

AN EFFICIENT ROAD REPRESENTATION FOR AUTONOMOUS VEHICLES
USING ARC-SPLINES WITH APPLICATION TO TRAJECTORY PLANNING

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ATAKAN SALIH BOLAT

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

JULY 2024

Approval of the thesis:

**AN EFFICIENT ROAD REPRESENTATION FOR AUTONOMOUS
VEHICLES USING ARC-SPLINES WITH APPLICATION TO
TRAJECTORY PLANNING**

submitted by **ATAKAN SALIH BOLAT** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Naci Emre Altun _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. İlkay Ulusoy _____
Head of Department, **Electrical and Electronics Engineering**

Prof. Dr. Klaus Werner Schmidt _____
Supervisor, **Electrical and Electronics Engineering, METU**

Examining Committee Members:

Prof. Dr. Afşar Saranlı _____
Electrical and Electronics Engineering, METU

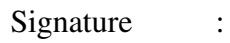
Prof. Dr. Klaus Werner Schmidt _____
Electrical and Electronics Engineering, METU

Assist. Prof. Dr. Ulaş Beldek _____
Mechatronics Engineering, Çankaya University

Date: 11.07.2024

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Atakan Salih Bolat

Signature : 

ABSTRACT

AN EFFICIENT ROAD REPRESENTATION FOR AUTONOMOUS VEHICLES USING ARC-SPLINES WITH APPLICATION TO TRAJECTORY PLANNING

Bolat, Atakan Salih

M.S., Department of Electrical and Electronics Engineering

Supervisor: Prof. Dr. Klaus Werner Schmidt

July 2024, 117 pages

Autonomous driving technology is the basis of future transportation, particularly on highways, where conditions are optimal due to steady traffic and minimal interruptions. Implementing autonomous driving on highways enhances safety, traffic flow, and environmental impact while benefiting long-distance travel and logistics.

Trajectory planning for autonomous vehicles (AVs) on highways is commonly based on high-definition (HD) maps, which aim to provide a highly accurate road representation with low memory requirements. Highways are typically designed with clothoid curves, which are characterized by a linear change in road curvature and can be followed by vehicles at high speeds. Since clothoids lack an analytical representation, this study develops an algorithm combining arc-splines and straight-line segments to approximate clothoid curves for a novel memory-efficient HD map representation. Using real road data from OpenStreetMap and HERE maps, we demonstrate that an analytical road representation using arc-splines and straight segments achieves approximately 3 centimeters of accuracy. A further benefit of our method is the computation of all lanes on a highway by parallel shifting a reference lane.

To showcase the usability of our road representation, we perform trajectory planning for AVs using Bézier curves and arc-splines under typical highway conditions. Our findings highlight that arc-spline trajectories are superior to Bézier curves in terms of controllability and computational efficiency. Overall, this research demonstrates that accurate road information, including geometric properties like position, heading, and curvature, can be achieved with low memory requirements and computational effort.

Keywords: Path planning, Autonomous vehicles, Highway modeling, Bézier Curves, arc-splines

ÖZ

OTONOM ARAÇLAR İÇİN VERİMLİ BİR OTOYOL MODELLEMESİ VE ROTA PLANLAMADA YAY-EĞRİLERİ KULLANIMI

Bolat, Atakan Salih

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Klaus Werner Schmidt

Temmuz 2024 , 117 sayfa

Otonom sürüş teknolojisi, özellikle trafiğin sabit ve kesintilerin az olduğu otoyollarda, geleceğin ulaşımının temelini oluşturmaktadır. Otoyollarda otonom sürüşün uygulanması, güvenliği, trafik akışını ve çevresel etkileri iyileştirirken, uzun mesafeli seyahat ve lojistik için de faydalı sağlar.

Otoyollarda otonom araçlar (AV'ler) için güzergah planlaması genellikle düşük bellek gereksinimleriyle son derece doğru bir yol modeli sağlamayı amaçlayan yüksek çözünürlüklü (HD) haritalara dayanır. Otoyollar genellikle yol eğriliğinde doğrusal bir değişiklikle karakterize edilen ve araçlar tarafından yüksek hızlarda takip edilebilen klotoid eğrilerle tasarılmıştır. Klotoidlerin analitik bir modeli olmadığından, bu çalışma, klotoid eğrilerini yenilikçi bir bellek verimli HD harita modeli için yaklaşık olarak tanımlamak amacıyla arc-splines ve düz çizgi segmentlerini birleştiren bir algoritma geliştirmektedir. OpenStreetMap ve HERE haritalarından alınan gerçek yol verilerini kullanarak, arc-splines ve düz çizgiler kullanarak yapılan analitik yol modeli ile yaklaşık 3 santimetre doğruluk elde edildiğini gösteriyoruz. Yöntemimizin bir diğer avantajı, referans bir şeridin paralel olarak kaydırılmasıyla otoyoldaki tüm

şeritlerin hesaplanabilmesidir.

Yol modeli kullanımını göstermek için, tipik otoyol koşulları altında Bézier eğrileri ve yay-eğrileri kullanarak AV'ler için güzergah planlaması yapıyoruz. Bulgularımız, yay-eğrisi güzergahlarının kontrol edilebilirlik ve hesaplama verimliliği açısından Bézier eğrilerinden üstün olduğunu vurgulamaktadır. Genel olarak, bu araştırma, konum, yön ve eğrilik gibi geometrik özellikler de dahil olmak üzere doğru yol bilgilerinin düşük bellek gereksinimleri ve hesaplama çabası ile elde edilebileceğini göstermektedir.

Anahtar Kelimeler: Yol planlama, Otonom araçlar, Otoyol modelleme, Bézier eğrileri, Yay eğrileri

To my beloved ones...

ACKNOWLEDGMENTS

I would like to thank my supervisor Prof. Dr. Klaus Werner Schmidt for his patience, support and assistance during this period.

I sincerely thank to my family and my friends for their support during the thesis work.

TABLE OF CONTENTS

ABSTRACT	v
ÖZ	vii
ACKNOWLEDGMENTS	x
TABLE OF CONTENTS	xi
LIST OF TABLES	xiv
LIST OF FIGURES	xv
CHAPTERS	
1 INTRODUCTION	1
2 BACKGROUND INFORMATION	9
2.1 Basic Concepts	9
2.1.1 Road Modelling	9
2.1.1.1 Clothoids	9
2.1.1.2 Clothoid Approximation with Arc Splines	13
2.1.2 Waypoint Data	16
2.1.2.1 OpenStreetMap Database	17
2.1.2.2 HERE Maps Database	18
2.1.3 Bézier Curve Based Trajectory Planning	18
2.1.4 Bi-elementary Paths	20

2.2	Contributions and Novelties	24
	Arc Spline-Based Road Modeling:	24
	Trajectory Planning Algorithm:	24
3	ROAD SEGMENT REPRESENTATION METHODOLOGY	25
3.1	Ground Truth Generation	27
3.2	Approximation	28
3.3	Combination	31
3.4	Generation of Additional Lanes from a Single Lane	37
3.5	Evaluation Metrics and Results	40
3.5.1	RMSE Spatial Coordinates	42
3.5.2	Maximum Error In Spatial Coordinates	49
3.5.3	Number of Segments	54
4	TRAJECTORY GENERATION METHODOLOGY	57
4.1	Bézier Trajectory Generation	57
4.1.1	Generating a Sufficient Number of Candidate Curves	58
4.1.2	Identifying the Best Trajectory	63
4.2	Arc-spline Trajectory Generation	63
4.2.1	Heading and Curvature Correction (HCC) Maneuver	64
4.2.1.1	Case 1 Solution	67
4.2.1.2	Case 2 Solution	69
4.2.2	Position Correction	73
4.2.3	Road Curvature Change Correction	73
4.2.4	Superposition of Maneuvers	73

4.3	Evaluation Metrics and Results	76
4.3.1	RMS and Maximum Error For Many Cases	77
4.3.2	Arc-spline Trajectory	82
4.3.3	Comparison with Bézier Curves	92
4.3.4	Trajectory Simulation	101
5	CONCLUSION	111
	REFERENCES	113

LIST OF TABLES

TABLES

Table 3.1 Road Segment Data	29
Table 3.2 Error Table	31
Table 3.3 Segments before and after combination operation for Autobahn 38 . .	54
Table 3.4 Segments before and after combination operation for Autobahn 1 . .	55
Table 4.1 Configuration Parameters for Candidate Bézier Curve Generation . .	60
Table 4.2 Different heading and curvature error configurations with the RMS and Maximum errors for +0.80 meter position error	78
Table 4.3 Different heading and curvature error configurations with the RMS and Maximum errors for +0.40 meter position error	79
Table 4.4 Different heading and curvature error configurations with the RMS and Maximum errors for -0.40 meter position error	80
Table 4.5 Different heading and curvature error configurations with the RMS and Maximum errors for -0.80 meter position error	81
Table 4.6 Error configurations for different cases	82
Table 4.7 Computation time for Bézier and arc-spline trajectories	100

LIST OF FIGURES

FIGURES

Figure 2.1	Curvature plot of a clothoid curve.	10
Figure 2.2	Plot of $\mathcal{C} = [0, 0, 0, 0, 0.01, 100]$ clothoid.	11
Figure 2.3	Plot of $\mathcal{C} = [0, 0, 0, 0.01, 0, 100]$ clothoid.	12
Figure 2.4	Plot of both $\mathcal{C} = [0, 0, 0, 0.01, 0, 100]$ and $\mathcal{C} = [0, 0, 0, 0, 0.01, 100]$.	12
Figure 2.5	Curvature plot of a clothoid curve and curvatures of arcs.	15
Figure 2.6	Approximation of $\mathcal{C} = [0, 0, 0, 0.1, 0.2, 10]$ with arc-spline.	15
Figure 2.7	Error for a 154 meter long road segment with an order 5 approximation.	16
Figure 2.8	A closeup view of an approximation and real road data.	17
Figure 2.9	OpenStreetMap data example of Autobahn 38 in Germany.	18
Figure 2.10	HERE Map example of Autobahn 4 in Germany.	19
Figure 2.11	Bézier curve example for $B = [(0, 0), (15, 10), 0, 45^\circ, 0, 0.01]$	21
Figure 2.12	Curvature and ΔY change of a Bi-elementary curve for $\lambda = 0.50, \gamma = 0.50$.	22
Figure 2.13	Curvature and ΔY change of a Bi-elementary curve for $\lambda = 0.50, \gamma = 0.25$.	22

Figure 2.14 Curvature and ΔY change of a Bi-elementary curve for $\lambda = 0.75, \gamma = 0.50.$	23
Figure 3.1 G1 clothoid fitting applied on 5 waypoints.	28
Figure 3.2 Error comparison of low and high order approximations for the clothoid $\mathcal{C} = [0, 0, 0, 0, 0.1, 50].$	30
Figure 3.3 Euclidian error between approximated reference lane and ground truth	31
Figure 3.4 Parallel shifting of a single basic arc.	38
Figure 3.5 Parallel shifting error of an arc-spline representing a real road segment.	38
Figure 3.6 Parallel shifting of a single basic arc.	39
Figure 3.7 Parallel shifting error of a line segment representing a real road segment.	40
Figure 3.8 Satellite image overlaid with G1 fitted clothoid road segments (plotted in light blue) on Autobahn 38.	41
Figure 3.9 Satellite image overlaid with G1 fitted clothoid road segments (plotted in dark blue) on Autobahn 1.	41
Figure 3.10 RMSE of a line approximated segment	42
Figure 3.11 Maximum error of a line approximated segment.	43
Figure 3.12 RMSE of a arc-spline approximated segment.	43
Figure 3.13 Maximum error of a arc-spline approximated segment.	44
Figure 3.14 RMSE for a 323 segment Autobahn 38 (overall length around 12.5 km).	45
Figure 3.15 RMSE and average curvature for 323 segment Autobahn 38 . . .	46

Figure 3.16	RMSE and segment length for 323 segment Autobahn 38	46
Figure 3.17	RMSE for 278 segment Autobahn 1 (overall length around 8 km)	47
Figure 3.18	RMSE and average curvature for 278 segment Autobahn 1	48
Figure 3.19	RMSE and segment length for 278 segment Autobahn 1	48
Figure 3.20	General view of the approximation compared to ground truth	49
Figure 3.21	Maximum error of a line approximated segment	50
Figure 3.22	Maximum error of a arc-spline approximated segment	50
Figure 3.23	Maximum error values for 323 segment Autobahn 38	51
Figure 3.24	Maximum error and average curvature for 323 segment Autobahn 38	52
Figure 3.25	Maximum error and segment length for 323 segment Autobahn 38	52
Figure 3.26	Maximum error values for 278 segment Autobahn 1	53
Figure 3.27	Maximum error and average curvature 278 segment Autobahn 1	53
Figure 3.28	Maximum error and segment length for 278 segment Autobahn 1	54
Figure 4.1	Effect of varying P1 while keeping the other parameters same.	60
Figure 4.2	Effect of varying P2 while keeping the other parameters same.	61
Figure 4.3	Effect of varying P3 while keeping the other parameters same	62
Figure 4.4	Effect of varying P4 while keeping the other parameters same	62
Figure 4.5	An example of position, heading, and curvature error.	64
Figure 4.6	Curvature plot of a road segment with a heading and curvature correcting maneuver curvature plot for case 1.	65
Figure 4.7	Curvature plot of a road segment with a heading and curvature correcting maneuver curvature plot for case 2	66

Figure 4.8	Curvature error correction with a clothoid	67
Figure 4.9	Curvature and heading error correction with a clothoid.	68
Figure 4.10	Effect of an example HCC maneuver on the position error.	69
Figure 4.11	Curvature error correction with two clothoids.	70
Figure 4.12	Curvature and heading error correction with two clothoids	71
Figure 4.13	Curvature changes over <i>HCClength</i> as an example. This data is simulated for visualization.	76
Figure 4.14	Position error over an arc-spline trajectory for case 1	83
Figure 4.15	Heading error over an arc-spline trajectory for case 1	83
Figure 4.16	Curvature error over an arc-spline trajectory for case 1	84
Figure 4.17	Arc-spline trajectory on Autobahn 38 case 1	84
Figure 4.18	Position error over an arc-spline trajectory for case 2	85
Figure 4.19	Heading error over an arc-spline trajectory for case 2	86
Figure 4.20	Curvature error over an arc-spline trajectory for case 2	86
Figure 4.21	Arc-spline trajectory on Autobahn 38 case 2	87
Figure 4.22	Position error over an arc-spline trajectory for case 3	88
Figure 4.23	Heading error over an arc-spline trajectory for case 3	88
Figure 4.24	Curvature error over an arc-spline trajectory for case 3	89
Figure 4.25	Arc-spline trajectory on Autobahn 38 case 3	89
Figure 4.26	Curvature error over an arc-spline trajectory for case 2 with $\sigma = 0.0010m^{-2}$	90
Figure 4.27	Arc-spline trajectory for case 2 with $\sigma = 0.0010m^{-2}$	91

Figure 4.28 Curvature error over an arc-spline trajectory for case 2 with $\sigma = 0.00001m^{-2}$	91
Figure 4.29 Arc-spline trajectory for case 2 with $\sigma = 0.00001m^{-2}$	92
Figure 4.30 Position error comparison for a set of Bézier curves and arc-spline trajectory for case 1	93
Figure 4.31 Heading error comparison for a set of Bézier curves and arc-spline trajectory for case 1	93
Figure 4.32 Curvature error comparison for a set of Bézier curves and arc-spline trajectory for case 1	94
Figure 4.33 Comparison of Bézier trajectories and arc-splines with lane boundaries on the road for case 1	94
Figure 4.34 Position error comparison for a set of Bézier curves and arc-spline trajectory for case 2	95
Figure 4.35 Heading error comparison for a set of Bézier curves and arc-spline trajectory for case 2	96
Figure 4.36 Curvature error comparison for a set of Bézier curves and arc-spline trajectory for case 2	96
Figure 4.37 Comparison of Bézier trajectories and arc-splines with lane boundaries on the road for case 2	97
Figure 4.38 Position error comparison for a set of Bézier curves and arc-spline trajectory for case 3	97
Figure 4.39 Heading error comparison for a set of Bézier curves and arc-spline trajectory for case 3	98
Figure 4.40 Curvature error comparison for a set of Bézier curves and arc-spline trajectory for case 3	98

Figure 4.41 Comparison of Bézier trajectories and arc-splines with lane boundaries on the road for case 3	99
Figure 4.42 Computation time of Bézier and Arc-spline trajectories	100
Figure 4.43 Trajectory simulation block diagram	101
Figure 4.44 Reference trajectory and the actual trajectory that is followed by vehicle for case 1	102
Figure 4.45 Position error of followed trajectory for case 1	103
Figure 4.46 Curvature plot of trajectory and maximum velocity allowed over trajectory for case 1	103
Figure 4.47 Acceleration limit and the actual acceleration of the vehicle over trajectory for case 1	104
Figure 4.48 Reference trajectory and the actual trajectory that is followed by vehicle for case 2	105
Figure 4.49 Position error of followed trajectory for case 2	105
Figure 4.50 Curvature plot of trajectory and maximum velocity allowed over trajectory for case 2	106
Figure 4.51 Acceleration limit and the actual acceleration of the vehicle over trajectory for case 2	106
Figure 4.52 Reference trajectory and the actual trajectory that is followed by vehicle for case 3	107
Figure 4.53 Position error of followed trajectory for case 3	108
Figure 4.54 Curvature plot of trajectory and maximum velocity allowed over trajectory for case 3	108
Figure 4.55 Acceleration limit and the actual acceleration of the vehicle over trajectory for case 3	109

CHAPTER 1

INTRODUCTION

The field of autonomous driving has been studied for many years [1] because of its potential to change transportation and make our lives easier. Autonomous vehicles are an important part of this research, offering benefits like improved road safety and efficiency [2]. A key part of developing these vehicles is how we define and model roads. The way roads are represented affects the performance of trajectory planning algorithms, which are important for the safe and effective operation of autonomous cars [3, 4, 5, 6]. This thesis study looks at different road modeling techniques and how various trajectory planning algorithms can be used with these models to help advance autonomous driving technology.

Autonomous driving presents numerous challenges and opportunities, particularly in the area of mapping and trajectory generation [6, 7]. Accurate and up-to-date maps are fundamental for the vehicle to understand its environment and navigate safely. Trajectory generation, which involves planning the path the vehicle will take, is important for ensuring that the vehicle smoothly follows its lane and avoids obstacles. These processes must be highly precise and reliable to maintain safety and efficiency. Advanced algorithms and robust data sources play a significant role in overcoming these challenges and harnessing the full potential of autonomous driving technologies.

The existing literature provides a multitude of methods for map generation and high-definition map representation. In particular, HD maps are essential for autonomous vehicles, providing highly accurate and detailed road information. [8] provides an overview of HD maps, emphasizing their precision and reliability, which range from centimeters to millimeters. This high level of detail is fundamental for safe and effi-

cient autonomous driving, enabling vehicles to navigate complex environments more accurately. In [9], the authors propose a highly detailed map representation for autonomous vehicle navigation, focusing on urban areas. This map includes three basic traffic elements: roads, lanes, and lane markings. Using ArcGIS and LIDAR high-precision maps, the researchers generate a comprehensive map database that captures both geometrical and topological information. The map facilitates a two-class path planning process, including road class and lane class planning, enabling intelligent vehicles to navigate intersections safely and efficiently. The work in [3] proposes a map generation algorithm designed to create precise roadway maps for autonomous cars. The algorithm involves three steps: data acquisition, data processing, and road modeling. Data from GPS and onboard sensors is refined using a fixed-interval optimal smoothing theory to improve accuracy. The refined road geometry data is then represented using a B-spline model. This method is verified through experimental studies, demonstrating its accuracy and reliability in various road conditions. Similarly, [10] employs cubic B-spline curves in a hierarchical motion planning framework that ensures safe and comfortable navigation in complex urban environments by addressing both static and dynamic obstacles. [11] presents a methodology for identifying road alignments (curves, straights, and clothoids) and their corresponding curvature values based on UTM coordinates obtained from field data. The procedure reconstructs road geometry using a cubic spline that identifies geographically referenced singular points, providing curvature values for each road element. The authors of [12] evaluate a mapping strategy that uses smooth arc-splines to represent road segments. The Smooth arc-splines Approximation Method (SMAP) generates splines with minimal curve segments while maintaining high accuracy. The method is assessed for its performance regarding accuracy, data volume, and curvature characteristics across different road types, including rural and highway roads. The results show that arc-spline approximation generally outperforms polygonal representations, particularly in terms of computational complexity and data storage requirements. [13] describes a novel approach for extracting the centerline geometry of road and railway alignments using traditional design elements such as straight lines, circle arcs, and clothoids. The method relies on data from a ground-based mobile mapping system and involves manipulating the bearing diagram and its derivatives. The approach includes dynamically tuned filters that adhere to the fractal properties of the centerline

location data, ensuring highly accurate and consistent results. A comprehensive review of high-definition (HD) map representation techniques crucial for autonomous vehicle navigation is provided in [4]. HD maps offer high precision, capturing detailed spatial information for automated driving. The paper covers various methodologies and technologies used to create HD maps, emphasizing accuracy, completeness, and extensibility. The authors also discuss the challenges of implementing HD mapping in urban environments, where complex road structures and the need for real-time updates pose significant difficulties.

Several studies focus on open-source map data. [14] presents an algorithm for reconstructing detailed carriageway maps from road centrelines and open-access areal representations such as polygons. This methodology, applied in different countries with varied urban environments, aims to transform basic road centerline data from sources like OpenStreetMap into more detailed carriageway representations. The authors validate their approach through a delivery routing problem, demonstrating the algorithm's effectiveness in generating accurate and practical carriageway maps for global applications. Road modeling for autonomous vehicles in urban city environments with OpenStreetMap data is discussed in [15]. However, this work does not implement the algorithm in highway environments. Highway characteristics are much different than city road networks. Moreover, OpenStreetMap data is not as reliable as HERE Maps data as it will be discussed in this research. Another study that explores road modeling is [16]. This paper focuses on generating lane boundaries from OpenStreetMap data. The paper does not focus on modeling the centerline. The reference for trajectory planning algorithms is the road centerline, which should be modeled. Similar to [15], OpenStreetMap data is not as accurate as HERE Maps data. In [17], the authors discuss using OpenStreetMap (OSM) geodata for autonomous robot navigation. OSM provides a wealth of information, including street names, types, and widths, as well as public speed limits. However, the paper highlights significant challenges, such as the inaccuracies and incompleteness of OSM data. These shortcomings can lead to difficulties in robot localization and path planning, which can be critical in dynamic environments. The paper emphasizes the need for more accurate and reliable data sources to improve navigation performance. A probabilistic approach to autonomous robot navigation using OpenStreetMap data is presented in [18]. They address the

problem of map inaccuracies and the uncertainty of the robot’s position relative to the map. The study combines 3D-LiDAR data with OSM tracks to improve the alignment of the vehicle’s pose with the map data. Despite these enhancements, the authors note that OSM data often lacks the detailed information required for precise navigation, such as lane-level data, which is important for accurate and safe autonomous driving in complex environments.

We next review the background on trajectory planning methods for autonomous driving. In [19], the authors describe an algorithm that creates smooth trajectories for autonomous vehicles. They model paths and lane segments using straight lines and circular arcs for simplicity and elegance. The process involves developing the path in an idealized space, computing path length, and generating a one-dimensional trajectory that meets speed and position targets. [20] explores the Reachability-based Trajectory Design (RTD) algorithm for real-time trajectory planning, which ensures safety in autonomous driving. RTD consists of an offline computation of the Forward Reachable Set (FRS) of the vehicle, parameterizing tracking trajectories, and an on-line optimization using the FRS to map obstacles to constraints safely. The work in [21] proposes a trajectory generation method using G2 cubic Bézier spiral smoothing for high-speed autonomous vehicles in structured on-road environments, such as highways. This method creates smooth trajectories based on the current path and the centerline model of the desired lane. The use of Model Predictive Control (MPC) for trajectory generation in autonomous vehicles has been explored in various innovative approaches. [22] developed a method to mimic skilled human driving by maximizing road width usage while maintaining vehicle stability. [23] integrated deep learning with MPC, using recurrent neural networks to predict the trajectories of nearby vehicles, thus enhancing collision avoidance and decision-making. [24] proposed a unified path planning approach that automatically decides maneuvers and incorporates collision avoidance constraints, ensuring smooth and safe navigation. Additionally, [25] combined potential field methods with MPC to treat different obstacles and road structures distinctly, optimizing path planning while maintaining vehicle stability and adherence to road regulations. These advancements highlight the versatility and effectiveness of MPC in improving the safety and performance of autonomous vehicles.

The literature on mapping and trajectory generation for autonomous vehicles reveals

several gaps that need addressing to enhance accuracy, safety, and efficiency. Specifically, one under-explored area in road modeling for autonomous vehicles is the implementation of memory efficiency techniques. Efficient data structures and algorithms that optimize memory usage are important to ensure real-time performance without overwhelming the system's computational resources. Detailed evaluation metrics and error analysis are crucial for assessing the effectiveness of road modeling approaches. Many existing studies lack comprehensive error metrics, such as RMSE and other statistical analyses, which are vital for understanding model performance. [26] emphasize the need for robust evaluation frameworks, defining methods to compare algorithms objectively. Their work outlines the importance of scenario-based evaluations and comprehensive error metrics to ensure the reliability and safety of autonomous vehicles. Considering trajectory planning for autonomous vehicles, MPC is effective for trajectory planning in autonomous vehicles but comes with several challenges. Its computational complexity often requires substantial processing power, leading to delays that hinder real-time performance. MPC also relies on accurately tuned parameters and precise models, making it sensitive to inaccuracies. The extensive parameter tuning and complex implementation further add to its development time. Simpler analytical methods offer faster computation and deterministic performance, making them more suitable for real-time applications [27]. There are also trajectory planning algorithms utilizing machine learning, such as [28], [29], [30] and [31]. The first issue that comes with machine learning algorithms is the collection of required training data and the reliability of the trained models in safety-critical applications such as autonomous driving. Overall, existing studies collectively underline the importance of advancing both mapping and trajectory generation methods to address the existing shortcomings in autonomous vehicle navigation. As a common problem, it has to be highlighted that existing methods use different curve representations for mapping and trajectory planning.

In this thesis, we propose novel methods for road modeling and trajectory planning using arc-splines. Assuming that highways are designed with clothoid curves, we leverage this characteristic to represent roads with arc-splines. Our primary objective is to minimize the number of parameters required to accurately depict a highway. To achieve this, we implement predefined error metrics that balance the trade-off be-

tween the number of parameters and the accuracy of the model. Additionally, our approach concatenates multiple road segments into single segments wherever feasible. By utilizing arc-spline definitions, our methodology also facilitates the generation of additional lanes from a single lane by parallel shifting, significantly reducing the overall number of parameters required for road representation. In addition to our road modeling method, we propose a trajectory planning algorithm based on arc-splines, similar to our road modeling approach. This algorithm employs an analytical approach, ensuring low computational power and reduced computation time. The algorithm can compute a trajectory that achieves zero error at the destination point by using arc-splines for both the road model and the trajectory. For a comprehensive analysis, this study also implements Bézier curves, allowing for a comparison between Bézier curve-based trajectories and those based on arc-splines. This comparative analysis aims to highlight the strengths and potential advantages of using arc-splines in trajectory planning for autonomous vehicles.

The main contributions of our methodology are as follows:

1. **Arc-spline-Based Road Modeling:** We propose a novel method for modeling highways using arc-splines, assuming highways are designed with clothoid curves.
2. **Parameter Minimization:** Our algorithm minimizes the number of parameters required to accurately represent a highway by implementing predefined error metrics that balance parameter count and model accuracy.
3. **Segment Concatenation:** The method concatenates multiple road segments into single segments wherever feasible, further reducing the number of parameters.
4. **Lane Generation:** By utilizing arc-spline definitions, the methodology allows for the generation of additional lanes from a single lane, significantly decreasing the overall parameter count.
5. **Trajectory Planning Algorithm:** We introduce a trajectory planning algorithm based on arc-splines, which is computationally efficient due to its analytical approach.

6. **Error-Free Destination Point:** The trajectory planning algorithm ensures zero error at the destination point by aligning both the road model and trajectory on arc-splines.
7. **Comparative Analysis with Bézier Curves:** The study includes an implementation of Bézier curves for trajectory planning, providing a comparative analysis to highlight the advantages of arc-spline-based trajectories.

CHAPTER 2

BACKGROUND INFORMATION

This thesis focuses on the efficient representation and trajectory planning for autonomous vehicles on highways. The first part of the thesis develops a new method for effectively representing roads as arc and line segments to capture their geometric properties. By merging these segments where possible, the methodology minimizes the data needed while keeping important information like position, heading, and curvature. The second part of the thesis focuses on using the developed road representation for trajectory planning.

This Chapter provides the required background for the thesis. Section 2.1 presents basic concepts about road modeling and the collection of accurate waypoint data for roads. In addition, Bézier curves are explained since they are useful in generating smooth paths between specified points. We use Bézier curves for comparison in this thesis. The main contributions of the thesis are summarized in Section 2.2.

2.1 Basic Concepts

2.1.1 Road Modelling

2.1.1.1 Clothoids

In modern highway design, clothoids are widely used [?, ?]. Known as the Euler or transition curve, a clothoid keeps the same curvature rate along its length. This feature makes them very useful for road construction to ensure smooth transitions between straight and curved sections. Therefore, clothoids also become important for setting

a reliable path for autonomous vehicles on highways.

The mathematics of clothoids are based on Fresnel integrals, specifically the Fresnel Sine and Cosine integrals. The curvature $\kappa(s)$ of a clothoid in terms of the arc length s is defined as:

$$\kappa(s) = k_1 + \frac{(k_2 - k_1)s}{L} \quad (2.1)$$

where k_1 is the initial curvature, k_2 is the final curvature. L is the total length of the trajectory. Figure 2.1 shows an example of the linear curvature change of a clothoid.

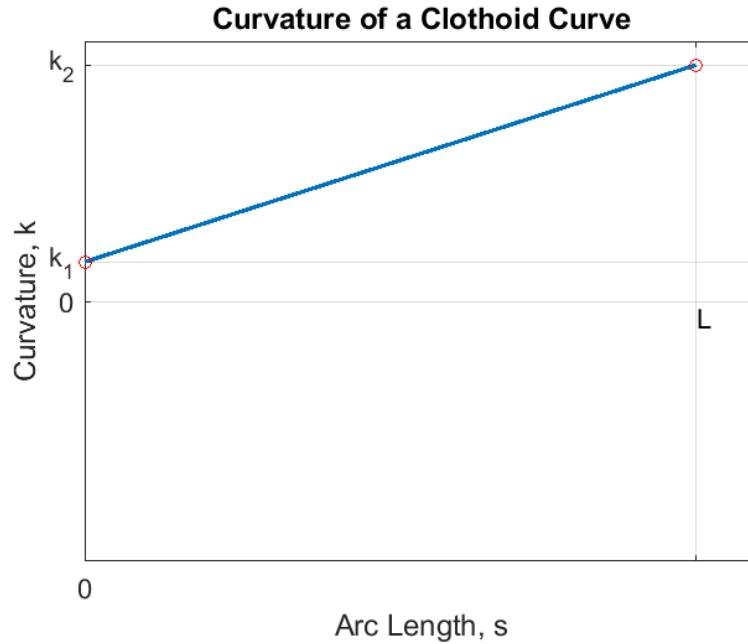


Figure 2.1: Curvature plot of a clothoid curve.

The integral of clothoid curve's curvature gives the heading change $\theta(s)$ along the arc length of a clothoid curve. The heading at any point depends on curvature and path length:

$$\theta(t) = \theta_0 + \int_0^t \kappa(s) ds \quad (2.2)$$

where θ_0 is the initial heading and Equation (2.1) gives $\kappa(s)$. This integrates to:

$$\theta(s) = \theta_0 + k_1 s + \frac{k_2 s^2}{2L} - \frac{k_1 s^2}{2L} \quad (2.3)$$

which shows how the heading changes as a function of the distance s along the curve. Then, the x and y coordinates of the clothoid are determined by the following inte-

grals:

$$\begin{aligned} x(s) &= \int_0^s \cos(\theta(t)) dt + x_0 \\ y(s) &= \int_0^s \sin(\theta(t)) dt + y_0 \end{aligned} \quad (2.4)$$

Clothoids have a few different parametrizations. In this thesis $\mathcal{C} = [x_i, y_i, \theta_i, \kappa_i, \kappa_f, \mathcal{L}]$ parametrization and notation is used. x_i, y_i represents the initial position, θ_i is the initial heading of the trajectory, κ_i and κ_f is the initial and final curvature of the trajectory respectively while \mathcal{L} is the length of the trajectory.

Figure 2.2 shows a plot of clothoid trajectory whose initial values are defined as zero.

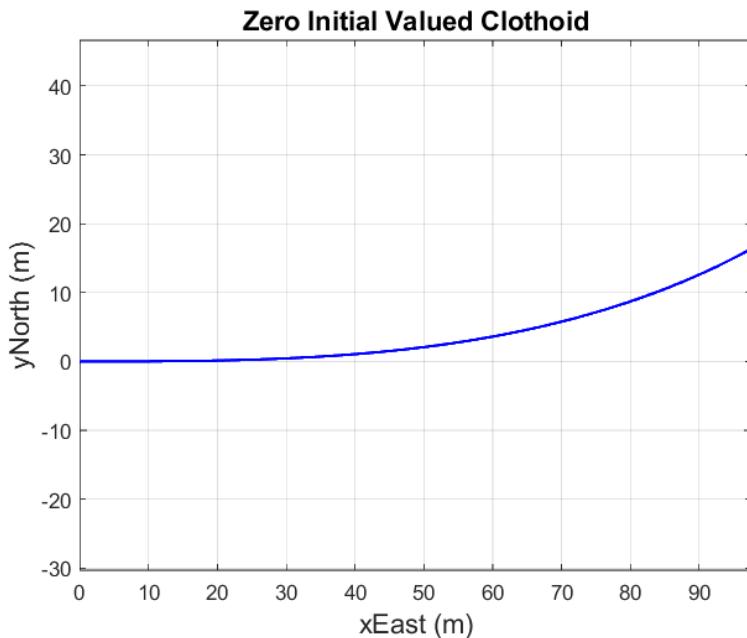


Figure 2.2: Plot of $\mathcal{C} = [0, 0, 0, 0, 0.01, 100]$ clothoid.

A clothoid does not have to have increasing curvature along the trajectory. Curvature can also linearly decrease. Figure 2.3 gives an example for this type of clothoids.

Even though the clothoids in Figure 2.2 and Figure 2.3 have similar parameter values the trajectories are very different. This difference may be observed in Figure 2.4.

Clothoids are particularly useful in road construction due to their property of having a curvature that changes linearly with arc length, facilitating smooth transitions between straight sections and curves. The ability of clothoids to seamlessly con-

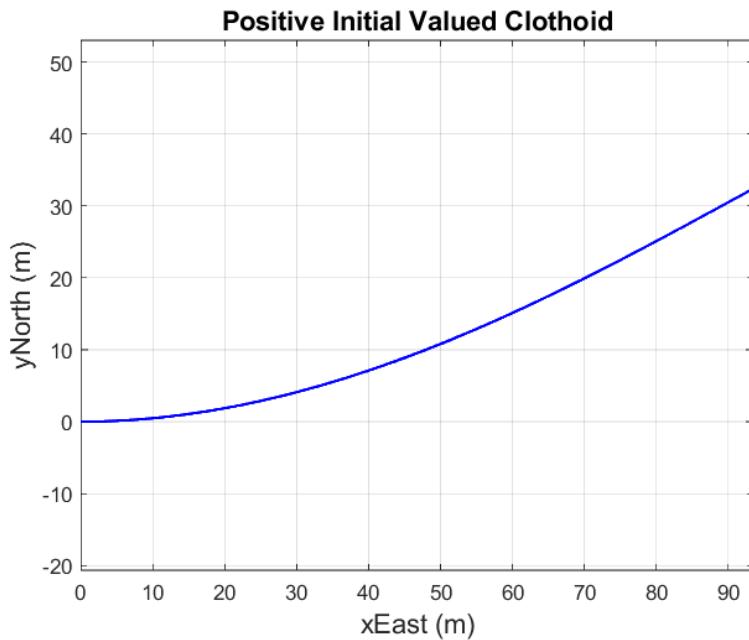


Figure 2.3: Plot of $\mathcal{C} = [0, 0, 0, 0.01, 0, 100]$ clothoid.

