LANE CHANGE PREDICTION FOR AUTONOMOUS VEHICLES USING TRAJECTORY TEMPLATES

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

BY

İSMAİL HAKKI YILDIZ

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

AUGUST 2024

Approval of the thesis:

LANE CHANGE PREDICTION FOR AUTONOMOUS VEHICLES USING TRAJECTORY TEMPLATES

submitted by **İSMAİL HAKKI YILDIZ** 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. Umut Orguner
Electrical and Electronics Engineering, METU _____

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

Prof. Dr. Selahattin Çağlar Başlamışlı
Mechanical Engineering, Hacettepe University _____

Date: 26.08.2024

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: İsmail Hakkı Yıldız

Signature :

ABSTRACT

LANE CHANGE PREDICTION FOR AUTONOMOUS VEHICLES USING TRAJECTORY TEMPLATES

Yıldız, İsmail Hakkı
M.S., Department of Electrical and Electronics Engineering
Supervisor: Prof. Dr. Klaus Werner Schmidt

August 2024, 100 pages

The navigation of autonomous vehicles depends both on the desired motion of the ego-vehicle and on the motion of surrounding traffic such as vehicles. Accordingly, the fast and accurate trajectory prediction of vehicles is crucial for safe and efficient autonomous navigation. This thesis introduces a novel template-based approach for lane change trajectory prediction. The method utilizes clothoid-based bi-elementary paths, a compact and flexible mathematical model, to represent template lane change trajectories. The parameters of these templates are then estimated using least-squares fitting of measured vehicle motion data. In our method, we employ an efficient approximation of bi-elementary paths by arc-splines, that is, sequences of arc segments. The main advantage of this approximation is the representation of lane change trajectories by a very small number of parameters, which reduces the computational complexity of curve fitting. An additional advantage is the direct applicability of our method to lane changes on both straight and curved roads through appropriate coordinate transformations. A comprehensive evaluation on synthetic data as well as the real-world NGSIM dataset, using metrics like Root Mean Squared Error (RMSE) in spatial and temporal coordinates, demonstrates the method's competitive accuracy

and superior computational efficiency compared to state-of-the-art methods. Our approach is further characterized by its interpretability and real-time performance and, hence, offers a promising solution for enhancing the safety and decision-making capabilities of autonomous vehicles.

Keywords: Lane change prediction, Trajectory template, Autonomous vehicles, Clothoid, Curve fitting

ÖZ

YÖRÜNGE ŞABLONLARI KULLANARAK OTONOM ARAÇLAR İÇİN ŞERİT DEĞİŞTİRME TAHMİNİ

Yıldız, İsmail Hakkı
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Klaus Werner Schmidt

Ağustos 2024 , 100 sayfa

Otonom araçların navigasyonu, hem ego-aracın istenen hareketine hem de çevredeki trafiğin hareketine bağlıdır. Bu nedenle, çevredeki araçların hızlı ve doğru yörünge tahmini, güvenli ve verimli otonom navigasyon için kritik öneme sahiptir. Bu çalışma, otonom araç navigasyonunda şerit değiştirme yörüngelerinin tahmini için yeni bir model-tabanlı yaklaşım sunmaktadır. Yöntem, şerit değiştirme yörüngelerinin klotoid-tabanlı 'bi-elementary path' modeli olarak adlandırılan yörünge şablonları kullanılarak temsil edilmesine dayanmaktadır. Bu şerit değiştirme yörüngesi şablonlarının parametreleri daha sonra en küçük kareler yöntemi ile eğri uydurma uygulanarak tahmin edilmektedir. Yöntemimizde, 'bi-elementary path' yörüngeleri 'arc-splines' denilen yay segmentleriyle etkili bir şekilde yaklaşık temsil edilmektedir. Methodumuzun temel avantajı, şerit değiştirme yörüngelerini çok az sayıda parametre ile temsil edebilmesi ve böylece eğri uydurma işleminin hesaplama karmaşıklığının önemli ölçüde azalmasıdır. Bir diğer avantajı ise, uygun koordinat dönüşümleri aracılığıyla hem düz hem de eğimli yollardaki şerit değiştirmelerine doğrudan uygulanabilir olmasıdır. Yöntemin etkinliği, gerçek dünya ölçümlerine dayanan

'Next Generation SIMulation of Mobility (NGSIM)' veri seti kullanılarak değerlendirilmiştir. Burada, tahmin doğruluğunu değerlendirmek için çeşitli tahmin ufukları için konumsal ve zamansal koordinatlardaki Kare Kök Ortalama Hata (RMSE) gibi performans ölçütleri kullanılmıştır. Sonuçlar, yöntemin son teknoloji yöntemlere kıyasla rekabetçi doğruluğunu ve üstün hesaplama verimliliğini göstermektedir. Yorumlanabilirliği ve gerçek zamanlı performansı ile karakterize edilen bu yaklaşım, otonom araçların güvenliğini ve karar verme yeteneklerini geliştirmek için umut verici bir çözüm sunmaktadır.

Anahtar Kelimeler: Şerit değiştirme tahmini, Yörünge şablonu, Otonom araçlar, Klotoid, Eğri uydurma

To my beloved ones...

ACKNOWLEDGMENTS

First and foremost, I would like to express my deepest gratitude to my dear wife, Büşra, for her endless love, support, and patience throughout my thesis journey. Her encouragement and understanding were invaluable in helping me overcome challenges and stay motivated.

I am also deeply grateful to my supervisor, Prof. Dr. Klaus Werner Schmidt, for his exceptional guidance, mentorship, and insightful feedback.

I extend my sincere appreciation to my company, Roketsan, for their support of my academic career and for providing me the necessary time off to focus on my thesis. I'm particularly grateful to my colleagues for their understanding throughout this process.

Finally, I also thank my dear friends, Ömer and Atakan, for their friendship, support, and discussions during the thesis journey.

TABLE OF CONTENTS

ABSTRACT v

ÖZ vii

ACKNOWLEDGMENTS x

TABLE OF CONTENTS xi

LIST OF TABLES xiii

LIST OF FIGURES xiv

LIST OF ABBREVIATIONS xvii

CHAPTERS

1 INTRODUCTION 1

2 BACKGROUND INFORMATION 7

2.1 Lane Change Trajectory Templates 7

2.1.1 Clothoids 7

2.1.2 Clothoid Approximations using Arc-Splines 9

2.1.3 Bi-elementary Path 12

2.2 Coordinate Frames and Transformations 17

2.2.1 Transformation between Body frame and Cartesian frame Frame 18

2.2.2 Transformation between Cartesian and Frenet Frames 20

2.3 NGSIM Dataset 23

2.4 Least Squares Curve Fitting 25

3 METHODOLOGY 29

3.1 Lane Change Trajectories and Bi-elementary Paths 29

3.1.1 Trajectory Measurements 30

3.1.2 Illustration of Trajectory Behaviour with Time 33

3.2 Lane Change Trajectory Prediction using Least Squares Curve Fitting 35

3.2.1 Algorithmic Steps of the Prediction 35

3.2.2 Transformation from Spatial Coordinates to Temporal Coordinates 52

4 EVALUATION OF THE METHOD 61

4.1 Evaluation Metrics 61

4.2 Evaluation with Synthetic Trajectories 64

4.3 Evaluation with NGSIM Dataset 73

4.3.1 Parameter Tuning 75

4.3.2 Comparison with Straight-Line Model 79

4.3.3 Evaluation of Velocity Models 85

4.3.4 Comparison with the Existing Studies 89

4.4 Discussion 91

5 CONCLUSION 93

REFERENCES 95

LIST OF TABLES

TABLES

Table 4.1 Parameter Estimation Errors for a Synthetic Left Lane Change Trajectory with $w = 0$ 67

Table 4.2 Prediction Errors for a Synthetic Trajectory with $w = 0$ 67

Table 4.3 Parameter Estimation Errors for a Synthetic Trajectory with $w = 10$ cm 69

Table 4.4 Prediction Errors for a Synthetic Trajectory with $w = 10$ cm 69

Table 4.5 Overall Estimation Errors for Synthetic Trajectories 71

Table 4.6 Parameter comparison for heading threshold Φ_{th} 76

Table 4.7 Parameter comparison for lane offset threshold LO_{th} 77

Table 4.8 Parameter comparison for lateral displacement margin $m_{\Delta Y}$ 78

Table 4.9 Performance comparison of the bi-elementary path and straight-line models for lane change trajectory prediction on the NGSIM dataset across different prediction horizons 83

Table 4.10 Performance comparison of different velocity models for lane change trajectory prediction on the NGSIM dataset 86

Table 4.11 Comparison with the existing trajectory prediction methods on the NGSIM dataset 90

LIST OF FIGURES

FIGURES

Figure 2.1 Basic lane change maneuver. 7

Figure 2.2 Illustration of a clothoid. 8

Figure 2.3 Clothoid $\mathcal{C}(0, 0, 0, -0.02, 100)$ 9

Figure 2.4 Clothoid $\mathcal{C}(0, 0, 0, 0.5, 10)$ approximation, n=5 11

Figure 2.5 Illustration of a bi-elementray path 13

Figure 2.6 Illustration of a bi-elementray path curvature change 13

Figure 2.7 $\mathcal{B}(S, \Delta Y, \lambda, \gamma)$ for $S = 100$, $\Delta Y = 3.7$, $\gamma = 1$ and different λ . 15

Figure 2.8 $\mathcal{B}(S, \Delta Y, \lambda, \gamma)$ for $S = 100$, $\Delta Y = 3.7$, $\lambda = 0.5$ and different γ . 16

