approximation function as follows,

$$\begin{aligned} \alpha_{1i} &= \mathcal{G} \cdot (1/d_{\text{if}})^2, \\ \mathcal{G} &= -0.3352\ \delta^2 + 2.2111\ \delta - 0.0429 \end{aligned} \tag{B.6}$$

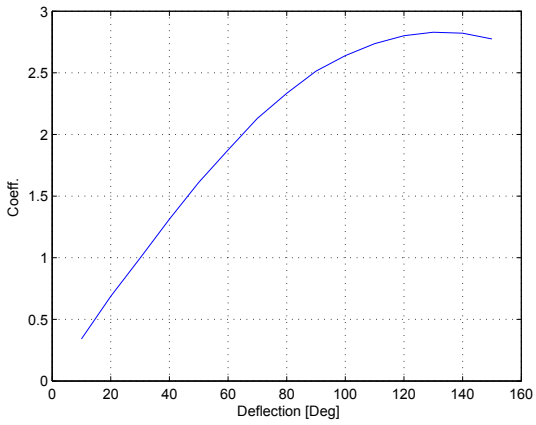
where the function approximates the relation between α_{1i} and d_{if} within 10^{-2}, where its coefficient $\mathcal{G}$ is determined using second-order polynomial fitting with the variable δ.

Figure B.4 shows the approximation function using empirical results, where (a) depicts the true result, and its appriximated result using the coefficient is shown in (b). From Fig. B.4, it is shown that Eq.(B.6) fits well for a d_{if} range of 1 m to 50 m as well as for a δ range of 0° to 160°, which are feasible in typical passenger vehicles. The derivative $d\alpha$ for the initial step can also be determined from differentiating equation Eq.(1.7) (p.28) as $\frac{\partial \alpha_{1i}}{\partial d_{\text{if}}}$ and by inserting ∂d_{if} and current d_{if} for the given configuration, for instance using $\partial d_{\text{if}} = d_{\text{if}}/100$.

[1] The MATLAB function-*polyfit* was used.



(a) Approximation of initial sharpness (α_{1i})

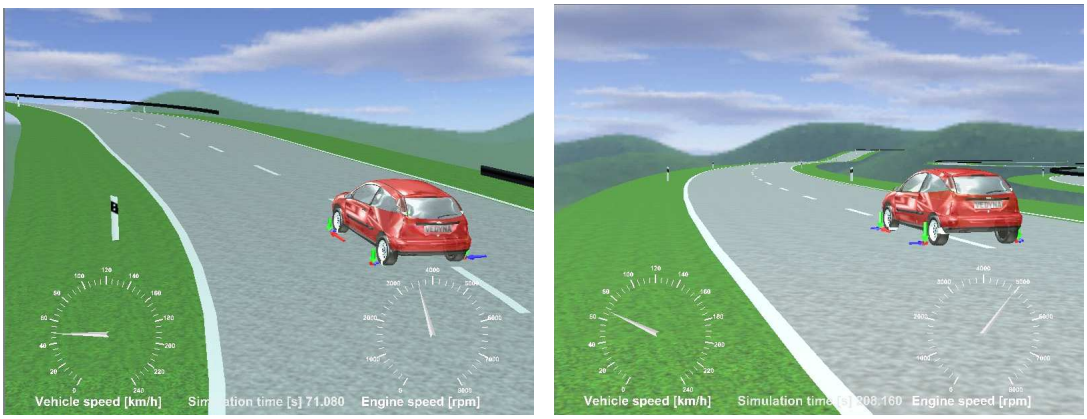


(b) $\mathcal{G}$ variation by δ

Figure B.4: Approximation function for initial sharpness

Appendix C

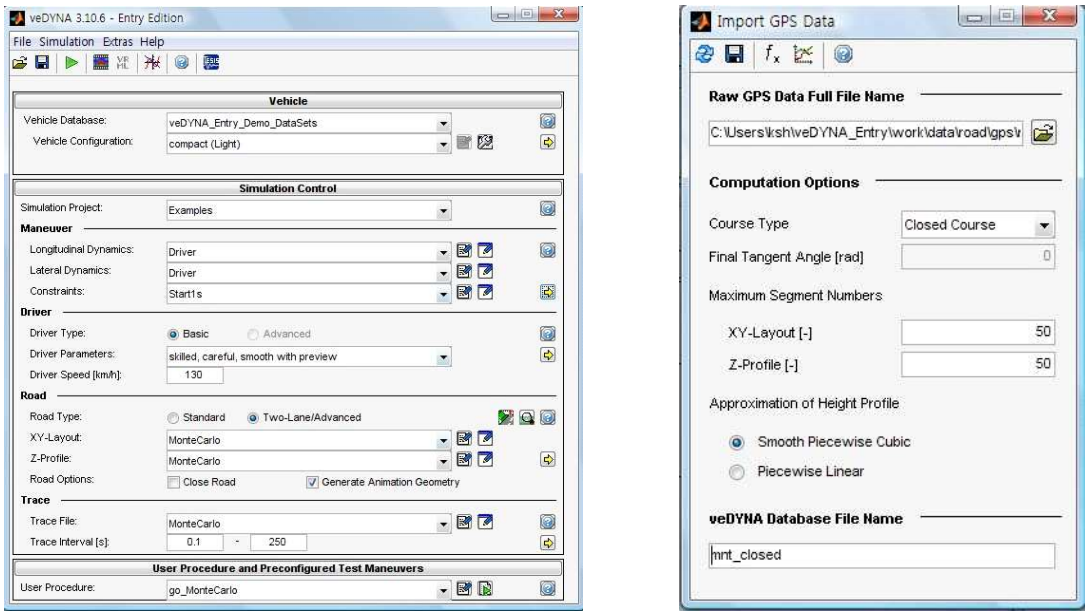
Raw data acquisition and conversion in ve-DYNA Entry®