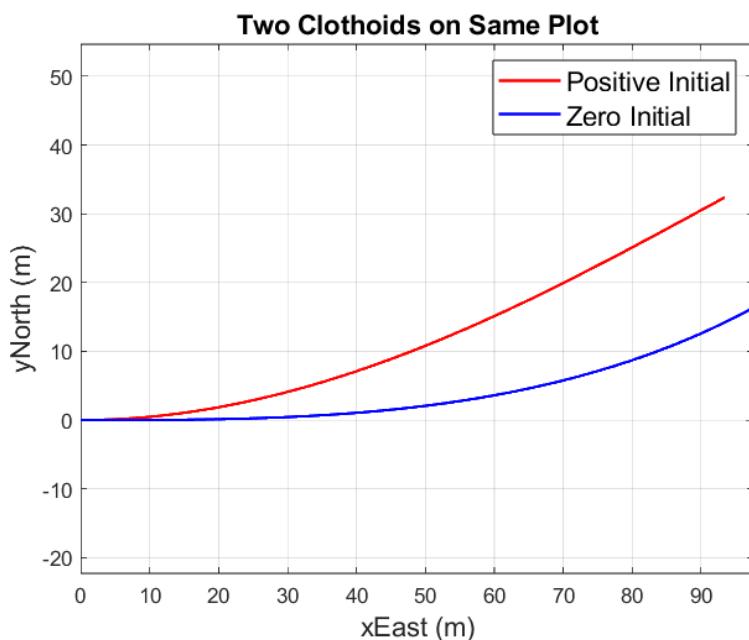


Figure 2.4: Plot of both $\mathcal{C} = [0, 0, 0, 0.01, 0, 100]$ and $\mathcal{C} = [0, 0, 0, 0, 0.01, 100]$.

nect different roadway segments minimizes abrupt changes in steering requirements, which is important for high-speed travel.

Despite these advantages for road design, the computational aspect of using clothoids has challenges. Clothoids do not have an analytical solution for their arc length, which means that calculating their coordinates from given parameters requires numerical integration. This process can be computationally intensive and less efficient for real-time applications, such as autonomous vehicle navigation systems that demand rapid calculations. The necessity for integration makes the implementation of clothoid-based paths more complex compared to simpler curve forms that offer analytical solutions.

2.1.1.2 Clothoid Approximation with Arc Splines

We use a method to obtain an analytical approximation of clothoid segments. Our main motivation for this approximation is to simplify and enhance the usability of these curves in practical applications. Representing a segment with arc splines offers several advantages: it facilitates the creation of additional lanes from a single lane by simply adjusting the turning radius of the arcs by the width of a lane. This method also allows for straightforward computation of coordinates along the trajectory, eliminating the need for complex integration processes.

In the approximation of a clothoid curve $\mathcal{C} = [x_i, y_i, \theta_i, \kappa_i, \kappa_f, \mathcal{L}]$, we assume that the initial curvature k_i and the final curvature k_f share the same sign. The approximation involves creating an arc spline of order n , which consists of $n + 1$ arc segments. The key parameters defining an arc spline are as follows:

- **Curvature increment h :** Defined as $h = \frac{k_f - k_s}{n}$.
- **Curvature of arc j** for $j = 0, \dots, n$: Given by $k_j = k_s + jh = k_s + j\frac{k_f - k_s}{n}$.
- **Length of the first and last arcs $j = 0, n$:** $S_0 = S_n = \frac{S}{2n}$.
- **Length of the remaining arcs $j = 1, \dots, n - 1$:** $S_j = \frac{S}{n}$.

Using these parameters, the arc spline representing the clothoid curve is parameterized accordingly.

For later use, we also list the memory required to store the different parameters of an arc-spline:

- Initial point (x_i, y_i): 2 floats, 8 bytes
- Initial heading (θ_i): 1 float, 4 bytes
- Initial curvature (κ_i): 1 float, 4 bytes
- Final curvature (κ_f): 1 float, 4 bytes
- Trajectory length: 1 float, 4 bytes
- Number of arcs: 1 byte

For further observation on curvatures, Figure 2.5 may be observed. Over the length of the original clothoid, a series of arcs is generated with curvature values at certain points. These points are related to the order of the approximation. In order to maintain second-order derivative continuity, consecutive arcs start with the heading of the preceding arc's heading.

Figure 2.6 shows an arc spline with order 5. When the order is 5, there are 6 arcs to approximate the curvature. Each arc's center may also be observed. This clothoid's curvature increases; therefore, the turning radius decreases towards the end of the spline.

The order of the approximation affects the error between the approximation and the actual clothoid. Specifically, smaller errors are obtained for higher-order approximations. To be memory efficient, the order is commonly decreased as much as possible while keeping a certain bound on the error.

When representing roads by arc-splines, the error metrics are RMS Euclidian distance and maximum Euclidian distance between approximation and ground truth over a whole road segment. That is, the minimum distance between each point of the approximation and the ground truth is computed. Then, the RMS and maximum errors

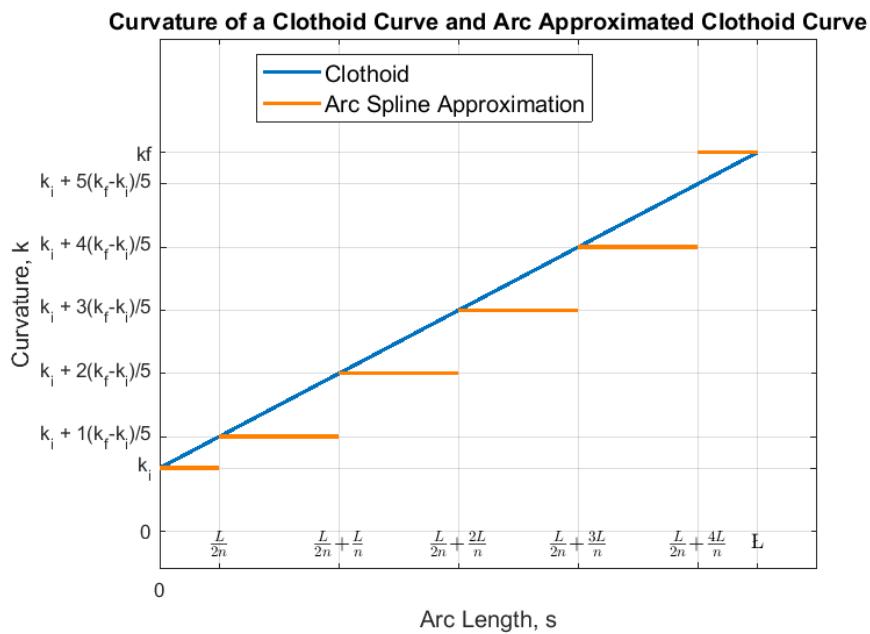


Figure 2.5: Curvature plot of a clothoid curve and curvatures of arcs.

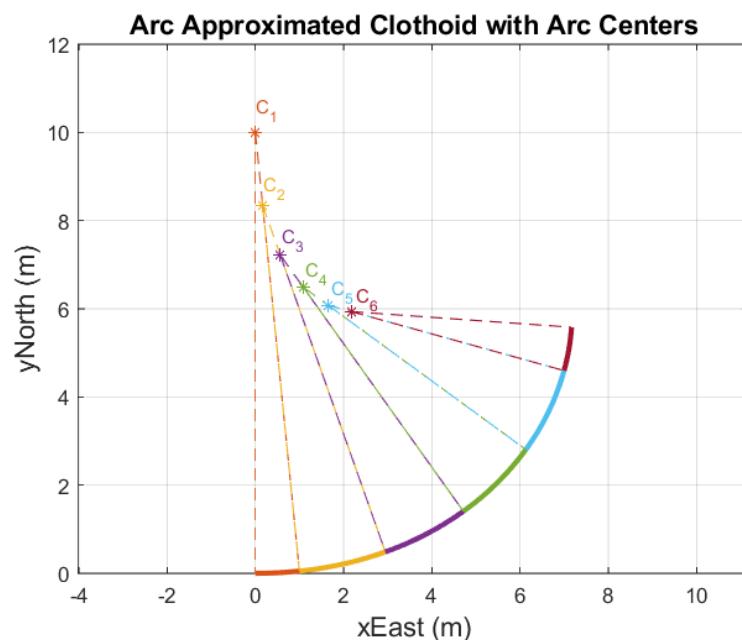


Figure 2.6: Approximation of $\mathcal{C} = [0, 0, 0, 0.1, 0.2, 10]$ with arc-spline.

are computed. Based on the accuracy objective, the order of approximation is either increased, decreased, or kept the same. Figure 2.7 shows an example of an error plot. The error oscillates since the road is a clothoid and makes a smooth turn, whereas the approximation is a series of arcs.

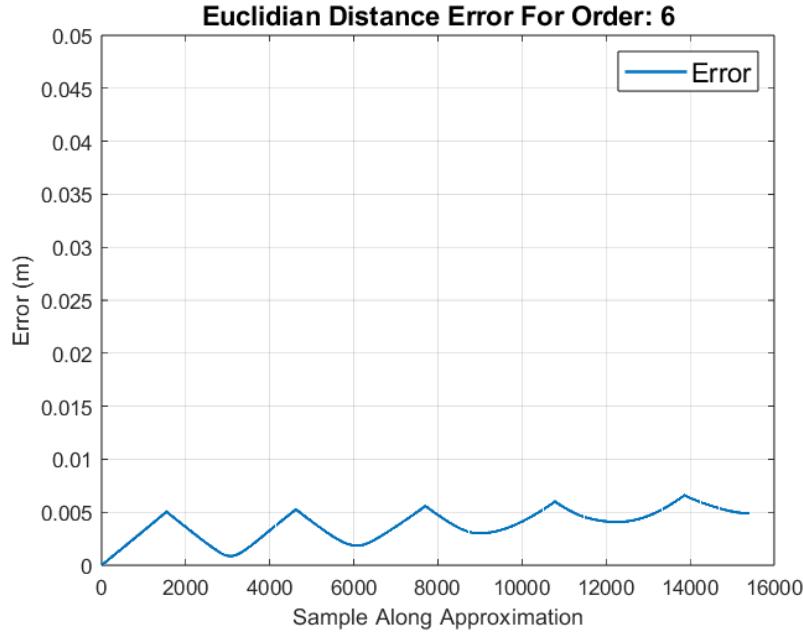


Figure 2.7: Error for a 154 meter long road segment with an order 5 approximation.

Figure 2.8 shows a ground truth and approximated curve. The Figure is zoomed in to observe the error better.

2.1.2 Waypoint Data

The input to our algorithm is a series of waypoints, where each waypoint includes position, heading, and curvature information. The position gives the exact location of the waypoint on the map, the heading tells us the direction the vehicle is facing at that point, and the curvature shows how sharply the road is turning at that location. While these three pieces of information provide some geometry information about the road, they are not enough to represent the entire road. To address this, we assume that every road segment is a clothoid. With this assumption, we can approximate segments with an arc spline as discussed in Section 2.1.1.2, allowing for a practical representation

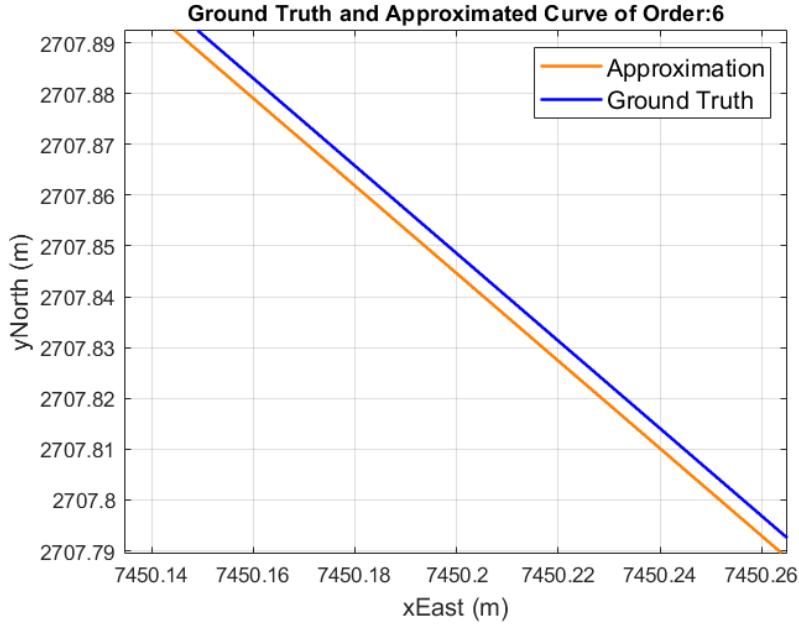


Figure 2.8: A closeup view of an approximation and real road data.

of the road. Moreover, to generate the intermediate ground truth road between two waypoints a G1 clothoid fitting algorithm [32] is used.

2.1.2.1 OpenStreetMap Database

The OpenStreetMap database consists of latitude and longitude pairs along the requested highway. For each direction in a road a separate latitude and longitude pair exists in the data. Lane information does not exist in this dataset.

Figure 2.9 shows an example OpenStreetMap data. Waypoints are connected to each other with lines.

For some part of the road the waypoints are at the center of the lane however for some part of the road, the waypoints are located at lane borders. This type of data would create issues while modeling the road.

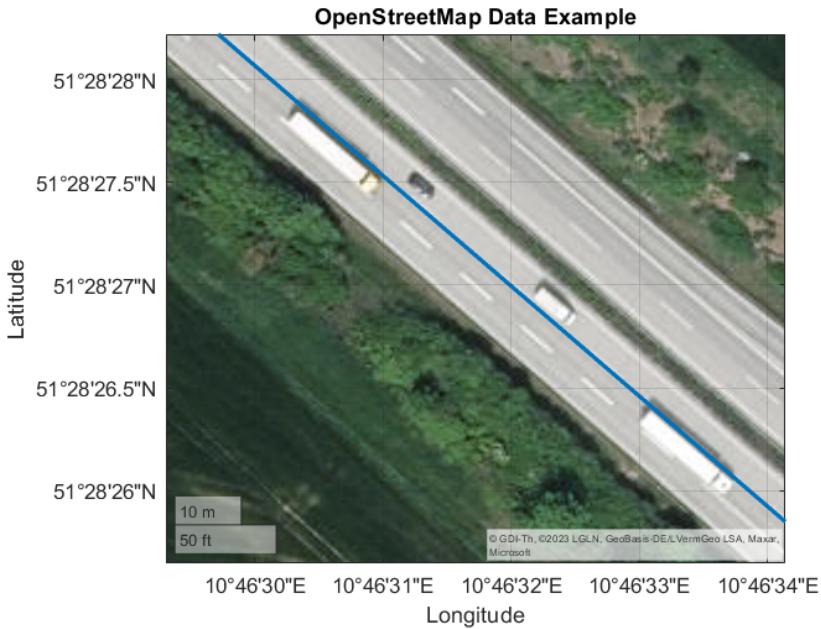


Figure 2.9: OpenStreetMap data example of Autobahn 38 in Germany.

2.1.2.2 HERE Maps Database

The HERE Maps database consists of latitude and longitude pairs along the requested highway. For each lane, a separate latitude and longitude pair exists in the data. This property allows for the computation of errors in the parallel shifting procedure, which will be discussed in the upcoming Sections. Figure 2.10 displays an example road segment where each color represents a different lane. Waypoints are connected to each other with a simple line.

The HERE Maps database is notably more accurate than the OpenStreetMap database. Unlike OpenStreetMap, it offers a higher frequency of waypoints and provides position information for each individual lane. These features make HERE Maps more useful for this research.

2.1.3 Bézier Curve Based Trajectory Planning

Bézier curves are practical tools used in computer graphics and vehicle design. They work by defining a curve with a few control points. These parametric curves are

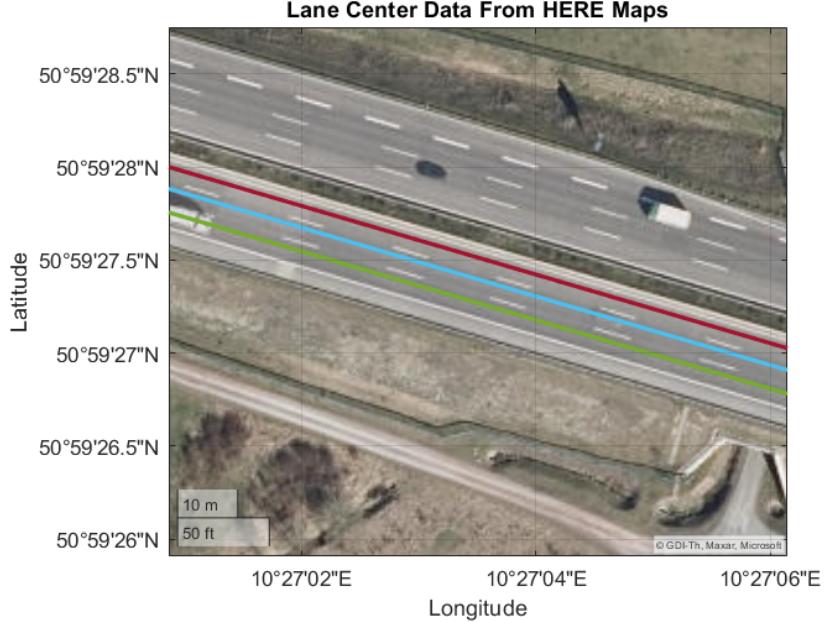


Figure 2.10: HERE Map example of Autobahn 4 in Germany.

beneficial for creating smooth shapes and paths in graphics and for designing smooth car bodies [33]. They give designers precise control over the shape of the curve, making them really useful in various applications where smooth, controlled curves are needed.

Given the detailed comparison between lower-order curves and quintic Bézier curves in [34], the choice is made to utilize quintic Bézier curves for generating the final path. One of the primary advantages of fifth-order Bézier curves over cubic ones is the enhanced control they offer in shaping the curve, as well as the ability to impose curvature at both ends of the curve.

Equation (2.5) is the definition of quintic Beziér curves.

$$B(t) = \sum_{i=0}^5 \binom{5}{i} (1-t)^{5-i} t^i P_i \quad (2.5)$$

A Quintic Bézier curve requires 6 control points by definition. To define the control points of a Bézier curve, we utilize the initial and final pose of the vehicle. P_0 and P_5 are directly the position of the vehicle and the destination position respectively. P_1 and P_4 are related to the heading of the vehicle and the heading value at the des-

tination, respectively. Equation (2.6) is the derivative of the Bézier curve. Similarly, P_2 and P_3 are related to the curvature of the vehicle and the heading value at the destination, respectively. Equation (2.7) is the second derivative of the Bézier curve.

$$B'(t) = \sum_{i=0}^4 5 \binom{4}{i} (1-t)^{4-i} t^i (P_{i+1} - P_i) \quad (2.6)$$

$$B''(t) = \sum_{i=0}^3 20 \binom{3}{i} (1-t)^{3-i} t^i (P_{i+2} - 2P_{i+1} + P_i) \quad (2.7)$$

It is possible to impose initial heading and curvature values to a Bézier curve by substituting $t = 0$ to Equations (2.6) and (2.7) respectively. Similarly, it is possible to impose destination heading and curvature values to a Bézier curve by substituting $t = 1$ to Equations (2.6) and (2.7) respectively. The methodology for computing the Bézier curves will be discussed in the upcoming Sections.

In this research, Bézier curves are defined by:

$$B = [(x_i, y_i), (x_f, y_f), \theta_i, \theta_f, \kappa_i, \kappa_f] \quad (2.8)$$

where (x_i, y_i) and (x_f, y_f) denote the initial and final positions, θ_i and θ_f represent the initial and final heading values, and κ_i and κ_f indicate the initial and final curvature values of the Bézier curve.

Figure 2.11 illustrates an example of a Bézier curve in the Cartesian coordinate system. The curve clearly demonstrates the specified initial and final positions and tangents. While the positions and tangents are straightforward to observe, understanding the curvature of the curve is not as intuitive for a human observer.

2.1.4 Bi-elementary Paths

Bi-elementary paths are paths that consists of 2 clothoids, one line segment and 2 more clothoids back to back [35], [36], [37]. These segments are named C_1, C_2, L, C_3, C_4 respectively.

γ parameter is the ratio of line segment's length to total maneuver length. λ is the ratio of first two clothoid's length to last two clothoid's length. Equation (2.9) governs these relations. S stands for the total maneuver length. S_{E_1}, S_{E_2} and S_L are the

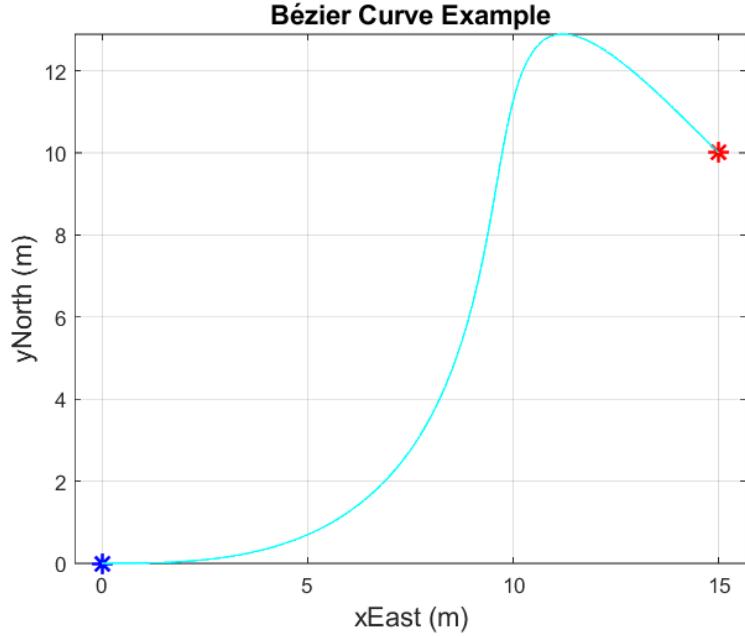


Figure 2.11: Bézier curve example for $B = [(0, 0), (15, 10), 0, 45^\circ, 0, 0.01]$

length of first two clothoids, last two clothoids and line segment respectively. First two clothoids have the same length, similarly last two clothoids have the same length.

$$\begin{aligned}
 S_{E_1} + S_{E_2} &= \gamma S \\
 S_L &= (1 - \gamma) \cdot S \\
 S_{E_1} &= \lambda \gamma S \\
 S_{E_2} &= (1 - \lambda) \gamma S
 \end{aligned} \tag{2.9}$$

In our work, we only work with symmetrical bi-elementary paths which means the line segment has the half length of the total maneuver. All clothoids has one quarter of the total maneuver length. This configuration is equivalent to $\lambda = 0.5$, $\gamma = 0.5$. An example Bi-elementary path is given in Figure 2.12 which shows a bi-elementary path's curvature and ΔY change over the trajectory.

Figure 2.13 shows the curvature and ΔY change over a Bi-elementary path with $\lambda = 0.50$ and $\gamma = 0.25$. As observed the length of line segment increases in this case. In fact most of the maneuver is a line segment. Since $\lambda = 0.5$, the length of clothoid segments are equal to each other.

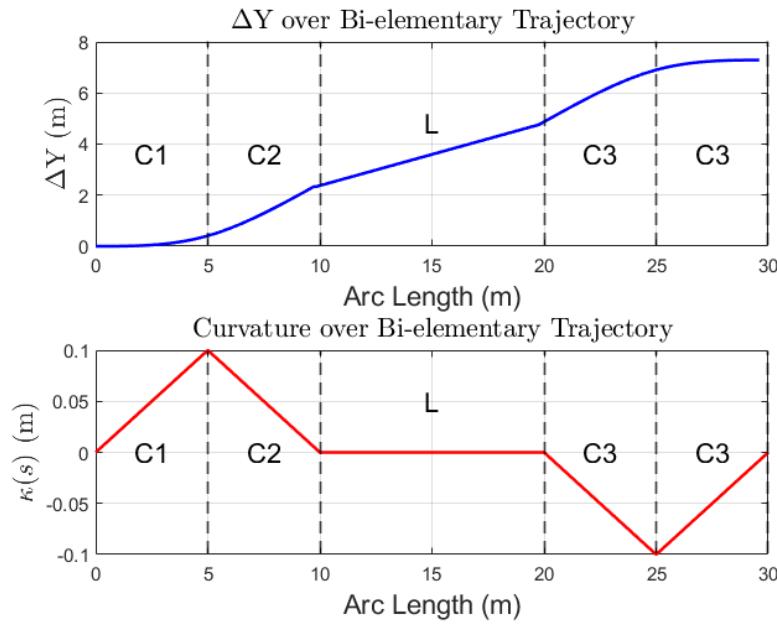


Figure 2.12: Curvature and ΔY change of a Bi-elementary curve for $\lambda = 0.50$,
 $\gamma = 0.50$.

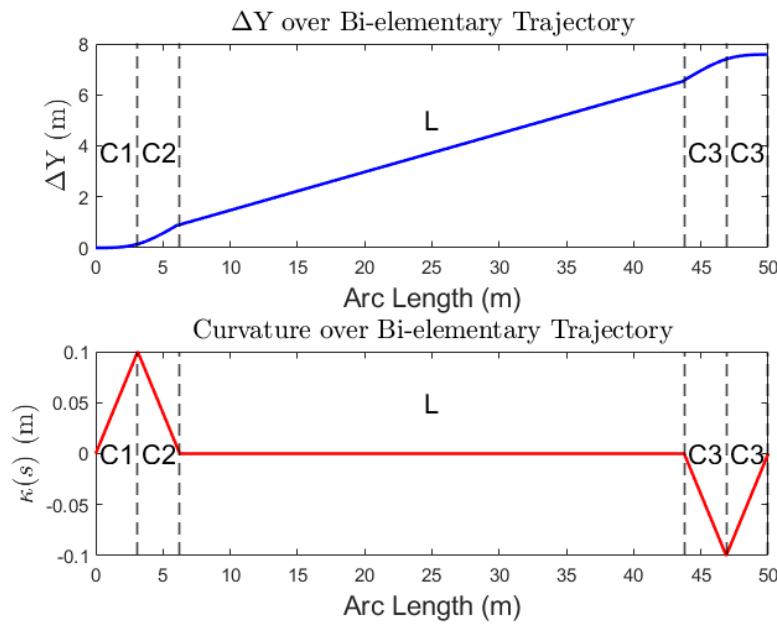


Figure 2.13: Curvature and ΔY change of a Bi-elementary curve for $\lambda = 0.50$,
 $\gamma = 0.25$.

Figure 2.14 shows the curvature and ΔY change over a Bi-elementary path with $\lambda = 0.75$ and $\gamma = 0.50$. Length of line segment is the same as Figure 2.12. However, the length ratio of clothoids change this time since $\lambda = 0.75$, first two clothoids longer than last two clothoids.

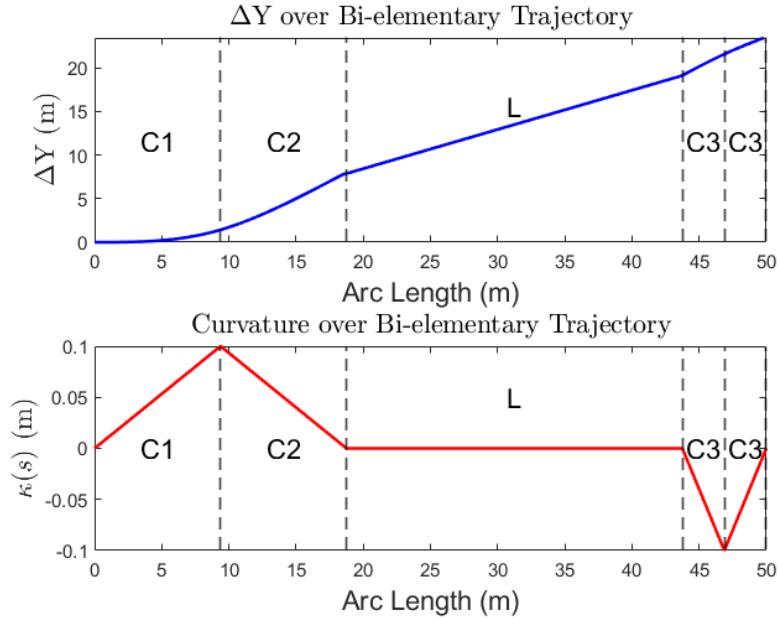


Figure 2.14: Curvature and ΔY change of a Bi-elementary curve for $\lambda = 0.75$,
 $\gamma = 0.50$.

Bi-elementary curves are especially used in lane change maneuvers. This is due to the fact that bi-elementary paths have the final curvature the same as the initial curvature. Similarly heading is preserved as well. These curves make the vehicle translate in the perpendicular direction to the heading of the vehicle which is denoted as ΔY .

In our case ΔY and arc length will be given which in that case Equation (2.10) is known by [38] and Equation (2.9). In this equation α and S are tangent angle change and arc length respectively.

$$\kappa_{peak} = \frac{8\alpha}{S} \quad (2.10)$$

The value of α is computed with Newton iteration method given in [38].

2.2 Contributions and Novelties

Using clothoid curves and their arc-spline approximations, the main contributions of this thesis are as follows:

Arc Spline-Based Road Modeling: We propose a novel method for efficiently modeling highways using arc splines, assuming highways are designed with clothoid curves.

- Parameter Minimization: Our algorithm minimizes the number of parameters required to accurately represent a highway by implementing predefined error metrics that balance parameter count and model accuracy.
- Road Segment Concatenation: Our method concatenates multiple road segments into single segments wherever feasible, further reducing the number of parameters.
- Lane Generation: By utilizing arc spline definitions, the methodology allows for generating additional lanes from a single lane by a simple parallel shift, significantly decreasing the overall parameter count.

The developed road representation method is described in Chapter 3.

Trajectory Planning Algorithm: We introduce a trajectory planning algorithm based on arc splines, which is computationally efficient due to its analytical approach.

- **Error-Free Destination Point**: The trajectory planning algorithm ensures zero error at the destination point by aligning both the road model and trajectory on arc splines.
- **Comparative Analysis with Bézier Curves**: The study includes an implementation of Bézier curves for trajectory planning, providing a comparative analysis to highlight the advantages of arc spline-based trajectories.

The trajectory planning algorithm is described in Chapter 4.

CHAPTER 3

ROAD SEGMENT REPRESENTATION METHODOLOGY

In trajectory planning for autonomous vehicles on highways, having an efficient and accurate road representation is very important. Roads are usually represented by waypoints, each containing information about its position, heading, and curvature. However, these waypoints are often spread out and do not provide a detailed picture of the road between them. To achieve an accurate road representation, waypoints must be stored closely together, which requires a significant amount of memory. Additionally, considering a multi-lane road, waypoints need to be stored for each lane separately, further increasing the memory needed.

To mitigate the observed problem, we suggest dividing roads into road segments, where each road segment defines the road between two waypoints. In addition, we focus on a clothoid-based road representation, noting that clothoids are already used for road construction [?]. The advantage of clothoids is that they require only a small number of parameters to represent a road segment: the coordinates, heading, and curvature of the starting point of the road segment; the change in curvature during the road segment; the arc-length of the road segment.

It has to be respected that clothoids do not have an analytical representation, which is a clear shortcoming in real-time autonomous driving applications. Because of this reason, we propose to use an arc-spline approximation of clothoids. Specifically, we can analytically compute accurate intermediate points between the starting and end points of any road segment. In addition, arc-splines can be parallel shifted, such that it is possible to use a single arc-spline for a multi-lane road. Overall, using arc-splines enables an efficient and accurate road representation with few parameters and, hence, a small amount of required memory.

In the following Sections, we explain how we use a clothoid fitting algorithm [32] to get the ground truth for the road centerline. Then, we apply arc-spline approximation to represent the road segments. As a result, we obtain an accurate and efficient road representation, which also helps improve the performance of trajectory planning for autonomous vehicles. Algorithm 1 summarizes the general flow of the methodology which is detailed in the subsequent Sections.

At first, waypoints are downloaded from HERE Maps for the given coordinates and stored.

Then, clothoids are fitted between each waypoints with G1Fit function. These clothoids are the ground truth for our work.

Each clothoid is approximated with an arc spline based on some error configuration $errCfg$.

Approximated arc spline segments are merged with the same $errCfg$.

Additional lanes are generated by parallel shifting the approximated segments. Lane width and number of lanes are given as input to genLanes function.

Algorithm 1 Road Segment Approximation Method

- 1: **Input:** $LU, RL, errCfg, laneW, nLanes$
 - 2: $wp = \text{downloadData}(LU, RL)$
 - 3: $clo = \text{G1Fit}(wp)$
 - 4: $approxSeg = \text{approxSeg}(clo, errCfg)$
 - 5: $mergedSeg = \text{mergeSeg}(approxSeg, errCfg)$
 - 6: $addLanes = \text{genLanes}(approxSeg, laneW, nLanes)$
-

The parameters use in the algorithm are summarized as follows:

- LU : Left upper latitude and longitude of the bounding box.
- RL : Right lower latitude and longitude of the bounding box.
- $errCfg$: Error configuration parameter.
- $laneW$: Width of a single lane.
- $nLanes$: Number of lanes.

- wp : Waypoints data.
- clo : Clothoids fitted to the waypoints.
- $approxSeg$: Approximated road segments.
- $mergedSeg$: Merged road segments.
- $addLanes$: Additional lanes generated.

We explain the details of this algorithm in the next Sections.

3.1 Ground Truth Generation

In order to generate the ground truth, a fast G¹ clothoid fitting algorithm is used [32]. Initially, the HERE maps database provides only the positions of waypoints, not the heading and curvature. After clothoid fitting, the information of each waypoint is extended by heading and curvature.

One of the notable properties of the clothoid fitting algorithm is the G^1 continuity, which guarantees that the first derivative of the curve, representing the tangent direction, is continuous. This continuity is important for ensuring smooth changes in the vehicle's heading, minimizing sudden jerks or deviations in the path. The algorithm iteratively adjusts the clothoid parameters to fit a given set of waypoints, optimizing the curve to minimize the deviation from the actual road path.

The distance between every road segment's endpoint and the next waypoint is computed. Consequently, the G^1 clothoid fitting algorithm is guaranteed to have an error lesser than 10^{-5} meters, which is an outstanding performance.

Figure 3.1 shows a small example of the fitting procedure. 5 waypoints are provided as input, and each color on the lane center represents a different road segment.

Table 3.1 shows a portion of sample data extracted from HERE Maps. As seen, the length of each segment vary significantly. Curvature information is obtained after G^1 clothoid fitting algorithm is applied.

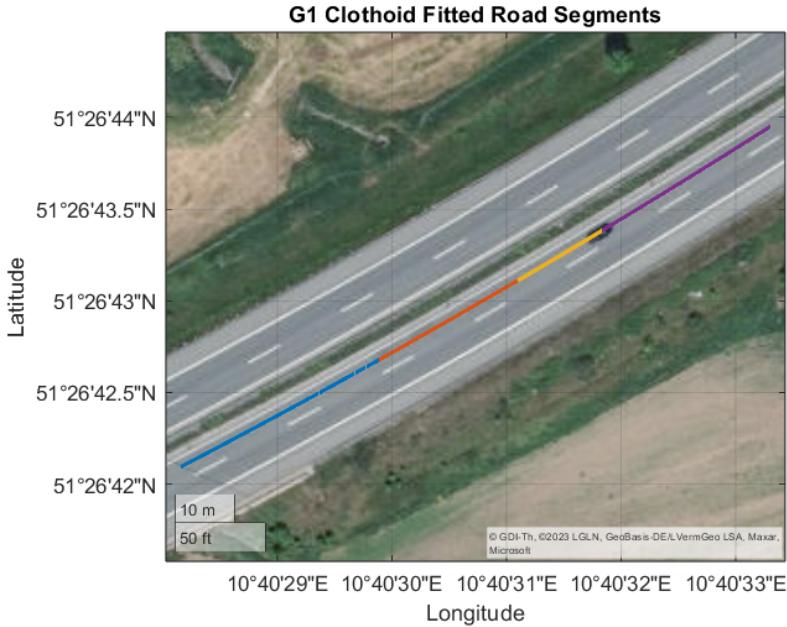


Figure 3.1: G1 clothoid fitting applied on 5 waypoints.