Figure 2.9 $\mathcal{B}(S, \Delta Y, \lambda, \gamma)$ for $S = 100$, $\lambda = 0.5$, $\gamma = 1$ and different ΔY . 16

Figure 2.10 $\mathcal{B}(S, \Delta Y, \lambda, \gamma)$ for $\Delta Y = 3.7$, $\lambda = 0.5$, $\gamma = 1$ and different S . . 17

Figure 2.11 Frenet, Body and Cartesian Coordinate Frames 18

Figure 2.12 Cartesian and body frames 19

Figure 2.13 Cartesian and Frenet Frames 21

Figure 2.14 I-80 24

Figure 2.15 US-101 24

Figure 2.16 Linear and Nonlinear Least Squares Curve Fittings 27

Figure 3.1 Starting accumulating measurements. 31

Figure 3.2 Accumulating measurements until the vehicle disappears. 31

Figure 3.3 Vehicle ID: 159 in I-80 Highway, Position and Velocity Data . . 32

Figure 3.4 Vehicle ID: 76 in US101 Highway, Position and Velocity Data . 32

Figure 3.5 Measurements and Trajectories for t=1 33

Figure 3.6 Measurements and Trajectories for t=3 34

Figure 3.7 Measurements and Trajectories for t=5 34

Figure 3.8 Suitable and unsuitable fitting data for straight sections 36

Figure 3.9 Suitable and unsuitable fitting data for a double lane change scenario 37

Figure 3.10 Target lanes for the vehicle in close proximity to the right boundary 39

Figure 3.11 Target lanes for the vehicle not in close proximity to the right boundary 40

Figure 3.12 Left lane change trajectory at t=1 s 42

Figure 3.13 Left lane change trajectory and the prediction at t=1 s 43

Figure 3.14 Predicted position errors at t=1 s for the left lane change trajectory 44

Figure 3.15 Left lane change trajectory at t=3 s 45

Figure 3.16 Left lane change trajectory and the prediction at t=3 s 46

Figure 3.17 Predicted position errors at t=3 s for the left lane change trajectory 46

Figure 3.18 Right lane change trajectory at t=1 s 47

Figure 3.19 Right lane change trajectory and the prediction at t=1 s 48

Figure 3.20 Predicted position errors at t=1 s for the right lane change trajectory 49

Figure 3.21 Right lane change trajectory and the prediction at t=6 s 50

Figure 3.22 Predicted position errors at t=6 s for the right lane change trajectory 50

Figure 3.23 Predicted velocities for different velocity models for the left lane change example at t=1 s 56

Figure 3.24 Predicted positions and RMS errors for the left lane change trajectory at t=1 s 57

Figure 3.25 Predicted velocities for different velocity models for the right lane change example at t=1 s 58

Figure 3.26 Predicted positions and RMS errors for the right lane change trajectory at t=1 s 59

Figure 4.1 Generated left lane trajectories 65

Figure 4.2 Generated right lane trajectories 65

Figure 4.3 Predictions for a Synthetic Trajectory with $w = 0$ 68

Figure 4.4 Predictions for a Synthetic Trajectory with $w = 10$ cm 70

Figure 4.5 NGSIM Dataset Trajectories 74

Figure 4.6 Predictions for a right lane change trajectory at t = 1s 79

Figure 4.7 Predictions for a right lane change trajectory at t = 2s 80

Figure 4.8 Predictions for a right lane change trajectory at t = 3s 81

Figure 4.9 Predictions for a right lane change trajectory at t = 4s 81

Figure 4.10 RMSE-T plots for bi-elementary path and straight-line models against prediction horizon 84

Figure 4.11 RMSE-T plots for different velocity models 88

LIST OF ABBREVIATIONS

ADAS	Advanced Driver Assistance Systems
AE	Absolute Error
AV	Autonomous Vehicle
CA	Constant Acceleration
CNN	Convolutional Neural Network
CV	Constant Velocity
DBN	Dynamic Bayesian Network
FN	False Negative
FP	False Positive
GAN	Generative Adversarial Network
GISNet	Graph-based Information Sharing Network
GP	Gaussian Process
GRIP	Graph Representation and Interaction Prediction
HMM	Hidden Markov Model
IMM	Interacting Multiple Model
IMM-KF	Interacting Multiple Model Kalman Filter
LO	Lane Offset
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MLP	Multi-Layer Perceptron
MPC	Model Predictive Control
NGSIM	Next Generation SIMulation
PA	Partial Acceleration
RMSE	Root Mean Square Error

RMSE-S	Root Mean Square Error in Spatial Coordinates
RMSE-T	Root Mean Square Error in Temporal Coordinates
RNN	Recurrent Neural Network
SGAN	Social Generative Adversarial Network
SVM	Support Vector Machine
TP	True Positive

CHAPTER 1

INTRODUCTION

Autonomous driving is attracting growing interest from both academia and industry due to its potential to revolutionize transportation by enhancing safety, reducing congestion, and improving energy efficiency [1, 2, 3, 4]. While significant progress has been made in developing perception, planning, and control systems for autonomous vehicles (AVs) [5, 6], ensuring their safety in complex and dynamic environments remains a critical challenge. Motion planning, a crucial aspect of autonomous driving, relies heavily on accurate predictions of surrounding vehicle trajectories [7, 8]. Among the various maneuvers that autonomous vehicles must anticipate, lane changes are particularly vital due to their dynamic nature and significant impact on traffic patterns [9, 10, 11]. Accurately predicting the lane change trajectories of surrounding vehicles enables autonomous vehicles to prepare for potential hazards, plan optimal trajectories, and ensure smooth traffic flow [12].

Researchers have explored a variety of methodologies to tackle the challenge of lane change trajectory prediction, each with its own set of strengths and limitations [13, 14]. Model-based approaches are based on the fundamental principles of physics and vehicle dynamics. They utilize mathematical models, often derived from kinematic or dynamic equations, to predict future trajectories based on the current state of the vehicle. Simplified kinematic models, such as the Constant Velocity (CV) and Constant Acceleration (CA) models [15, 16], or more sophisticated models that incorporate the yaw angle and yaw rate [17, 18] can be given as examples to this approach. Due to the complex nature of vehicle dynamics, simplified dynamic models are also preferred for trajectory prediction. In [19], the bicycle dynamic model is used to predict the future states of the vehicle. To improve the accuracy of model-based

predictions, researchers have explored combining sensor fusion with parametric interpolation techniques [20]. The proposed work aims to address the limitations of traditional CV/CA models in aggressive scenarios by fitting fused radar and camera data to a parametric polynomial function. However, the simplicity of polynomial functions may limit their ability to fully represent the complex vehicle maneuvers. Another study employed multiple model–based adaptive estimator (MMAE) to infer the lane change intention. Then, it utilizes cubic spline curves as trajectory templates for prediction using linearized recursive least-squares estimation (LRLSE) [21]. However, due to their lack of constraints specific to vehicle dynamics and lane change maneuvers, cubic spline curves can lead to predictions that deviate from realistic vehicle behavior. To deal with real-world noise and uncertainties, Kalman Filtering (KF) methods have been utilized [22]. KF models the uncertainty in vehicle state using a Gaussian distribution, allowing it to estimate average trajectories with associated uncertainty [23]. However, the unimodal Gaussian distribution may not be enough represent the diverse range of possible scenarios. To improve upon this, the Interacting Multiple Model Kalman Filter (IMM-KF) integrates multiple motion models and considers interactions between vehicles [24, 25], but its reliance on predefined models and assumptions about driver behavior can limit its adaptability to complex real-world traffic scenarios. While computationally efficient, these models often simplify the complexities of human driving behavior and may struggle with long-term predictions.

Maneuver-based approaches focus on recognizing and classifying specific lane change patterns utilizing probabilistic models and machine learning techniques. Support Vector Machines (SVMs), for example, can learn and recognize driver maneuvers based on characteristics such as steering wheel angle, position, and acceleration as presented in [26]. Hidden Markov Models (HMM) also have been widely used for maneuver recognition and prediction [27, 28, 29]. By modeling the sequence of events within a maneuver and learning the transition probabilities between them, HMMs can infer likely future trajectories based on observed behavior. To further enhance the accuracy of trajectory prediction, models should ideally consider both vehicle states and the interaction effects between traffic participants. As highlighted by Koller *et al.* [30], Dynamic Bayesian Networks (DBNs) offer a flexible framework for representing

probabilistic relationships between variables involved in maneuver decision-making. DBNs can incorporate various factors, such as vehicle states, road structure, and interactions with other vehicles, to predict the most likely future maneuver [31, 32]. However, the complexity of DBNs can grow significantly as the number of variables and dependencies increases.

Deep learning-based methods have recently revolutionized the field of trajectory prediction, with sophisticated neural network architectures demonstrating impressive capabilities in learning complex patterns and relationships from vast amounts of driving data [33, 34, 35]. Recurrent Neural Networks (RNNs), especially Long Short-Term Memory (LSTM) networks, have been widely adopted due to their ability to process sequential data and model temporal dependencies in trajectories [36, 37]. For example, Ding *et al.* [38] use an LSTM encoder that encodes the states of the target vehicle to predict its maneuver, and then achieve trajectory prediction using the predicted maneuver and map information. As proposed in [39], CS-LSTM method utilize LSTM encoder-decoder architectures with convolutional social pooling to learn interdependencies and generate multi-modal predictions. Convolutional Neural Networks (CNNs) have also been employed to extract spatial features and capture interactions between vehicles [40, 41, 42]. However, most methods applying the CNN framework require a bird's-eye image as input, often infeasible for real-time scenarios. Moreover, the integration of attention mechanisms and graph neural networks (GNNs) has further enhanced the ability to model complex relationships within the traffic environment [43]. Methods like GISNet [44] and GRIP [45] utilize graph representations and convolutional networks to model interactions and extract relevant features for prediction. To account for the inherent multi-modal nature of trajectory distributions, some researchers employ generative models like Generative Adversarial Networks (GANs) to generate multi-modal trajectory predictions, as exemplified by the Spatio-Temporal GAN (TS-GAN) proposed in [46]. While deep learning-based methods have shown great promise, they often come with high computational demands, requiring significant resources and potentially limiting real-time implementation. Additionally, their reliance on large training datasets and their "black-box" nature can create challenges in terms of generalizability and interpretability.