(a) Driving scene #1 (b) Driving scene #2

Figure C.1: Vehicle dynamics simulation in **LR-track**

Fig. C.1(a) is the GUI panel to set up a vehicle dynamics simulation, where necessary procedures to make a simulation on VRML environment are separated by several items such as vehicle, maneuver, driver, road and trace. In the buttom list menu box, *user procedure* can be chosen from a few items, where *go_ MonteCarlo* item is selected to load the database for the considered **LR-track**. Fig. C.2(b) is the subsequent GUI panel under the set up of Fig. C.2(a) to invoke the raw GPS position data and then convert into a interpolated smooth path for the vehicle to follow. The GPS data is provided by the program and the conversion procedure applies the *Piecewise Cubic Hermite Interpolating Polynomials* (PCHIP) algorithm under the constraint of 50 segments with closure



(a) Simulation setting panel (b) Data conversion panel

Figure C.2: The procedure for obtaining a raw data in ve-DYNA®

course. After the conversion procedure, the resultant file (in the buttom edit box) has the smooth path with 50 number of way points where each way point has three component of information of x, y and θ, i.e. orientation and tangential angle (orientation). More specific information with procedural function for every menu and item is described in *User manual* or *Road manual* of the ve-DYNA® documentation lists.

Bibliography

[Abele 16] Rudiger Abele. *High-precision maps for self-driving cars.* `https://www.mercedes-benz.com/en/mercedes-benz/next/connectivity/high-precision-maps-for-self-driving-cars/`, 2016. Accessed: 2016-1-14.

[Adouane 08] Lounis Adouane. *An Adaptive Multi-Controller Architecture for Mobile Robot Navigation.* 10th IAS, Intelligent Autonomous Systems, pp. 342–347, Baden-Baden, Germany, July 23-25 2008.

[Adouane 13] Lounis Adouane. *Toward Smooth and Stable Reactive Mobile Robot Navigation using On-line Control Set-points.* IEEE/RSJ, IROS'13, 5th Workshop on Planning, Perception and Navigation for Intelligent Vehicles, Tokyo-Japan, 3-7 November 2013.

[Adouane 16] Lounis Adouane. *Autonomous Vehicle Navigation: From Behavioral to Hybrid Multi-Controller Architectures.* Taylor & Francis CRC Press, ISBN: 9781498715584, 228 pages, April 2016.

[Adouane 17] Lounis Adouane. *Reactive versus Cognitive Vehicle Navigation based on Optimal Local and Global PELC.* Robotics and Autonomous Systems (RAS), vol. 88, pp. 51 – 70, 2017.

[Akcelik 01] Rahmi Akcelik & Mark Besley. *Acceleration and deceleration models.* Conf. of Australian Institutes of Transport Research, vol. Melbourne, Australia, pp. 1–9, 2001.

[Attia 14] Rachid Attia, Rodolfo Orjuela & Michel Basset. *Combined longitudinal and lateral control for automated vehicle guidance.* Vehicle System Dynamics, vol. 52(2), pp. 261–279, 2014.

[Avanzini 09] Pierre Avanzini, Benoit Thuilot & Philippe Martinet. *Obstacle avoidance and trajectory replanification for a group of communicating vehicles*. IEEE Int. Conf. on Intel. Transport Systems Telecommunications, pp. 267–272, Oct. 2009.

[Avanzini 12] Pierre Avanzini, Benoit Thuilot & Philippe Martinet. *Manual convoying of automated urban vehicles relying on monocular vision*. Int. Vehicles Symposium, pp. 19–24, June. 2012.

[Bae 13] Il Bae, Jin Hyo Kim & Shiho Kim. *Steering rate controller based on curvature of trajectory for autonomous vehicles*. IEEE Intel. Vehicles Symposium, pp. 1381–1386, 2013.

[Bastow 04] Donald Bastow, Geoffrey Howard & John P. Whitehead. Car suspension and handling. Society of Automotive Engineers Inc., 400 Commonwealth Drive, Warrendale, PA 15096, USA, 2004.

[Benzerrouk 14] Ahmed Benzerrouk, Lounis Adouane & Philippe Martinet. *Stable Navigation in Formation for a Multi-robot System Based on a Constrained Virtual Structure*. Robotics and Autonomous Systems (RAS), vol. 62, no. 12, pp. 1806 – 1815, 2014.

[Bevan 07] Geraomt Paul Bevan, Henrik Gollee & J. O'reilly. *Automatic lateral emergency collision avoidance for a passenger car*. Int. J. of Control, vol. 80(11), pp. 1751–1762, 2007.

[Bhuiyan 17] Johana Bhuiyan. *Uber's autonomous cars drove 20,354 miles and had to be taken over at every mile, according to documents*. `https://www.recode.net/2017/3/16/14938116/uber-travis-kalanick-self-driving-internal-metrics-slow-progress`, 2017. Accessed: 2017-3-23.

[Biagiotti 08] Luigi Biagiotti & Claudio Melchiorri. Trajectory planning for automatic machines and robots. Springer, ISBN 978-3-540-85628-3, Heidelberg, 2008.