3.2 Approximation

As indicated above, clothoids do not have an analytical representation and, hence, cannot be used in real-time autonomous driving applications. Accordingly, we next develop a method for approximating the computed clothoids by arc-splines.

The input to the approximation algorithm is a series of road segment clothoids. The approximation output is a series of road segments where each segment is either a line segment or an arc-spline segment. Therefore, the type of each road segment must be decided to make the approximation more efficient. In this context, efficiency refers to both the number of parameters required to represent the road and the accuracy of the representation. Line segments are more efficient than arc-spline segments as they require fewer parameters to store and are simpler to compute. Consequently, line segments will be used whenever possible to minimize data storage and computational effort. As mentioned in Section 2.1.1.2, it is possible to approximate clothoids with a variable number of arcs, balancing the trade-off between simplicity and accuracy. Simplicity here refers to the ease of computing waypoints for autonomous driving, as fewer arcs result in less computational complexity.

Table 3.1: Road Segment Data

Road Segment	Initial Curvature (m^{-1})	Ending Curvature (m^{-1})	Segment Length (m)
1	5.863553e-04	7.166330e-04	21.7
2	7.166330e-04	6.989685e-04	27.8
3	6.989685e-04	4.804532e-04	27.9
4	4.804532e-04	8.929795e-04	12.7
5	8.929795e-04	1.228206e-03	19.2
6	1.228206e-03	2.689868e-04	19.3
7	2.689868e-04	1.057045e-03	11.3
8	1.057045e-03	2.396309e-04	12.4
9	2.396309e-04	5.064884e-04	20.8
10	5.064884e-04	8.977215e-04	27.3
11	8.977215e-04	1.092975e-03	22.7
12	1.092975e-03	4.962459e-04	22.7
13	4.962459e-04	7.071046e-04	36.6
14	7.071046e-04	6.607312e-04	26.1
15	6.607312e-04	8.301637e-04	26.1

To determine whether a road segment can be approximated by a line segment, we first assess the heading change between the initial and destination points. If the heading difference between the two waypoints is sufficiently small, a straight line connecting the initial point to the final point is fitted. Subsequently, the Euclidean distance error between the approximation and the ground truth is calculated.

If the maximum error exceeds the predefined allowable error or if the Root Mean Square error (RMSE) exceeds the maximum allowable RMSE, the segment cannot be accurately represented by a straight line. In such cases, the segment must be fitted with an arc-spline instead. Conversely, if the error remains within acceptable limits, the segment is fitted successfully with a straight line.

Predefined error metrics must be established to approximate a road segment with an arc-spline. Like line segment approximation, the error metrics for arc-spline approximation include the RMSE and maximum errors. The arc-spline approximation

process is iterative, beginning with an initial order of 5 for the spline.

First, the segment is approximated using this initial order, and the Euclidean distance error is computed. If the error falls within the acceptable range, the order is reduced to decrease the number of parameters, thereby simplifying the representation. Conversely, if the error exceeds the acceptable range, the order is increased to achieve a more accurate representation of the road segment. This iterative approach ensures that each road segment is represented with optimal accuracy while minimizing the complexity of the spline.

Figure 3.2 shows the error along a trajectory for low and high-order approximations. It is clear that higher-order approximations are much more accurate.



Figure 3.2: Error comparison of low and high order approximations for the clothoid

$$\mathcal{C} = [0, 0, 0, 0, 0.1, 50].$$

Table 3.2 is given for a better understanding of the predefined error metrics. [39] also emphasizes the requirements for a high definition map. Endpoint degree deviation is the deviation from ground truth at the end waypoint of the segment. arc-splines do not have an endpoint degree deviation error by their nature.

The presented algorithmic procedures ensure that each road segment is represented

Table 3.2: Error Table

	Endpoint Degree Deviation	RMSE (m)	Maximum Error (m)
Line Segment	0.2°	0.10	0.15
arc-spline	-	0.10	0.15

with minimal error, maintaining accuracy and increasing efficiency.

Figure 3.3 shows the error of an example approximation of a real road. The RMSE and maximum error are computed for every 10-meter-long subsegment to better observe the evolution of error values.

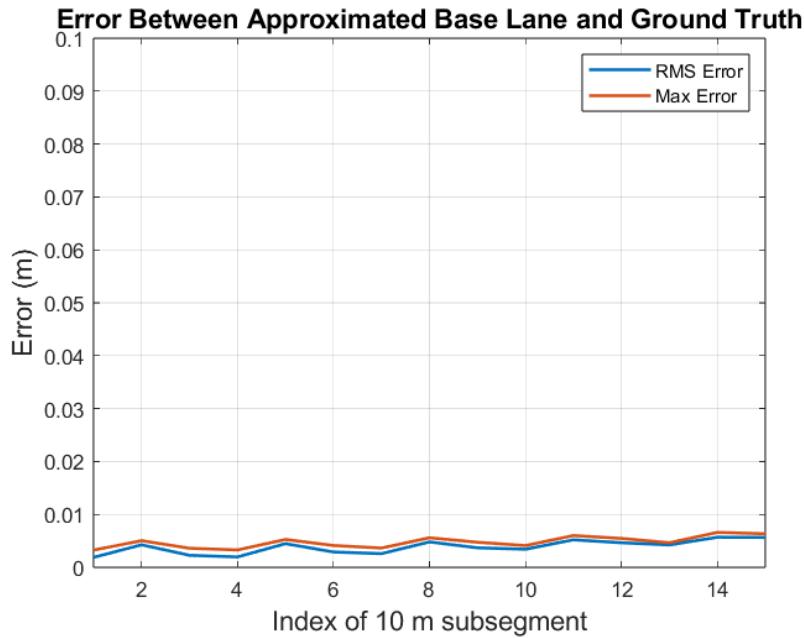


Figure 3.3: Euclidian error between approximated reference lane and ground truth

Pseudocode 2 gives the complete algorithm for road segment approximation algorithm.

3.3 Combination

A method to further decrease the number of parameters in a road representation is the process of merging consecutive road segments. In principle, if a road is entirely

Algorithm 2 Road Segment Approximation

```
1: for each road segment  $r$  do
2:    $\theta \leftarrow \text{headingChange}(r)$ 
3:   if  $\theta < \text{lineDegreeDeviation}$  then
4:      $\text{error} \leftarrow \text{fitLine}(r)$ 
5:     if  $\text{error} < \text{predefinedMetrics}$  then
6:        $\text{save}(\text{lineApproximation})$ 
7:       continue
8:     end if
9:   else
10:     $\text{order} \leftarrow 5$ 
11:     $\text{error} \leftarrow \text{fitArcSpline}(r, \text{order})$ 
12:     $\text{fitFlag} \leftarrow \text{False}$ 
13:    while  $\text{error} < \text{predefinedMetrics}$  do
14:       $\text{order} \leftarrow \text{order} - 1$ 
15:       $\text{error} \leftarrow \text{fitArcSpline}(r, \text{order})$ 
16:      if  $\text{error} > \text{predefinedMetrics}$  then
17:         $\text{save}(\text{order} + 1)$ 
18:         $\text{fitFlag} \leftarrow \text{True}$ 
19:        break
20:      end if
21:    end while
22:    while  $\text{error} > \text{predefinedMetrics}$  do
23:       $\text{order} \leftarrow \text{order} + 1$ 
24:       $\text{error} \leftarrow \text{fitArcSpline}(r, \text{order})$ 
25:      if  $\text{error} < \text{predefinedMetrics}$  then
26:         $\text{save}(\text{order})$ 
27:        break
28:      end if
29:    end while
30:  end if
31: end for
```

straight and has multiple waypoints, it is possible to represent the entire road with a single line segment. Specifically, for consecutive line segments, if the initial and final heading values of initial and final waypoints are close to each other, it may be possible to merge these line segments into a single line segment.

Similarly, a single arc-spline may represent a group of arc-splines if consecutive arc-spline segments have similar curvature rates. It is known that arc-splines have either increasing or decreasing curvature. Therefore, if we want to represent a few road segments with a single arc-spline, the road segments must have either linearly increasing or decreasing curvatures.

However, the decision for the merging process cannot rely solely on these criteria. The Euclidean distance error metric must also be considered. This metric ensures that the combined representation maintains an acceptable level of accuracy, balancing the reduction in parameters with the need for an accurate road representation.

Algorithms 3 and 4 describe the combination algorithms for arc-splines and line segments, respectively. The algorithms are applied according to the type of the segment at hand. *max_concat* is the maximum number of road segments that are allowed to be concatenated. This number may be increased to further decrease the number of segments. However, *max_concat* is rarely the limiting factor in this algorithm.

To start the concatenation process first the curvature derivative of the current road segment is taken.

If the next segment is an arc-spline, the ratio between the current segment's curvature derivative and the next segment's curvature derivative is computed. Else the loop is broken and the algorithm continues to next arc-spline while remembering that j number of arcs are concatenated.

If the ratio is below the tolerance, then this segment may be concatenated, and it is added to *arcSplineConcatenationList*. Else the loop is broken and the algorithm continues to the next arc-spline while remembering that j number of arcs are concatenated.

i is increased by j since j number of arc segments are concatenated, and these arc-

splines should not be taken into account while trying the concatenation in the next loop.

After computing every concatenation list, a temporary arc-spline for each concatenation group is generated.

Then the error between the temporary arc-spline and the ground truth is computed.

If the error is less than $errorCfg.concatenationError$, these arc-splines are added to $mergedArcSplineSegments$ list.

Algorithm 4 starts similar to Algorithm 3. max_concat is defined as 5.

Heading value of the current line segment is saved to θ .

Then, concatenation is tried between the current segments and upcoming segments.

If the next segment is a line segment, the heading difference between the current segment's initial heading and the next segment's final heading is compared. Else the loop is broken while keeping the j value and concatenation group.

If the error is below the defined error margin, the index of the next line segment is added to the concatenation group. Else the loop is broken while keeping the j value and concatenation group.

i is increased by j since j number of line segments are concatenated, and these line segments should not be taken into account while trying the concatenation in the next loop.

Similar to Algorithm 3, for each concatenation group, a temporary line segment is generated.

Then the error between the temporary line segment and ground truth is computed.

If the error is below the predefined error margin, the concatenation group is added to the $mergedLineSegments$.

Algorithm 3 arc-spline Combination Algorithm

```
1: Input: approximatedSegments, groundTruth, errorCfg
2: Output: mergedArcSplineSegments
3: max_concat  $\leftarrow$  5
4: i  $\leftarrow$  1
5: for each arc-spline s in approximatedSegments do
6:   kappa  $\leftarrow$  s.curvatureDerivative
7:   for j  $\leftarrow$  1 to max_concat do
8:     if segment i + j is arcSpline then
9:       ratio  $\leftarrow$  kappa/next segment's curvature
10:      if ratio  $<$  errorCfg.curvatureDerivativeTolerance then
11:        Append index to arcSplineConcatenationList
12:      else
13:        break
14:      end if
15:    else
16:      break
17:    end if
18:   end for
19:   i  $\leftarrow$  i + j + 1
20: end for
21: for each concatenation group g in arcSplineConcatenationList do
22:   temporarySpline  $\leftarrow$  fitArcSpline(g)
23:   error  $\leftarrow$  computeError(temporarySpline, groundTruth(g))
24:   if error  $<$  errorCfg.concatenationError then
25:     mergedArcSplineSegments  $\leftarrow$  g
26:   end if
27: end for
```

Algorithm 4 Line Segment Combination Algorithm

```
1: Input: approximatedSegments, groundTruth, errorCfg
2: Output: mergedLineSegments
3: max_concat  $\leftarrow 5
4: i  $\leftarrow 1
5: for each line segment l in approximatedSegments do
6:    $\theta \leftarrow l.startHeading$ 
7:   for j  $\leftarrow 1$  to max_concat do
8:     if segment i + j is line segment then
9:       headingError  $\leftarrow \theta -$  next segment's final heading
10:      if headingError  $< errorCfg.headingDeviationTolerance then
11:        Append index to lineSegmentConcatenationList
12:      else
13:        break
14:      end if
15:    else
16:      break
17:    end if
18:   end for
19:   i  $\leftarrow i + j + 1
20: end for
21: for each concatenation group g in lineSegmentConcatenationList do
22:   temporaryLine  $\leftarrow fitLine(g)$ 
23:   error  $\leftarrow computeError(temporaryLine, groundTruth(g))$ 
24:   if error  $< errorCfg.concatenationError then
25:     mergedLineSegments  $\leftarrow g$ 
26:   end if
27: end for$$$$$ 
```

3.4 Generation of Additional Lanes from a Single Lane

It is possible to generate adjacent lanes if the road width and lane information are known for a particular road segment. Lane width is usually standardized in every country. Lane information refers to the position of the base lane, such as whether it is the leftmost lane, the rightmost lane, or another lane in between. With this information, it is possible to derive the representation of the other lanes. This process further improves the efficiency of our road representation, making it possible to represent all lanes using only a single lane's information.

The procedure differs for arc-spline segments and line segments. For arc-spline segments, it is possible to obtain other lanes by changing the turning radius of each arc. From Section 2.1.1.2, it is known that an arc-spline consists of several consecutive arcs.

Equation (3.1) defines the relation between radius of an arc and curvature. An arc's radius is equivalent to a vehicle's turning radius for autonomous driving. Therefore, curvature is the inverse of turning radius

$$R = \frac{1}{\kappa} \quad (3.1)$$

Figure 3.4 shows a parallel shifting process where the base arc has a turning radius (R) of 10 meters. The base arc's turning radius is increased by the lane width, which is 3.7 meters, using Equation (3.2). In this equation, the curvature is positive when the vehicle is turning left by definition. In this case, the shifted arc has a turning radius of 13.7 meters. For this example, it is convenient to think of the "Original Lane" as the leftmost lane, and we are trying to get the lane center coordinates of the right lane, which corresponds to the "Shifted Lane" in Figure 3.4. This procedure can be applied to every arc of an arc-spline.

$$R_{shifted} = R_{original} + sign(curvature) \cdot LaneWidth \quad (3.2)$$

Figure 3.5 shows the error between the shifted lane and the ground truth. The Euclidean distance error is computed by dividing the road segment to subsegments, where

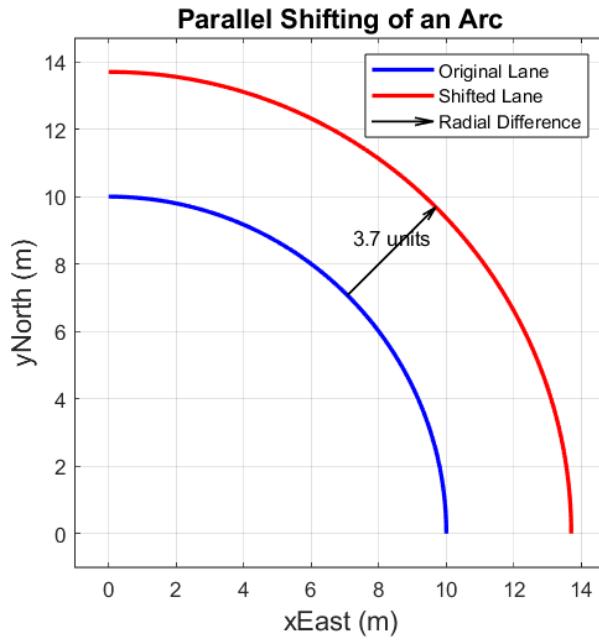


Figure 3.4: Parallel shifting of a single basic arc.

each subsegment is 10 meters long. For each subsegment, RMSE and maximum value of the error are computed.

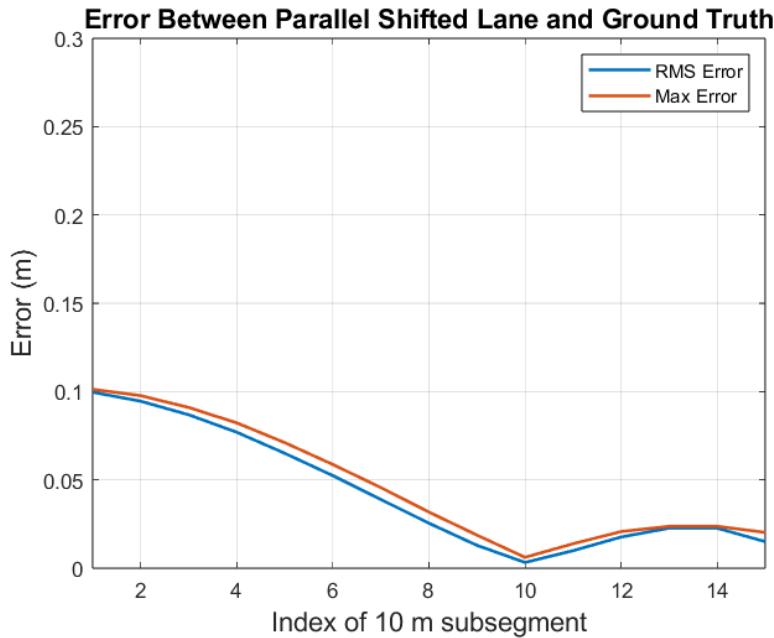


Figure 3.5: Parallel shifting error of an arc-spline representing a real road segment.

For line segments, the parallel shifting procedure is fairly simple. The line segment's

initial and final position is shifted in the direction perpendicular to the heading angle of the line segment as given in Equation (3.3). The exact direction depends on the relative position of the base lane and desired lane. Figure 3.6 demonstrates a basic example for line parallel shifting.

Given two points (x_i, y_i) and (x_f, y_f) , the new coordinates after shifting the line perpendicular to its original direction by a distance d are given by:

$$\begin{aligned} x'_i &= x_i + d \cdot \frac{-(y_f - y_i)}{\sqrt{(y_f - y_i)^2 + (x_f - x_i)^2}} \\ y'_i &= y_i + d \cdot \frac{(x_f - x_i)}{\sqrt{(y_f - y_i)^2 + (x_f - x_i)^2}} \\ x'_f &= x_f + d \cdot \frac{-(y_f - y_i)}{\sqrt{(y_f - y_i)^2 + (x_f - x_i)^2}} \\ y'_f &= y_f + d \cdot \frac{(x_f - x_i)}{\sqrt{(y_f - y_i)^2 + (x_f - x_i)^2}} \end{aligned} \quad (3.3)$$

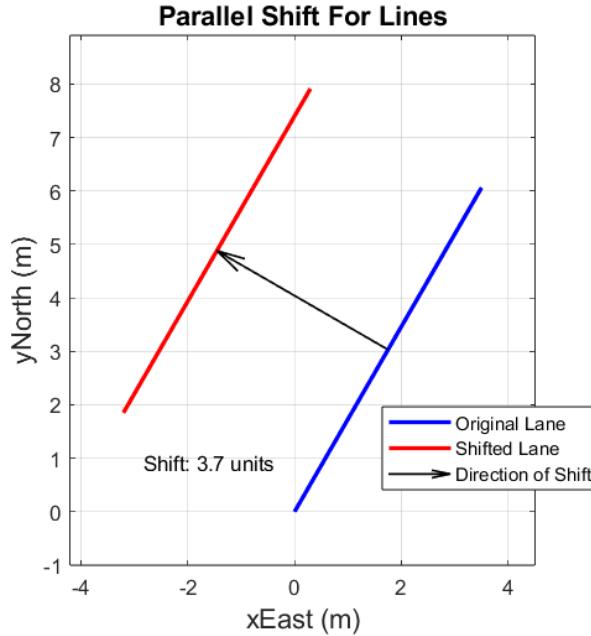


Figure 3.6: Parallel shifting of a single basic arc.

Figure 3.7 shows the error between the shifted lane and ground truth for an example road segment. The Euclidian distance error is computed by dividing the road segment to subsegments where each subsegment is 10 meters long. For each subsegment, RMSE and maximum value of the error are computed.

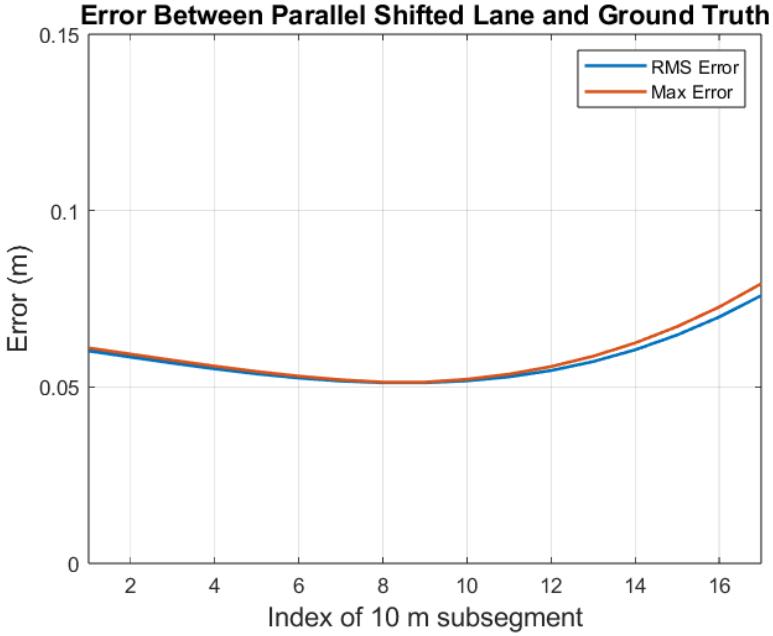


Figure 3.7: Parallel shifting error of a line segment representing a real road segment.

3.5 Evaluation Metrics and Results

Autobahn 38 in Germany is selected to evaluate the proposed method. Data is downloaded from HERE maps, and clothoids are fitted between the waypoints as described in Section 3.1. Satellite imagery together with clothoids is given in Figure 3.8. For trajectory planning applications, a road segment from Autobahn 38 is picked.

Additionally, a Section of Autobahn 1 is also approximated. Satellite imagery together with clothoids is given in Figure 3.9.

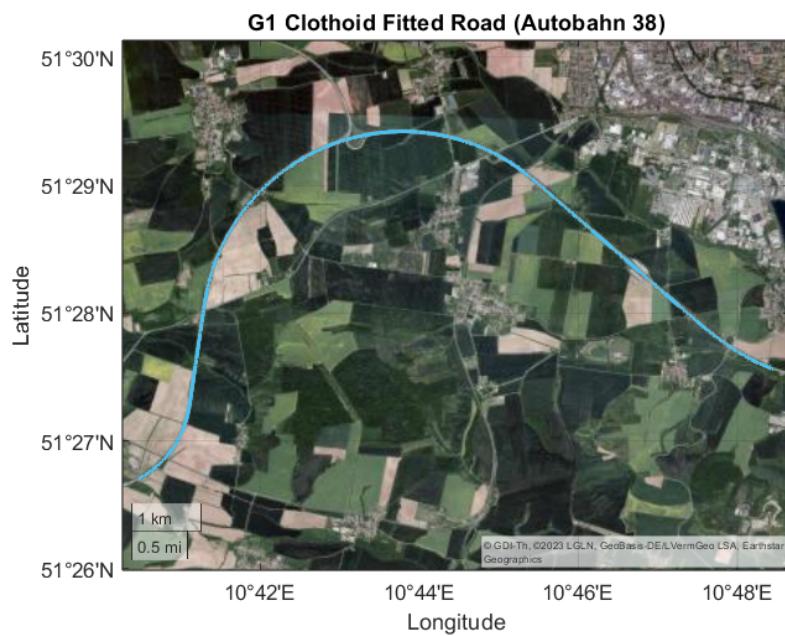


Figure 3.8: Satellite image overlaid with G1 fitted clothoid road segments (plotted in light blue) on Autobahn 38.

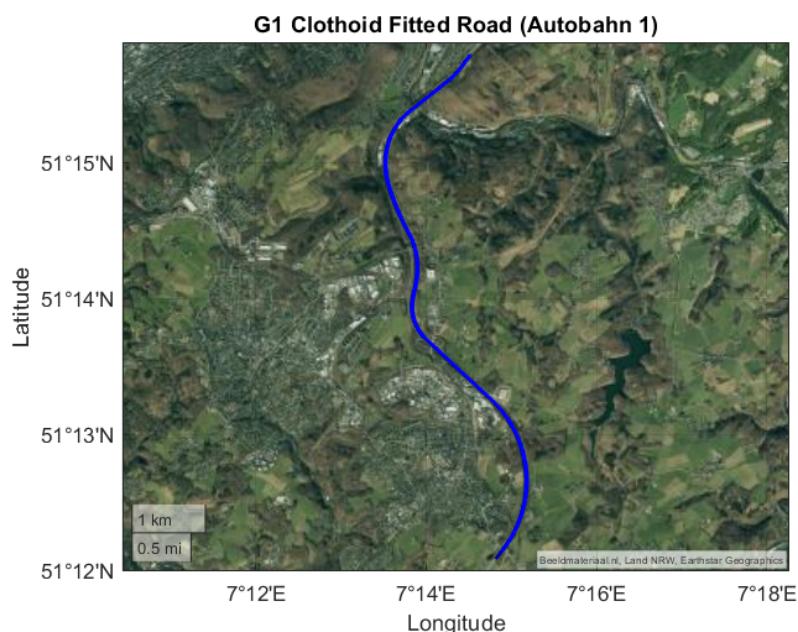


Figure 3.9: Satellite image overlaid with G1 fitted clothoid road segments (plotted in dark blue) on Autobahn 1.

3.5.1 RMSE Spatial Coordinates

To evaluate the accuracy of the approximations, RMSE is computed. RMSE is useful since it is possible to see the overall performance of the approximation. Road segments have lengths varying from 20 meters to 400 meters; therefore, it would not be meaningful to directly compute the RMSE for each segment. We might need to inspect some parts of segments to observe the approximation in more detail. Moreover, the RMSE for long sequential data may hide some error peaks since the overall error would be low. Therefore, RMSE is computed for 10 meter long subsegments. In particular, each road segment is divided into 10-meter-long subsegments, and RMSE is computed for these subsegments.

Figure 3.10 shows the RMSE plot over an approximately 370-meter-long segment. For every 10-meter-long segment, the RMSE is computed. There is a data point for each centimeter for line segments. The RMSE reaches 9 centimeters at most which would be accurate enough for autonomous driving.

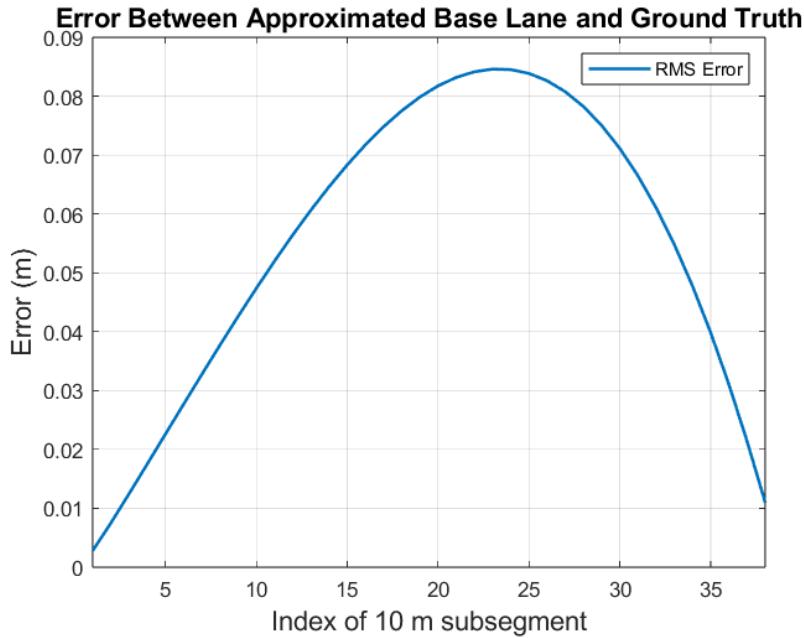


Figure 3.10: RMSE of a line approximated segment

Figure 3.11 shows the maximum error plot over the same 370-meter-long line segment. Maximum error is as well at acceptable levels.

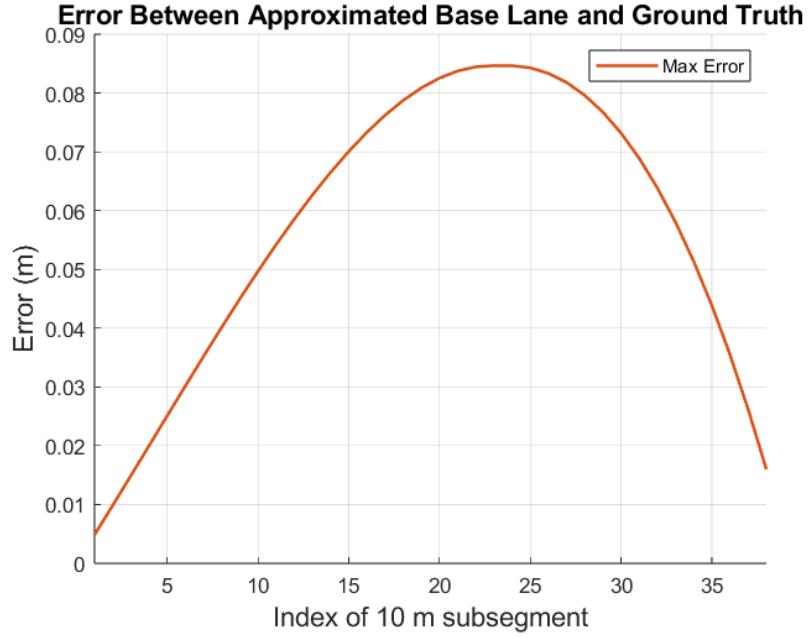


Figure 3.11: Maximum error of a line approximated segment.

Figure 3.12 shows the RMSE plot over an approximately 150-meter-long arc-spline segment. There is a data point for each centimeter for arc-spline segments. The order of arc-spline approximation for this segment is 6, which means that this segment's curvature rate is relatively high for a highway.

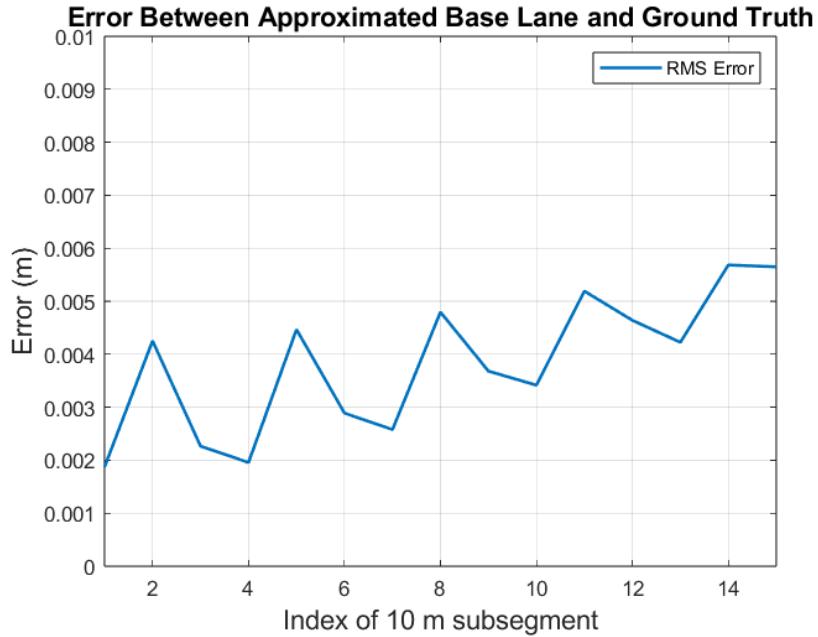


Figure 3.12: RMSE of a arc-spline approximated segment.

Figure 3.13 shows the maximum error plot over the same 150-meter-long arc-spline segment. There is a data point for each centimeter for arc-spline segments. The values of the maximum error is close to RMS errors which is at acceptable levels.

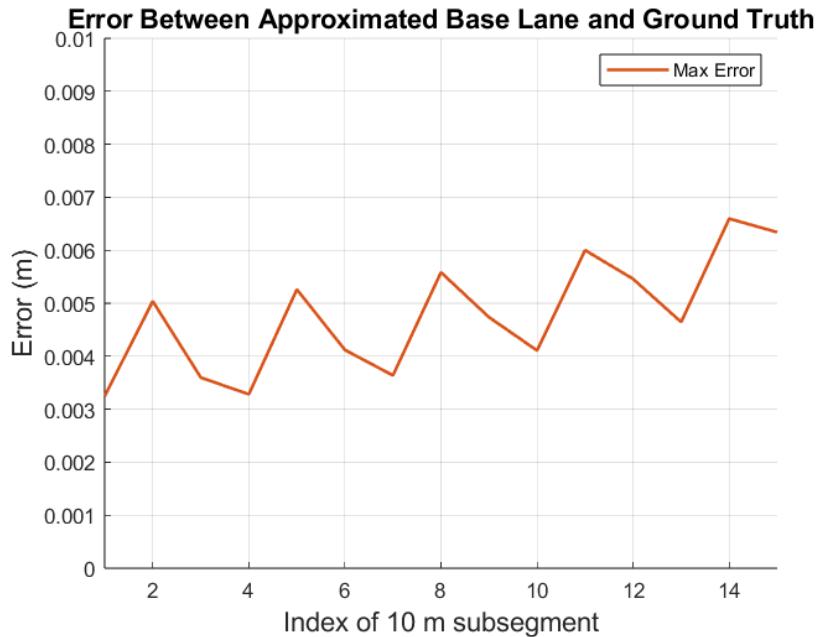


Figure 3.13: Maximum error of a arc-spline approximated segment.

Figure 3.14 shows the RMSE plot over an approximately 12 500 meters long road. For each segment, the RMSE is computed. As mentioned before, the length of each road segment can vary from 20 meters to 400 meters. In this plot, segments are classified as either arc-spline or line segments. The RMSE over an entire segment never exceeds 0.1 meters. This result is promising given that there are 323 segments, and this road includes every type of section a highway would have, such as curves and straight parts.

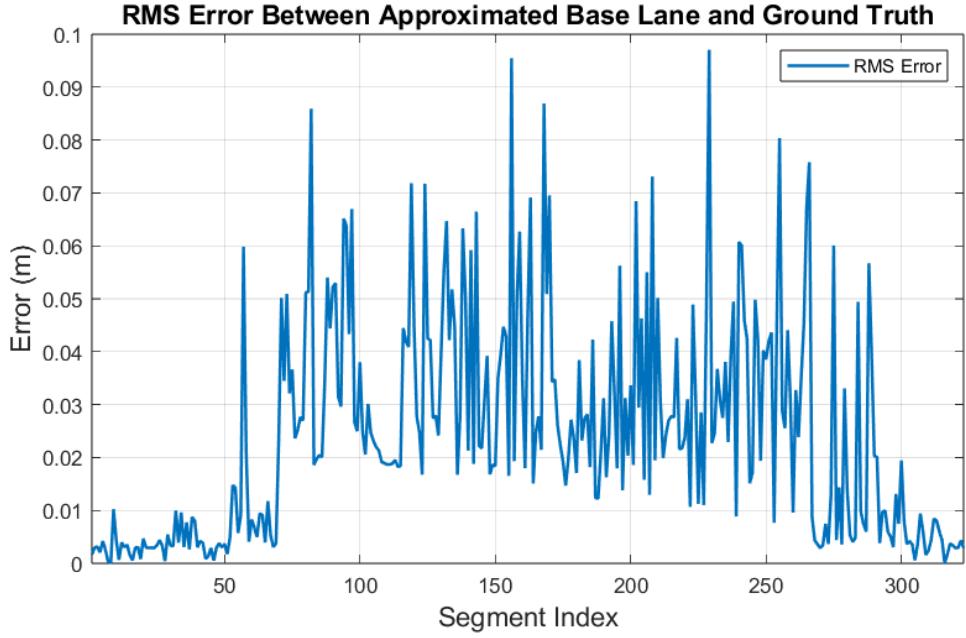


Figure 3.14: RMSE for a 323 segment Autobahn 38 (overall length around 12.5 km).