In this thesis, we propose a novel model-based approach for lane change trajectory

prediction that utilizes data-driven optimization to enhance its adaptability and accuracy. Our approach is based on the representation of lane change trajectories using a mathematical model known as a bi-elementary path model, originally introduced for optimal control applications [47]. This model, composed of combined clothoid segments efficiently approximated by arc-splines, offers a flexible and compact representation of the geometric characteristics of lane changes, capturing their essential features with a small set of parameters. To estimate the parameters of the bi-elementary path model for a given lane change instance, we employ a least-squares curve fitting technique. This data-driven optimization step allows our model to adapt to different scenarios and capture the diverse nature of lane changes observed in real-world driving. By minimizing the error between the observed trajectory data and the model's output, we obtain optimal parameter values that best represent the given lane change. Our approach presents several potential advantages. First, its reliance on a well-defined mathematical model provides insights into the underlying factors influencing the predicted trajectory, enhancing interpretability compared to purely learning-based methods. Second, by using a compact model and efficient curve fitting techniques, our method can potentially achieve real-time performance even on standard hardware, addressing a key limitation of many deep learning approaches. Moreover, the inherent flexibility of our approach allows it to effectively handle lane changes on both straight and curved roads through appropriate coordinate transformations. We note that the developed method is the first to utilize trajectory templates based on clothoids for lane change prediction. The only other methods employing template curves are those fitting fused sensor data to parametric polynomial functions [20] and those utilizing cubic spline curves [21]. While polynomial functions offer simplicity, their limitations in capturing the complex dynamics of vehicle maneuvers can restrict their prediction accuracy. Similarly, cubic spline curves, despite their flexibility, lack the inherent constraints associated with vehicle dynamics and the specific characteristics of lane-change maneuvers, potentially leading to predictions that deviate from realistic vehicle behavior. In contrast, our clothoid-based bi-elementary paths, explicitly designed to model the smooth and continuous changes in curvature characteristic of lane changes, offer a more dynamically realistic and interpretable representation. Combined with efficient arc-spline approximations and optimized parameter estimation, our method presents a promising solution for accurate and real-time lane change

trajectory prediction. The primary objective of this thesis is to develop and comprehensively evaluate this proposed approach, aiming to achieve accuracy and computational efficiency that will significantly contribute to advancing autonomous driving technology.

The main contributions of the thesis are as follows:

- **Adaptation of the Bi-elementary Path Model:** We successfully adapted and applied the bi-elementary path model and arc-spline representation, originally developed for path control, to the distinct domain of lane change trajectory prediction.
- **Development of a novel prediction approach:** We developed a novel approach for lane change trajectory prediction that integrates the geometric insights of the bi-elementary path model with least-squares curve fitting for data-driven parameter estimation.
- **Efficient Computation:** We utilized arc-spline approximation and a compact parameter set for the bi-elementary path model, enabling efficient computation and real-time performance for lane change prediction.
- **Bi-elementary Path Parameter Estimation:** We employed least-squares curve fitting to achieve precise parameter estimation and data-driven adaptation of the bi-elementary path model, enabling it to handle diverse lane change scenarios effectively.
- **Coordinate Frame Transformations:** We developed and implemented well-defined coordinate transformations between the vehicle's body frame, the Cartesian frame, and the Frenet frame, ensuring applicability to both straight and curved roads.
- **Spatial-to-Temporal Transformation:** We implemented an effective methodology to transform predicted trajectories from arc-length-based representations to time-based predictions, enabling the prediction of future vehicle positions at specific time instants.
- **Performance Evaluation:** We conducted a detailed evaluation of the proposed method using both synthetically generated trajectories and real-world driving

data from the Next Generation Simulation (NGSIM) dataset [48], demonstrating its robustness and adaptability.

- **Comparative Analysis:** We performed a comprehensive comparison of our method's performance with state-of-the-art approaches, showcasing its competitive accuracy and superior computational efficiency.

The thesis is organized as follows: Chapter 2 provides the necessary background information, including an overview of lane change trajectory templates, coordinate frames and transformations, the NGSIM dataset, and least-squares curve fitting. Chapter 3 details the proposed methodology, encompassing the prediction algorithm and spatial-to-temporal transformation. Chapter 4 presents a comprehensive evaluation of the method using both synthetic and real-world data, comparing its performance with existing approaches. Finally, Chapter 5 concludes the thesis by summarizing the key contributions, discussing the limitations, and outlining potential directions for future research.

CHAPTER 2

BACKGROUND INFORMATION

2.1 Lane Change Trajectory Templates

Lane changing is a vehicle maneuver where the vehicle transitions from its current lane of travel to an adjacent lane.

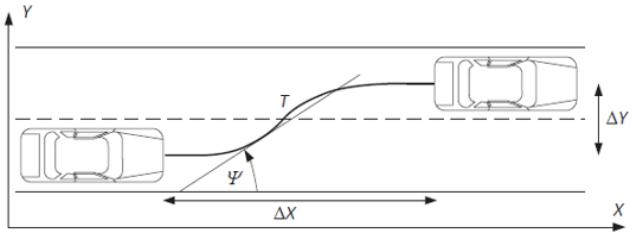

Figure 2.1: Basic lane change maneuver.

Due to vehicle dynamics, a typical lane change maneuver results in a smooth, curved path with varying curvature. In the following subsections, the mathematical process of generating the lane change trajectory templates will be explained.

2.1.1 Clothoids

A clothoid, also known as Euler spiral, is a spiral curve whose curvature changes linearly with its arc length. This property makes clothoids well-suited for modeling road and vehicle trajectories where a smooth and gradual change in curvature is desired, ensuring comfortable and safe transitions for the vehicles.

Figure 2.2 illustrates the behavior of a clothoid curve with increasing curvature.

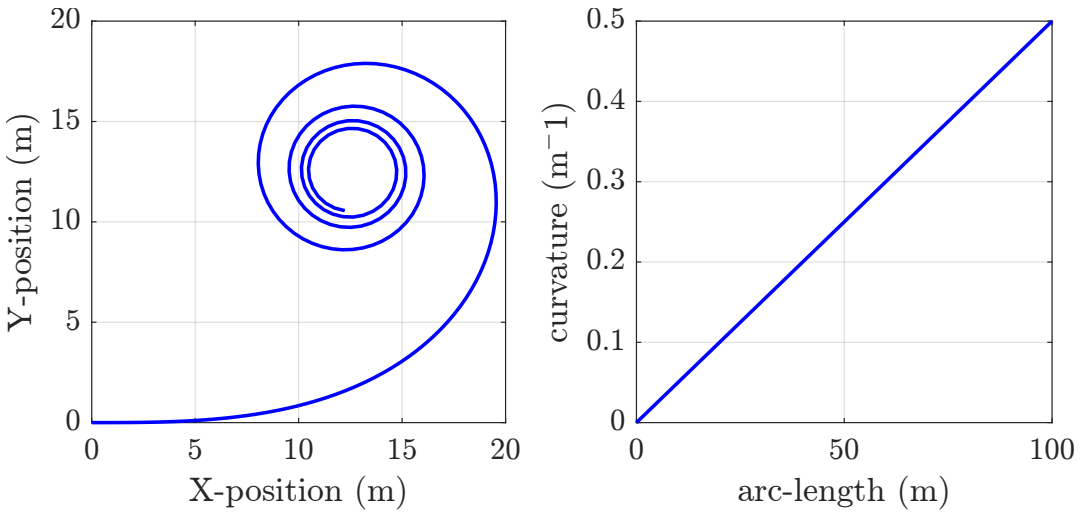

Figure 2.2: Illustration of a clothoid.

The clothoid coordinates X, Y, the tangent angle ψ and the curvature value k, as a function of the arc-length s, can be calculated as follows:

$$X(s) = \int_0^s \cos\left(\psi(z)\right) dz + X(0) \tag{2.1}$$

$$Y(s) = \int_0^s \sin\left(\psi(z)\right) dz + Y(0) \tag{2.2}$$

$$\psi(s) = \int_0^s k(z)\, dz + \psi(0) \tag{2.3}$$

and

$$k(s) = k(0) + \frac{\left(k(S) - k(0)\right) s}{S} \tag{2.4}$$

where S is the total arc-length.

We adopt the following notation to represent clothoids:

$$C = \mathcal{C}(P_i, \psi_i, k_i, k_f, S) \tag{2.5}$$

These parameters can uniquely characterize a clothoid, which are the initial position $P_i = [X(0), Y(0)]$, initial tangent angle $\psi_i = \psi(0)$, initial curvature $k_i = k(0)$, final curvature $k_f = k(S)$ and arc-length S.

Figure 2.3 illustrates a clothoid curve generated with specific parameter values.