[Blog 17] Idwu Blog. *Self-driving Uber? Self-Driving Cars Pros and Cons*. `http://www.idrivewithuber.com/self-driving-uber-self-driving-cars-pros-cons/`, 2017. Accessed: 2017-3-20.

[Brambley 11] Galem Brambley & N. L. Fulton. *Specific length trajectories optimised for maximum acceleration using conic parameterised clothoids*. IEEE Tran. on Aerospace and Electronic Systems, vol. 47(4), pp. 2627–2636, 2011.

[Brezak 13] Misel Brezak & Ivan Petrovic. *Real time approximation of clothoids with bounded error for path planning applications*. IEEE Trans. on Robotics, vol. PP(99), pp. 1–9, 2013.

[Canudas 96] Carlos Canudas, Bruno Sicliano & Georges Bastin. Theory of robot control. Springer, ISBN 3-540-76054-7, Berlin Heidelberg, 1996.

[Chakravarthy 98] Animesh Chakravarthy & Debasish Ghose. *Obstacle avoidance in a dynamic environment: A collision cone approach*. IEEE trans. on Systems, Man and Cybernetics-Part A: Systems and Humans, vol. 28(5), pp. 562–574, 1998.

[Chakravarthy 11] Animesh Chakravarthy & Debasish Ghose. *Collision cones for quadric surfaces*. IEEE Trans. on Robotics, vol. 27(6), pp. 1159–1166, 2011.

[Chakravarthy 12] Animesh Chakravarthy & Debasish Ghose. *Generalization of the collision cone approach for motion safety in 3-D environments*. Autonomous Robot, vol. 32, pp. 243–266, 2012.

[Chebly 15] A. Chebly, G. Tagne, R. Talj & A. Charara. *Local Trajectory Planning and Tracking For Autonomous Vehicle Navigation Using Clothoid Tentacles Method*. International IEEE Conference on Intelligent Vehicles Symposium (IV), pp. 674–679, Seoul, South Korea, June 2015.

[Chen 13] Jiajia Chen, Pan Zhao, Tao Mei & Huawei Liang. *Lane change path planning based on piecewise Bezier curve for autonomous vehicle*. IEEE Int. Conf. on Vehicular Electronics and Safety, pp. 17–22, 2013.

[Chevallereau 02] Christine Chevallereau & Lounis Adouane. *On line reference trajectory adaptation for the control of a planar biped*. 5th International Conference on CLimbing And WAlking Robots (CLAWAR), pp. 427–436, Paris-France, 25-27, September, 2002.

[Choi 10] Ji Wung Choi, Renwick E. Curry & Gabriel H. Elkaim. *Curvature-continuous trajectory generation with corridor constraint for autonomous ground vehicles*. IEEE conf. on Decision and Control, pp. 7166–7171, 2010.

[Chung 09] Woojin Chung, Chang bae Moon & Jae-Bok Song et al. *Safe Navigation of a Mobile Robot Considering Visibility of Environment*. IEEE Transactions on Industrial Electronics, vol. 56, no. 10, pp. 3941–3950, 2009.

[Ctvnews 17] Ctvnews. *Uber's self-driving cars encountering technical issues: report*. `http://www.ctvnews.ca/autos/uber-s-self-driving-cars-encountering-technical-issues-report-1.3333530`, 2017. Accessed: 2017-3-23.

[DeLuca 98] A. DeLuca, G. Oriolo & C. Samson. Feedback control of a nonholonoic car-like robot (robot motion planning and control). Springer-Verlag Berlin Heidelberg, 1998.

[DenBerg 08] Jur Van DenBerg, Ming Lin & Dinesh Manocha. *Reciprocal velocity obstacles for real-time multi-agent navigation*. IEEE Int. Conf. on Robotics and Automation, pp. 1928–1935, 2008.

[Dolgov 10] Dmitri Dolgov, Sebastian Thrun, Michael Montemerlo & James Diebel. *Path planning for autonomous vehicles in unknown semi-structured environments*. The International Journal of Robotics Research, vol. 29, pp. 485–501, 2010.

[Doumiati 13] Moustapha Doumiati, Ali Charara, Alessandro Victorino & Daniel Lechner. Vehicle dynamics estimation using kalman filtering. ISTE Ltd and John Wiley & Sons, Inc., 27-37 St George's Road, London SW 19 4EU, UK, 2013.

[Dubins 57] L. E. Dubins. *On curves of minimal lengt with a constraint on average curvature, and with prescribed initial and terminal positions and tangents*. American Journal of Mathematics, vol. 79, pp. 497–516, 1957.

[Duda 72] Richard O. Duda & Peter E. Hart. *Use of the Hough Transformation to detect lines and curves in pictures*. Graphics and Image Processing, vol. 15(1), pp. 11–15, 1972.

[Duda 01] Richard O. Duda, Peter E. Hart & David G. Stork. Pattern classification. Wiley-Interscience, 2001.

[Dudek 10] Gregory Dudek & Michael Jenkin. Computational principles of mobile robotics, vol. 32 Avenue of the Americas, New York, USA. 2010.

[Edelmann 07] Johannes Edelmann, Manfred Plochl, Werner Reinalter & Werner Tieber. *A passenger car driver model for higher lateral accelerations*. Vehicle System Dynamics, vol. 45(12), pp. 1117–1129, 2007.

[Elfes 89] Alberto Elfes. *Occupancy Grids: A probabilistic framework for robot perception and navigation*. Thèse de doctorat, Department of Electrical and Computer Engineering, Carneigie Mellon University, Carnegie Mellon University Pittsburgh, PA, USA, 1989.