Figure 3.15 shows the RMSE and average curvature on the same plot. As mentioned in Section 2.1.2, every waypoint has the curvature data stored in it, and road segments are the structures that connect waypoints. To define a curvature for a road segment, the average curvature values at both ends are computed. High-curvature parts of the road seem to have low RMSE. This could be because the algorithm uses higher-order arc-spline approximation for the curved sections of the road.

Figure 3.16 shows the RMSE and segment length on the same plot. It is observed that the error is not correlated with the segment length. Long segments do not have significantly high error values. Similarly, short segments do not have low error values.

The memory usage of arc-spline segments is independent of the order of the arc-spline which is 25 bytes for a single arc-spline.

Line segments are simpler than arc-splines, they require only the starting and end point of the segment with single precision floating point for each of them. Therefore each line segment takes 16 bytes of memory.

As a result, Autobahn 38 section given in Figure 3.8 takes 5081 bytes in memory. This road segment is 13400 meters long. Considering the length of the highway section,

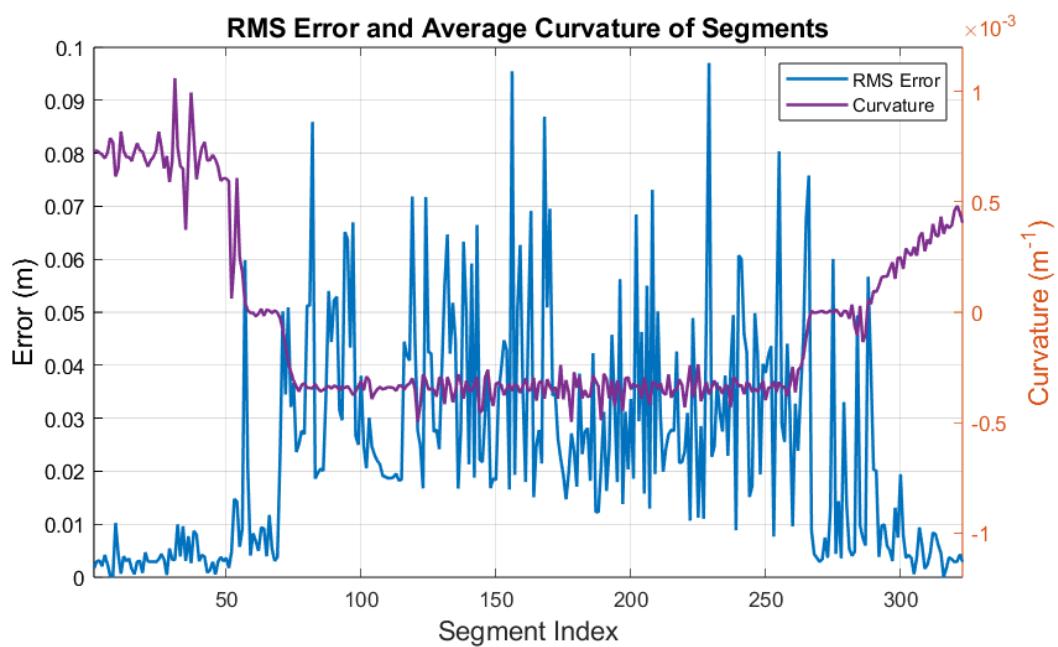


Figure 3.15: RMSE and average curvature for 323 segment Autobahn 38

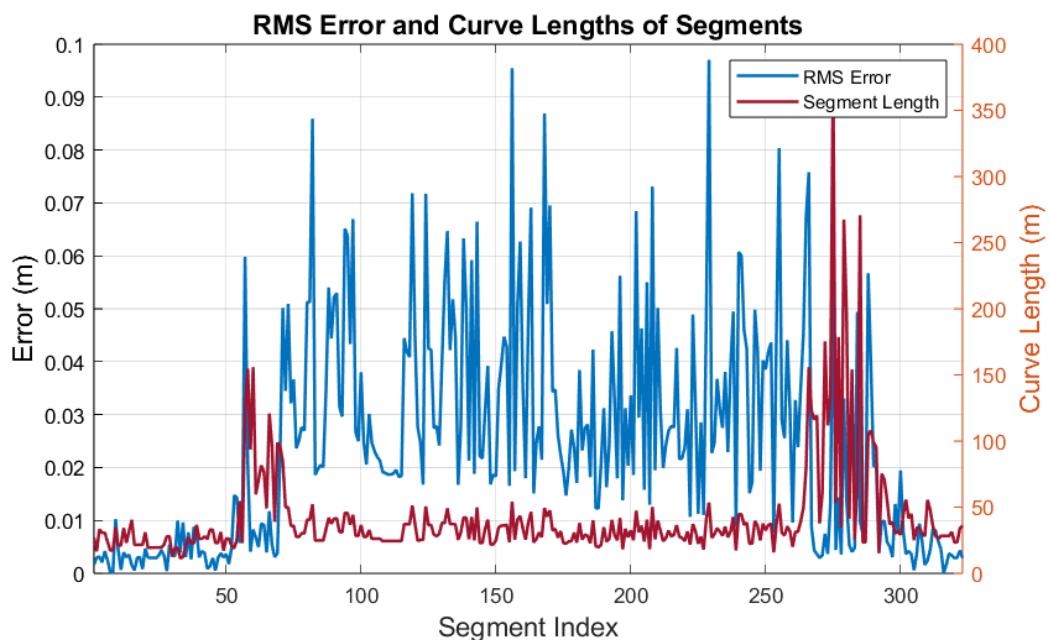


Figure 3.16: RMSE and segment length for 323 segment Autobahn 38

memory usage is significantly low.

Figure 3.17 shows the RMSE plot over an approximately 8 000 meters long road. For each segment, the RMSE is computed. As mentioned before, the length of each road segment can vary from 15 meters to 150 meters. In this plot, segments are classified as either arc-spline or line segments. The RMSE over an entire segment never exceeds 0.1 meters.

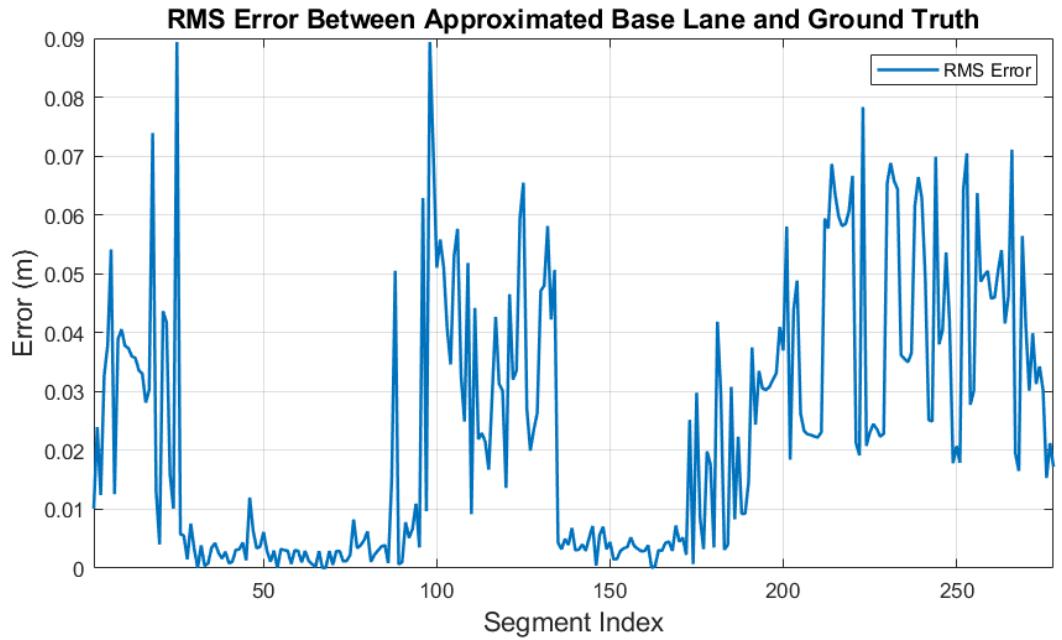


Figure 3.17: RMSE for 278 segment Autobahn 1 (overall length around 8 km)

Figure 3.18 shows the RMSE and average curvature of each segment on the same plot for Autobahn 1. Similar to previous result, high curvature parts of the road seem to have low RMSE.

Figure 3.19 shows the RMSE and segment length on the same plot. Similar to previous result it is observed that the error is not correlated with the segment length.

Autobahn 1 section given in Figure 3.9 takes 4649 bytes in memory. This road segment is 8000 meters long.

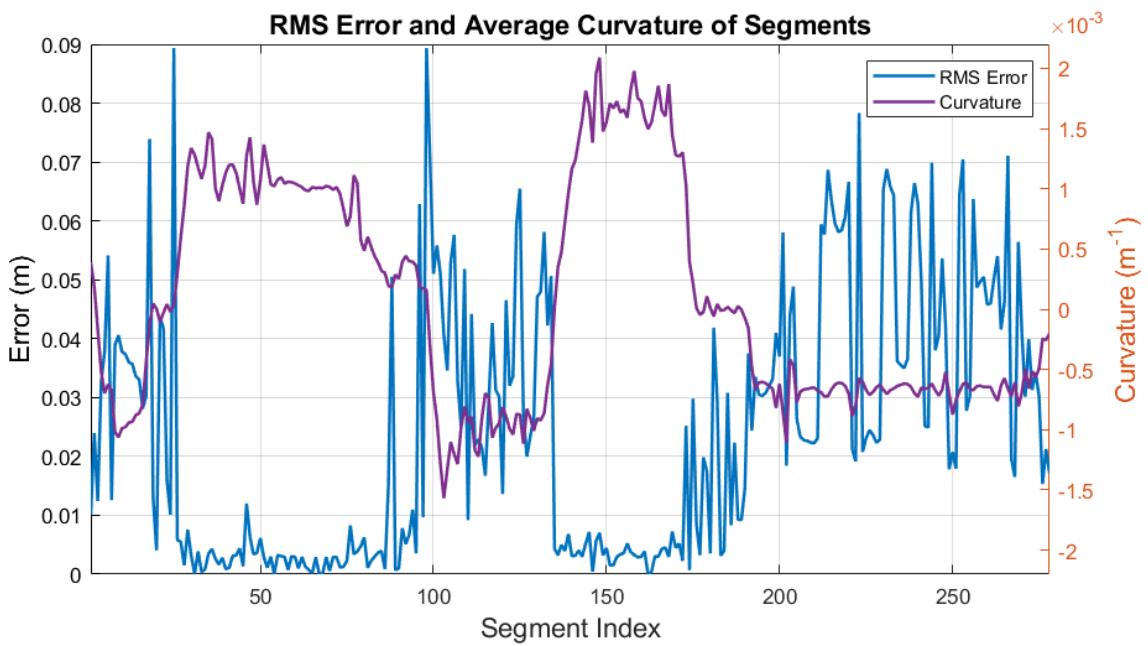


Figure 3.18: RMSE and average curvature for 278 segment Autobahn 1

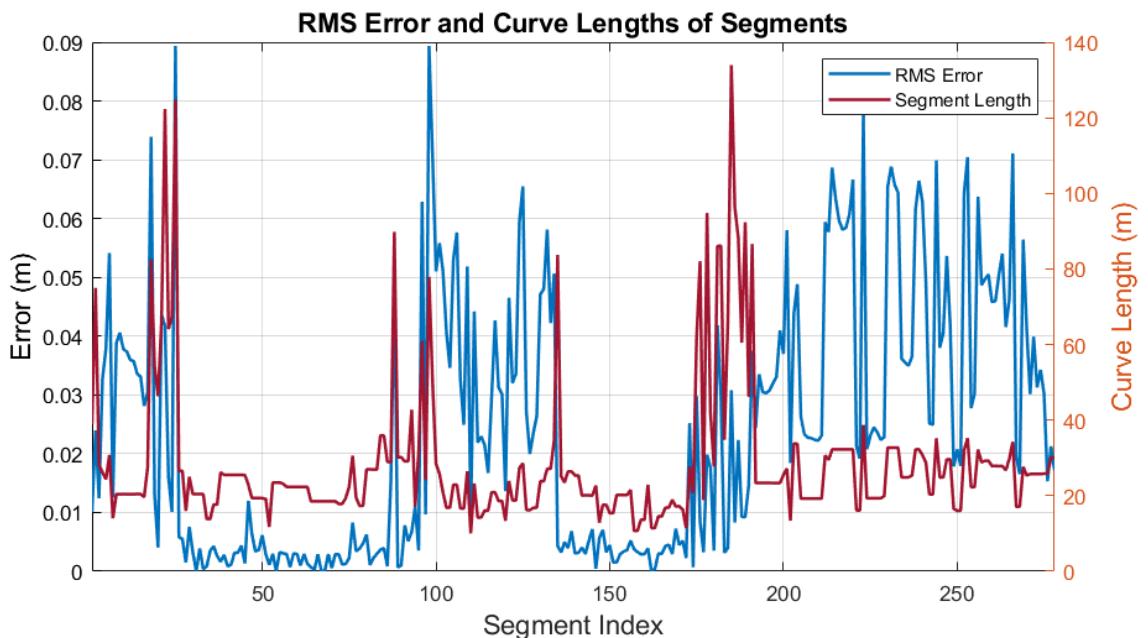


Figure 3.19: RMSE and segment length for 278 segment Autobahn 1

3.5.2 Maximum Error In Spatial Coordinates

To further evaluate the practicality of the approximations, maximum error is computed. maximum error is an important metric since it means that if a vehicle follows the approximated road, it will be guaranteed to not leave the lane. Given that road segments can range in length from 20 meters to 400 meters, directly calculating the maximum error for an entire segment is impractical. It may be necessary to examine specific parts of segments to observe the approximation closely. Additionally, calculating the maximum error for long data sequences might obscure other error peaks, as it only highlights the highest value. Therefore, the maximum error is computed for 10-meter-long subsegments, i.e., each segment is divided into 10-meter-long subsegments, and the maximum error is computed for these subsegments.

Figure 3.20 shows the overall road with a magnified segment. The error is in the order of few centimeters as expected.

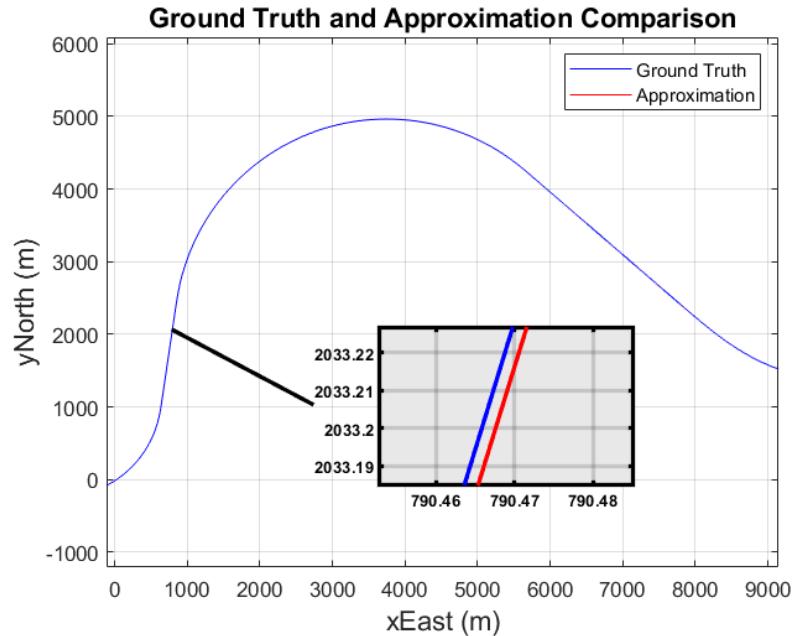


Figure 3.20: General view of the approximation compared to ground truth

Figure 3.21 shows a maximum error plot over a ~370 meters long segment. For every 10-meter-long segment, the maximum error is computed.

Figure 3.22 shows an maximum error plot over a ~150 meters long segment. For

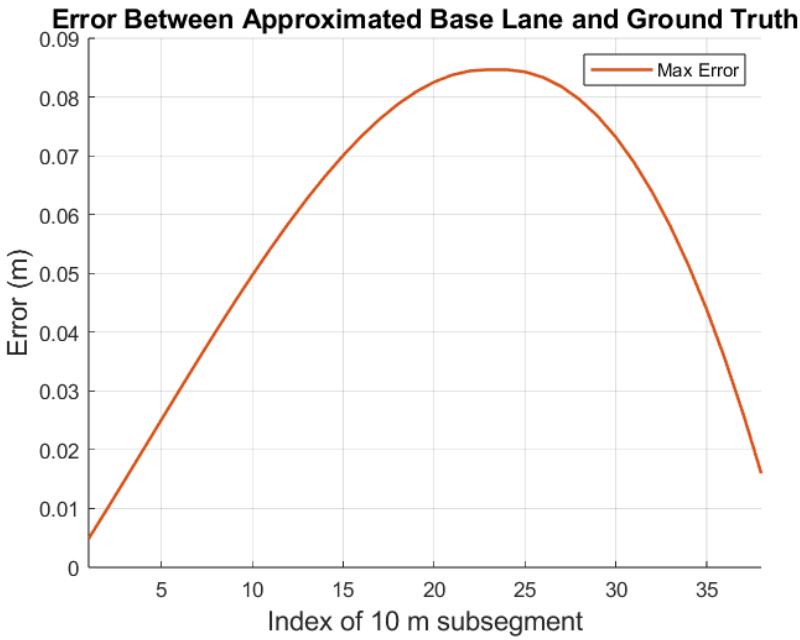


Figure 3.21: Maximum error of a line approximated segment

every 10-meter-long segment, the maximum error is computed.

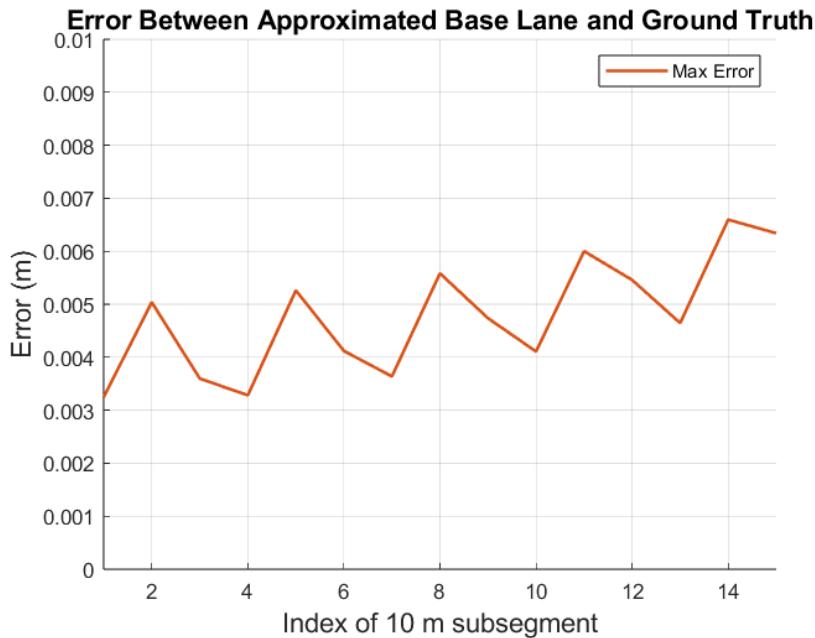


Figure 3.22: Maximum error of a arc-spline approximated segment

Figure 3.23 shows a maximum error plot over an approximately 12 500 meters long road. For each segment, maximum error is computed. As mentioned before, the

length of each road may vary from 20 meters to 400 meters. The segments may be classified as arc-spline or line segments in this plot.

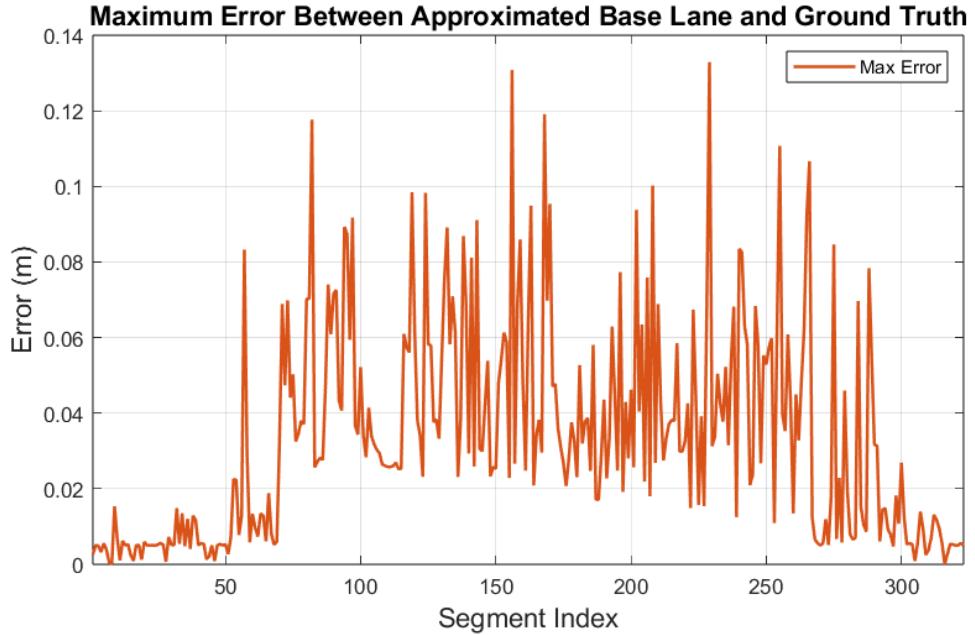


Figure 3.23: Maximum error values for 323 segment Autobahn 38

Figure 3.24 shows an maximum error and average curvature on the same plot. As mentioned in 2.1.2 every waypoint has the curvature data stored in it and road segments are the structures that connect waypoints. To define a curvature for a road segment, the average of curvature values at both ends of a segment is computed.

Figure 3.25 shows the maximum error and segment length on the same plot. It is observed that the error is not correlated with the segment length. Long segments do not have significantly high error values. Similarly short segments do not have low error values.

Figure 3.26 shows a maximum error plot over an approximately 8 000 meters long road. For each segment, maximum error is computed. As mentioned before, the length of each road may vary from 20 meters to 400 meters. The segments may be classified as arc-spline or line segments in this plot.

Figure 3.27 shows an maximum error and average curvature on the same plot.

Figure 3.28 shows the maximum error and segment length on the same plot. It is

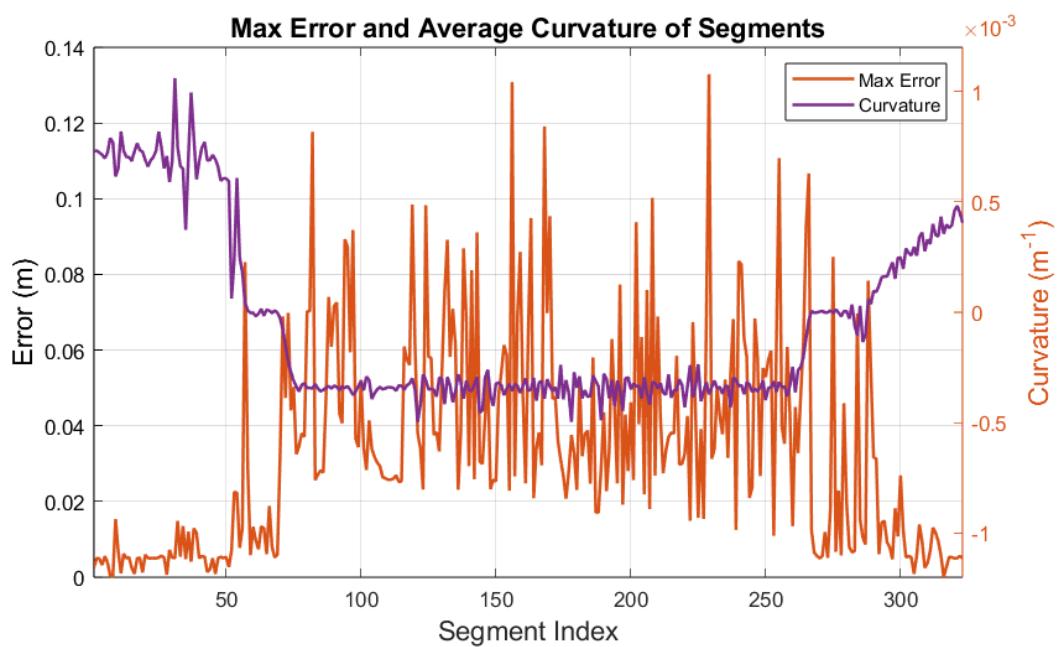


Figure 3.24: Maximum error and average curvature for 323 segment Autobahn 38

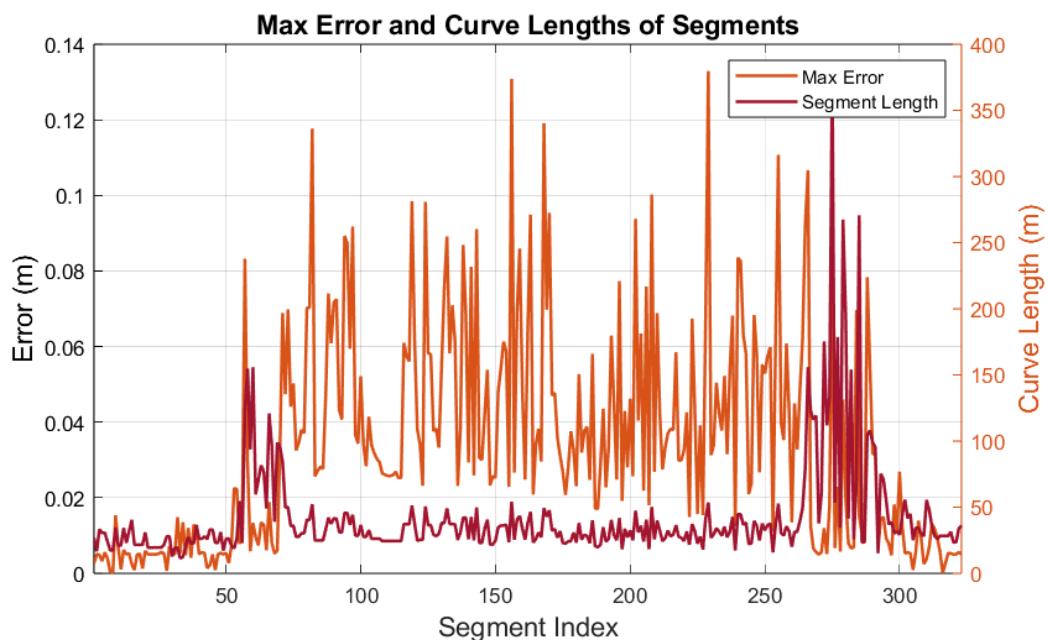


Figure 3.25: Maximum error and segment length for 323 segment Autobahn 38

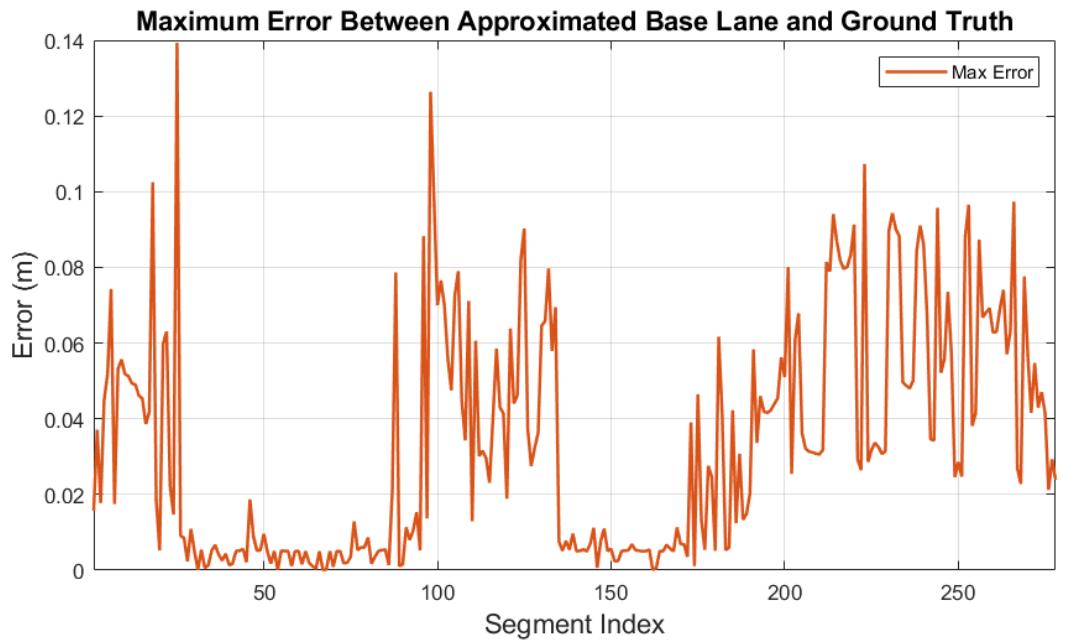


Figure 3.26: Maximum error values for 278 segment Autobahn 1

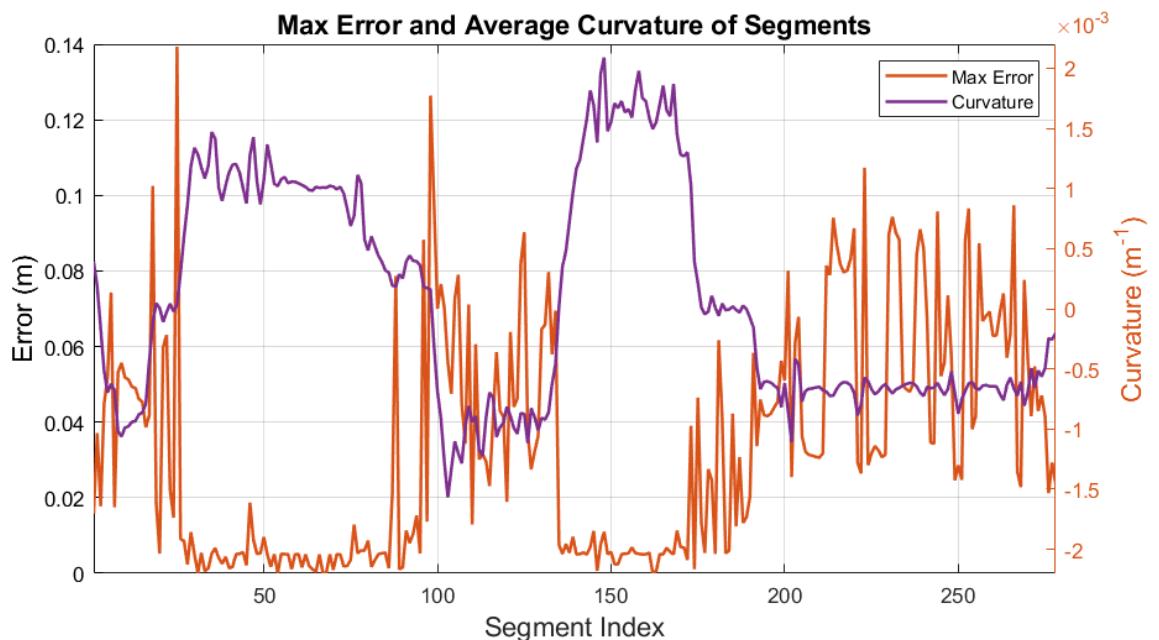


Figure 3.27: Maximum error and average curvature 278 segment Autobahn 1

observed that the error is not correlated with the segment length.

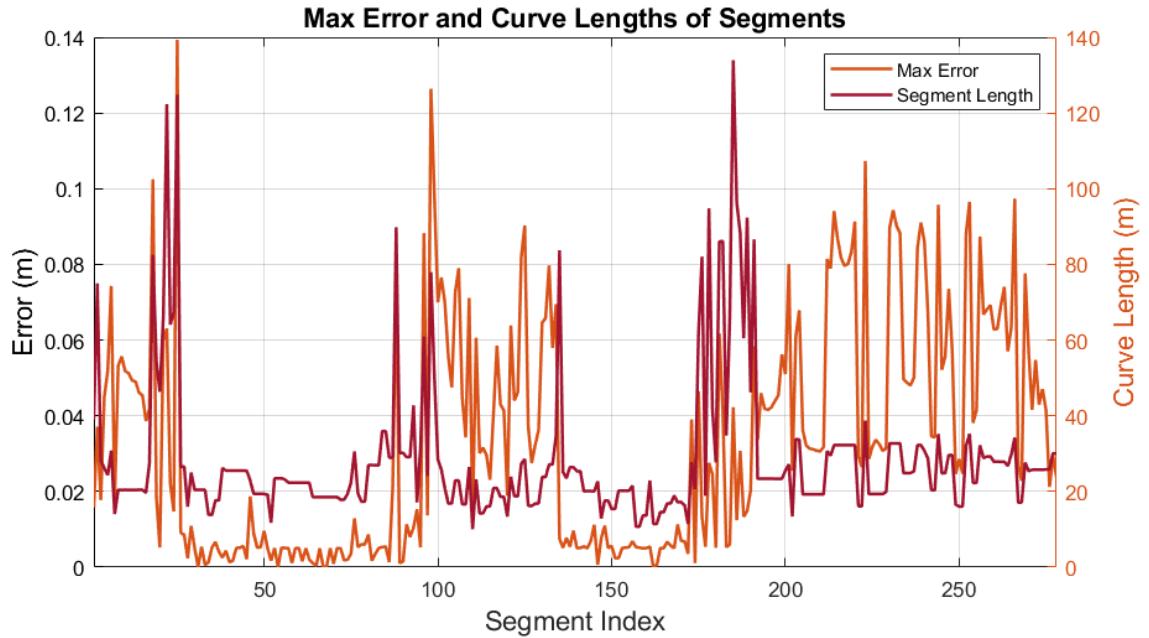


Figure 3.28: Maximum error and segment length for 278 segment Autobahn 1

3.5.3 Number of Segments

As mentioned in 3.3 segments may be merged on some error configurations. This would result in reduced number of segments and parameters which would decrease the memory usage of the offline map.

Table 3.3 shows the number of segments before and after combination operation. As seen on the table, both the number of arc-splines and line segments decrease. Line segments are combined more than arc-splines. This combination process makes the algorithm even more memory efficient.

Table 3.3: Segments before and after combination operation for Autobahn 38

	Number of arc-splines	Number of Line Segments	Total Segments	Memory Usage (bytes)
Before merge	86	237	323	5942
After merge	81	191	277	5081

Table 3.4 shows the number of segments before and after combination operation for Autobahn 1. As seen on the table, both the number of arc-splines and line segments decrease.

Table 3.4: Segments before and after combination operation for Autobahn 1

	Number of arc-splines	Number of Line Segments	Total Segments	Memory Usage (bytes)
Before merge	113	165	277	5465
After merge	97	139	236	4649

CHAPTER 4

TRAJECTORY GENERATION METHODOLOGY

This Chapter presents the methodologies employed for trajectory generation, focusing on the comparative analysis of Bézier curves and arc-splines. Bézier curves, while frequently utilized for their smoothness and controllability, face limitations in precise control over complex paths and can incur high computational costs in real-time applications. In contrast, arc-splines, although less commonly used due to the difficulty in adjusting endpoints, offer significant advantages in terms of vehicle dynamics suitability. This Chapter introduces a novel algorithm for trajectory planning using arc-splines, demonstrating their potential to enhance trajectory stability and control. The efficacy of the proposed method in generating accurate and reliable trajectories is examined by comparative metrics and simulations.