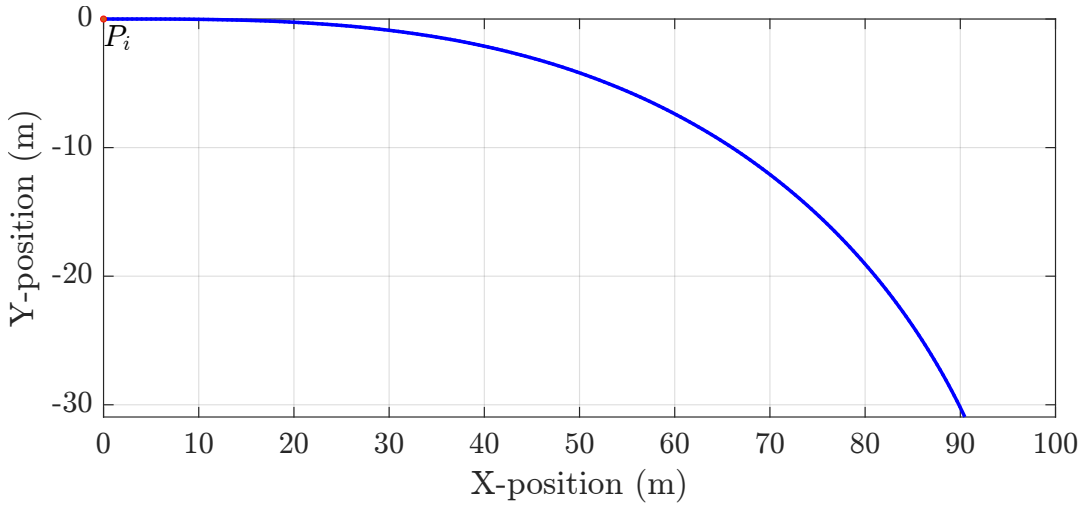

Figure 2.3: Clothoid $\mathcal{C}(0, 0, 0, -0.02, 100)$

The main limitation of clothoids is the absence of an analytical expression for their coordinate values. The integrals in equations (2.1), (2.2), and (2.3) requires numerical solutions to obtain the coordinates, making their direct use computationally inefficient for practical implementations.

2.1.2 Clothoid Approximations using Arc-Splines

To address the computational challenges associated with clothoids, we employ concatenated arc-segments, referred to as arc-splines, to approximate them.

An arc-spline consists of multiple arc segments, each characterized by a unique curvature value. The curvatures of the initial and final arc-segments are denoted as k_i and k_f respectively. The total number of arc-segments is $n+1$, where n represents the approximation order. A higher approximation order generally leads to a more accurate representation of the clothoid but at the cost of increased computational complexity.

The necessary parameters for the clothoid approximation are described as follows:

- The arc-segments are indexed from start to end as $j = 0, 1, ..., n$
- Curvature increment: $h = \dfrac{k_f - k_i}{n}$
- Curvature of the arc-segment j: $k_j = k_i + jh$

- Angle spanned by the first arc-segment: $w_0 = \frac{Sk_i}{2n}$
- Angle spanned by the last arc-segment: $w_n = \frac{Sk_n}{2n}$
- Angle spanned by the arc-segments in between: $w_j = \frac{Sk_j}{n}$
- Length of the first and last arc-segments: $S_0 = S_n = \frac{S}{2n}$
- Length of the arc-segments in between: $S_j = \frac{S}{n}$
- Angle at the end of each arc-segment: $\psi_j = \psi_i + \sum_{i=0}^{j} w_j$

Using these defined parameters, clothoids are approximated and parameterized by arc-splines with the following calculations:

First segment ($j = 0$) If the curvature of the first segment is 0, i.e. $k_i = 0$, the first segment is a line segment. Hence, the position calculation along this segment is

$$P(s) = P_i + s \begin{bmatrix} \cos(\psi_i) \\ \sin(\psi_i) \end{bmatrix}, \quad s \in [0, \frac{S}{2n}] \tag{2.6}$$

If $k_i \neq 0$, the first segment is an arc with curvature k_i. The position and tangent angle calculations along the segments are as follows:

$$P(s) = P_i + \frac{1}{k_i} \begin{bmatrix} \sin(\psi(s)) - sin(\psi_i) \\ \cos(\psi_i) - \cos(\psi(s)) \end{bmatrix}, \quad s \in [0, \frac{S}{2n}] \tag{2.7}$$

$$\phi(s) = \phi_i + sk_i \tag{2.8}$$

Segments 1 **to** $n-1$

These segments are represented by arcs with the respective curvature $k_j, \quad j = 1, 2, ...n-1$. The parametrization is

$$P(s) = P_{j-1} + \frac{1}{k_j} \begin{bmatrix} \sin(\psi(s)) - sin(\psi_{j-1}) \\ \cos(\psi_{j-1}) - \cos(\psi(s)) \end{bmatrix}, \quad s \in [\frac{S}{2n} + (j-1)\frac{S}{n}, \frac{S}{2n} + j\frac{S}{n}] \tag{2.9}$$

$$\phi(s) = \phi_{j-1} + (s - \frac{S}{2n} - (j-1)\frac{S}{n})k_j \tag{2.10}$$

where P_{j-1} and ψ_{j-1} are the ending position and tangent angle of the previous segment.

Last segment ($j = n$) If the curvature of the first segment is 0, i.e. $k_f = 0$, the last segment is a line segment. Hence, the position calculation along this segment is

$$P(s) = P_{n-1} + (s - S + \frac{S}{2n}) \begin{bmatrix} \cos(\psi_{n-1}) \\ \sin(\psi_{n-1}) \end{bmatrix}, \quad s \in [S - \frac{S}{2n}, S] \tag{2.11}$$

If $k_f \neq 0$, the final segment is an arc with curvature k_f. The parametrization is

$$P(s) = P_{n-1} + \frac{1}{k_f} \begin{bmatrix} \sin(\psi(s)) - sin(\psi_{n-1}) \\ \cos(\psi_{n-1}) - \cos(\psi(s)) \end{bmatrix}, \quad s \in [S - \frac{S}{2n}, S] \tag{2.12}$$

$$\phi(s) = \phi_{n-1} + (s - S + \frac{S}{2n})k_f \tag{2.13}$$

Figure 2.4 illustrates an example clothoid approximation using arc-splines.

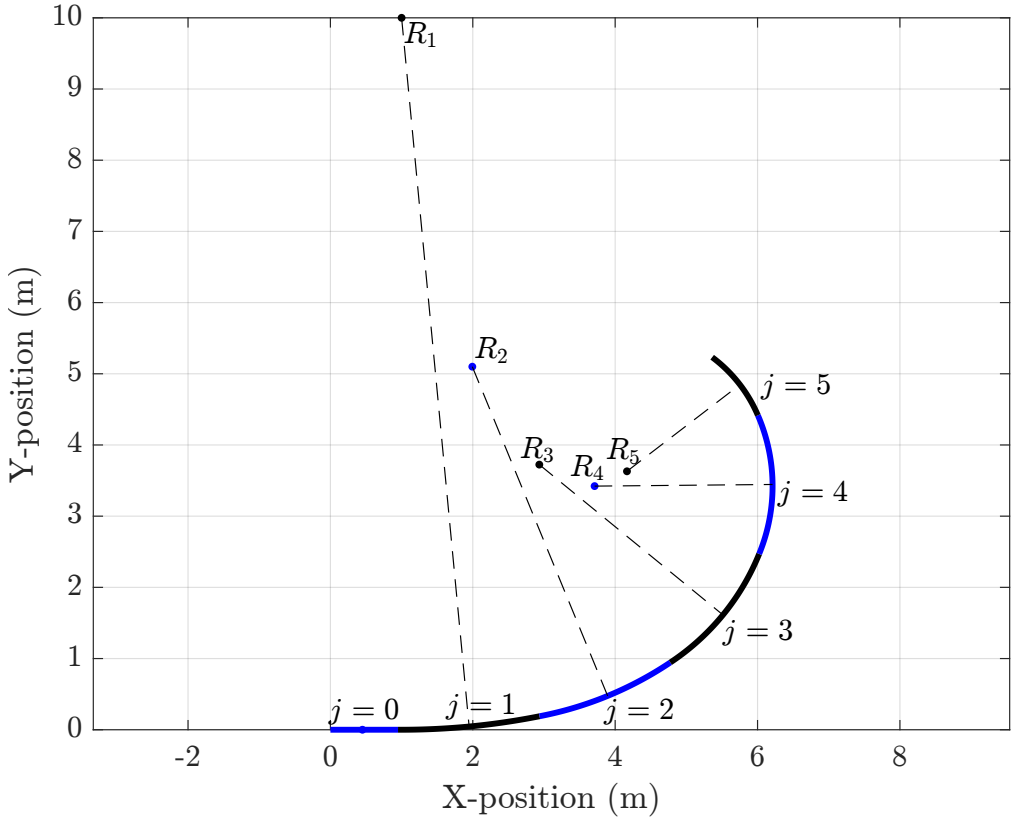

Figure 2.4: Clothoid $\mathcal{C}(0, 0, 0, 0.5, 10)$ approximation, n=5

For this approximation, the approximation order n is selected as 5. Since $k_i = 0$, the first segment is line and the following segments are arcs with radius $R_j, j = 1, 2, .., 5$.

2.1.3 Bi-elementary Path

To represent lane change trajectories, multiple clothoid segments are combined in a specific configuration, resulting in a curve known as a bi-elementary path. This model is well-suited for representing lane-change maneuvers due to its ability to capture the smooth transitions and varying curvatures characteristic of such maneuvers.