[ESRI 16] ESRI. *ArcGIS*. `http://www.esri.com/software/arcgis`, 2016. Accessed: 2016-10-25.

[Fiorini 98] Paolo Fiorini & Zvi Shiller. *Motion planning in dynamic environments using velocity obstacles*. Int. J. of Robotics Research, vol. 17(7), pp. 760–772, 1998.

[Fraichard 03] Thierry Fraichard & Hajime Asama. *Inevitable Collision States. A Step Towards Safer Robots?* IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, vol. Oct. 27-31 Las Vegas, US, pp. 388–393, 2003.

[Fraichard 04] Thierry Fraichard & Alexis Scheuer. *From Reeds and Shepp's to continuous curvature paths*. IEEE Trans. on Robotics, vol. 20, pp. 1025–1035, 2004.

[Fulgenzi 07] Chiara Fulgenzi, Anne Spalanzani & Christian Laugier. *Dynamic obstacle avoidance in uncertain environment combining PVOs and Occupancy Grid*. IEEE Int. Conf. on Robotics and Automation, pp. 1610–1616, 2007.

[Ghita 12] Narcis Ghita & Marius Kloetzer. *Trajectory planning for a car-like robot by environment abstraction*. Robotics and Autonomous Systems, vol. 60, pp. 609–619, 2012.

[Giesbrecht 04] J. Giesbrecht. *Global path planning for unmanned ground vehicles*. 2004. Technical Memorandum, DRDC SUFFIELD-TM-2004-272.

[Gillespie 92] Thomas D. Gillespie. Fundamentals of vehicle dynamics. Society of Automotive Engineering, Inc., Warrendale, USA, 1992.

[Gim 12] Suhyeon Gim, Ilyas Meo, Yongjin Park & Sukhan Lee. *Drivable road recognition by multilayered LiDAR and Vision*. Proc. of the Int. Conf. IAS-12, Jeju Island, Korea, 26-29, June 2012.

[Gim 14a] Suhyeon Gim, Lounis Adouane, Sukhan Lee & Jean-Pierre Derutin. *Parametric Continuous Curvature Trajectory for Smooth Steering of the Car-like Vehicle*. 13th Int. Conf. on Intelligent Autonomous Systems, pp. 1327–1342, 15-19, July 2014.

[Gim 14b] Suhyeon Gim, Lounis Adouane, Sukhan Lee & Jean-Pierre Derutin. *Smooth Trajectory Generation with 4D Space Analysis for Dynamic Obstacle Avoidance*. Special Session on Intelligent Vehicle Controls and Intelligent Transportaton Systems, Int. Conf. on Informatics in Control, Automation and Robotics, vol. 2, pp. 802–809, September 2014.

[Gim 17] Suhyeon Gim, Lounis Adouane, Sukhan Lee & Jean-Pierre Derutin. *Clothoids composition method for smooth path generation of car-like vehicle navigation*. Journal of Intelligent and Robotic Systems (JIRS), Accepted., 2017.

[Glauz 80] William Glauz, Andrew StJohn & Douglas Harwood. *Projected vehicle characteristics*. Transportation Research Report, vol. 772, pp. 37–44, 1980.

[Goldman 15] David Goldman. *Uber to develop self-driving cars*. `http://money.cnn.com/2015/02/03/technology/innovationnation/uber-self-driving-cars/`, 2015. Accessed: 2017-3-5.

[Gonzalez 14] David Gonzalez, Joshue Perez, Ray Lattarulo, Vicente Milanes & Fawzi Nashashibi. *Continuous curvature planning with obstacle avoidance capabilities in urban scenar-*

ios. IEEE Int. Conf. on Intelligent Transportation Systems, vol. Oc. Qingdao, China, pp. 1430–1435, 2014.

[Goodman 16] Paul Goodman. *Advantages and Disadvantages of Driveless Cars*. `https://axleaddict.com/safety/Advantages-and-Disadvantages-of-Driverless-Cars`, 2016. Accessed: 2017-3-2.

[GP 16] Monaco F1 GP. *Monaco Grand Prix 2017*. `http://www.gpdeluxe.com/monaco-gp-circuit-layout/`, 2016. Accessed: 2016-10-20.

[Guo 14] Chunzhao Guo, Hun ichi Meguro, Yoshiko Kojima & Takashi Naito. *Automatic Lane-level map generation for advanced driver assistance systems using low-cost sensors*. IEEE International Conference on Control Applications, vol. May, Hong Kong, China, pp. 3975–3982, 2014.

[Hayashi 12] R. Hayashi, J. Isogai, P. Raksincharoensak & M. Nagai. *Autonomous collision avoidance system by combined control of steering and braking using geometrically optimised vehicular trajectory*. Vehicle System Dynamics, vol. 50, pp. 151–168, 2012.

[Hundelshausen 08] Felix V. Hundelshausen, Michael Himmelsbach, Falk Hecker, Andre Mueller & Hans-Joachim Wuensche. *Driving with tentacles: Integral structures for sensing and motion*. Journal of Field Robotics, vol. 25(9), pp. 640–673, 2008.

[IPDS 16] IPDS. *The Institut Pascal Data Sets*. `http://ipds.univ-bpclermont.fr/`, 2016. Accessed: 2016-10-20.

[Jo 14] Kichun Jo & Myoungho Sunwoo. *Generation of a precise roadway map for autonomous cars*. IEEE Trans on Intelligent Transportation systems, vol. 15(3), pp. 925–937, 2014.