4.1 Bézier Trajectory Generation

Bézier curves are frequently used for trajectory planning due to their smooth and controllable nature; however, they also have certain disadvantages, such as difficulty in achieving precise control over complex paths and the potential for high computational cost in real-time applications. Bézier curves require the initial pose of the ego vehicle and a final pose to generate a trajectory. In our case, the initial pose is already given. We can give any point along a road segment as the final pose since we have already represented the whole road with either arc-splines or line segments. Thanks to this road representation method we can compute the required pose for a vehicle at any point on the road segment.

Bézier curves have the advantage of allowing precise control over the start and end

points, as well as their tangents and curvature. However, the main disadvantage is that the path between these points can be unpredictable. This means we need to generate many candidate Bézier curves for collision checking with the lane boundaries because a single curve might cross the lane boundaries. By having multiple candidates, we can find a path that keeps the vehicle on its lane. Additionally, generating many candidates allows us to choose a path with the desired curvature characteristics, which helps enable the vehicle to follow the path comfortably. This method balances safety and computational efficiency, allowing us to maintain real-time performance while exploring different trajectory options. Thus, having a variety of Bézier curve candidates is essential for reliable trajectory planning in autonomous vehicles.

4.1.1 Generating a Sufficient Number of Candidate Curves

As mentioned in Section 2.1.3 quintic Bezier curves have 6 control points. It is not possible to change the position of the first and last control points since they directly affect the initial and final points of the curve. However, it is possible to slightly vary P_1 , P_2 , P_3 , and P_4 while maintaining the initial and final pose [34].

Control point computations are performed according to Equation (4.1), (4.2), (4.3), and (4.4). In these Equations, t_0 and t_f correspond to the initial and final velocity vectors, respectively. Similarly, a_0 and a_f vectors correspond to initial and final acceleration vectors, respectively. To have a variety of Bézier curves we need to have a variety of control points. To achieve this we can change the magnitude of velocity and acceleration vectors. It should be noted that, the directions of velocity and acceleration vectors are directly related to pose hence the directions of these vectors cannot be altered.

$$P_1 = P_0 + \frac{1}{5} \vec{t}_0 \quad (4.1)$$

$$P_2 = \frac{1}{20} \vec{a}_0 + 2P_1 - P_0 \quad (4.2)$$

$$P_3 = \frac{1}{20} \vec{a}_f + 2P_4 - P_0 \quad (4.3)$$

$$P_4 = P_5 - \frac{1}{5} \vec{t}_f - P_5 \quad (4.4)$$

Equation (4.5) gives the formulation for \vec{t}_0 candidate computation. d is the distance between the starting point and the destination point, m_t is the tangential multiplier, and \vec{t}_0^u is the unit vector in the direction of the initial heading.

m_t is a design parameter for Bézier curve candidates. Possible values of m_t are given in Equation (4.6). m_t^{\min} and m_t^{\max} are design parameters that affect the shape of Bézier curves, and N_t is the number of candidates for each tangential vector.

$$\vec{t}_0 = d \cdot m_{tn} \cdot \vec{t}_0^u \quad (4.5)$$

$$\forall m_{tn} \in [m_t^{\min}, m_t^{\max}] \quad n = 1, \dots, N_t \quad (4.6)$$

Similar to \vec{t}_0 , Equation (4.7) gives the formulation for \vec{t}_f candidate computation. \vec{t}_f^u is the unit vector in the direction of the destination heading. For both \vec{t}_0 and \vec{t}_f same m_{tn} candidates are used.

$$\vec{t}_f = d \cdot m_{tn} \cdot \vec{t}_f^u \quad (4.7)$$

Equation (4.8) gives the formulation for \vec{a}_0 candidate computation. m_κ^n is the acceleration multiplier and \vec{t}_0^n is the unit vector in the direction of initial curvature. Initial curvature direction is the vector connecting the vehicle's current point to turning center. m_κ^n is a design parameter for Bézier curve candidates. Possible values are given in Equation (4.9). m_k^{\min} and m_k^{\max} are design parameters that affect the shape of Bézier curves, and N_k is the number of candidates for each acceleration vector.

$$\vec{a}_0 = \frac{d}{5} \cdot m_\kappa^n \cdot \vec{t}_0^n + d^2 \cdot \kappa_i \cdot \vec{t}_0^n \quad (4.8)$$

$$\forall m_\kappa^n \in [m_k^{\min}, m_k^{\max}] \quad n = 1, \dots, N_k \quad (4.9)$$

Similar to \vec{a}_0 , Equation (4.10) gives the formulation for \vec{a}_f candidate computation. \vec{t}_f^n is the unit vector in the direction of destination curvature. For both \vec{a}_0 and \vec{a}_f same m_κ^n candidates are used.

$$\vec{a}_f = \frac{d}{5} \cdot m_\kappa^n \cdot \vec{t}_f^n + d^2 \cdot \kappa_f \cdot \vec{t}_f^n \quad (4.10)$$

Table 4.1 gives the configuration parameters used in this thesis. The values of N_t and N_k define the number of curve candidates which is 225 for these values. The limits of tangential and acceleration multipliers are picked to be low since this research involves highway trajectory planning and trajectories on highways cannot tolerate high curvatures.

Table 4.1: Configuration Parameters for Candidate Bézier Curve Generation

Parameter	Value
N_t	5
N_k	3
m_t^{\min}	0.3
m_t^{\max}	1.7
m_k^{\min}	0
m_k^{\max}	5

For better understanding of these parameters, a simulation is given with $B = [(0, 0), (15, 10), 0, -45^\circ, 0, 0.01]$ and some control points are varied while keeping the others same.

Figure 4.1 demonstrates the effect of varying P_1 . As the magnitude of t_0 increases, the trajectory moves in the direction of initial tangent more. This parameter can be associated with initial velocity of the ego vehicle.

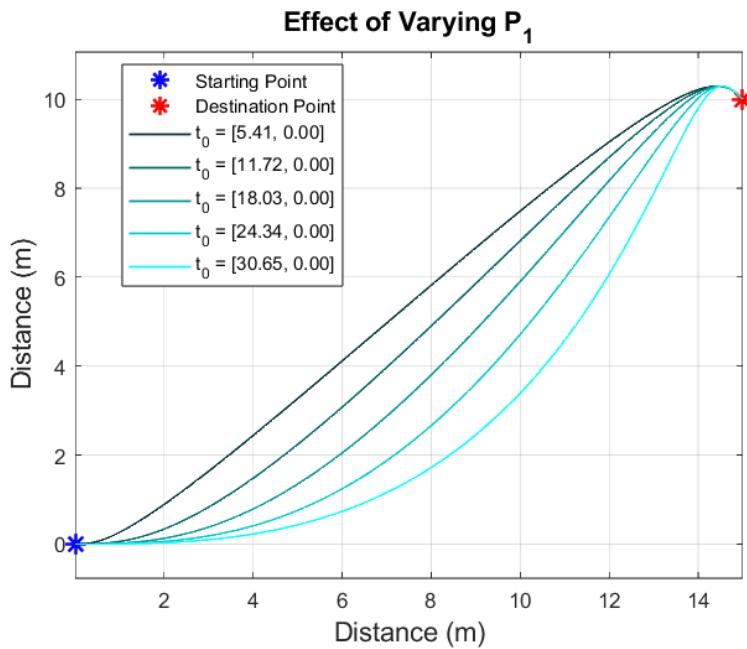


Figure 4.1: Effect of varying P_1 while keeping the other parameters same.

Figure 4.2 demonstrates the effect of varying P_2 . As the magnitude of a_0 increases,

the shape of the trajectory slightly changes close to the start point. The effect is minor since the (4.8) has a damping factor for \vec{a}_0 . This parameter can be associated with the initial curvature of the ego vehicle.

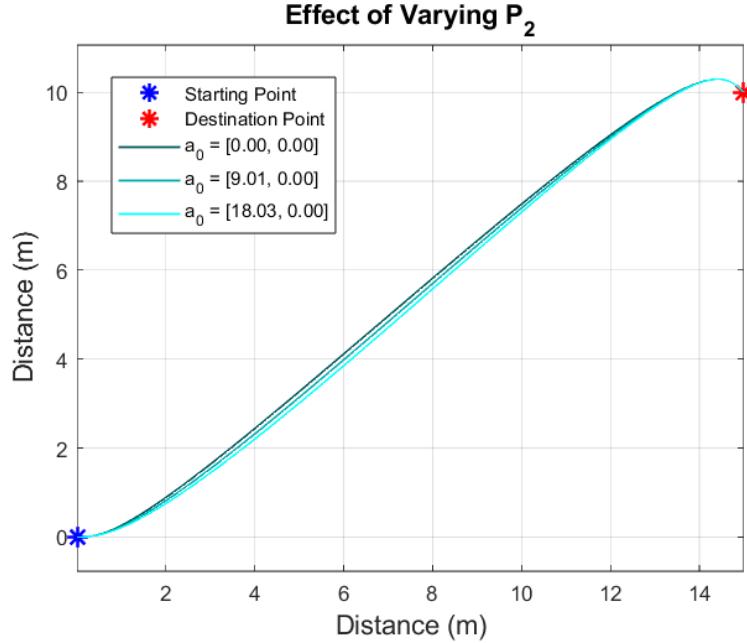


Figure 4.2: Effect of varying P_2 while keeping the other parameters same.

Figure 4.3 demonstrates the effect of varying P_3 . As the magnitude of a_f increases, the trajectory slightly changes towards the end of the trajectory. The effect is minor since the (4.10) has a damping factor for \vec{a}_f .

Figure 4.4 demonstrates the effect of varying P_4 . As the magnitude of t_f increases, the trajectory moves in the direction of destination tangent more. This parameter can be associated with destination velocity.

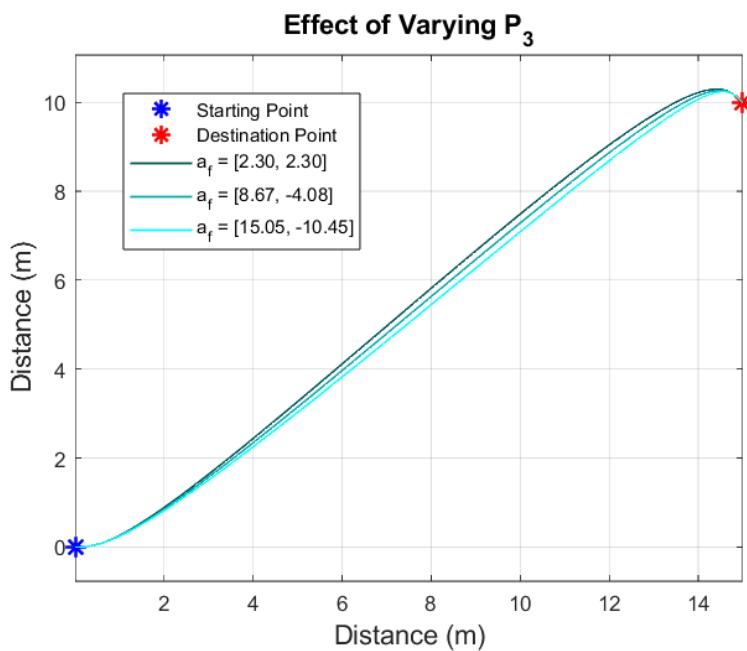


Figure 4.3: Effect of varying P₃ while keeping the other parameters same

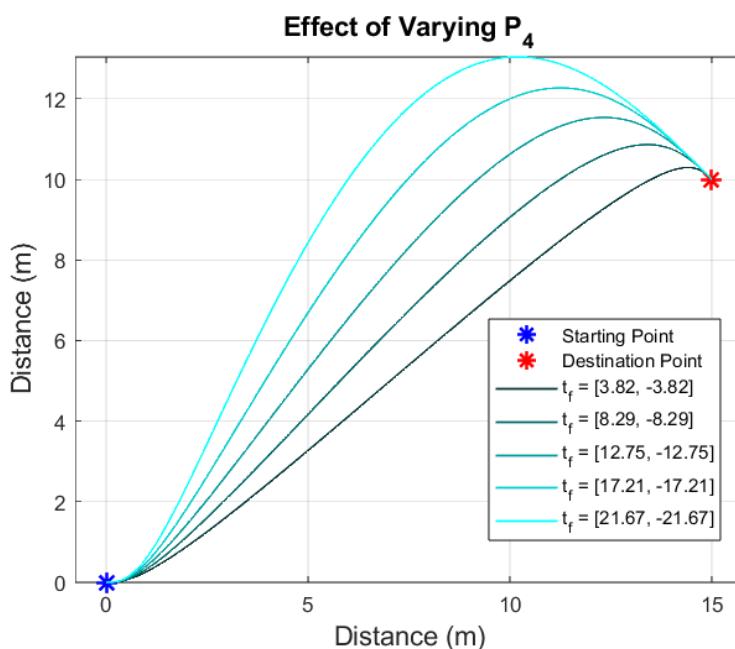


Figure 4.4: Effect of varying P₄ while keeping the other parameters same

4.1.2 Identifying the Best Trajectory

As mentioned in Section 4.1.1 there are many Bézier curves that satisfy the initial and final pose conditions. We need to find a method to pick the best of them based on some predefined conditions.

To accomplish this, the cost function in Equation (4.11) is introduced similar to [34]. In this Equation, s_0 and s_f indicate the beginning and end of the trajectory. $\kappa(s)$ is the curvature at arc length s . $w_{\ddot{\kappa}}$ is a regularization term for $\ddot{\kappa}(s)$.

$$J_p = \int_{s_0}^{s_f} \dot{\kappa}(s)^2 + w_{\ddot{\kappa}} \ddot{\kappa}(s)^2 ds \quad (4.11)$$

This cost function penalizes the trajectories that have high curvature rate and jerk. Note that [34] also uses the curve length to normalize the cost function. However, our study observed that the length is not a significant parameter when it comes to highway trajectory planning. Curvature rate and jerk are more significant since these directly affect passenger comfort.

4.2 Arc-spline Trajectory Generation

Arc-splines and clothoids are not frequently used for trajectory planning due to the difficulty in adjusting the end point. However, we have developed a new method for trajectory planning using arc-splines. A significant advantage of using arc-splines and clothoids is their suitability for vehicle dynamics, as these splines consist of consecutive arcs, making it easy to determine the steering input required to follow the trajectory. Arc-spline trajectories require an initial pose and a clothoid road as input, resulting in a series of waypoints that can be connected with arc-splines.

To compute an arc-spline for a vehicle that is outside the road centerline, it is necessary to correct the heading, curvature, and position. Our method is divided into two parts: the first part describes the heading and curvature correction algorithm, while the second part details the position correction algorithm.

Figure 4.5 shows an example of a vehicle that is away from the road centerline. The

position error is 0.55 meters, and the heading error is 2° . Curvature error is not easy to display with a figure since it is related to the rate of change of the heading angle.

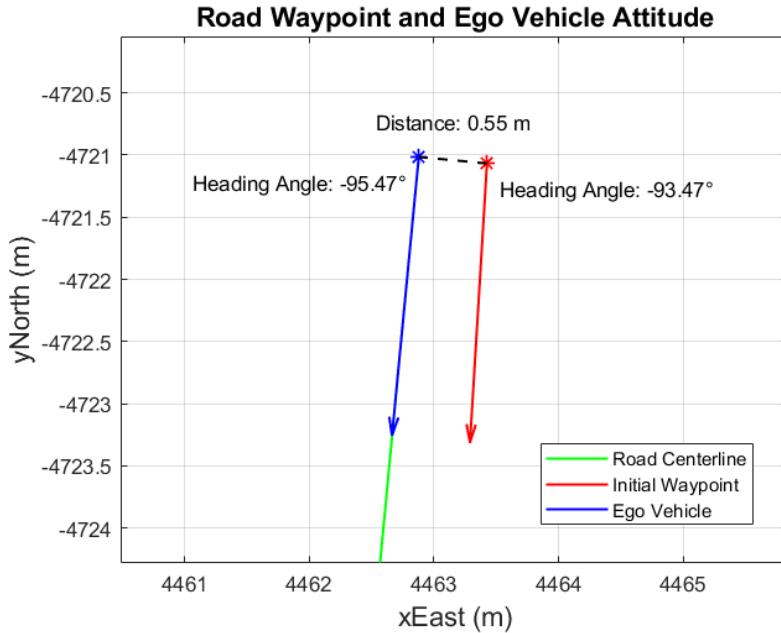


Figure 4.5: An example of position, heading, and curvature error.

4.2.1 Heading and Curvature Correction (HCC) Maneuver

In this part heading and curvature correction maneuvers will be described. Let's assume that a vehicle is exactly on the road centerline; however, it has heading and curvature error. The initial heading and curvature error can be compensated by cleverly computing the required curvature rate along some length. There are 4 cases for correction.

Figure 4.6 shows the curvature of a road segment where k_{ri} , k_{rf} , k_i and L stand for initial road curvature, final road curvature, initial vehicle curvature and arc length respectively. Since the vehicle starts from k_i , at the end of the road, it is desired to match the road's curvature. Therefore, we can draw a line connecting k_i to k_{rf} to create a clothoid-like maneuver. The arc length L is free if we only want to correct for curvature but we also desire to correct the heading. The heading error at the initial point is known, and the area A in Figure 4.6 gives the heading difference caused by this maneuver. If we can make the heading error equal to heading difference over

some arc length, it is possible to correct for both heading and curvature with a single maneuver.

Case 1 is defined with the Equation set (4.12).

$$\begin{aligned}\theta_{error} &= \theta_i - \theta_{ri} > 0 \\ \kappa_{error} &= \kappa_i - \kappa_{ri} < 0\end{aligned}\tag{4.12}$$

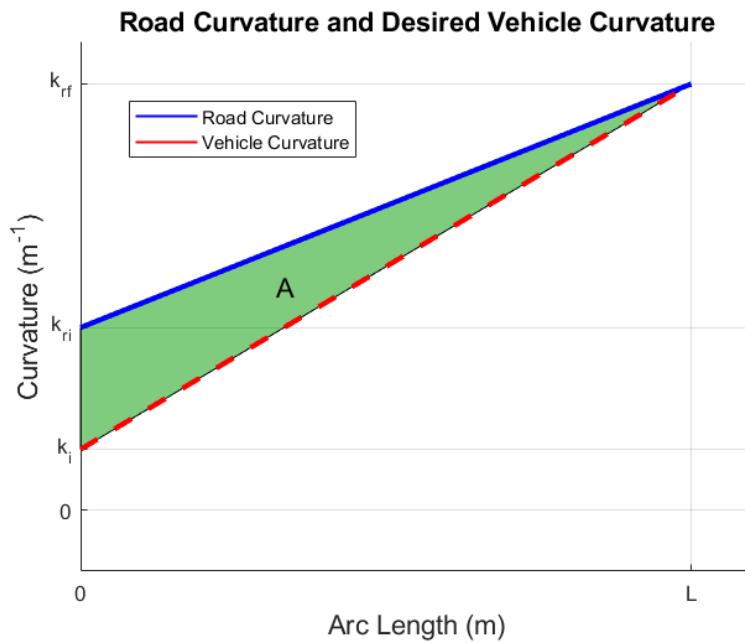


Figure 4.6: Curvature plot of a road segment with a heading and curvature correcting maneuver curvature plot for case 1.

Figure 4.7 shows the curvature of a road segment where k_{int} and l_1 are intermediate curvature and first maneuver length, respectively. In this case, $l_2 = L - l_1$ is also defined to clearly have an arc length for the second maneuver, and it is known that the Equation set (4.13) defines the problem at hand. A counter maneuver is needed to compensate for a positive curvature error while also compensating for a positive heading error. To summarize the problem again, we need to equalize the heading error to the areas in Figure 4.7. The Equation would be $\theta_{error} = -(A_1 + A_2 + A_3)$. However, this problem has more degrees of freedom than case 1. To avoid ambiguity in the solutions, for case 2 we have decided to fix σ value to $0.001 \frac{1}{m^2}$.

$$\begin{aligned}\theta_{error} &= \theta_i - \theta_{ri} > 0 \\ \kappa_{error} &= \kappa_i - \kappa_{ri} > 0\end{aligned}\tag{4.13}$$

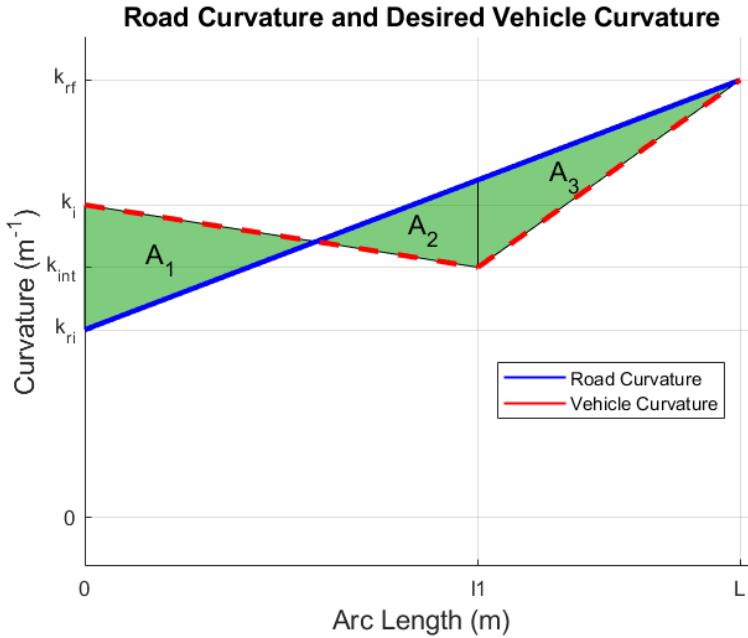


Figure 4.7: Curvature plot of a road segment with a heading and curvature correcting maneuver curvature plot for case 2

Case 3 and 4 solutions are asymmetrical to Case 1 and Case 2 solutions in the sense that the Equations of case 2 can be used to solve for case 3 with the conditions (4.14).

$$\begin{aligned}\theta_{error} &= \theta_i - \theta_{ri} < 0 \\ \kappa_{error} &= \kappa_i - \kappa_{ri} < 0\end{aligned}\tag{4.14}$$

Similarly, to solve case 4 with conditions (4.15), case 1 equations can be used.

$$\begin{aligned}\theta_{error} &= \theta_i - \theta_{ri} < 0 \\ \kappa_{error} &= \kappa_i - \kappa_{ri} > 0\end{aligned}\tag{4.15}$$

4.2.1.1 Case 1 Solution

The first case is where the initial heading error (θ_{error}) is positive and the initial curvature error (κ_{error}) is negative. We want to make the curvature and heading error zero with a clothoid. The required curvature plot only for the correction maneuver is given in Figure 4.8.



Figure 4.8: Curvature error correction with a clothoid

While correcting the curvature, it is necessary to correct the heading as well since they are closely related to each other. It is known that heading change along a curve is the integral of curvature. With this knowledge, Equation (4.16) is constructed. σ and L stand for curvature rate and curve length, respectively.

$$\theta_{error} + \kappa_{error} \cdot L + \frac{\sigma \cdot L^2}{2} = 0 \quad (4.16)$$

Since Figure 4.8 is a clothoid, it has a constant curvature rate depending on κ_{error} and L . The relation is given in Equation (4.17).

$$\sigma = \frac{\kappa_{error}}{L} \quad (4.17)$$

Substituting (4.17) into (4.16) and solving for L and σ yields the Equations (4.18) and (4.19) respectively.

$$L = -\frac{2 \cdot \theta_{error}}{\kappa_{error}} \quad (4.18)$$

$$\sigma = \frac{\kappa_{error}^2}{2 \cdot \theta_{error}} \quad (4.19)$$

With these Equations, we have the required curve length and curvature rate for heading and curvature correcting maneuver. With the assumption of a completely straight road, it is possible to generate the next waypoint that has zero curvature and heading error at the next endpoint.

Figure 4.9 shows the change of curvature and heading along an arc-spline.

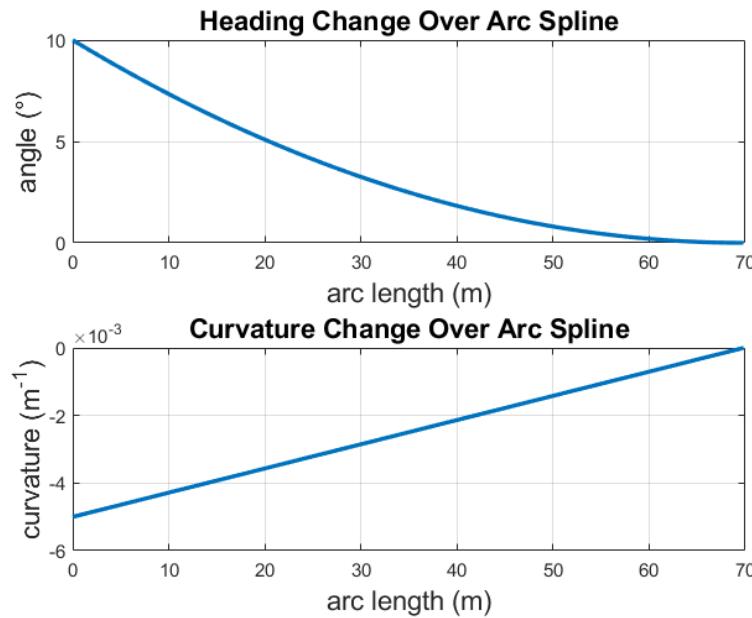


Figure 4.9: Curvature and heading error correction with a clothoid.

We should also consider the effect of position error while making the HCC maneuver. Figure 4.10 shows the evolution of position error when we apply the HCC maneuver. This maneuver changes the initial position error; therefore, we also need to take this effect into account while removing the position error. The position error may increase or decrease; this solely depends on the particular HCC maneuver.

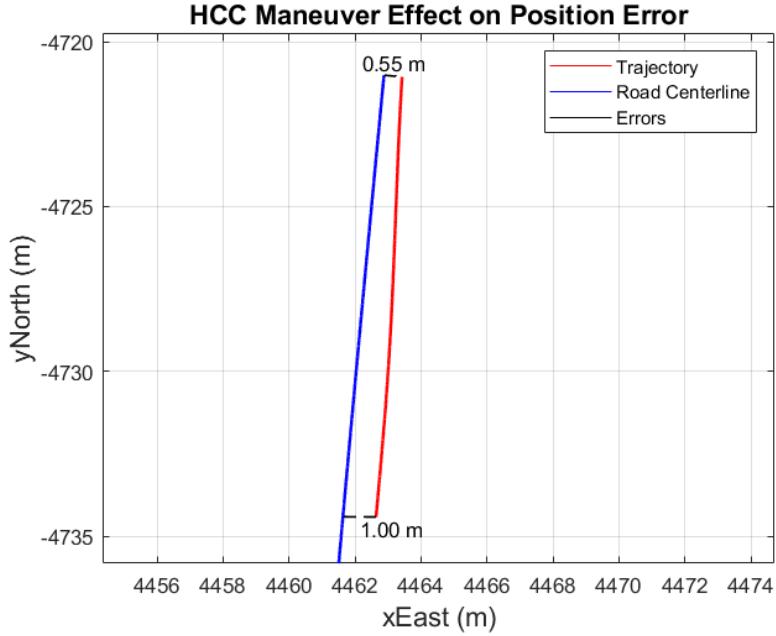


Figure 4.10: Effect of an example HCC maneuver on the position error.

4.2.1.2 Case 2 Solution

The second case is where the initial heading error (θ_{error}) is positive, and the initial curvature error (κ_{error}) is also positive. We want to make the curvature and heading error zero with two concatenated clothoids. The curvature plot is given in Figure 4.8.

In this case it is not possible to correct the heading and curvature with a single clothoid. Two clothoids with opposite curvature rate are needed which is called a counter maneuver. Figure 4.11 shows the required heading and curvature change along an arc-spline. In this plot, two clothoids have curvature rates of equal magnitude but opposite sign. This constraint can be expressed together with curvature correction by the Equation (4.20).

$$k_{error} - \sigma \cdot l_1 + \sigma \cdot l_2 = 0 \quad (4.20)$$

Similar to case 1, θ_{error} must be zero at the end of the arc-spline. The equation for this constraint is given in Equation (4.21), where l_1 and l_2 are the lengths of the first and second clothoid, respectively.

$$\theta_{error} + \kappa_{error} \cdot l_1 - \frac{\sigma}{l_1^2} + (\kappa_{error} - \sigma \cdot l_1) \cdot l_2 + \frac{\sigma \cdot l_2^2}{2} = 0 \quad (4.21)$$

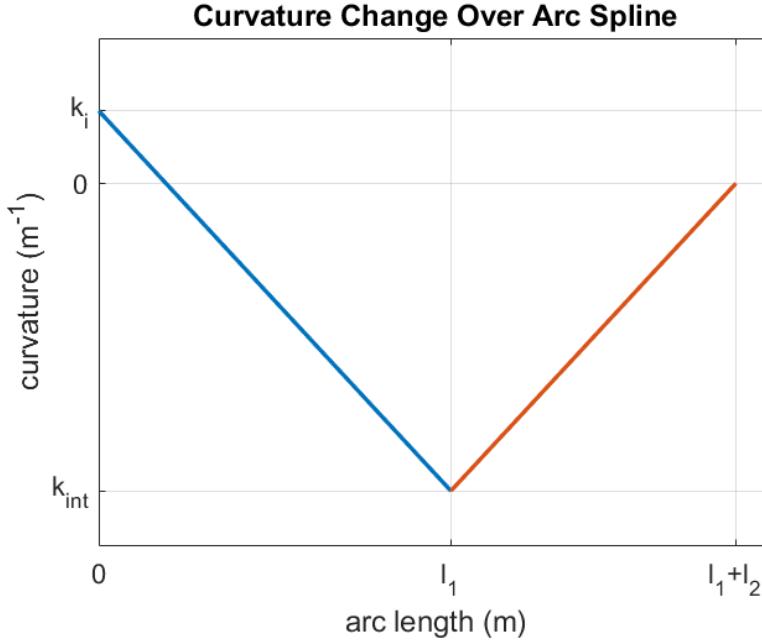


Figure 4.11: Curvature error correction with two clothoids.

Equation (4.21) must be solved for l_1 , l_2 , and σ to define the subsequent waypoints. However, this equation has more freedom than case 1; therefore, it makes sense to fix a variable to reduce the degree of freedom. Fixing l_1 and l_2 is not beneficial, since constraining trajectory length is risky. The best choice is to fix σ since it depends on vehicle dynamics. In this implementation, the value of σ is chosen to be 0.001. Substituting Equation (4.20) into (4.21) yields the solution for l_1 and l_2 given in (4.22) and (4.23) respectively.

$$l_1 = \frac{\kappa_{error}}{\sigma} + \sqrt{\frac{\kappa_{error}^2}{2 \cdot \sigma^2} + \frac{\theta_{error}}{\sigma}} \quad (4.22)$$

$$l_2 = l_1 - \frac{\kappa_{error}}{\sigma} \quad (4.23)$$

When these clothoids are imposed on the initial pose, heading and curvature change over the ego vehicle can be observed in Figure 4.12. Similar to case 1, this maneuver alters the initial position error.

Case 3 ($\theta_{error} < 0, \kappa_{error} < 0$), has the same solution as case 2.

Similarly case 4 ($\theta_{error} < 0, \kappa_{error} > 0$), has the same solution as case 1.

By applying these trajectories it is possible to correct heading and curvature along

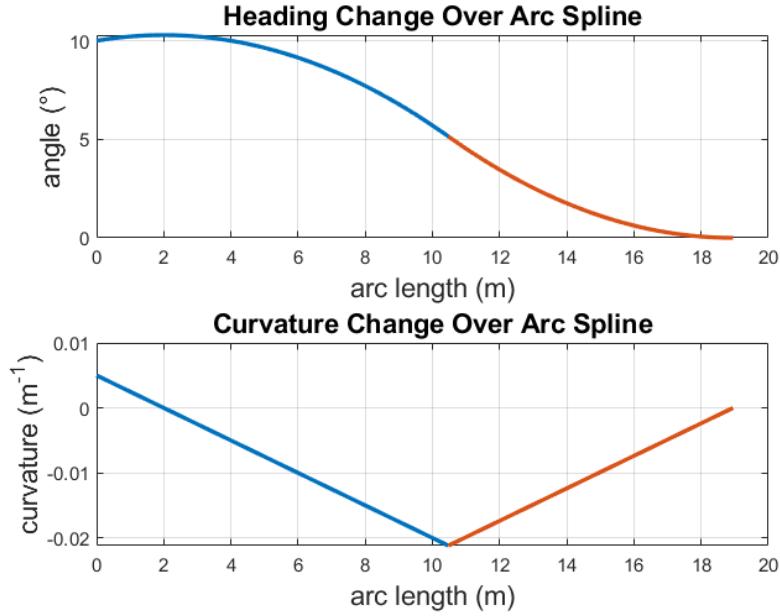


Figure 4.12: Curvature and heading error correction with two clothoids

a completely straight road. However, this method introduces an additional position error, which will be taken care of in Section 4.2.2.

Algorithm 5 describes the overall HCC maneuver. *correctionManeuver* is the heading and curvature correction clothoid maneuver. The algorithm is written for cases 1 and 2 only. Case 3 and 4 are asymmetrical to case 1 and case 2.

Algorithm starts by getting errors then determining the case accordingly.

If the problem at hand is the first case, a minimum function is applied to sigma. Sometimes the formula creates a low sigma value which causes the maneuver to be too long. Therefore a minimum sigma value is imposed.

The length of the maneuver is computed using κ_{error} and σ for both cases.

After getting the parameters for both cases, heading and curvature correction arc-splines are computed. These arc-splines use the initial parameters provided together with the computed σ and length values.

Algorithm 5 Heading and Curvature Error Correction

```
1: procedure COMPUTEHCCMANEUVER( $x_0, y_0, \theta_0, \kappa_0, clothoidRoadSegments$ )
2:    $\theta_{error} \leftarrow \theta_i - \theta_{ri}$ 
3:    $\kappa_{error} \leftarrow \kappa_i - \kappa_{ri}$ 
4:   if  $\theta_{error} > 0$  and  $\kappa_{error} < 0$  then
5:      $case \leftarrow 1$ 
6:      $\sigma \leftarrow min(0.001, \frac{\kappa_{error}^2}{2 \cdot \theta_{error}})$ 
7:      $hccLength = \frac{\kappa_{error}}{\sigma}$ 
8:   else if  $\theta_{error} > 0$  and  $\kappa_{error} > 0$  then
9:      $case \leftarrow 2$ 
10:     $\sigma \leftarrow 0.001$ 
11:     $l_1 \leftarrow \frac{\kappa_{error}}{\sigma} + \sqrt{\frac{\kappa_{error}^2}{2 \cdot \sigma^2} + \frac{\theta_{error}}{\sigma}}$ 
12:     $l_2 \leftarrow l_1 - \frac{\kappa_{error}}{\sigma}$ 
13:   end if
14:   if  $case = 1$  then
15:      $\kappa_1 \leftarrow \kappa_0 + \sigma \cdot hccLength + roadCurvature \cdot hccLength$ 
16:      $arcSpline \leftarrow \mathcal{C} = [x_0, y_0, \theta_0, \kappa_0, \kappa_1, hccLength]$ 
17:   else if  $case = 2$  then
18:      $\kappa_1 \leftarrow \kappa_0 - \sigma \cdot l_1 + roadCurvature \cdot l_1$ 
19:      $arcSpline(1) \leftarrow \mathcal{C} = [x_0, y_0, \theta_0, \kappa_0, \kappa_1, l_1]$ 
20:      $x_1, y_1 \leftarrow endPoint(arcSpline(1))$ 
21:      $\theta_1 \leftarrow endHeading(arcSpline(1))$ 
22:      $\kappa_2 \leftarrow \kappa_1 + \sigma \cdot l_2 + roadCurvature \cdot l_2$ 
23:      $arcSpline(2) \leftarrow \mathcal{C} = [x_1, y_1, \theta_1, \kappa_1, \kappa_2, l_2]$ 
24:   end if
25:   return  $HCCarcSpline, positionError$ 
26: end procedure
```

4.2.2 Position Correction

Position correction with arc-splines can be achieved by applying bi-elementary paths as described in Section 2.1.4. Bi-elementary paths take position error and trajectory length as input.