A bi-elementary path consist of four clothoid curves and a line segment in the middle. These curves can be denoted as:

- First curve $C_1 = \mathcal{C}(0, 0, 0, k_1, S_{E_1}/2)$
- Second curve $C_2 = \mathcal{C}(P_{C_1,end}, \psi_{C_1,end}, k_1, 0, S_{E_1}/2)$
- Line segment $L = \mathcal{C}(P_{C_2,end}, \psi_{C_2,end}, 0, 0, S_L)$
- Third curve $C_3 = \mathcal{C}(P_{L,end}, \psi_{L,end}, 0, k_2, S_{E_2}/2)$
- Fourth curve $C_4 = \mathcal{C}(P_{C_3,end}, \psi_{C_3,end}, k_2, 0, S_{E_2}/2)$.

Here, $[C_1, C_2]$ and $[C_3, C_4]$ pairs are also called as elementary paths. Therefore, a bi-elementary path is a combination of two elementary paths with a line segment in between. S_{E_1},S_{E_2} and S_L are the arc-lengths of the first elementary path, the second elementary path and the length of the line segment respectively.

Figure 2.5 illustrates a bi-elementary path for the parameters $S_{E_1} = 40$, $S_{E_2} = 50$, $S_L = 20$, $k_1 = 0.05$ and $k_2 = -0.04$. Also, the curvature change of this bi-elementary path can be observed in Figure 2.6

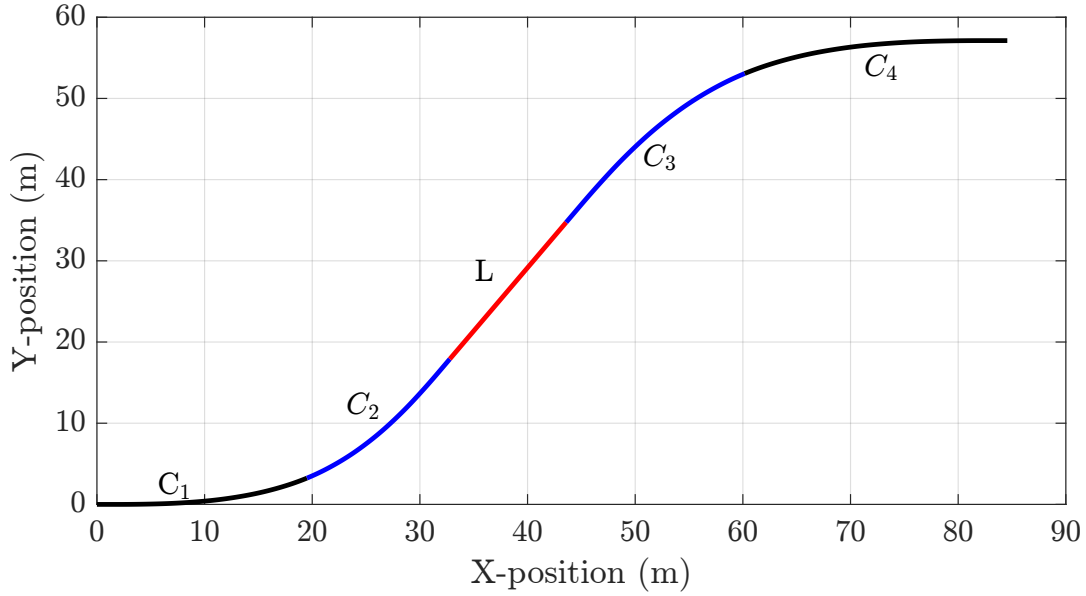

Figure 2.5: Illustration of a bi-elementray path

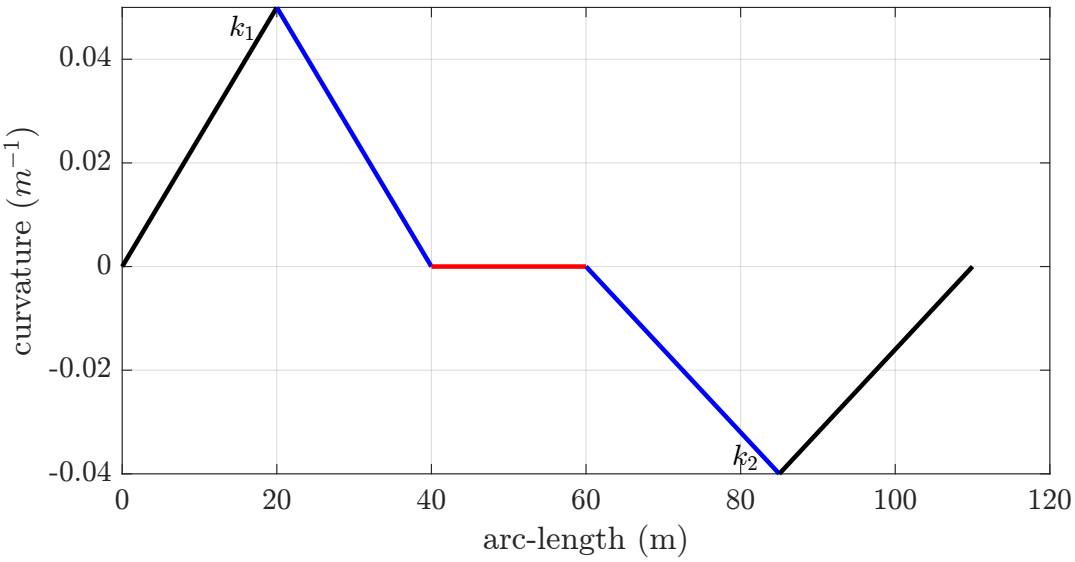

Figure 2.6: Illustration of a bi-elementray path curvature change

For an elementary path, the change in the tangent angle can be formalized as $\alpha = \frac{S_E\, k}{2}$, where S_E is the length of the elementary path and k is the maximum curvature.

In the case of a bi-elementary path, the initial and final tangent angles need to be zero, Therefore, the change in the tangent angles of the both elementary paths must

be equal. Hence, it must hold that

$$\alpha_{E1} = \frac{S_{E_1}\, k_1}{2} = -\frac{S_{E_2}\, k_2}{2} = -\alpha_{E2} \tag{2.14}$$

which yields that

$$k_2 = -\frac{S_{E_1}\, k_1}{S_{E_2}} \tag{2.15}$$

To have a more convenient parametrization, λ and γ parameters are introduced. γ represents the length of the elementary paths in relation to the total arc-length, whereas λ is the ratio of arc-lengths of the elementary paths. The mathematical expressions are as below:

$$S_{E_1} + S_{E_2} = \gamma S \tag{2.16}$$

$$S_L = (1-\gamma)S \tag{2.17}$$

$$S_{E_1} = \lambda\gamma S \tag{2.18}$$

$$S_{E_2} = (1-\lambda)\gamma S \tag{2.19}$$

$$\lambda \in (0,1),\ \gamma \in (0,1] \tag{2.20}$$

Using the parameters λ, γ, k_1 and S, a bi-elementary path can be uniquely characterized. The notation used is

$$B = \mathcal{B}(S, k_1, \lambda, \gamma) \tag{2.21}$$

For lane-change applications, ΔY, lateral change in y-position, can be used instead of one of the above parameters since it's more intuitive as a parameter of a lane-change trajectory.

ΔY can be obtained using the formula below:

$$\Delta Y = S(\gamma D(\alpha)\sin(\alpha/2) + (1-\gamma)\sin(\alpha))) \tag{2.22}$$

$$\alpha = \frac{\lambda\gamma S k_1}{2} \tag{2.23}$$

where $D(\alpha)$ is a tight polynomial approximation function for the computation purpose.

$$D(\alpha) = p_0 + p_1\alpha + p_2\alpha^2 + p_3\alpha^3 \tag{2.24}$$

with the parameters $p_0 = 1$, $p_1 = 1.34.10^{-4}$, $p_2 = -6.75.10^{-2}$, $p_3 = 1.64.10^{-3}$.

The effect of each parameter on the shape of a bi-elementary path is illustrated on the following figures. For these examples, ΔY parameter is used instead of k_1.

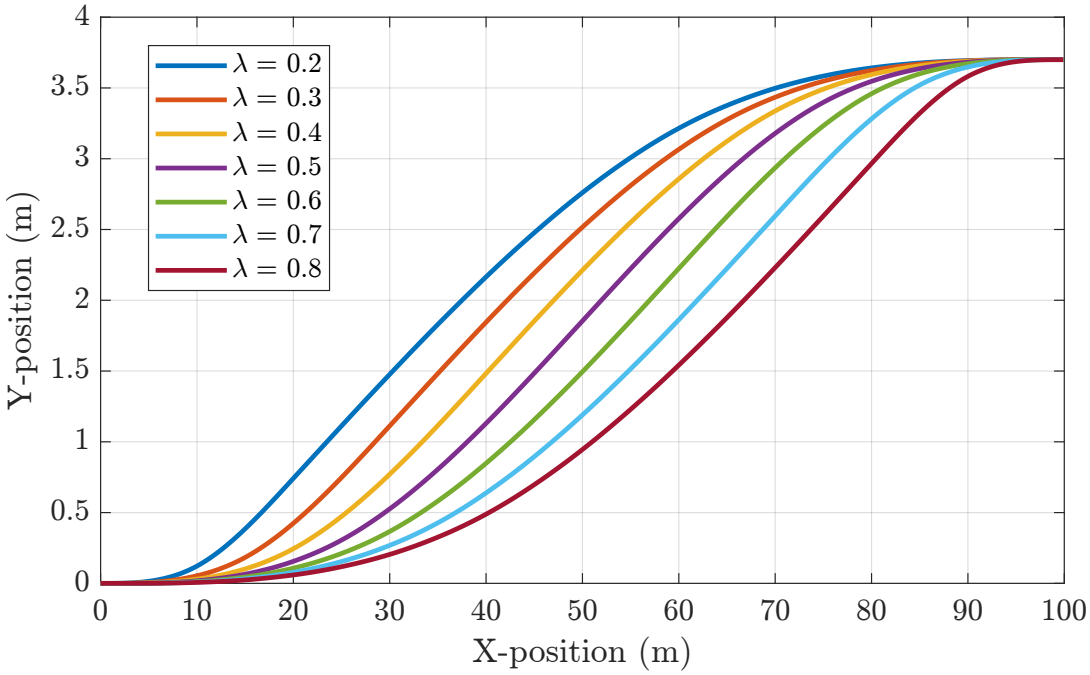

Figure 2.7: $\mathcal{B}(S, \Delta Y, \lambda, \gamma)$ for $S = 100$, $\Delta Y = 3.7$, $\gamma = 1$ and different λ