[Kelly 03] Alonzo Kelly & Bryan Nagy. *Reactive Nonholonomic Trajectory Generation via Parametric Optimal Control*. The International Journal of Robotics Research, vol. 22, no. 7 - 8, pp. 583 – 601, July 2003.

[Khatib 86a] Oussama Khatib. *The Potential Field Approach and Operational Space Formulation in Robot Control*. pp. 367–377, 1986.

[Khatib 86b] Oussama Khatib. *Real-time obstacle avoidance for manipulators and mobile robots*. The int. J. of Robotics Research, vol. 5(1), pp. 90–98, 1986.

[Kim 16] HyungJun Kim. *FingerEyes*. `http://www.gisdeveloper.co.kr/tag/FingerEyes-Xr`, 2016. Accessed: 2016-1-25.

[Kimia 03] Benjamin B. Kimia, Ilana Frankel & Ana-Maria Popescu. *Euler spiral for shape completion*. Int. J. of Computer Vision, vol. 54, pp. 159–182, 2003.

[Kiusalaas 07] Jaan Kiusalaas. Numerical methods in engineering with matlab. Cambridge University Press UK., Shaftesbury Road, Cambridge CB2 8BS, UK., 2007.

[Labakhua 08] Larissa Labakhua, Urbano Nunes, Rui Rodrigues & Fatima S. Leite. *Smooth trajectory planning for fully automated passengers vehicles: Spline and Clothoid based methods and its simulation*. Andrade Cetto et al., editeur, Informatics in Control Automation and Robotics, vol. 15 of *Lecture Notes Electrical Engineering*, pp. 159–182, 2008. [doi:10.1007/978-3-540-79142-3-14].

[Lamiraux 01] Florent Lamiraux & J. P. Laumond. *Smooth motion planning for car-like vehicles*. IEEE Trans. on Robotics and Automation, vol. 17(4), pp. 498–501, 2001.

[Lan 15] Xiaodong Lan & Stefano DiCairano. *Continuous curvature path planning for semi-autonomous vehicle maneuvers using RRT*. European Controll Conference, vol. July, Linz, Austria, pp. 2360–2365, 2015.

[Lavalle 99] Steven M. Lavalle & Jr. James J. Kuffner. *Randomized kinodynamic planning*. IEEE Int. Conf. on Robotics and Automation, vol. 1, pp. 473–479, 1999.

[Lavalle 06] Steven M. Lavalle. Planning algorithms. Cambride University Press, Shartesbury Road, Cambridge, UK, 2006.

[Lee 01] Hyeongcheol Lee & Masayoshi Tomizuka. *Coordinated longitudinal and lateral motion control of vehicles for IVHS*. Journal of Dynamic Systems, Measurement, and Control, vol. 123, pp. 535–543, 2001.

[Likhachev 09] Maxim Likhachev & Dave Ferguson. *Planning long dynamically feasible maneuvers for autonomous vehicles*. Int. Journal of Robotics Research, vol. 28(8), pp. 933–945, 2009.

[Lin 14] Yucong Lin & Srikanth Saripalli. *Path planning using 3D Dubins curve for unmanned aerial vehicles*. Int. Conf. on Unmanned Aircraft Systems, vol. Orlando, USA, pp. 296–304, 2014.

[Lozenguez 11] Guillaume Lozenguez, Lounis Adouane, Aurelie Beynier, Philippe Martinet & Abdel Illah Mouaddib. *Map Partitioning to Approximate an Exploration Strategy in Mobile Robotics*. PAAMS 2011, 9th International Conference on Practical Applications of Agents and Multi-Agent Systems, Published after in Advances on Practical Applications of Agents and Multiagent Systems Advances in Intelligent and Soft Computing Volume 88, 2011, pp 63-72, Salamanca-Spain, 6-8th April 2011.

[Macadam 10] Charles C. Macadam. *Understanding and Modeling the Human Driver*. Vehicle System Dynamics, vol. 40(1-3), pp. 101–134, 2010.

[Macek 06] Kristijan Macek, Marcelo Becker & Roland Siegward. *Motion planning for car-like vehicles in dynamic urban scenarios*. Int. Conf. on Intelligent Robots and Systems, vol. Oct. China, pp. 4375– 4380, 2006.

[Macek 09] Kristijan Macek, Roland Philippsen & Roland Siegwart. *Path following for autonomous vehicle navigation based on kinodynamic control*. Journal of Computing and Information Technology, vol. 17, pp. 17–26, 2009.

[Marie 13] Romain Marie, Ouiddad Labbani-Igbida, Pauline Merveilleux & El Mustapha Mouaddib. *Autonomous robot exploration and cognitive map building in unknown environments using omnidirectional visual information*

only. IEEE Workshop on Robot Vision, pp. 191–196, Clearwater Beach-FL, USA, 15-17 Jan. 2013.

[Markkula 14] Gustav Markkula, O. Benderius & M. Wahde. Comparing and validating models of driver steering behaviour in collision avoidance and vehicle stabilisation, vol. Taylor and Francis). 2014.

[Montes 07] Nicolas Montes, Marta C. Mora & Josep Tornero. *Trajectory generation based on rational Bezier curves as clothoids*. IEEE Intel. Vehicles Symposium, pp. 505–510, 2007.