Even if the position error of a vehicle's initial pose is known, the position error is introduced by heading and curvature correction maneuver as mentioned in Section 4.2.1.

In order to compensate for the initial error and the error introduced by the heading and curvature correction maneuver, the position error after the heading and curvature error correction maneuver is given as input to the bi-elementary path parameter computation.

The length of the bi-elementary path is chosen to be the same as the heading and curvature correction maneuver. With this design decision, it is possible to execute both maneuvers simultaneously. The superposition of both maneuvers is described in Section 4.2.4.

4.2.3 Road Curvature Change Correction

It is known that the given road is a sequence of clothoid road segments with a constant curvature rate along each road segment. Therefore, it is possible to compensate for curvature error along the road by simply integrating the curvature rate of the road.

The road's curvature rate may change while making the attitude correction maneuvers. In this case the change of the curvature rate should also be taken into account.

4.2.4 Superposition of Maneuvers

This is the Section where the arc-spline trajectory is generated by superposing the maneuvers described in Sections 4.2.1, 4.2.3 and 4.2.3.

Generating an arc-spline requires well defined consecutive *waypoints*. Each *waypoint*

consists of position, heading, curvature and length information. Length information is the arc-spline length connecting the previous *waypoint* to current *waypoint*. Length is determined by heading and curvature correction maneuver.

At the very beginning of the algorithm, the case at hand is determined as described in Section 4.2.1. When the case is determined, automatically σ and length of arc-spline is also determined.

Then, the required parameters for position correction maneuver is also computed with the output of heading and curvature correction maneuver.

Depending on the positions of waypoints, the road's curvature is taken into account as well.

After computing the required waypoints for heading and curvature correction maneuver, position correction maneuver and road curvature change correction maneuver it is possible to superpose these maneuvers. The superposition requires ordering of waypoints with respect to their positions along a road.

Algorithm 6 describes the arc-spline trajectory computation process. First the number of different road segments is determined. Every time a road segment is changed, the curvature rate coming from the nature of the road segment changes. The curvature rate of the road is defined as *roadRate*.

Number of waypoints are the number of heading and curvature correction waypoints, number of bi-elementary path waypoints and number of road segments passed until the end of the heading and curvature correction maneuver length which is named *HCClength*.

Each waypoint consists of position, heading, curvature information. Additionally, for computational ease, waypoints store the required arc-spline length from the previous waypoint.

lengthSoFar is the current arc-spline's length. It increases in for loop until it reaches *HCClength*. This variable is like the arc length pointer on Figure 4.13.

findClosestWp function gets the closest waypoint in all the waypoints given the

$lengthSoFar$. Then the required arc length until the next waypoint l_i is stored.

The curvature rates are determined from the waypoint sets. At this point we know the position of our length pointer and we just obtain the curvature rate for each maneuver along that segment.

Next curvature rate is computed with the rates and the length between the current waypoint and next waypoint by summation. This is the part where superposition is applied.

A clothoid (it is approximated with an arc-spline) is constructed with the known parameters.

Finally, arc-spline is added to the list and waypoint list is also updated.

Algorithm 6 Fit arc-spline

```

1: procedure FITARCSPLINE(waypoints)
2:   Input:  $x_0, y_0, \theta_0, \kappa_0, HccWps, biElemWps, clothoidRoad$ 
3:   Output: waypoints, arcSplines
4:    $num(roadSegments) \leftarrow num(RoadSegments)$  until  $HCClength$ 
5:    $numWaypoints \leftarrow num(HccWps) + num(biElemWps) + num(roadSegments) + 1$ 
6:    $waypoints(0) = [x_0, y_0, \theta_0, \kappa_0, 0]$ 
7:    $lengthSoFar \leftarrow 0$ 
8:   for  $i \leftarrow 1$  to  $numWaypoints$  do
9:      $closestWp \leftarrow findClosestWp(lengthSoFar, HccWps, biElemWps, roadSegments)$ 
10:     $l_i \leftarrow length(closestWp)$ 
11:     $hccRate \leftarrow getHccCurvatureRate(lengthSoFar + l_i)$ 
12:     $biElemRate \leftarrow getBiElemCurvatureRate(lengthSoFar + l_i)$ 
13:     $roadRate \leftarrow getRoadCurvatureRate(lengthSoFar + l_i)$ 
14:     $lengthSoFar = lengthSoFar + l_i$ 
15:     $\kappa_i \leftarrow k_{i-1} + hccRate \cdot l_i + biElemRate \cdot l_i + roadRate \cdot l_i$ 
16:     $tempClothoid \leftarrow \mathcal{C} = [x_{i-1}, y_{i-1}, \theta_{i-1}, \kappa_{i-1}, \kappa_i, l_1]$ 
17:     $arcSplines(i) \leftarrow tempClothoid$ 
18:     $x_i, y_i \leftarrow tempClothoid.lastPosition$ 
19:     $waypoints(i) = [x_i, y_i, \theta_i, \kappa_i, l_i]$ 
20:   end for
21:   return arcSpline
22: end procedure

```

Figure 4.13 shows the superposition of HCC maneuver, Bi-elementary maneuver and road curvature on the same plot. Superposed curvature is the sum of all curvatures at every point along the trajectory. This data is generated for visualization purposes.

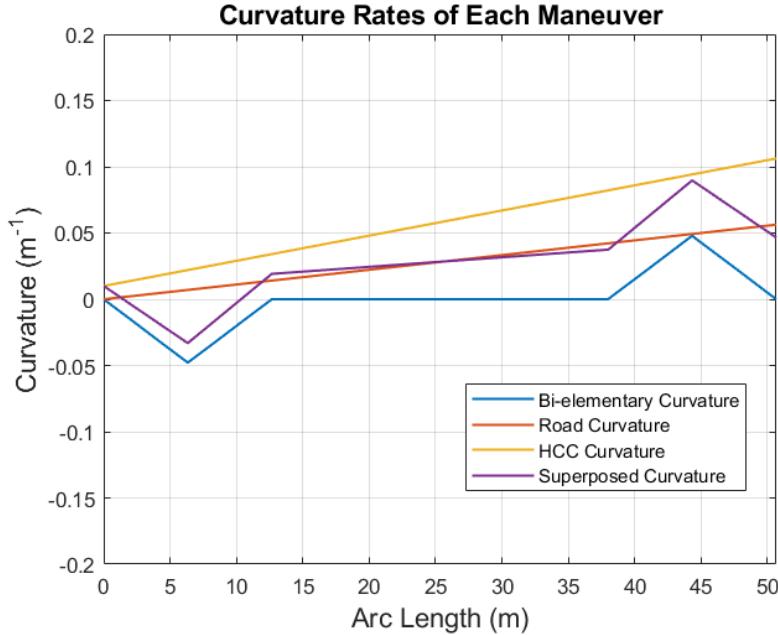


Figure 4.13: Curvature changes over $HCClength$ as an example. This data is simulated for visualization.

4.3 Evaluation Metrics and Results

Evaluation metrics for trajectory algorithms include position, heading, and curvature errors. Position error is defined as the Euclidean distance between the planned trajectory and the road centerline along the path. Heading error represents the difference in orientation between the reference road and the computed trajectory. Curvature error measures the discrepancy in curvature between the road centerline and the computed trajectory.

The road centerline is a real road in Germany, specifically Autobahn 38. The error is artificially introduced by shifting the position of the road by desired position error. Similarly, heading and curvature error is also introduced by adding up the initial heading and curvature with heading error and curvature error respectively.

4.3.1 RMS and Maximum Error For Many Cases

In this Section, the RMS and Maximum position error of arc-spline trajectories for a variety of cases is inspected. Without classifying the errors as case 1,2,3 or 4 all of the error configurations are solved with arc-spline trajectory generation algorithm. Over the trajectory, RMS and maximum error is computed and written on the tables. Each table has a different position error where a positive position error means that the vehicle is on the left side of the road centerline and vice versa. In each Table, possible heading and curvature error values are swept between reasonable values.

Table 4.2 has the error configurations with +0.80 meters error. In some cases error exceeds 1 meter which means that the vehicle might leave the lane boundary. This effect is closely related to error configurations. With these error configurations it might not be possible to plan a trajectory without leaving the lane boundary at highway speeds.

Table 4.3 has the error configurations with +0.40 meters error. With these error configurations it is observed that the positive heading error causes the vehicle to depart from road centerline initially. In this case the errors are safer than the previous case.

Table 4.4 has the error configurations with -0.40 meters error. Now it is known that the vehicle is to the right side of road centerline. With this configuration, when the initial heading error is positive it is observed that the maximum error is equal to initial position error. This means that over the trajectory, the position error kept getting closer and closer to zero. Similar effects can be observed in all error tables.

Table 4.5 has the error configurations with -0.80 meters error. Similar to the previous case, with this configuration, when the initial heading error is positive it is observed that the maximum error is equal to initial position error. Negative heading values cause the vehicle initially get further from the road centerline. At all cases, the final error is zero but it takes time to recover from heavy errors. It is worth noting that the RMS error at the bottom of the table is even more than the initial error, meaning that the vehicle took too much time to reach road centerline.

Table 4.2: Different heading and curvature error configurations with the RMS and Maximum errors for +0.80 meter position error

Position Error	Heading Error	Curvature Error	RMS Error	Max Error
+0.80m	+4.0°	+0.0100 m^{-1}	0.8156	1.0951
+0.80m	+4.0°	+0.0050 m^{-1}	0.6876	0.9807
+0.80m	+4.0°	-0.0050 m^{-1}	0.5234	0.8271
+0.80m	+4.0°	-0.0100 m^{-1}	0.5485	0.8555
+0.80m	+2.0°	+0.0100 m^{-1}	0.6556	0.9239
+0.80m	+2.0°	+0.0050 m^{-1}	0.5736	0.8602
+0.80m	+2.0°	-0.0050 m^{-1}	0.5150	0.8113
+0.80m	+2.0°	-0.0100 m^{-1}	0.5208	0.8153
+0.80m	-2.0°	+0.0100 m^{-1}	0.4563	0.8000
+0.80m	-2.0°	+0.0050 m^{-1}	0.4877	0.8000
+0.80m	-2.0°	-0.0050 m^{-1}	0.4154	0.8000
+0.80m	-2.0°	-0.0100 m^{-1}	0.3589	0.8000
+0.80m	-4.0°	+0.0100 m^{-1}	0.4057	0.8000
+0.80m	-4.0°	+0.0050 m^{-1}	0.4791	0.8000
+0.80m	-4.0°	-0.0050 m^{-1}	0.3287	0.8000
+0.80m	-4.0°	-0.0100 m^{-1}	0.2967	0.8000

Table 4.3: Different heading and curvature error configurations with the RMS and Maximum errors for +0.40 meter position error

Position Error	Heading Error	Curvature Error	RMS Error	Max Error
+0.40m	+4.0°	+0.0100 m^{-1}	0.5851	0.7534
+0.40m	+4.0°	+0.0050 m^{-1}	0.4493	0.6101
+0.40m	+4.0°	-0.0050 m^{-1}	0.2724	0.4318
+0.40m	+4.0°	-0.0100 m^{-1}	0.3017	0.4653
+0.40m	+2.0°	+0.0100 m^{-1}	0.4190	0.5456
+0.40m	+2.0°	+0.0050 m^{-1}	0.3303	0.4733
+0.40m	+2.0°	-0.0050 m^{-1}	0.2635	0.4135
+0.40m	+2.0°	-0.0100 m^{-1}	0.2719	0.4187
+0.40m	-2.0°	+0.0100 m^{-1}	0.2100	0.4000
+0.40m	-2.0°	+0.0050 m^{-1}	0.2354	0.4000
+0.40m	-2.0°	-0.0050 m^{-1}	0.1752	0.4000
+0.40m	-2.0°	-0.0100 m^{-1}	0.1579	0.4000
+0.40m	-4.0°	+0.0100 m^{-1}	0.1733	0.4000
+0.40m	-4.0°	+0.0050 m^{-1}	0.2268	0.4000
+0.40m	-4.0°	-0.0050 m^{-1}	0.1389	0.4000
+0.40m	-4.0°	-0.0100 m^{-1}	0.2319	0.4000

Table 4.4: Different heading and curvature error configurations with the RMS and Maximum errors for -0.40 meter position error

Position Error	Heading Error	Curvature Error	RMS Error	Max Error
-0.40m	+4.0°	+0.0100m ⁻¹	0.2312	0.4000
-0.40m	+4.0°	+0.0050m ⁻¹	0.1389	0.4000
-0.40m	+4.0°	-0.0050m ⁻¹	0.2322	0.4000
-0.40m	+4.0°	-0.0100m ⁻¹	0.1960	0.4000
-0.40m	+2.0°	+0.0100m ⁻¹	0.1577	0.4000
-0.40m	+2.0°	+0.0050m ⁻¹	0.1753	0.4000
-0.40m	+2.0°	-0.0050m ⁻¹	0.2407	0.4000
-0.40m	+2.0°	-0.0100m ⁻¹	0.2273	0.4000
-0.40m	-2.0°	+0.0100m ⁻¹	0.2979	0.4211
-0.40m	-2.0°	+0.0050m ⁻¹	0.2690	0.4138
-0.40m	-2.0°	-0.0050m ⁻¹	0.3303	0.4734
-0.40m	-2.0°	-0.0100m ⁻¹	0.4191	0.5457
-0.40m	-4.0°	+0.0100m ⁻¹	0.3596	0.4802
-0.40m	-4.0°	+0.0050m ⁻¹	0.2777	0.4324
-0.40m	-4.0°	-0.0050m ⁻¹	0.4494	0.6102
-0.40m	-4.0°	-0.0100m ⁻¹	0.5857	0.7542

Table 4.5: Different heading and curvature error configurations with the RMS and Maximum errors for -0.80 meter position error

Position Error	Heading Error	Curvature Error	RMS Error	Max Error
-0.80m	+4.0°	+0.0100m ⁻¹	0.2966	0.8000
-0.80m	+4.0°	+0.0050m ⁻¹	0.3287	0.8000
-0.80m	+4.0°	-0.0050m ⁻¹	0.4844	0.8000
-0.80m	+4.0°	-0.0100m ⁻¹	0.4430	0.8000
-0.80m	+2.0°	+0.0100m ⁻¹	0.3590	0.8000
-0.80m	+2.0°	+0.0050m ⁻¹	0.4154	0.8000
-0.80m	+2.0°	-0.0050m ⁻¹	0.4929	0.8000
-0.80m	+2.0°	-0.0100m ⁻¹	0.4764	0.8000
-0.80m	-2.0°	+0.0100m ⁻¹	0.5453	0.8163
-0.80m	-2.0°	+0.0050m ⁻¹	0.5205	0.8114
-0.80m	-2.0°	-0.0050m ⁻¹	0.5736	0.8603
-0.80m	-2.0°	-0.0100m ⁻¹	0.6556	0.9241
-0.80m	-4.0°	+0.0100m ⁻¹	0.6022	0.8624
-0.80m	-4.0°	+0.0050m ⁻¹	0.5287	0.8274
-0.80m	-4.0°	-0.0050m ⁻¹	0.6876	0.9808
-0.80m	-4.0°	-0.0100m ⁻¹	0.8162	1.0953

Table 4.6: Error configurations for different cases

	Position Error	Heading Error	Curvature Error
Case 1	+0.35	+2.0°	$-0.0010m^{-1}$
Case 2	+0.20	+3.0°	$0.0015m^{-1}$
Case 3	-0.35	-4.0°	$-0.0090m^{-1}$

4.3.2 Arc-spline Trajectory

Table 4.6 shows the introduced errors for different cases. A positive position error indicates that the ego vehicle is to the left of the road centerline. A positive heading error signifies that the vehicle is oriented to the right. Similarly, a positive curvature error means that the vehicle is turning to the right more sharply than required. These cases are further investigated in the upcoming sections.

Figure 4.14 displays the position error over an arc-spline for case 1. The algorithm makes the position error zero at the end. Figure 4.16 shows the curvature error over arc-spline. The curvature error plot resembles a bi-elementary path's curvature plot. This is due to the curvature error's low value compared to the position error. Figure 4.21 displays the trajectory and road centerline. Figure 4.15 displays the heading error which approaches zero at the end of the trajectory.

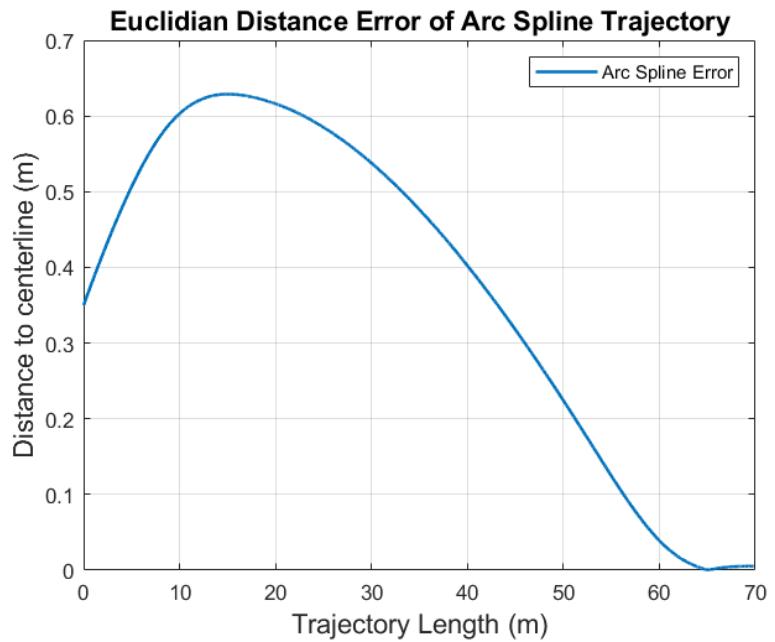


Figure 4.14: Position error over an arc-spline trajectory for case 1

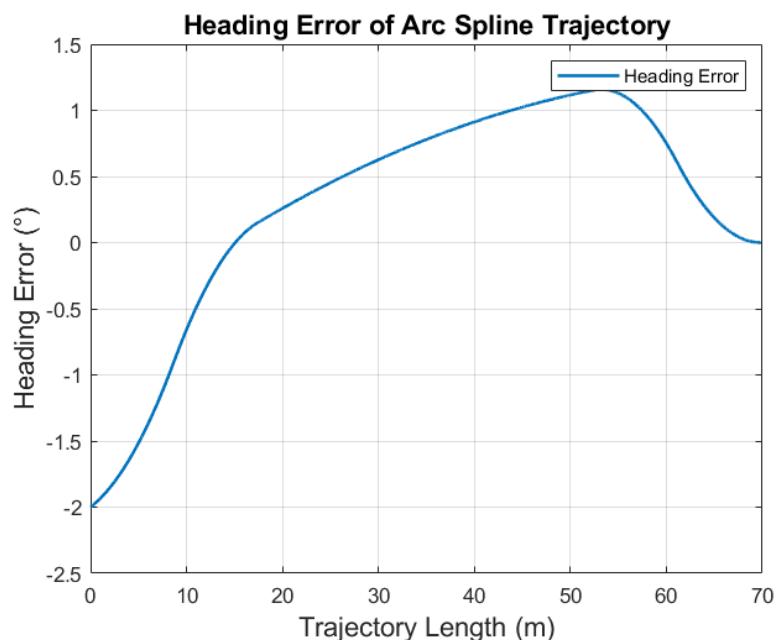


Figure 4.15: Heading error over an arc-spline trajectory for case 1

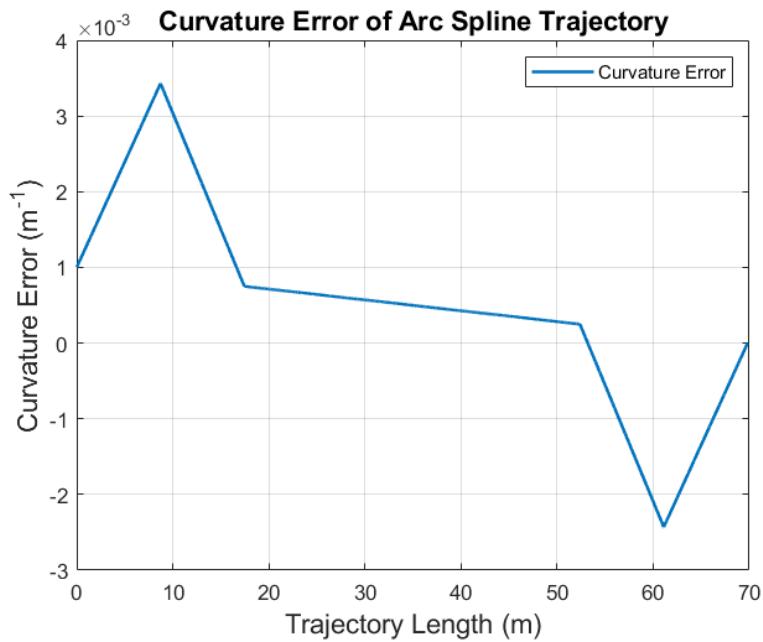


Figure 4.16: Curvature error over an arc-spline trajectory for case 1

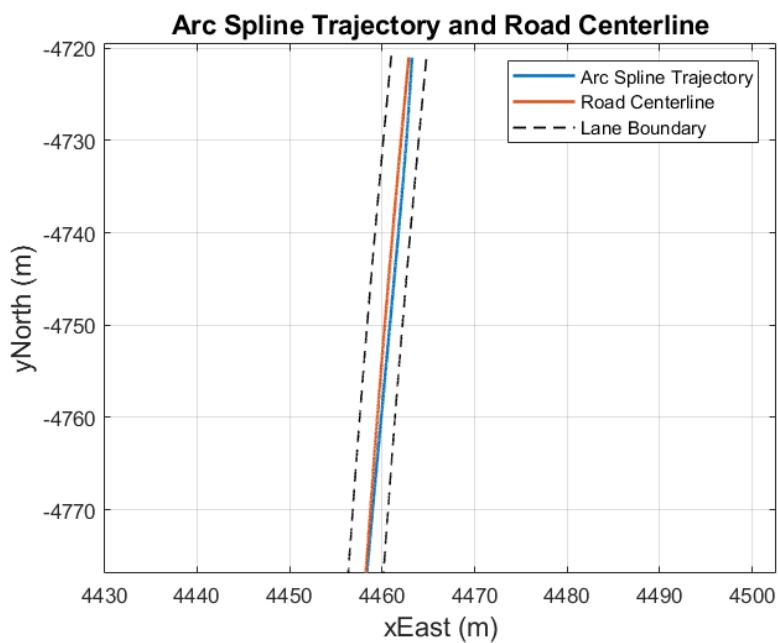


Figure 4.17: Arc-spline trajectory on Autobahn 38 case 1

Figure 4.18 displays the position error over an arc-spline for case 2. The algorithm makes the position error zero at the end. However, while making the maneuver the position error increases initially, this is due to curvature error. Figure 4.20 shows the curvature error over arc-spline. Due to the initial negative curvature error, the position error increases at the start. Figure 4.21 displays the trajectory and road centerline. Figure 4.19 displays the heading error which approaches zero at the end of the trajectory similar to case 1.

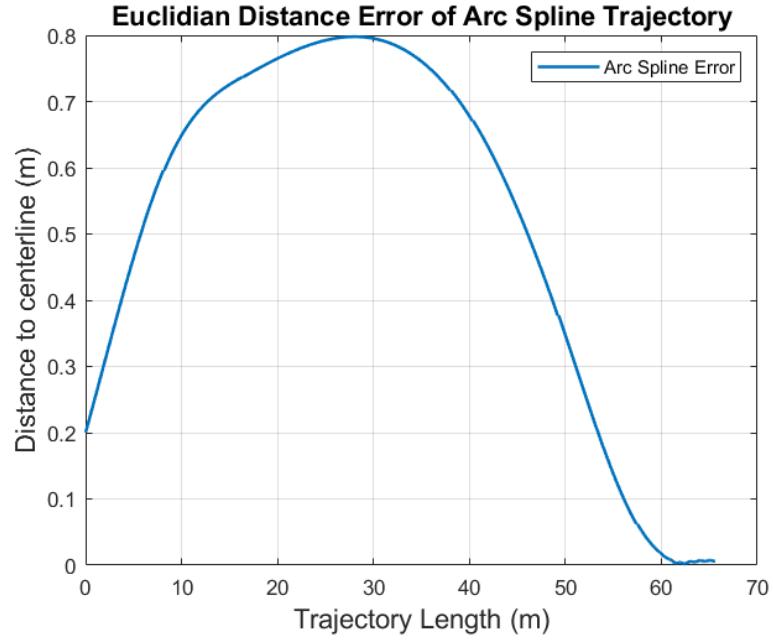


Figure 4.18: Position error over an arc-spline trajectory for case 2

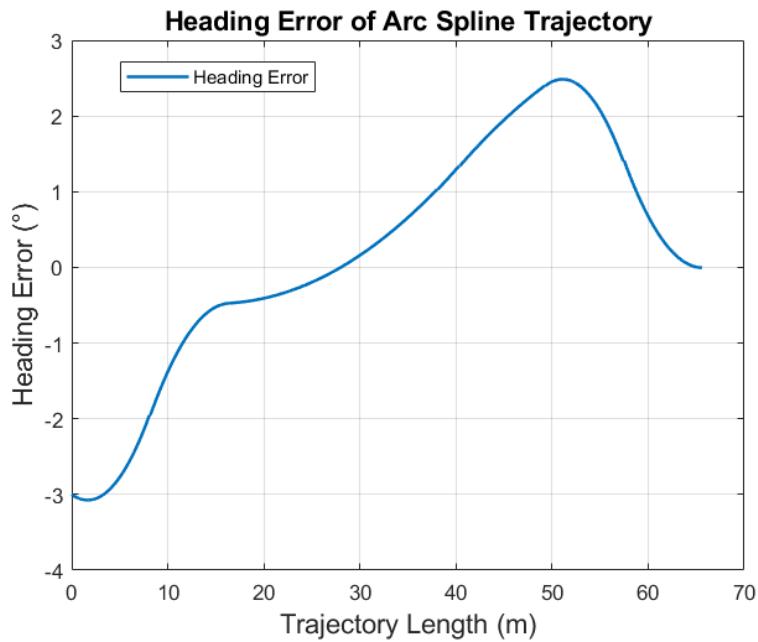


Figure 4.19: Heading error over an arc-spline trajectory for case 2

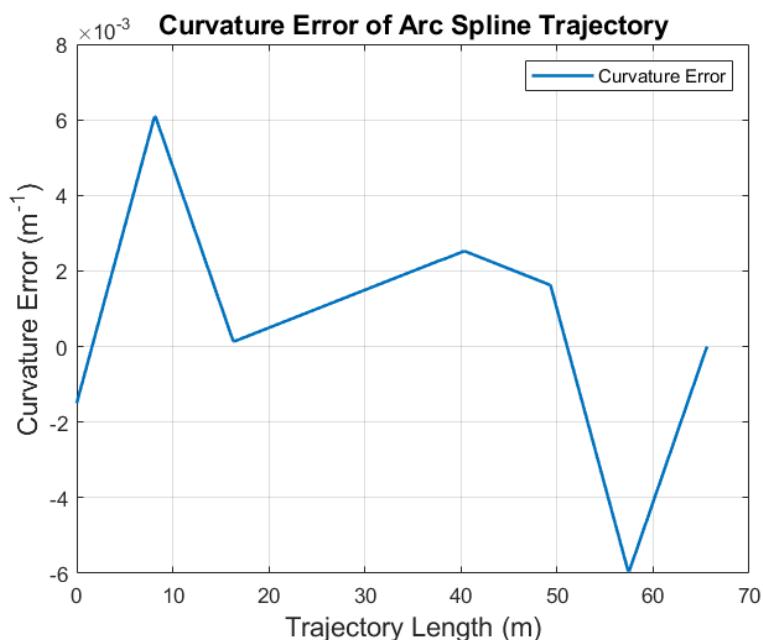


Figure 4.20: Curvature error over an arc-spline trajectory for case 2

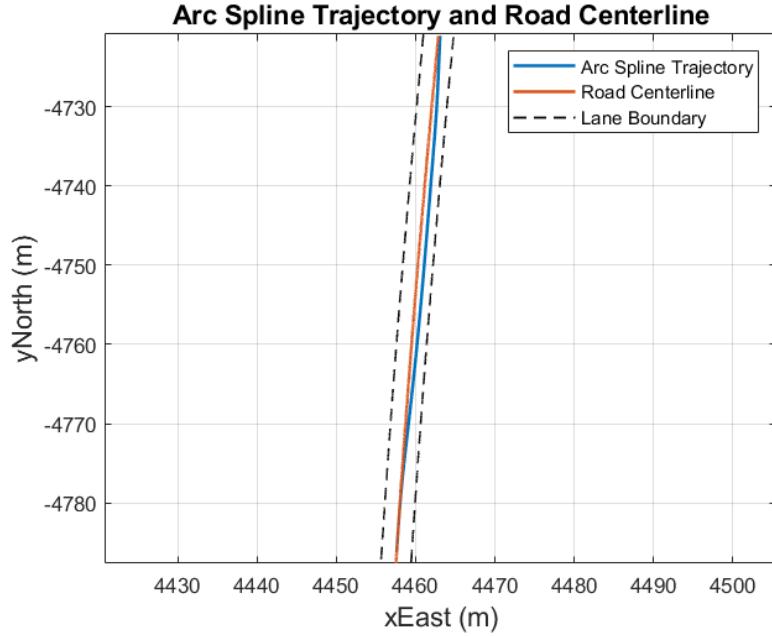


Figure 4.21: Arc-spline trajectory on Autobahn 38 case 2

Figure 4.22 displays the position error over an arc-spline for case 3. Figure 4.24 shows the curvature error over arc-spline. Figure 4.25 displays the trajectory and road centerline. Because the vehicle starts on the opposite side of the road centerline compared to case 1 and case 2, it initially has an opposite curvature value. Figure 4.23 displays the heading error, which approaches zero at the end of the trajectory, similar to the previous cases.

Case 2 solution requires a predefined sigma value. Due to the nature of the Equations (4.22) and (4.23), it is clearly seen that as σ value increases, the total path length of HCC decreases. Decreasing the path length of HCC forces the position correction maneuver to make sharper turns. Therefore, it is not desired. Increasing the path length, on the other hand, makes the vehicle travel too much distance while making the maneuver, which may not be desired. The default value for σ is chosen to be $0.001m^{-1}$ for this research.

Figure 4.26 shows the evolution of the curvature along the trajectory with $\sigma = 0.0010m^{-2}$. In this case, the path length significantly decreases. In contrast, the curvature error is unacceptably high. Figure 4.27 illustrates the resulting arc-spline trajectory.

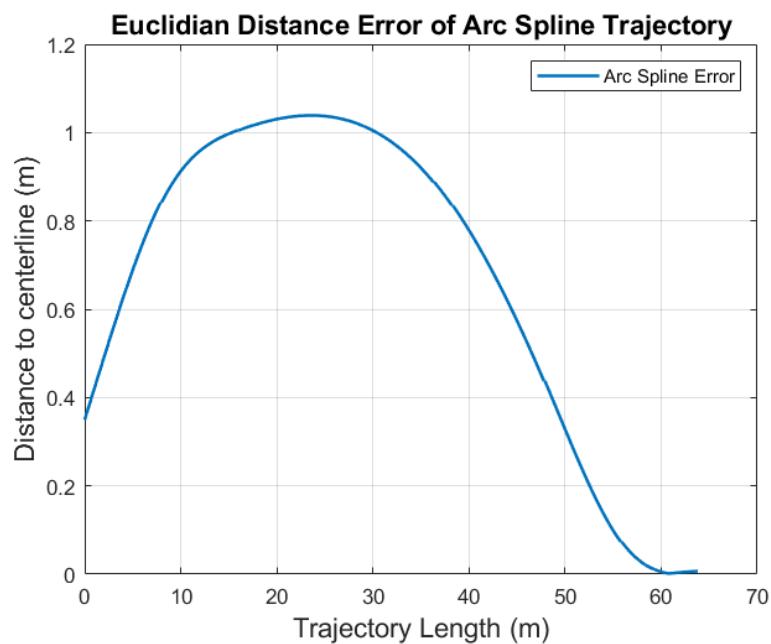


Figure 4.22: Position error over an arc-spline trajectory for case 3

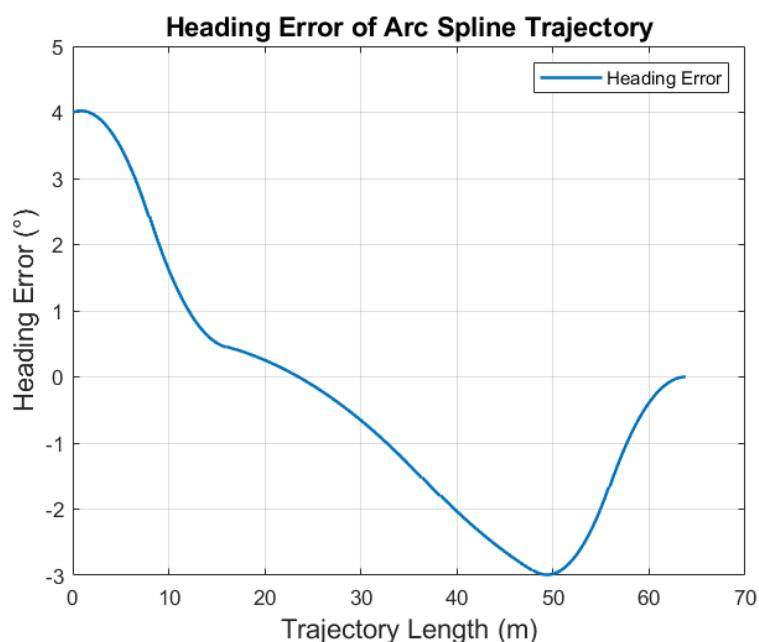


Figure 4.23: Heading error over an arc-spline trajectory for case 3

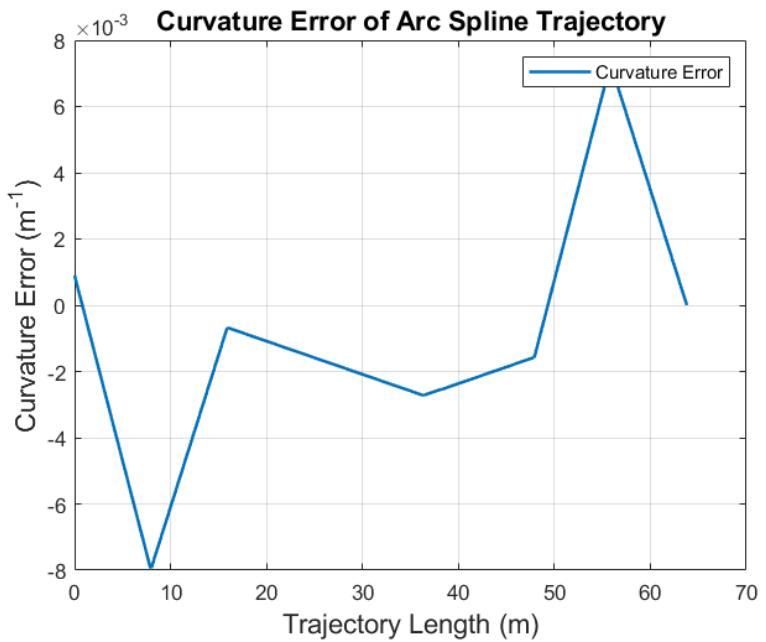


Figure 4.24: Curvature error over an arc-spline trajectory for case 3

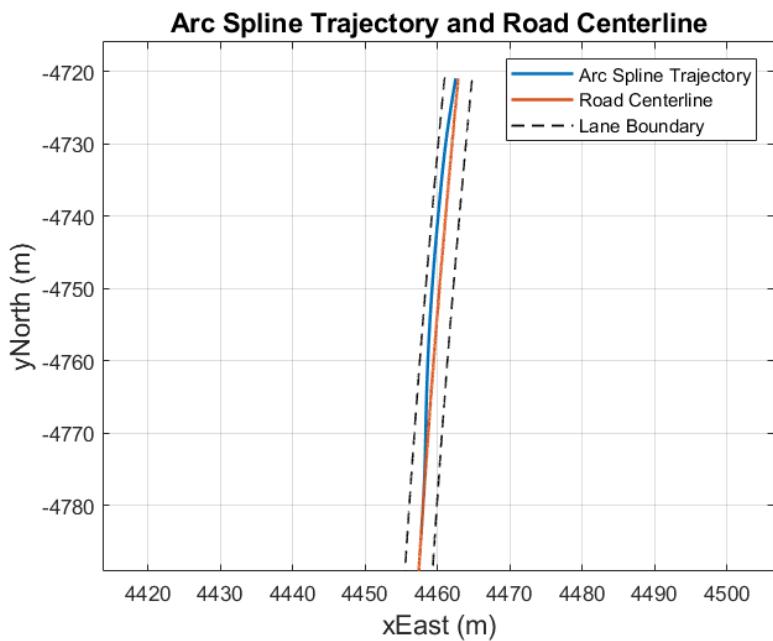


Figure 4.25: Arc-spline trajectory on Autobahn 38 case 3

Figure 4.28 shows the evolution of the curvature along the trajectory with $\sigma = 0.00005m^{-2}$. It makes sense to compare this figure to 4.20. As observed, the path length increases while lowering the peak values of curvature error. Figure 4.29 illustrates the resulting arc-spline trajectory.