As illustrated in Figure 2.7, variation of λ represents the ratios of the lengths of elementary paths. So, when λ is equal to 0.5, the arc-lengths of both elementary paths S_{E_1} and S_{E_2} are equal. When λ is larger than 0.5, it means that the arc-length of the first elementary path is larger, which means that the maximum curvature of the first elementary path is smaller compared to that of the second elementary path. The exact opposite statements can be made for λ being smaller than 0.5.

γ is inversely proportional to the portion of the line segment in the bi-elementary path. Therefore $\gamma = 1$ means that there is no line segment and the bi-elementary path consists of two elementary paths. For a lane changing vehicle this means that the steering is never kept still during the maneuver. As γ decreases, the portion of the line segment increases as observed in Figure 2.8.

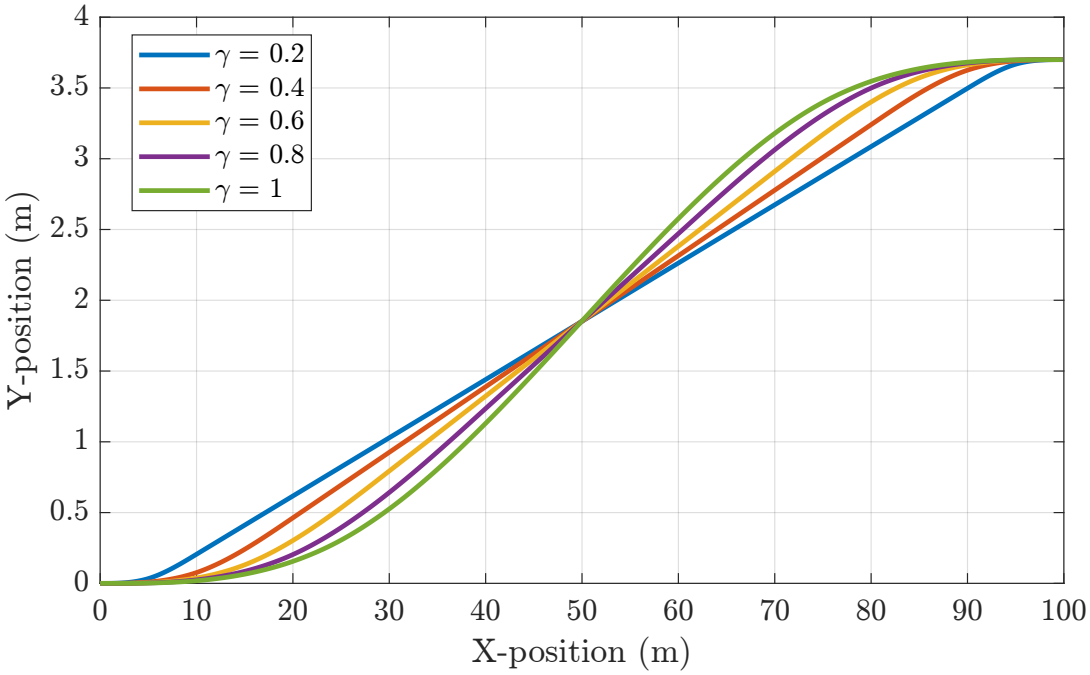

Figure 2.8: $\mathcal{B}(S, \Delta Y, \lambda, \gamma)$ for $S = 100$, $\Delta Y = 3.7$, $\lambda = 0.5$ and different γ

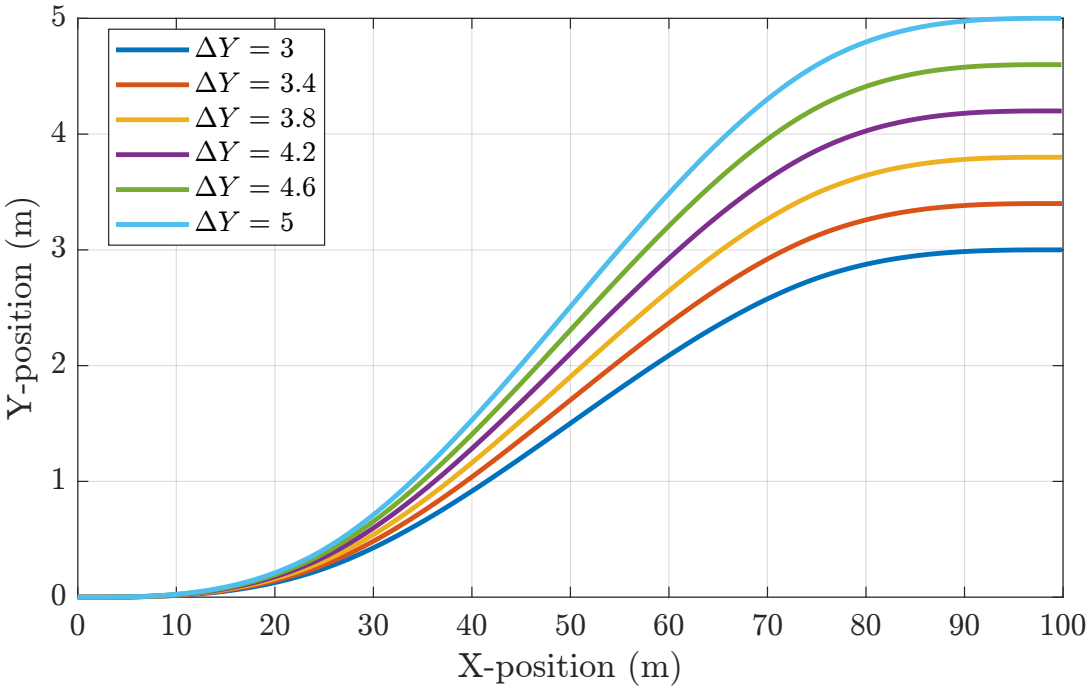

Figure 2.9: $\mathcal{B}(S, \Delta Y, \lambda, \gamma)$ for $S = 100$, $\lambda = 0.5$, $\gamma = 1$ and different ΔY

ΔY represents the width of the lane change maneuver. For typical highway roads ΔY is 3.7 m. In Figure 2.9 bi-elementary paths for different ΔY can be seen.

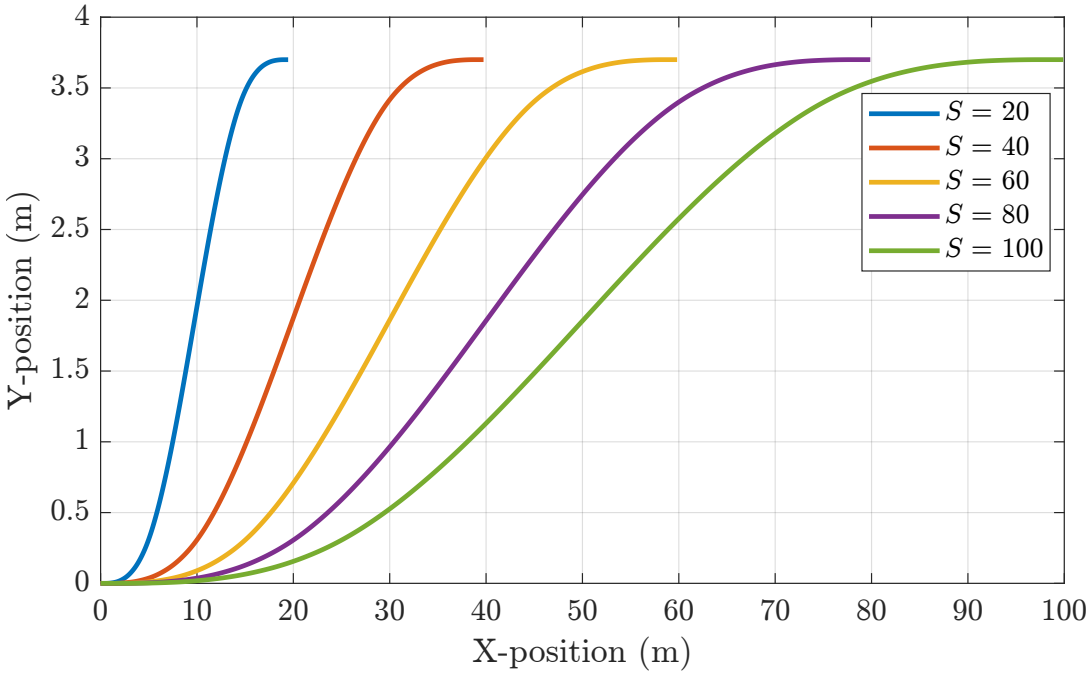

Figure 2.10: $\mathcal{B}(S, \Delta Y, \lambda, \gamma)$ for $\Delta Y = 3.7$, $\lambda = 0.5$, $\gamma = 1$ and different S

In Figure 2.10 bi-elementary paths for different total arc-lengths can be seen. When all the other parameters $(\Delta Y, \lambda, \gamma)$ kept constant, increase in S results in decrease in the curvatures k_1 and k_2. Therefore the path gets smoother.