[Mouad 12] Mehdi Mouad, Lounis Adouane, Djamel Khadraoui & Philippe Martinet. *Mobile Robot Navigation and Obstacles Avoidance based on Planning and Re-Planning Algorithm*. International IFAC Symposium on Robot Control (SYROCO'12), 5-7, September 2012.

[Muddhor 16] A. S. Muddhor, N. Valantasis-Kanellos & Dr. Eoin Plant. *Self-Driving Vehicles in Road Freight Transport*. `http://linklinejournal.com/self-driving-vehicles-road-freight-transport/`, 2016. Accessed: 2016-12-21.

[Muoio 16] Daniel Muoio. *Uber's self-driving cars are impressive-but there's still a lot they can't do*. `http://www.businessinsider.com/uber-driverless-car-problems-2016-9/#a-big-problem-for-self-driving-cars-is-simply-having-to-deal-with-other-human-drivers-on-the-road-1`, 2016. Accessed: 2016-12-23.

[Nelson 89] Wayne Nelson. *Continuous-curvature paths for autonomous vehicles*. IEEE Int. Conf. on Robotics and Automation, vol. 3, pp. 1260–1264, 1989.

[Nister 06] David Nister, Oleg Naroditsky & James Bergen. *Visual odometry for ground vehicle applications*. Journal of Field Robotics, vol. 23(1), pp. 3–20, 2006.

[Nunez 07] Pedro Nunez, Ricardo Vazquez & Jose C. del Toro et al. *A Curvature based Method to Extract Natural Landmarks*

for Mobile Robot Navigation. IEEE Int. Symposium on Intelligent Signal Processing, vol. Oct., pp. 432–437, 2007.

[Pacejka 92] Hans Pacejka. Tire and vehicle dynamics (3rd ed.). Elsevier Ltd., 225 Wyman Steet, Waltham, MA 02451, USA, 1992.

[Parlangeli 10] Gianfranco Parlangeli & Giovanni Indiveri. *Dubins inspired 2D smooth paths with bounded curvature and curvature derivative*. Proc. of the 7th IFAC Symposium on Intelligent Autonomous Vehicles, pp. 252–257, 2010.

[Petrov 13] Plamen Petrov & Fawzi Nashashibi. *Adaptive steering control for autonomous lane change maneuver*. IEEE Intelligent Vehicles Symposium, vol. Australia, pp. 835–840, 2013.

[Poussot-Vassal 11] C. Poussot-Vassal, O. Sename, L. Dugard & S. M. Savaresi. *Vehicle dynamics stability improvements through gain-scheduled steering and braking control*. Vehicle System Dynamics, vol. 49(10), pp. 1597–1621, 2011.

[Premebida 07] Christiano Premebida, Goncalo Monteiro, Urbano Nunes & Paulo Peixoto. *A Lidar and vision-based approach for pedestrian and vehicle detection and tracking*. IEEE Intel. Transportation Systems Conf., pp. 1044–1049, 2007.

[Rajamani 06] Rajesh Rajamani. Vehicle dynamics and control. Springer, Spring Street, New York , NY 10013, USA, 2006.

[Reeds 90] James A. Reeds & Lawrence Alan Shepp. *Optimal paths for a car that goes both forwards and backwards*. Pacific Journal of Mathematics, vol. 145(2), pp. 367–393, 1990.

[Rill 11] Georg Rill. Vehicle dynamics: Fundamentals and modeling. CRC Press, Taylor and Francis Group, Mortimer House 37-41, Mortimer Street,London, W1T3JH, UK, 2011.

[Rothhamel 14] Malte Rothhamel, Jolle IJkema & Lars Drugge. *Influencing driver chosen cornering speed by means of modified steering feel*. Vehicle System Dynamics, vol. 52(4), pp. 522–538, 2014.

[Samson 95] Claude Samson. *Control of chained systems: application to path following and time-varying point stabilization of mobile robots.* IEEE Trans. on Automatic Control, vol. 40(1), pp. 64–77, 1995.

[Scheuren 12] Stephan Scheuren, Stefan Stiene, Ronny Hartanto & Joachim Hertzberg. Approximating reference trajectories for autonomous vehicles using motion primitives. Springer-Verlag Berlin Heidelberg, 2012.

[Seo 14] Dongwook Seo & Kanghyun Jo. *Inverse Perspective Mapping based Road Curvature Estimation.* IEEE/SICE Int. Symposium on System Integration, vol. December, Tokyo, Japan, pp. 480–483, 2014.

[Sharp 00] R.S. Sharp, D.Casanova & P.Symonds. *A Mathematical model for driver steering control, with design, tuning and performance results.* Vehicle System Dynamics, vol. 33, pp. 289–326, 2000.

[Siciliano 16] Bruno Siciliano & Oussama Khatib. Springer handbook of robotics. Springer, Berlin, Heidelberg, 2016.

[Siegwart 11] Roland Siegwart & Illah R. Nourbakhsh. Introduction to autonomous mobile robots, vol. Cambridge, Massachusetts, London, England. 2011.

[Smith 08] Nicholas D. Smith. *Understanding parameters influencing tire modeling.* Colorado State University, 2004 Formula SAE Platform, 2008.

[Soetanto 98] Danny Soetanto, Lionel Lapierre & Antonio Pascoal. *Adaptive, non-singular path-following control of dynamic wheeled robots.* Proc. of IEEE Conf. on Decision and Control, pp. 1765–1770, 1998.

[Solea 06] Razvan Solea & Urbano Nunes. *Trajectory planning with velocity planner for fully-automated passenger vehicle.* Intel. Transportation Systems Conf., pp. 474–480, Sep. 2006.