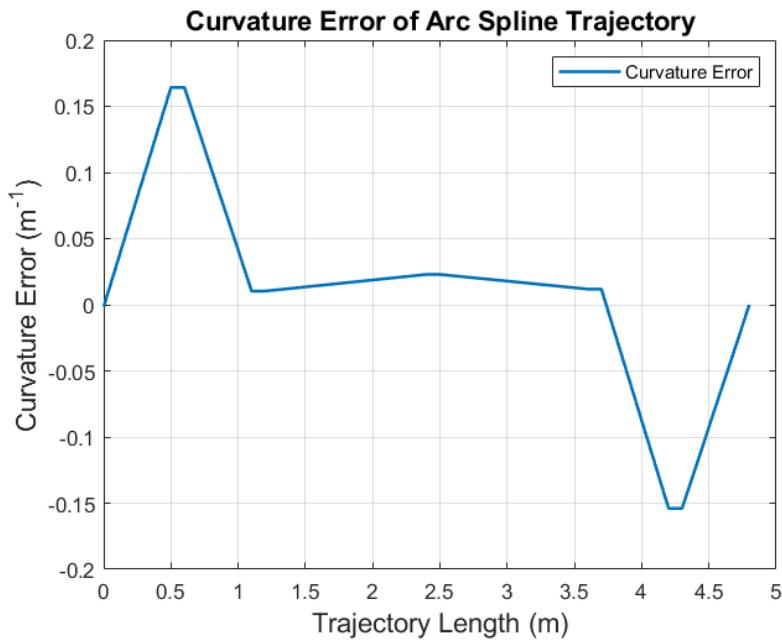


Figure 4.26: Curvature error over an arc-spline trajectory for case 2 with
 $\sigma = 0.0010m^{-2}$

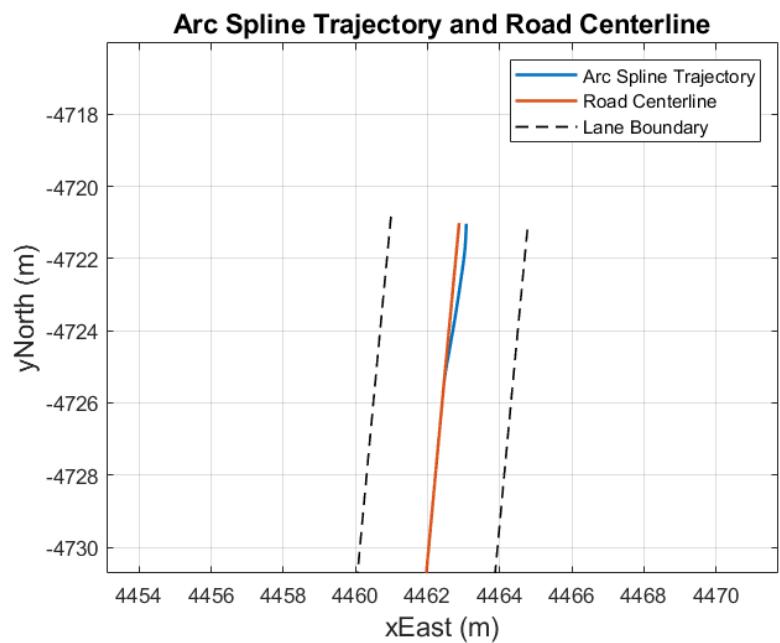


Figure 4.27: Arc-spline trajectory for case 2 with $\sigma = 0.0010m^{-2}$

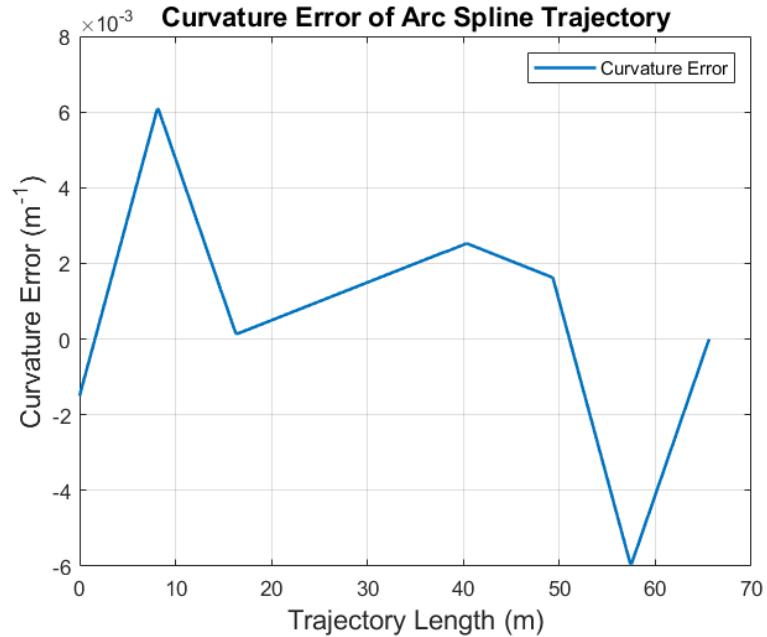


Figure 4.28: Curvature error over an arc-spline trajectory for case 2 with
 $\sigma = 0.00001m^{-2}$

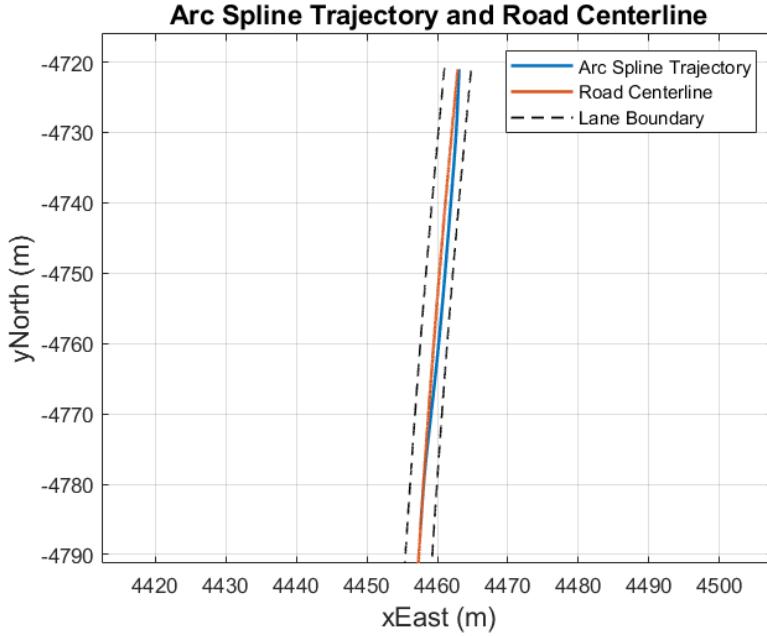


Figure 4.29: Arc-spline trajectory for case 2 with $\sigma = 0.00001 m^{-2}$

4.3.3 Comparison with Bézier Curves

In this section, we investigate case 1 to 3 as discussed before, and use Bézier curves as described in Section 4.1 for trajectory computation. For a better comparison, 5 different length Bézier curves are generated for each case. The lengths of each Bézier trajectory are given as

$$\left[\frac{8 \cdot L}{10} \quad \frac{9 \cdot L}{10} \quad L \quad \frac{11 \cdot L}{10} \quad \frac{12 \cdot L}{10} \right] \quad (4.24)$$

where L is the length of the arc-spline trajectory for the case at hand.

Figures 4.33, 4.37 and 4.41 displays the trajectories for case 1, case 2 and case 3 respectively. The plots show the road centerline and road lane boundaries as well.

Figure 4.30 displays the position error over Bézier trajectories and arc-spline for case 1. It is clearly seen that the maximum error of arc-spline trajectory is lower in all cases. The closest Bézier curve in the sense of position error is the Bézier trajectory with the least length. As observed Figure 4.32 the Bézier trajectory with the least length has higher peak curvature error value. This result shows that Bézier curves tend to have high curvature rates to compensate for position error. Figure 4.31 shows the heading error over Bézier trajectory, as expected it approaches zero at the end of

the trajectory.

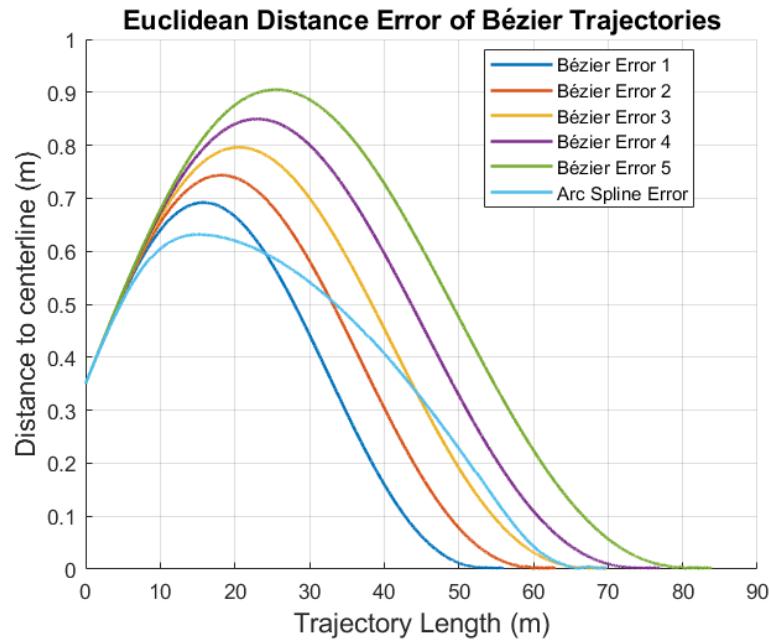


Figure 4.30: Position error comparison for a set of Bézier curves and arc-spline trajectory for case 1

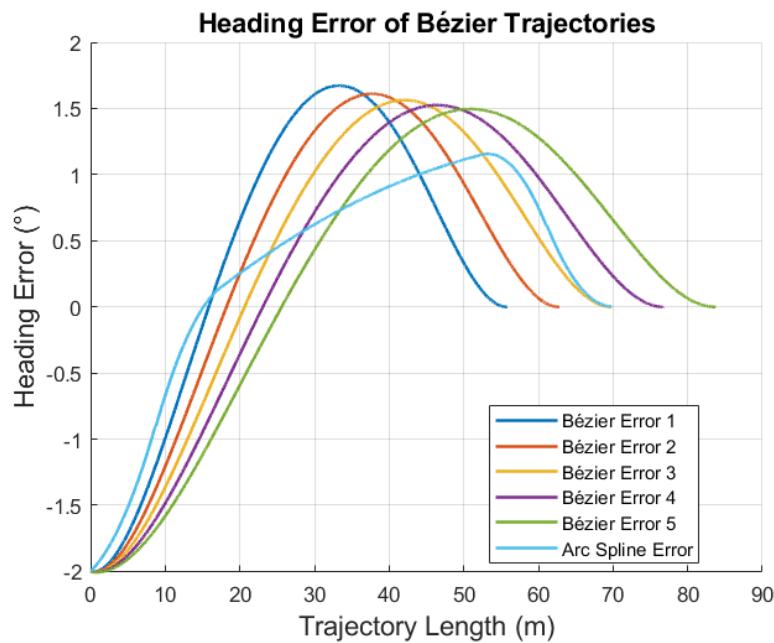


Figure 4.31: Heading error comparison for a set of Bézier curves and arc-spline trajectory for case 1

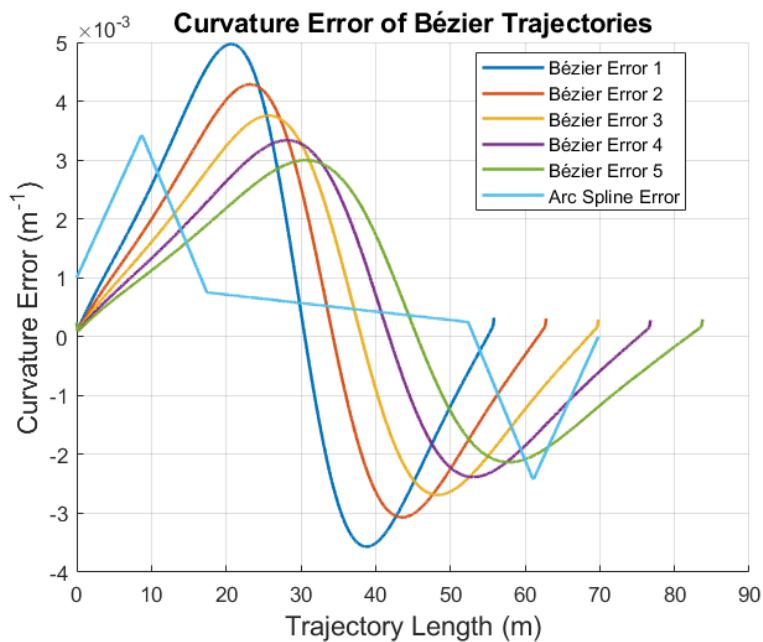


Figure 4.32: Curvature error comparison for a set of Bézier curves and arc-spline trajectory for case 1

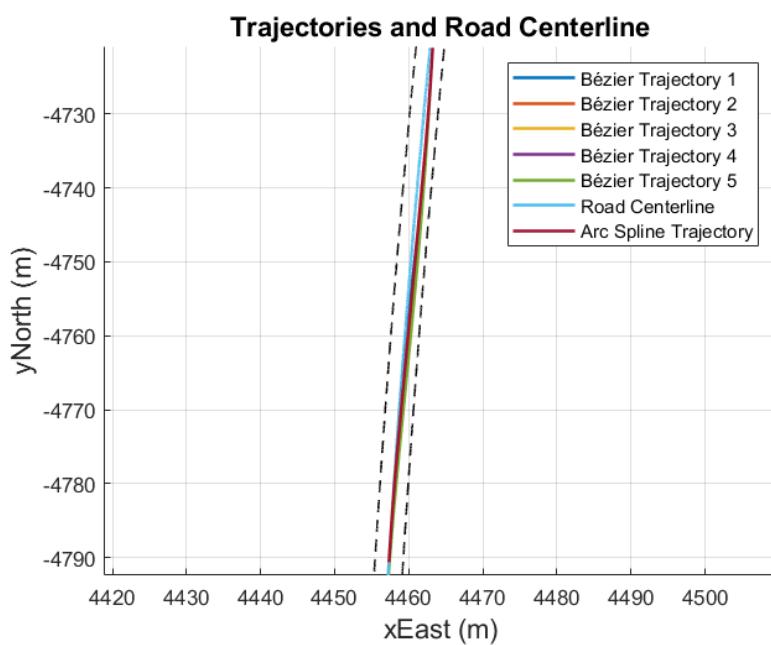


Figure 4.33: Comparison of Bézier trajectories and arc-splines with lane boundaries on the road for case 1

Figure 4.34 displays the position error over Bézier trajectories and arc-spline for case 2. The algorithm makes the position error zero at the end. However, while making the maneuver the position error significantly increases, for some cases these cases are not acceptable since that vehicle may leave the road. There are 3 other Bézier trajectories that have lower position error. It is necessary to inspect the curvature plot as well, Figure 4.36 shows the curvature error over Bézier trajectories. These 3 lanes with lower position error values have significantly high curvature rates. Figure 4.35 shows the heading error over Bézier trajectory, as expected it approaches zero at the end of the trajectory similar to case 1.

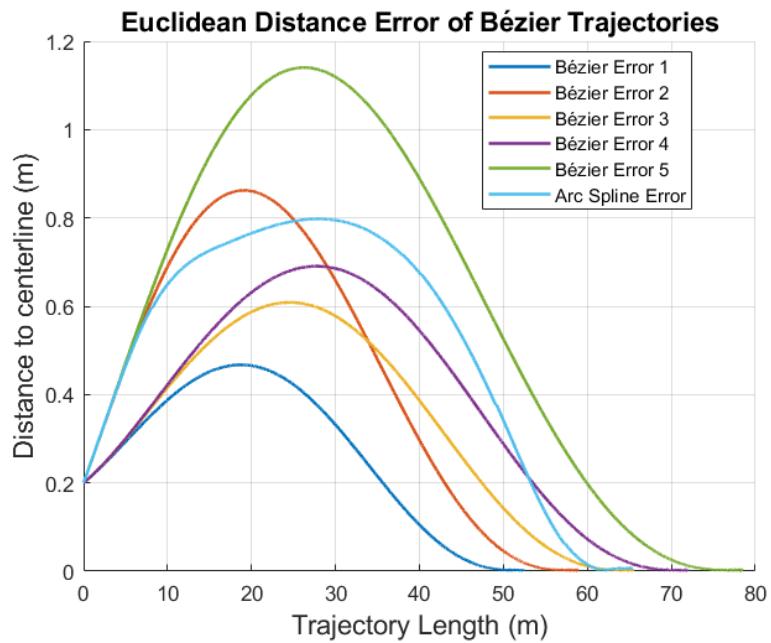


Figure 4.34: Position error comparison for a set of Bézier curves and arc-spline trajectory for case 2

Figure 4.38 displays the position error over Bézier trajectories and arc-spline for case 3. It is seen that the position error of the arc-spline is either better or comparable to Bézier trajectories. Similar to previous cases, the position error is comparable with the shortest Bézier curves. Again, similar to previous cases the shortest Bézier trajectory has very high curvature rates compared to arc-spline trajectory as observed in Figure 4.40. Figure 4.39 shows the heading error over Bézier trajectory, as expected it approaches zero at the end of the trajectory similar to previous cases.

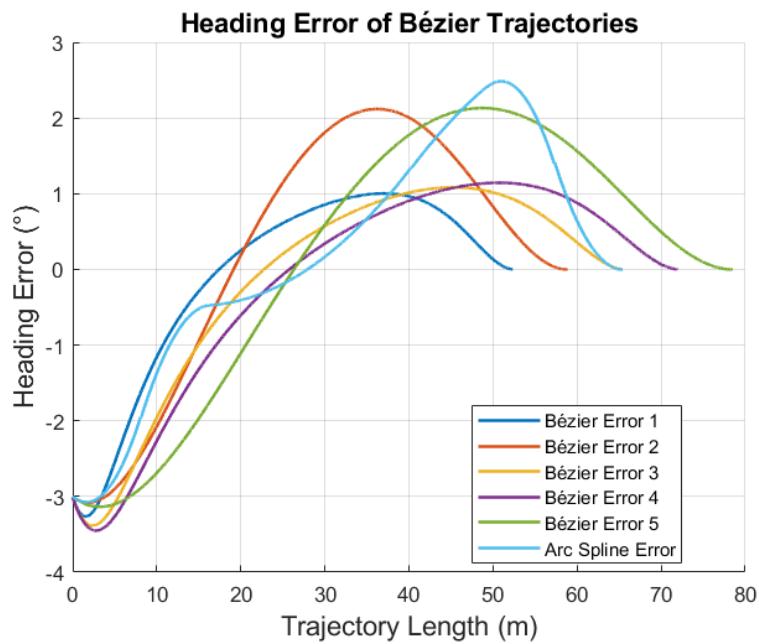


Figure 4.35: Heading error comparison for a set of Bézier curves and arc-spline trajectory for case 2

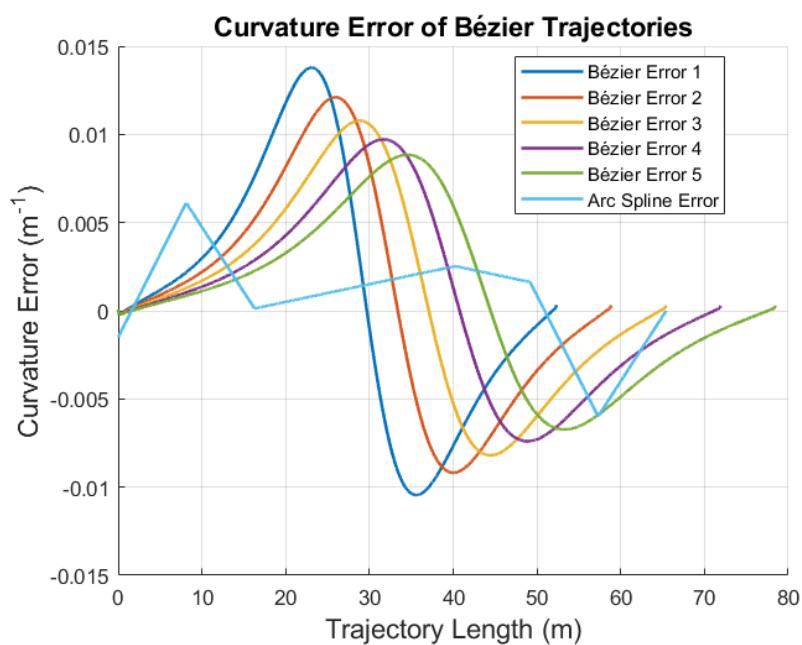


Figure 4.36: Curvature error comparison for a set of Bézier curves and arc-spline trajectory for case 2

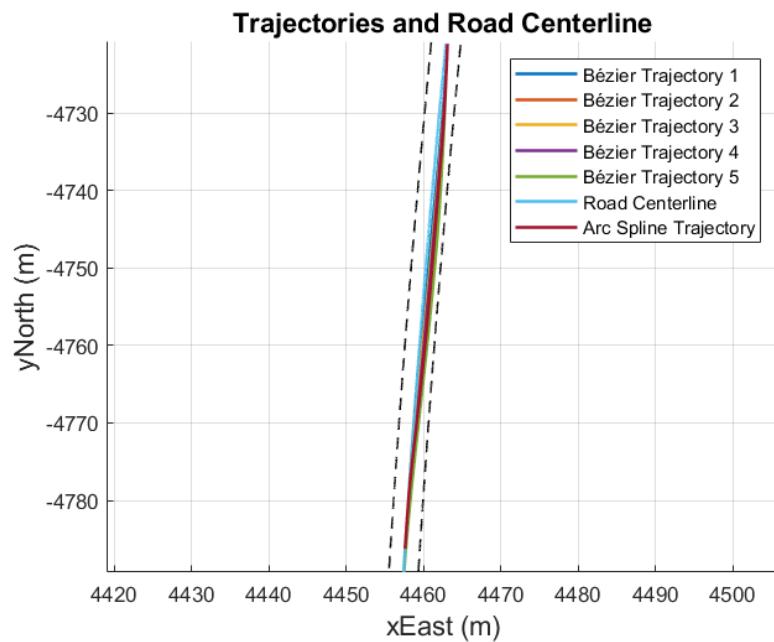


Figure 4.37: Comparison of Bézier trajectories and arc-splines with lane boundaries on the road for case 2

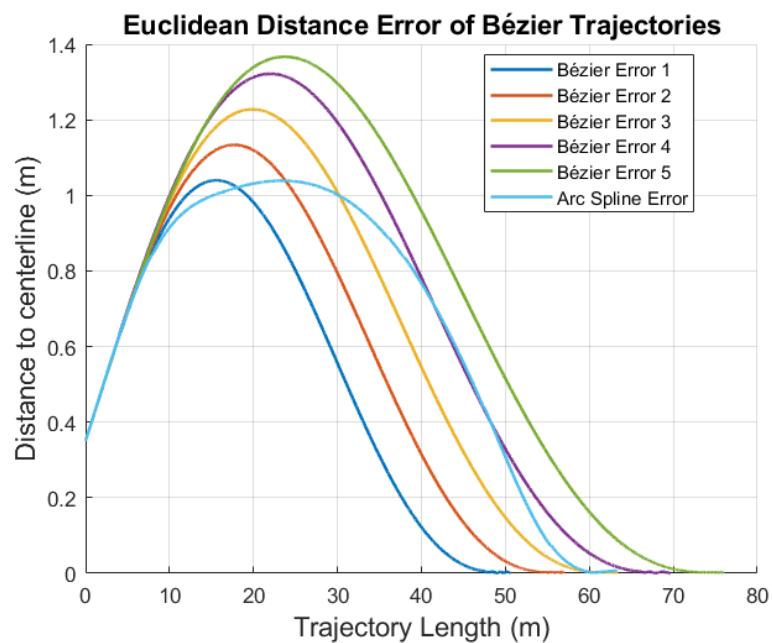


Figure 4.38: Position error comparison for a set of Bézier curves and arc-spline trajectory for case 3

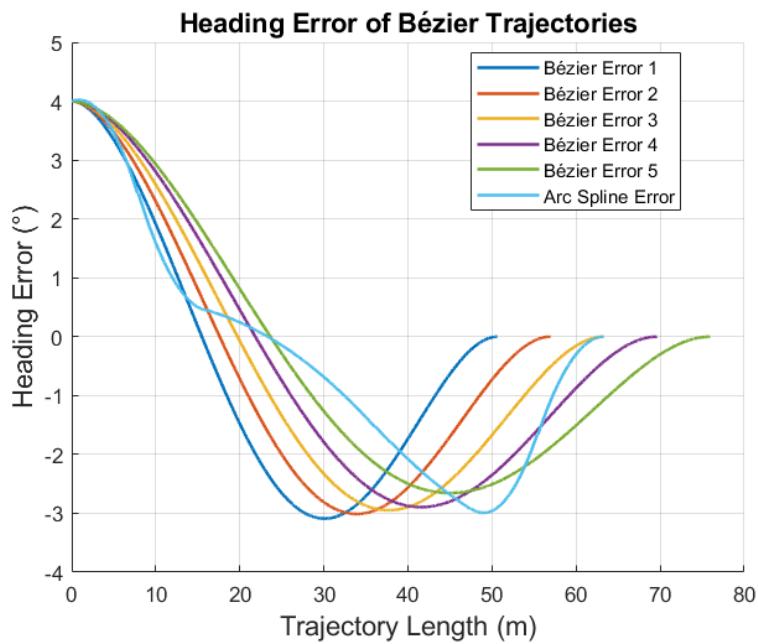


Figure 4.39: Heading error comparison for a set of Bézier curves and arc-spline trajectory for case 3

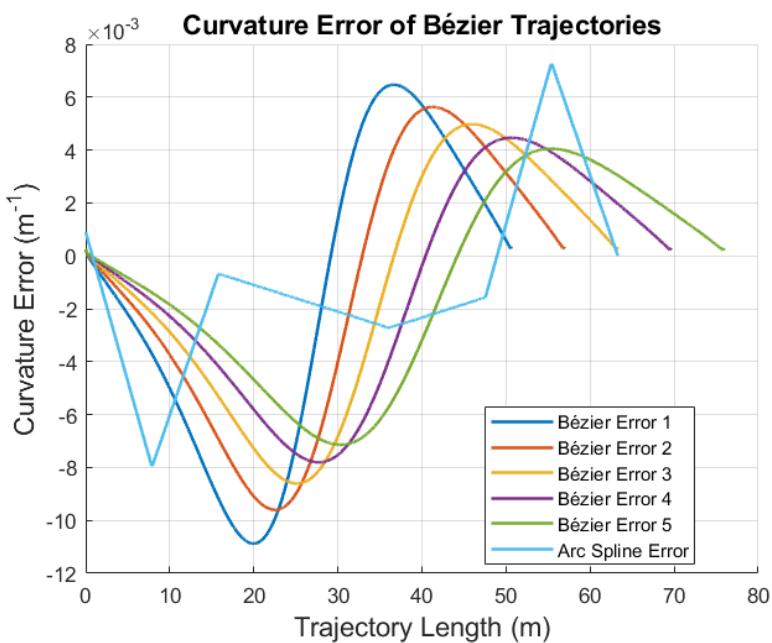


Figure 4.40: Curvature error comparison for a set of Bézier curves and arc-spline trajectory for case 3

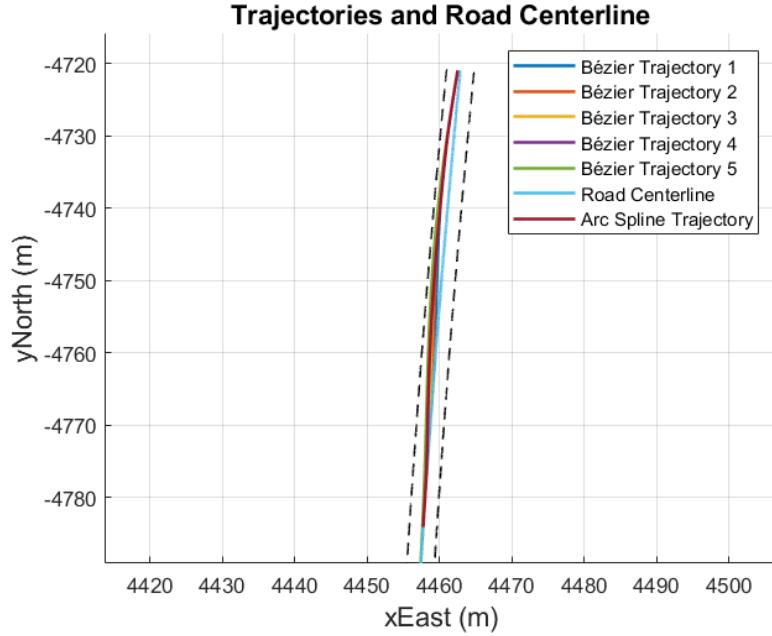


Figure 4.41: Comparison of Bézier trajectories and arc-splines with lane boundaries on the road for case 3

It is evident that arc-spline trajectories directly produce suitable trajectories, with errors decreasing to zero. In contrast, Bézier curves often exhibit either high positional errors or high curvature values combined with high curvature rates. arc-spline trajectories generally have lower curvature rates compared to Bézier curves, resulting in a more comfortable travel experience. To achieve an optimal Bézier curve, multiple iterations with different curve lengths are required, which can lead to computational time issues in real-time applications. Conversely, arc-splines offer a single solution without requiring iterative operations, making them more efficient and reliable for real-time trajectory planning.

In autonomous vehicles, the ability to process and react to information in real-time is important for ensuring safety and efficiency. This Subsection examines the computation time required for generating arc-spline trajectories versus Bézier trajectories. Real-time application demands quick and efficient computation to allow autonomous vehicles to navigate complex environments seamlessly. Comparing these methods will highlight their suitability for real-time navigation and help determine which algorithm provides the optimal balance between computational efficiency and trajectory

accuracy.

Table 4.7: Computation time for Bézier and arc-spline trajectories

Trajectory Type	Computation time (msec)
Bézier Trajectory	78
Arc-spline Trajectory	5.5

Table 4.7 shows that arc-spline trajectories are computed in 5 milliseconds, whereas Bézier trajectories require approximately 78 milliseconds. The arc-spline trajectory generation is an analytical method, making it significantly faster. In contrast, Bézier curves involve evaluating multiple candidate trajectories against a cost function, which increases computation time.

Figure 4.42 displays the computation time of Bézier and Arc-spline trajectories for the cases given in Tables 4.2, 4.3, 4.4 and 4.5. As observed, Arc-spline computation times are much lower than Bézier curve computation time.

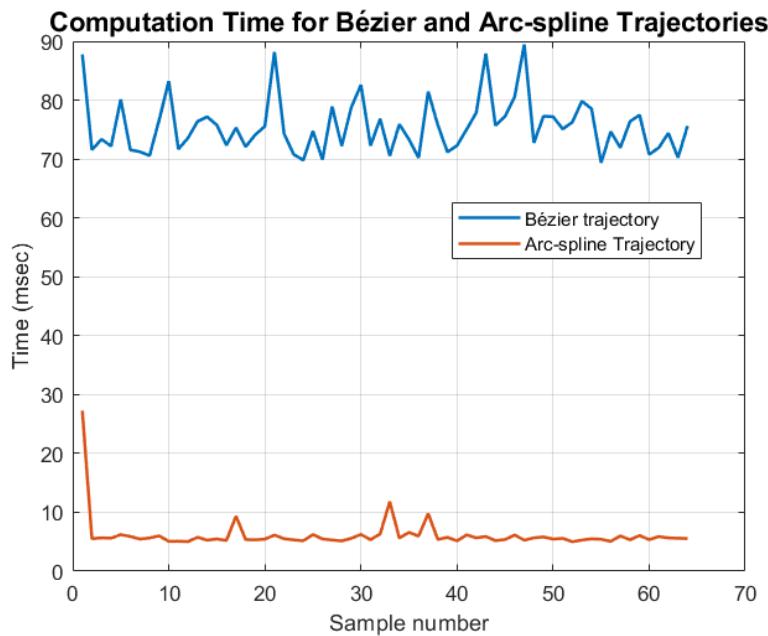


Figure 4.42: Computation time of Bézier and Arc-spline trajectories

4.3.4 Trajectory Simulation

The results in the previous sections are developed under the assumption that clothoid-based trajectories are suitable for the dynamics of autonomous vehicles. In order to justify this assumption, the generated arc-spline trajectories are next validated through a trajectory simulation algorithm to ensure that the ego vehicle can accurately follow the trajectory. The block diagram of this process is presented in Figure 4.43. In this algorithm, the trajectory serves as the input, providing reference position, heading and curvature information for the simulation. The simulation initiates with a pre-defined initial velocity, set at 130, km/h, which is the standard speed for highways.

The simulation algorithm also includes a standard sedan vehicle dynamics. Errors are fed into a PID controller. The details of the controller are given in [40].

Case studies are conducted with the same cases as previous Sections.