2.2 Coordinate Frames and Transformations

In this thesis study, we utilize three distinct coordinate frames for specific purposes:

1. **Frenet Frame :** This frame provides an intuitive representation of the relationship between a vehicle and the road. A vehicle's position is expressed as an (s, d) pair as shown in Figure 2.11, where s is the arc length along the reference curve to the closest point to the vehicle, and d is the lateral offset from that point. By mapping vehicles coordinates onto a simplified curvilinear coordinate system aligned with the road, the Frenet frame simplifies trajectory planning and prediction tasks, particularly on curved roads.

2. **Body Frame :** Body frame is a reference system fixed to the center of the host vehicle. X-axis points forward in the direction of the vehicle's travel and Y-

axis points to the right, perpendicular to the X-axis. In Figure 2.11, x_b and y_b axes form the body frame. We assume that all measurements related to surrounding vehicles are obtained from the host vehicle's onboard sensors and are initially expressed in this body frame.

3. **Cartesian Frame :** This frame serves as an intermediate frame for the transformation from the body frame to the Frenet frame. The origin of this frame is fixed at the starting point of the reference curve. As illustrated in Figure 2.11, the orthogonal x_c and y_c axes form the Cartesian frame.

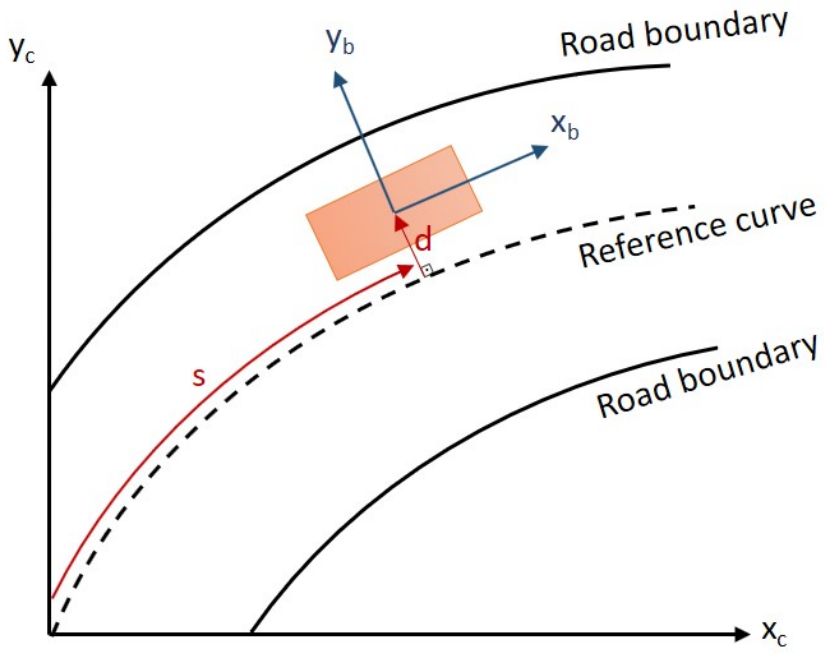

Figure 2.11: Frenet, Body and Cartesian Coordinate Frames

2.2.1 Transformation between Body frame and Cartesian frame Frame

Since measurements are acquired in the body frame, they need to be transformed into the Frenet frame for compatibility with our prediction algorithm. As mentioned earlier, the Cartesian frame serves as an intermediate step in this transformation. This subsection details the conversion from body frame to Cartesian frame coordinates.

In Figure 2.12, the body and Cartesian frames are illustrated.

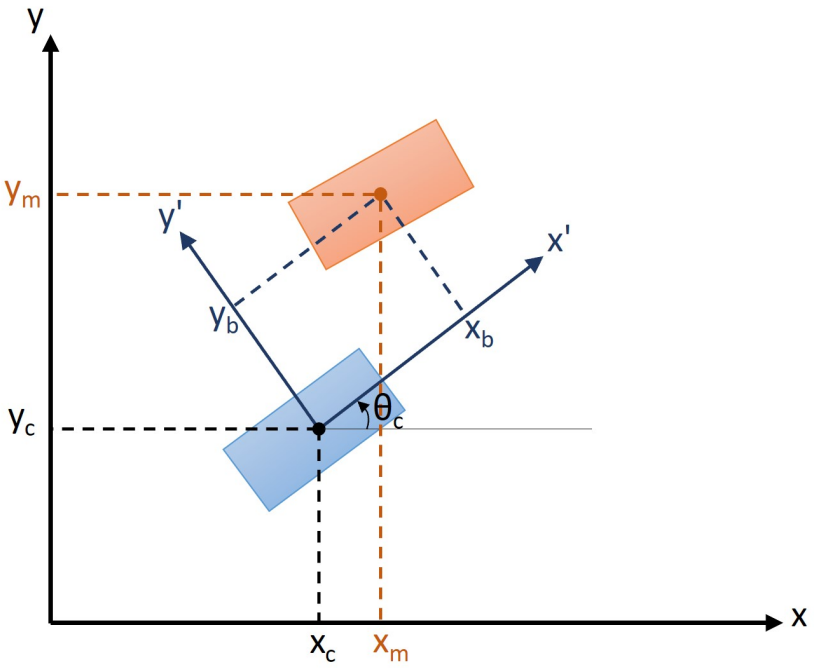

Figure 2.12: Cartesian and body frames

Let (x_b, y_b) represent the position measurement of the surrounding vehicle in body frame. To transform this and obtain the measurement in Cartesian frame, (x_m, y_m), following calculations need to be applied:

$$\begin{aligned} x_m &= x_c + x_b \cos\theta_c - y_b \sin\theta_c \\ y_m &= y_c + x_b \sin\theta_c + y_b \cos\theta_c \end{aligned} \tag{2.25}$$

where:

(x_c, y_c) : position of the host vehicle in Cartesian frame
θ_c : heading angle of the host vehicle in Cartesian frame
(x_b, y_b) : position measurement of the surrounding vehicle in body frame
(x_m, y_m) : position measurement of the surrounding vehicle in Cartesian frame

The velocity information required for the prediction algorithm is the magnitude of the velocity, as direction of motion is inferred from the shape of the fitted bi-elementary path. Assuming the velocity components (v_{x_b}, v_{y_b}) of the surrounding vehicle are measured in the host vehicle's body frame, the magnitude can be extracted as:

$$v = \sqrt{v_{x_b}^2 + v_{y_b}^2} \tag{2.26}$$

where:

v : the magnitude of the velocity

(v_{x_b}, v_{y_b}) : velocity measurement of the surrounding vehicle in body frame

In addition to position and velocity measurements, we also assume that the heading in body frame, the angle between the host vehicle's x-axis and the predicted vehicle's x-axis (θ_b), is measured. To obtain the heading angle measurement in the Cartesian frame (θ_m) we utilize the following relationship:

$$\theta_m = \theta_c + \theta_b \tag{2.27}$$

where:

θ_m : heading angle of the predicted vehicle in Cartesian frame

θ_c : heading angle of the host vehicle in Cartesian frame

θ_b : measured heading angle of the predicted vehicle in body frame

It is essential for predicted trajectories, which are generated in the Frenet frame, to be converted back into the body frame of the host vehicle for practical use in decision-making and control. Therefore, the necessary intermediate transformation from Cartesian to body frame can be applied using the following equations:

$$\begin{aligned} x_b &= (x_m - x_c)\cos\theta_c + (y_m - y_c)\sin\theta_c \\ y_b &= -(x_m - x_c)\sin\theta_c + (y_m - y_c)\cos\theta_c \end{aligned} \tag{2.28}$$

2.2.2 Transformation between Cartesian and Frenet Frames

After obtaining the transformation from body frame to Cartesian frame as explained in Section 2.1.1, the transformation from Cartesian frame to Frenet frame is applied.

Let P_C be the position of the vehicle in Cartesian frame. To transform this point to Frenet frame, we first identify the closest point R_C on the reference curve. The arc length at R_C , which is assumed to known from the reference curve points, determines the s coordinate of the transformed point P_F. The signed distance from R_C to P_C determines the d coordinate of P_F .