[Sprunk 08] Christoph Sprunk. Planning motion trajectories for mobile robots using splines. Albert-Ludwigs-Universitat Freiburg, 2008.

[Tesis-DYNAware 16] Tesis-DYNAware. *veDYNA: Real-Time Simulation of Vehicle Dynamics*. `http://www.tesis-dynaware.com/`, 2016. Accessed: 2016-10-20.

[Thrun 06a] Sebastian Thrun, Wolfram Burgard & Dieter Fox. Probabilistic robotics. MIT Press, 55 Hayward Street Cambridge, USA, 2006.

[Thrun 06b] Sebastian Thrun, Mike Montemerlo & et. al. *Stanley: The robot that won the DARPA Grand Challenge*. Journal of Field Robotics, vol. 23(9), pp. 661–692, 2006.

[Trepagnier 06] Paul G. Trepagnier, Jorge Nagel & et. al. *KAT-5: Robust systems for autonomous vehicle navigation in challenging and unknown terrain*. Journal of Field Robotics, vol. 25(9), pp. 640–673, 2006.

[Truax 08] Robert D. Truax. *Characterization of Side-slip dynamics in Land Rover LR3 for improved high speed autonomous control*. Bachelor Thesis of Massachusetts Institute of Technology, 2008.

[Ulsoy 12] A. Galip Ulsoy, Huei Peng & Melih Cakmakci. Automotive control systems. Cambridge University Press, 32 Avenue of the Americas, New York, NY 10013-2473, USA, 2012.

[Urmson 06] Chris Urmson, Charlie Ragusa & et. al. *A robust approach to high-speed navigation for unrehearsed desert terrain*. Journal of Field Robotics, vol. 23(8), pp. 467–508, 2006.

[Vilca 13] Jose-Miguel Vilca, Lounis Adouane & Youcef Mezouar. *Reactive navigation of a mobile robot using elliptic trajectories and effective online obstacle detection*. Gyroscopy and Navigation, vol. 4(1), pp. 14–25, 2013.

[Vilca 15] Jose-Miguel Vilca, Lounis Adouane & Youcef Mezouar. *Optimal Multi-Criteria Waypoint Selection for Autonomous Vehicle Navigation in Structured Environment*. Journal of Intelligent and Robotic Systems (JIRS), Accepted and Published online: 15th April 2015, vol. 82, no. 2, pp. 301–324, 2015.

[Villagra 12] Jorge Villagra, Vicente Milantes, Joshue Perez & Jorge Godoy. *Smooth path and speed planning for an automated public transport vehicle.* Robotics and Autonomous Systems, vol. 60, pp. 252–265, 2012. [doi:10.1016/j.robot.2011.11.001].

[Walton 05] D. J. Walton & D. S. Meek. *A controlled clothoid spline.* Computers and Graphics, vol. 29, pp. 353–363, 2005.

[Walton 09] D. J. Walton & D. S. Meek. *G^1 interpolation with a single Cornu spiral segment.* J. of Computational and Applied Mathematics, vol. 220, pp. 86–96, 2009.

[Wang 98] Yongji Wang & M. P. Cartmell. *Trajectory generation for a four wheel steering tractor-trailer system: a two-step method.* Robotica, vol. 16, pp. 381–386, 1998.

[Wang 04] Jun Wang, Karen Dixon, Hainan Li & Jennifer Ogle. *Normal acceleration behavior of passenger vehicles starting from rest at all-way stop-controlled intersections.* Transportation Research Record 1883, vol. Melbourne, Australia, pp. 158–166, 2004.

[Webster 06] Robert J. Webster, JinSeob Kim, Noah J. Cowan, Gregory S. Chirikjian & Allison M. Okamura. *Nonholonomic Modeling of Needle Steering.* The International Journal of Robotics Research, vol. 25(5-6), pp. 509–525, 2006.

[Wikipedia 17] Wikipedia. *Manifold.* `https://en.wikipedia.org/wiki/Manifold/`, 2017. Accessed: 2017-3-20.

[Wilde 09] Doran K. Wilde. *Computing clothoid segments for trajectory generation.* IEEE/RSJ Int. Conf. on Intel. Robots and Systems, pp. 2440–2445, Oct. 2009.

[Wilkie 09] David Wilkie, Jur van den Berg & Dinesh Manocha. *Generalized velocity obstacles.* IEEE Int. Conf. on Intelligent Robots and Systems, pp. 5573–5578, 2009.

[Wu 11] Paul P. Y. Wu, Duncan Campbell & Torsten Merz. *Multi-objective Four-Dimenstional vehicle motion planning in large dynamic environments.* IEEE Trans. on Systems, Man, and Cybernetics-Part B:Cyberntetics, vol. 41(3), pp. 621–634, 2011.

[Xu 12] Wenda Xu, Junqing Wei, John M. Dolan, Huijing Zhao & Hongbin Zha. *A Real-time motion planner with trajectory optimization for autonomous vehicles*. IEEE Int. Conf. on Robotics and Automation, pp. 2061–2067, 2012.

[Yamakado 08] Makoto Yamakado & Masato Abe. *An experimentally confirmed driver longitudinal acceleration control model combined with vehicle lateral motion*. Vehicle System Dynamics, vol. 46(supplement), pp. 129–149, 2008.

[Yang 13] Kwangjin Yang, Daehan Jung & Salah Sukkarieh. *Continuous curvature path-smoothing algorithm using cubic Bezier spiral curves for non-holonomic robots*. Advanced Robotics, vol. 27(4), pp. 247–258, 2013.