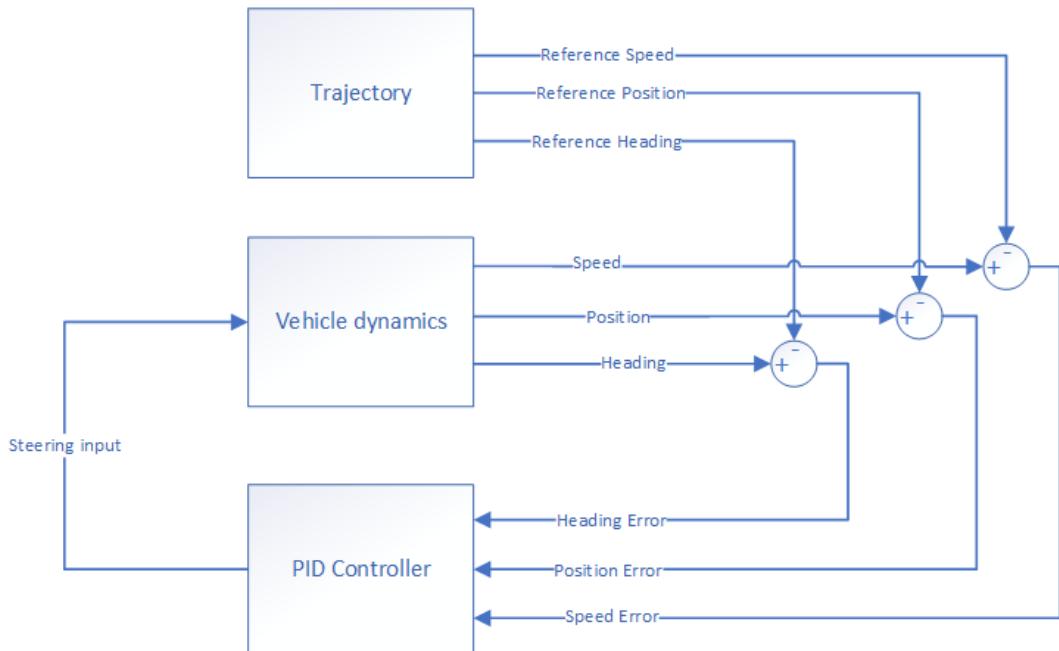


Figure 4.43: Trajectory simulation block diagram

Figures 4.44, 4.48 and 4.52 display the generated trajectory and the actual trajectory that is followed by the ego vehicle. For evaluation, inspecting these Figures are not enough. We also need to consider the vehicle limitations.

Figure 4.45 shows the Euclidian distance error between the trajectory and the simu-

lation. As observed trajectory following error is in the order of 3 centimeters. Figure 4.46 shows the speed limit over the trajectory depending on the current curvature of the road and vehicle dynamics. Even if the curvature increases at some points of the trajectory, the vehicle does not exceed the maximum allowed speed. Similarly Figure 4.47 may be observed for acceleration limits. Acceleration limit is not exceeded as well which is a metric for safety and comfort. The acceleration and speed limits for a car depend on the maximum available traction forces on the road as described in [41].

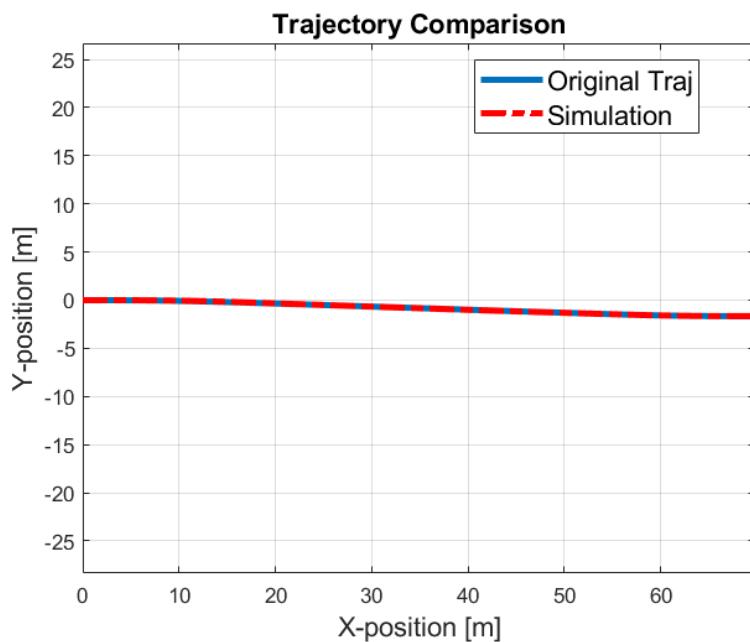


Figure 4.44: Reference trajectory and the actual trajectory that is followed by vehicle for case 1

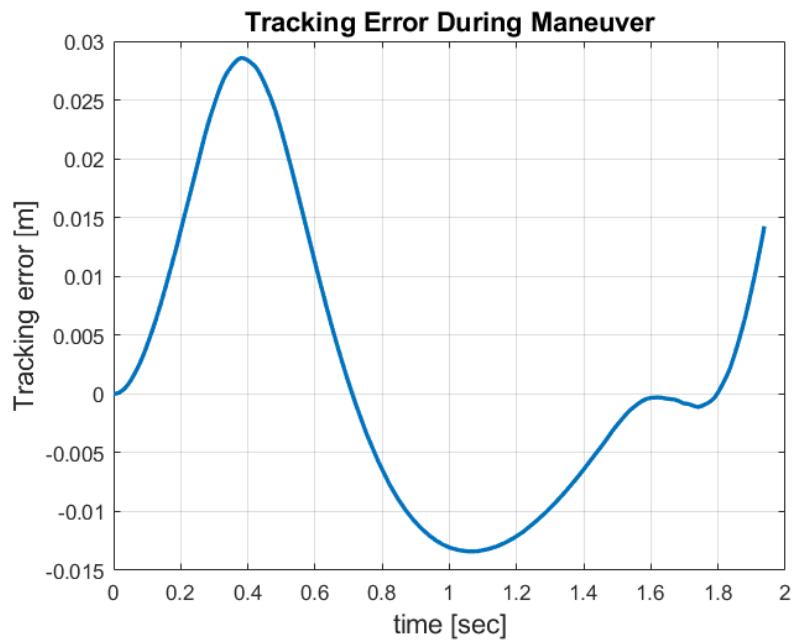


Figure 4.45: Position error of followed trajectory for case 1

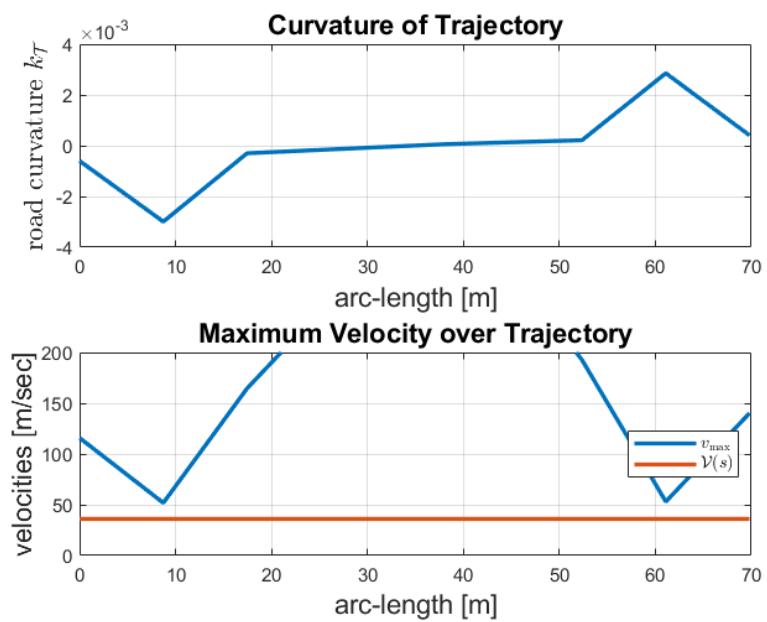


Figure 4.46: Curvature plot of trajectory and maximum velocity allowed over trajectory for case 1

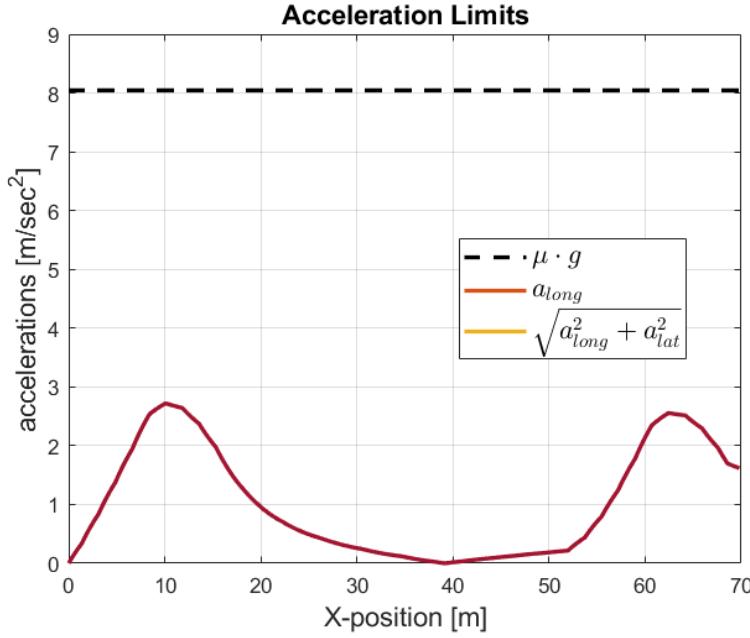


Figure 4.47: Acceleration limit and the actual acceleration of the vehicle over trajectory for case 1

Figure 4.49 shows the Euclidian distance error between the trajectory and the simulation. As observed trajectory following error is in the order of 8 centimeters. Figure 4.50 shows the speed limit over the trajectory depending on the current curvature of the road and vehicle dynamics. Even if the curvature increases at some points of the trajectory, the vehicle does not exceed the maximum allowed speed. Similarly Figure 4.51 may be observed for acceleration limits. Acceleration limit is not exceeded as well which is a metric for safety and comfort.

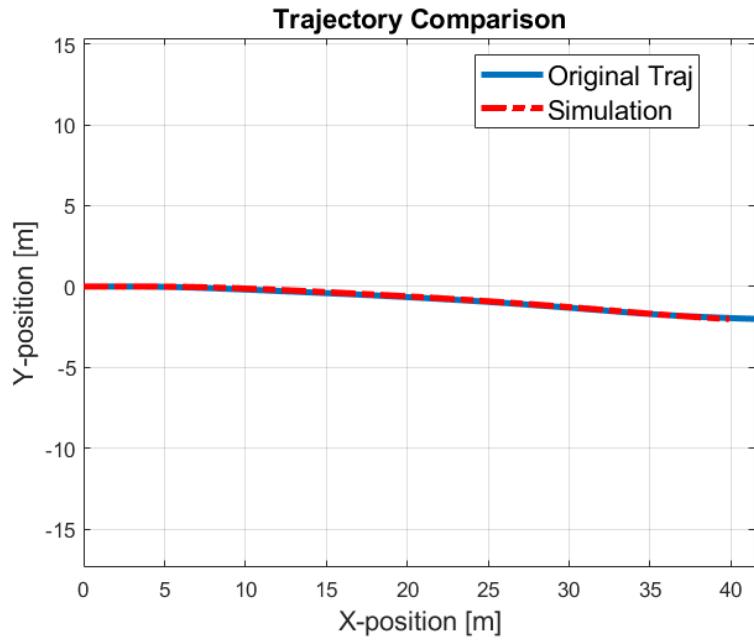


Figure 4.48: Reference trajectory and the actual trajectory that is followed by vehicle for case 2

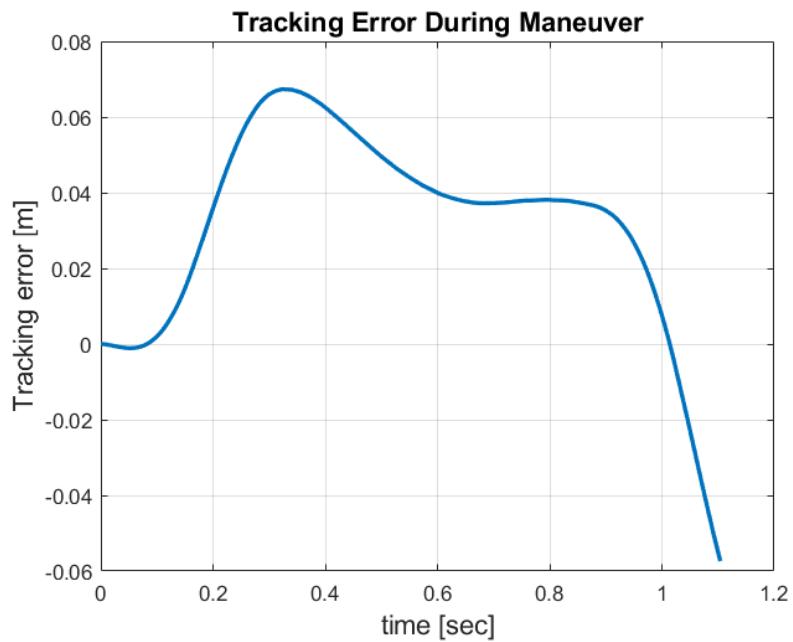


Figure 4.49: Position error of followed trajectory for case 2

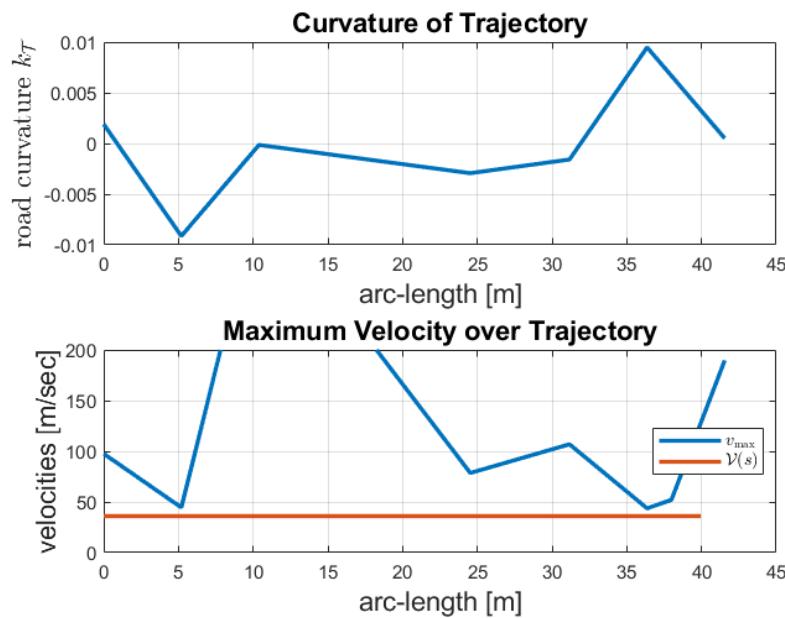


Figure 4.50: Curvature plot of trajectory and maximum velocity allowed over trajectory for case 2

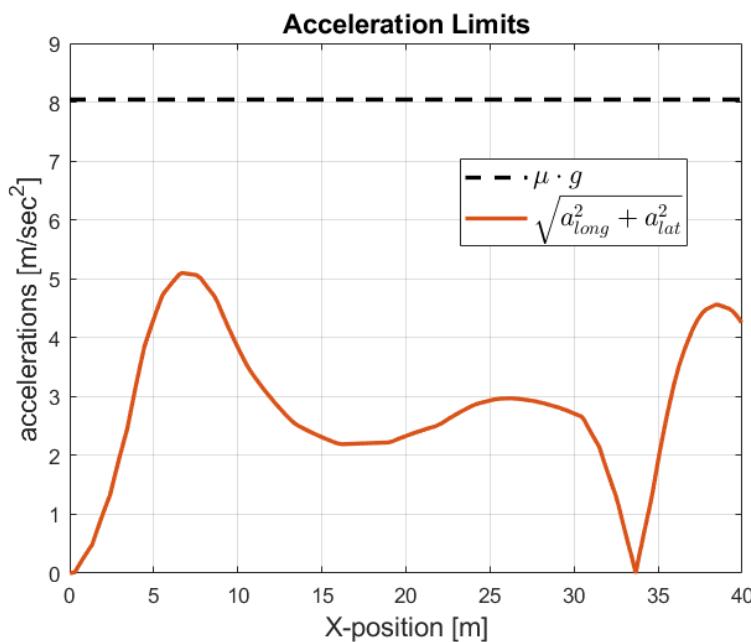


Figure 4.51: Acceleration limit and the actual acceleration of the vehicle over trajectory for case 2

Figure 4.53 shows the Euclidian distance error between the trajectory and the simulation. As observed trajectory following error is at most 18 centimeters. This error is tolerable in highway conditions considering the initial errors. Figure 4.54 shows the speed limit over the trajectory depending on the current curvature of the road and vehicle dynamics. In this case, the vehicle gets close to upper limits due to sharp maneuver. Similarly Figure 4.55 may be observed for acceleration limits. Acceleration limit also gets closer to upper limit.

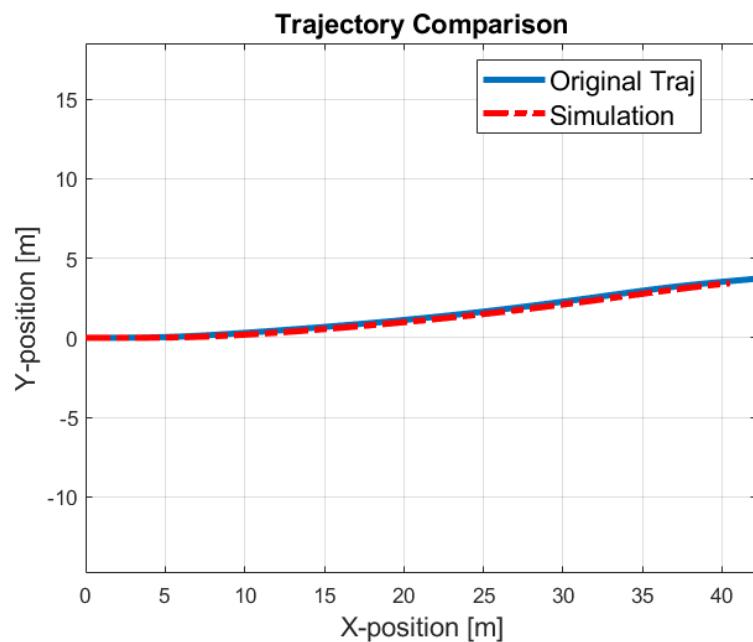


Figure 4.52: Reference trajectory and the actual trajectory that is followed by vehicle for case 3

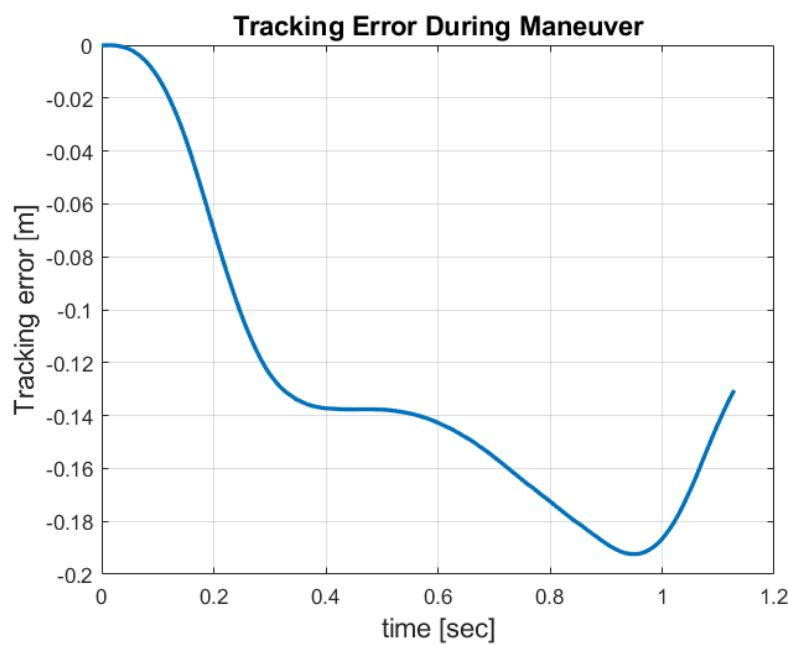


Figure 4.53: Position error of followed trajectory for case 3

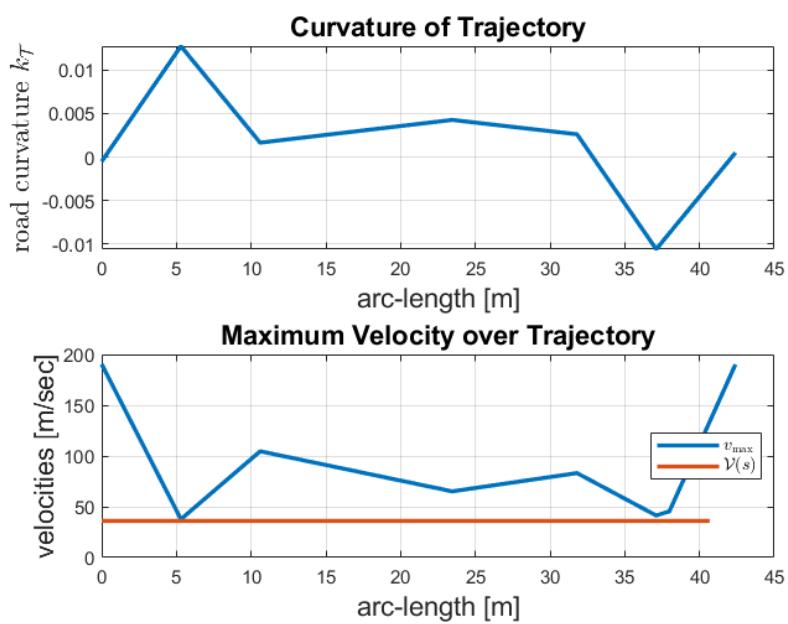


Figure 4.54: Curvature plot of trajectory and maximum velocity allowed over trajectory for case 3

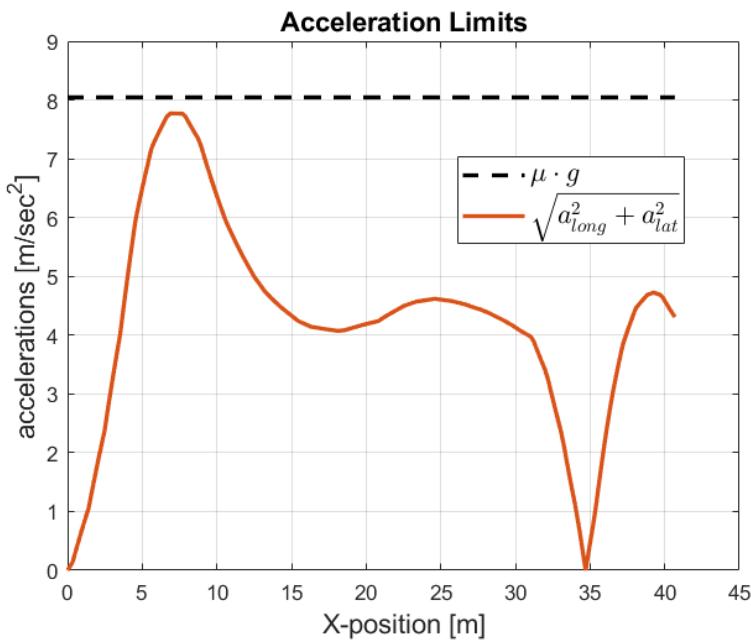


Figure 4.55: Acceleration limit and the actual acceleration of the vehicle over trajectory for case 3

The figures illustrate the position error, as well as the speed and acceleration limits for a vehicle. The simulation demonstrates that the generated trajectories can be followed by the vehicle while remaining within safe speed and acceleration limits. Additionally, it shows that the vehicle can accurately follow the trajectory in terms of position.

CHAPTER 5

CONCLUSION

This thesis has significantly improved road modeling and trajectory planning for autonomous vehicles specifically designed for highway environments. Our research considers two important aspects in the common framework of clothoid-based curves: firstly, how to represent the road itself (modeling), and secondly, how to plan the optimal path for the vehicle to follow the road (trajectory planning). Since clothoid curves do not have an analytical representation, we use arc-spline approximations for computational efficiency.

The first main contribution of the thesis is a novel approach for highway modeling using arc-splines. Since arc-splines are approximations of clothoids, they align well with highways due to their inherent ability to represent clothoid curves, a fundamental component of highway design. Our methodology facilitates the creation of accurate road representations with a minimal set of parameters. The focus on parameter minimization translates to reduced computational complexity, ensuring faster processing times for real-world applications. Additionally, predefined error metrics were implemented to achieve a balance between model accuracy and the number of required parameters. This ensures the generated road model faithfully reflects the actual highway while maintaining computational efficiency.

Beyond parameter minimization, our road modeling approach offers further benefits. The proposed method facilitates the concatenation of multiple, smaller road segments into single, larger ones wherever feasible. This concatenation process further reduces the overall number of parameters needed to represent the entire highway. Furthermore, our methodology leverages an inherent property of arc-splines to generate additional lanes from a single-lane definition by simple parallel-shifting. This capability

proves particularly valuable for highways with multiple lanes in each direction. Road modeling experiments with ground truth data from real high-definition maps demonstrate that our approach offers a comprehensive and efficient solution for modeling highways in autonomous vehicle applications.

The trajectory planning algorithm we developed seamlessly complements our arc-spline-based road modeling method. Similar to the road model, the trajectory planning algorithm utilizes the analytical nature of arc-splines. This translates to low computational requirements and reduced execution time, making it well-suited for real-time implementation on autonomous vehicles. Moreover, the algorithms are easily integrated by aligning the road model and the trajectory on arc-splines. This guarantees a smooth and precise arrival at the desired location, enhancing the overall safety and reliability of autonomous driving.

To provide a more comprehensive analysis and highlight the potential of arc-splines, this study also implements a Bézier curve-based trajectory planning algorithm. The comparative analysis between the two approaches highlights the strengths of using arc-splines. By demonstrating the ability to achieve zero error at the destination point and the inherent efficiency of the analytical approach, this thesis establishes arc-splines as a strong contender for trajectory planning in autonomous vehicles, particularly for highway environments.

In conclusion, this thesis presents an advancement in the field of autonomous driving technology. The proposed arc-spline-based road modeling and trajectory planning algorithms offer efficiency in terms of memory usage and computational effort, accuracy, and guaranteed performance. These advancements pave the way for a future where autonomous vehicles can navigate highway environments with greater precision, safety, and reliability. Future research can further explore the integration of this methodology with real-time sensor data for dynamic adaptation to changing road conditions. Additionally, investigating the performance of arc-splines in more complex highway scenarios with features like exits, merges, and intersections will be valuable for further refining the approach.

REFERENCES

- [1] D. Parekh, N. Poddar, A. Rajpurkar, M. Chahal, N. Kumar, G. P. Joshi, and W. Cho, “A Review on Autonomous Vehicles: Progress, Methods and Challenges,” *Electronics*, vol. 11, p. 2162, Jan. 2022. Number: 14 Publisher: Multi-disciplinary Digital Publishing Institute.
- [2] A. Faisal, M. Kamruzzaman, T. Yigitcanlar, and G. Currie, “Understanding autonomous vehicles,” *Journal of transport and land use*, vol. 12, no. 1, pp. 45–72, 2019.
- [3] K. Jo and M. Sunwoo, “Generation of a Precise Roadway Map for Autonomous Cars,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, pp. 925–937, June 2014. Conference Name: IEEE Transactions on Intelligent Transportation Systems.
- [4] B. Soorchaei, M. Razzaghpoor, R. Valiente Romero, A. Raftari, and Y. Falalah, “High-Definition Map Representation Techniques for Automated Vehicles,” *Electronics*, vol. 11, p. 3374, Oct. 2022.
- [5] G. Abdelkader, T. Alghamdi, K. Elgazzar, and A. Khamis, “Hd maps for connected and automated vehicles: Enabling technologies and future directions,” in *2023 IEEE International Conference on Smart Mobility (SM)*, pp. 91–97, IEEE, 2023.
- [6] G. Elghazaly, R. Frank, S. Harvey, and S. Safko, “High-definition maps: Comprehensive survey, challenges and future perspectives,” *IEEE Open Journal of Intelligent Transportation Systems*, 2023.
- [7] S. Teng, X. Hu, P. Deng, B. Li, Y. Li, Y. Ai, D. Yang, L. Li, Z. Xuanyuan, F. Zhu, *et al.*, “Motion planning for autonomous driving: The state of the art and future perspectives,” *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 6, pp. 3692–3711, 2023.

- [8] B. Bučko, K. Zábovská, J. Ristvej, and M. Jánošíková, “HD Maps and Usage of Laser Scanned Data as a Potential Map Layer,” in *2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO)*, pp. 1670–1675, Sept. 2021. ISSN: 2623-8764.
- [9] L. Liu, T. Wu, Y. Fang, T. Hu, and J. Song, “A smart map representation for autonomous vehicle navigation,” in *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pp. 2308–2313, Aug. 2015.
- [10] X. Li, Z. Sun, D. Cao, Z. He, and Q. Zhu, “Real-Time Trajectory Planning for Autonomous Urban Driving: Framework, Algorithms, and Verifications,” *IEEE/ASME Transactions on Mechatronics*, vol. 21, pp. 740–753, Apr. 2016. Conference Name: IEEE/ASME Transactions on Mechatronics.
- [11] L. Garach, J. De Oña, and M. Pasadas, “Mathematical formulation and preliminary testing of a spline approximation algorithm for the extraction of road alignments,” *Automation in Construction*, vol. 47, pp. 1–9, Nov. 2014.
- [12] S. Brummer, F. Janda, G. Maier, and A. Schindler, “Evaluation of a mapping strategy based on smooth arc splines for different road types,” in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pp. 160–165, Oct. 2013. ISSN: 2153-0017.
- [13] V. Gikas and J. Stratakos, “A Novel Geodetic Engineering Method for Accurate and Automated Road/Railway Centerline Geometry Extraction Based on the Bearing Diagram and Fractal Behavior,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, pp. 115–126, Mar. 2012. Conference Name: IEEE Transactions on Intelligent Transportation Systems.
- [14] S. Vitalis, A. Labetski, H. Ledoux, and J. Stoter, “From road centrelines to carriageways—A reconstruction algorithm,” *PLOS ONE*, vol. 17, no. 2, p. e0262801, 2022. Publisher: Public Library of Science.
- [15] A. Jafari, A. Both, D. Singh, L. Gunn, and B. Giles-Corti, “Building the road network for city-scale active transport simulation models,” *Simulation Modelling Practice and Theory*, vol. 114, p. 102398, Jan. 2022.

- [16] J. Godoy, A. Artuñedo, and J. Villagra, “Self-Generated OSM-Based Driving Corridors,” *IEEE Access*, vol. PP, pp. 1–1, Feb. 2019.
- [17] M. Hentschel and B. Wagner, “Autonomous robot navigation based on OpenStreetMap geodata,” in *13th International IEEE Conference on Intelligent Transportation Systems*, pp. 1645–1650, Sept. 2010. ISSN: 2153-0017.
- [18] B. Suger and W. Burgard, “Global outer-urban navigation with OpenStreetMap,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1417–1422, May 2017.
- [19] J. Horst and A. Barbera, “Trajectory generation for an on-road autonomous vehicle - art. no. 62302J,” *Proc SPIE*, p. 82, June 2006.
- [20] S. Vaskov, U. Sharma, S. Kousik, M. Johnson-Roberson, and R. Vasudevan, “Guaranteed Safe Reachability-based Trajectory Design for a High-Fidelity Model of an Autonomous Passenger Vehicle,” in *2019 American Control Conference (ACC)*, pp. 705–710, July 2019. ISSN: 2378-5861.
- [21] B. Park, Y.-C. Lee, and W.-Y. Han, “Trajectory generation method using Bézier spiral curves for high-speed on-road autonomous vehicles,” *IEEE International Conference on Automation Science and Engineering*, vol. 2014, pp. 927–932, Aug. 2014.
- [22] Y. Irie and D. Akasaka, “Trajectory Generation Using Model Predictive Control for Automated Vehicles: -Maximize Usage of Road Width Like A Skilled Human Driver-,” *International Journal of Automotive Engineering*, vol. 12, pp. 24–31, Feb. 2021.
- [23] N. Dang, Z. Zhang, J. Liu, M. Leibold, and M. Buss, “Incorporating Target Vehicle Trajectories Predicted by Deep Learning Into Model Predictive Controlled Vehicles,” Oct. 2023. arXiv:2310.02843 [cs].
- [24] C. Liu, S. Lee, S. Varnhagen, and H. E. Tseng, “Path planning for autonomous vehicles using model predictive control,” in *2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 174–179, June 2017.

- [25] Y. Rasekhipour, A. Khajepour, S.-K. Chen, and B. Litkouhi, “A Potential Field-Based Model Predictive Path-Planning Controller for Autonomous Road Vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, vol. PP, pp. 1–13, Sept. 2016.
- [26] P. Farkas, L. Szoke, and S. Aradi, “Defining metrics for scenario-based evaluation of autonomous vehicle models,” in *2022 IEEE 1st International Conference on Cognitive Mobility (CogMob)*, pp. 000155–000160, Oct. 2022.
- [27] D. Zhang and B. Chen, “Path Planning and Predictive Control of Autonomous Vehicles for Obstacle Avoidance,” pp. 1–6, Nov. 2022.
- [28] E. Zhang, R. Zhang, and N. Masoud, *Predictive Trajectory Planning for Autonomous Vehicles at Intersections Using Reinforcement Learning*. Feb. 2023.
- [29] Z. Wang, J. Tu, and C. Chen, “Reinforcement Learning based Trajectory Planning for Autonomous Vehicles,” in *2021 China Automation Congress (CAC)*, pp. 7995–8000, Oct. 2021. ISSN: 2688-0938.
- [30] A. Chaulwar, H. Al-Hashimi, M. Botsch, and W. Utschick, “Efficient Hybrid Machine Learning Algorithm for Trajectory Planning in Critical Traffic Scenarios,” pp. 196–202, Sept. 2019.
- [31] S. Grigorescu, B. Trasnea, L. Marina, A. Vasilcoi, and T. Cocias, “NeuroTrajectory: A Neuroevolutionary Approach to Local State Trajectory Learning for Autonomous Vehicles,” *IEEE Robotics and Automation Letters*, vol. PP, pp. 1–1, July 2019.
- [32] E. Bertolazzi and M. Frego, “Fast and accurate G1 fitting of clothoid curves,” June 2013.
- [33] M. E. Mortenson, *Mathematics for Computer Graphics Applications*. Industrial Press Inc., 1999. Google-Books-ID: YmQy799flPkC.
- [34] A. Artuñedo, J. Villagra, and J. Godoy, “Real-Time Motion Planning Approach for Automated Driving in Urban Environments,” *IEEE Access*, vol. 7, pp. 180039–180053, Dec. 2019.

- [35] A. Scheuer and T. Fraichard, “Collision-free and continuous-curvature path planning for car-like robots,” in *Proceedings of International Conference on Robotics and Automation*, vol. 1, pp. 867–873 vol.1, Apr. 1997.
- [36] S. Kousik, S. Vaskov, M. Johnson-Roberson, and R. Vasudevan, “Safe Trajectory Synthesis for Autonomous Driving in Unforeseen Environments,” Apr. 2017.
- [37] J. Funke and J. Gerdes, “Simple Clothoid Lane Change Trajectories for Automated Vehicles Incorporating Friction Constraints,” *Journal of Dynamic Systems Measurement and Control*, vol. 138, Feb. 2016.
- [38] A. Kahiya and A. H. Mohammed, “Optimal control problems for safe and efficient lane changes of self-driving vehicles,” Master’s thesis, Sept. 2017. Accepted: 2022-04-04T12:44:39Z Journal Abbreviation: Otonom araçların emniyetli ve verimli şerit değişikliklerine yönelik optimal kontrol problemleri.
- [39] K. Wong, Y. Gu, and S. Kamijo, “Mapping for Autonomous Driving: Opportunities and Challenges,” *IEEE Intelligent Transportation Systems Magazine*, vol. 13, no. 1, pp. 91–106, 2021. Conference Name: IEEE Intelligent Transportation Systems Magazine.
- [40] R. Marino, S. Scalzi, and M. Netto, “Nested PID steering control for lane keeping in autonomous vehicles,” *Control Engineering Practice*, vol. 19, pp. 1459–1467, Dec. 2011.
- [41] F. Wei, Y. Guo, P. Liu, Z. Cai, Q. Li, and L. Chen, “Modeling Car-Following Behaviour of Turning Movements at Intersections with Consideration of Turning Radius,” *Journal of Advanced Transportation*, vol. 2020, pp. 1–9, Aug. 2020.