[Yuhara 01] Naohiro Yuhara & Jun Tajima. *Advanced steering system adaptable to lateral control task and driver's intention*. Vehicle System Dynamics, vol. 36(2-3), pp. 119–158, 2001.

[Zhu 14] Gang Zhu, Ming Yang, Bing Wang & Chunxiang Wang. *A New Approach for Autonomous Vehicle Navigation in Urban Scenarios based on Roadway Magnets*. IEEE Intelligent Vehicles Symposium (IV), vol. June, Dearborn, Michigan, USA, pp. 432–437, 2014.

[Ziegler 08] Julius Ziegler, Moritz Werling & J. Schroeder. *Navigating car-like robots in unstructured environments using and obstacle sensitive cost function*. Proc. of IEEE Intelligent Vehicles Symposium (IV), pp. 787–791, 2008.

[Zucker 10] Matt Zucker, J. Andrew Bagnell, Christopher G. Arkeson & James Kuffner. *An optimization approach to rough terrain locomotion*. IEEE Int. Conf. on Robotics and Automation, pp. 3589–3595, 2010.

Résumé

La génération de chemins lisses pour les voitures intelligentes est l'une des conditions les plus importantes pour faire accepter et faciliter la navigation autonome de ces véhicules. Cette th èse propose plusieurs méthodes de génération de chemins lisses pour les véhicules non-holonomes qui permet une continuité intrinsèque de la courbure de navigation et offre par ailleurs une flexibilité accrue pour diverses conditions aux limites. Le chemin de courbure continue est construit en composant plusieurs clothoids, comprenant notamment des segments de lignes et/ou d'arcs, et où chaque clothoid est obtenue par une régulation appropriée de ses paramètres. À partir de ces propriétés, le chemin obtenu est nommé **pCCP** (parametric Continuous Curvature Path). Le **pCCP** fournit un diagramme de courbure qui facilite une commande en orientation du véhicule, ce qui permet dâ€™obtenir une évolution lisse de sa trajectoire. Le problème du **pCCP** local est défini par des configurations initiales et finales (caractérisées pour chacune par une posture et un angle de braquage). Le problème a été étendu pour être aussi général que possible en incluant plusieurs cas. La génération locale de **pCCP**s, pour des cibles statiques, est spécifiquement décrite, les problèmes ont été divisés en trois problèmes et chaque problème a été décomposé par la suite en plusieurs sous-classes possibles. Pour avoir une flexibilité importante des **pCCP**s proposés, des cibles dynamiques ont été considérées, obtenant ainsi le dynamic-**pCCP** (**d-pCCP**). Un cadre simple mais efficace pour analyser lâ€™état futur de lâ€™évitement des obstacles est appliqué en configuration 4D (3D avec l'ajout d'un axe temporel) en mettant en exergue deux manÅ“uvres d'évitement possibles, car les évolutions avant et arrière sont appliquées et validées avec plusieurs exemples. Selon une méthodologie similaire pour atteindre les critères de performance liés à la génération des **pCCP**s, le **h-CCP** (pour human-**pCCP**) est proposé en utilisant des modèles expérimentaux comportementaux d'échantillons de conducteurs humains. À partir de quelques sous-expériences, le modèle de conduite humain pour l'évitement d'obstacles, les changements de voie et les mouvements en virage sont extraits et ces modèles ont été inclus pour créer ainsi le **h-CCP** (obtenu d'une manière similaire au **pCCP** mais avec différents critères d'optimisation) qui permet d'améliorer considérablement le confort des passagers.

Mots-clés : *Chemin de courbure continue, Véhicule non-holonome et autonome, Évitement d'obstacles, Espace de configuration 4D, Modèle de conduite humaine.*

Abstract

Smooth path generation for car-like vehicles is one of the most important requisite to facilitate the broadcast use of autonomous navigation. This thesis proposes a smooth path generation method for nonholonomic vehicles which has inherently continuity of curvature and having important flexibility for various boundary conditions. The continuous curvature path is constructed by composing multiple clothoids including lines and/or arc segments, and where each clothoid is obtained by parameter regulation. From those properties the path is named **pCCP** (parametric Continuous Curvature Path) and provides curvature diagram which facilitates a smooth steering control for path following problem. Local **pCCP** problem is defined by initial and final tuple configurations (vehicleâ€™s posture and steering angle). The problem is expanded to be as general as possible by including several cases. The local **pCCP** generation for steady target pose is specifically described, where the problem is divided into three problems and each problem is also decomposed into several sub-cases. To give more flexibility to the proposed **pCCP**, dynamic target is considered to obtain dynamic-**pCCP** (**d-CCP**). A simple but efficient framework to analyze the future status of obstacle avoidance is applied in 4D (3D with the addition of time axis) configuration and two avoidance maneuvers as front and rear avoidance are applied and validated with several examples. Under the similar methodology in performance criteria of **pCCP** generation, the human-**CCP** (**h-CCP**) is derived from experimental patterns of human driver samples. From several sub-experiments, human driving pattern for obstacle avoidance, lane change and cornering motion are extracted and those pattern were included to make the **h-CCP** (which is obtained with similar way as **pCCP** but with different optimization criteria) to enhance considerably the passenger comfort.

Keywords: *Continuous Curvature Path, Nonholonomic and autonomous car-like vehicle, Obstacle avoidance, 4D configuration space, Human driving pattern.*