Mathematically, the Frenet coordinates (s, d) are defined as:

$$\begin{aligned} s &= \text{arc-length of the reference curve at } R_C \\ d &= \text{signed distance from } R_C \text{ to } P_C \end{aligned} \tag{2.29}$$

where:

P_C : Position of the car in the Cartesian frame

R_C : Closest point to P_C on the reference curve, in the Cartesian frame

P_F : Position of the car, in the Frenet frame (s, d)

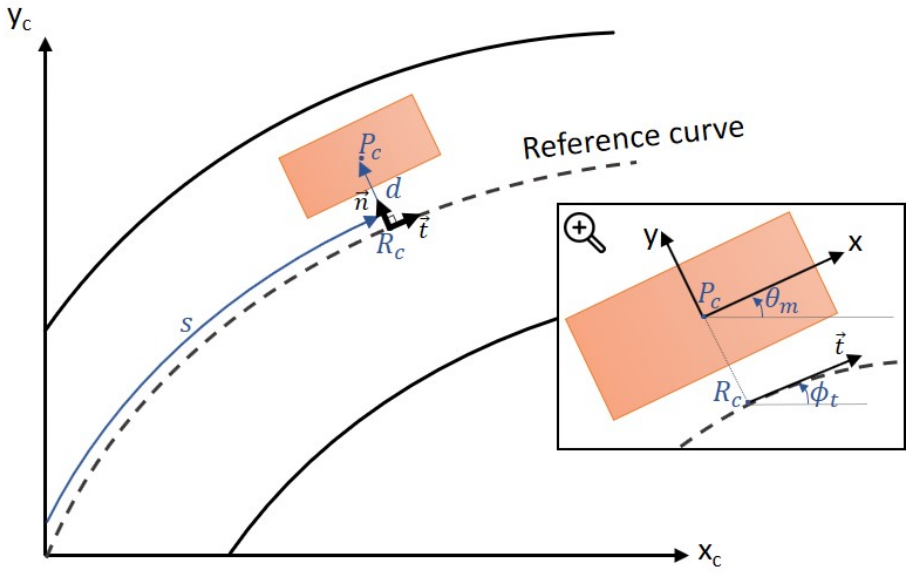

Figure 2.13: Cartesian and Frenet Frames

The heading angle of the vehicle in the Frenet frame (θ_F) is defined as the angle between the vehicle's x-axis and the tangent line of the reference path at the vehicle's closest point. Assuming we know the tangent angle of the reference path (ϕ_t) and the heading angle of the predicted vehicle in the Cartesian frame (θ_m), we can obtain θ_F as follows:

$$\theta_F = \theta_m - \phi_t \tag{2.30}$$

where:

θ_F : heading angle of the predicted vehicle in Frenet frame

θ_m : heading angle of the predicted vehicle in Cartesian frame

ϕ_t : tangent angle of the reference path at the vehicle's closest point in Cartesian frame

Conversely, to transform a point $P_F = (s, d)$ from the Frenet frame back to the Cartesian frame, we follow these steps:

1. **Find the reference point:** Locate the point R_C on the reference curve corresponding to the arc length s .

2. **Compute the tangent and normal vectors:** Calculate the unit tangent vector $\overrightarrow{t}$ and the unit normal vector $\overrightarrow{n}$ at R_C.

3. **Obtain the Cartesian coordinates:** The Cartesian coordinates $P_C = (x_c, y_c)$ are then given by the equation:

$$P_C = R_C + d \cdot \overrightarrow{n} \tag{2.31}$$

   where:

   R_C : Point on the reference curve at arc length 's', in the Cartesian frame

   $\overrightarrow{t}$: Unit tangent vector at R_C

   $\overrightarrow{n}$: Unit normal vector at R_C

In summary, our prediction algorithm relies on measurements of the surrounding vehicle's position and velocity, initially captured in the host vehicle's body frame. We first transform these measurements into the Cartesian frame then convert these coordinates into the Frenet frame to obtain the (s, d) representation, which is then utilized by our prediction algorithm. The resulting predicted trajectories, initially generated in the Frenet frame, are then transformed back into the host vehicle's body frame to provide actionable information for decision-making and control. This accurate transformation process ensures that our model is applicable even on curved roads and it also generates results in a format readily usable by the host vehicle's control systems.

2.3 NGSIM Dataset

The Next Generation Simulation (NGSIM) dataset, provided by the U.S. Federal Highway Administration, serves as the primary source of real-world driving data for this thesis study. This publicly available dataset offers detailed vehicle trajectory data collected from real-world traffic scenarios on various road types. It has been widely used for validation and verification of prediction algorithms.

The dataset contains two primary highway segments, US-101 and I-80, where lane-changing behaviors can be distincly observed.

- **NGSIM I-80:** This dataset is collected from the traffic on Interstate 80 in the San Francisco Bay Area, California. The study area is spans approximately 500 meters in length with six mainline lanes as shown in Figure 2.14. It contains trajectories of 5678 individual vehicles, with 1851 of them exhibiting lane-changing behavior.
- **NGSIM US-101:** This dataset is collected from the traffic on Hollywood Freeway, Los Angeles. The study area covers approximately 640 meters in length with five mainline lanes, as shown in Figure 2.15. It contains trajectories of 4824 individual vehicles, with 753 of them performing lane changes.

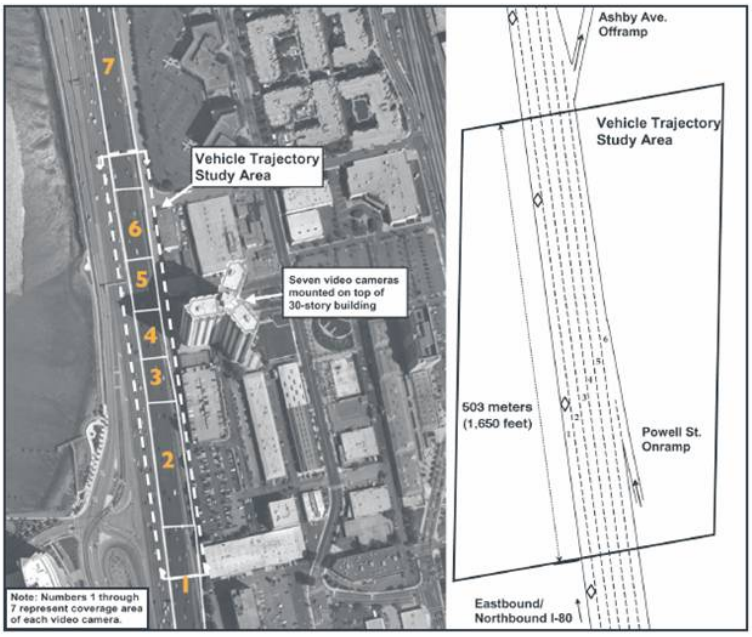

Figure 2.14: I-80

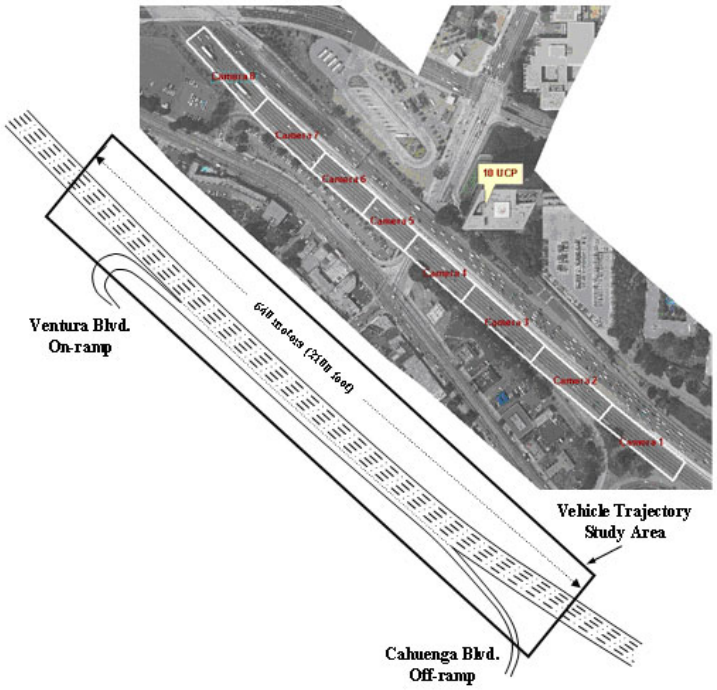

Figure 2.15: US-101

The trajectories within the NGSIM dataset were captured using vision-based monitoring systems. Through comprehensive video analysis, the dataset provides fundamen-

tal information for each recorded vehicle, including identification number, position, velocity, acceleration, and current lane ID, all sampled at a frequency of 10 Hz.

For our lane change trajectory prediction task, we extract the following data from the NGSIM dataset for each vehicle:

- **Position Data:** A sequence of (x, y) coordinates in the Frenet frame, representing the vehicle's position over time at 10 Hz. This data serves as the measurement data for our curve fitting algorithm and also will be used as ground truth for evaluating the performance.
- **Velocity Data:** The vehicle's instantaneous longitudinal velocity at each time step. This information is crucial for transforming our arc-length-based predictions into time-based trajectories.
- **Acceleration Data:** The longitudinal acceleration of the vehicle. Similar to velocity data, this information is used for the transformation into time-based coordinates.
- **Lane ID:** The vehicle's current lane identification at each time step. This is used to determine the vehicles performing lane changes for evaluation part.

By utilizing this extracted data, our aim is to develop and evaluate our proposed lane change trajectory prediction method, demonstrating its effectiveness in accurately predicting vehicle movements in real-world scenarios.

2.4 Least Squares Curve Fitting

Curve fitting is a fundamental technique used to find a mathematical function that best represents a set of observed data points. Among various methods, least squares fitting is a widely used approach due to its simplicity and effectiveness in finding the "best fit" line or curve through a given dataset.

The principle of least squares is to minimize the sum of the squares of the residuals, where residuals are the differences between the observed data points y_i and the corresponding values predicted by the model $f(x_i; \theta)$:

$$\min_{\theta} \sum_{i=1}^{n} \left(y_i - f(x_i; \theta)\right)^2$$

Here, θ represents the parameters of the model f, and n is the number of data points.

In the case of linear models, the model $f(x; \theta)$ is expressed as a linear combination of the parameters θ:

$$f(x; \theta) = \theta_1 x + \theta_0$$

The parameters θ_0 and θ_1 are determined by minimizing the sum of squared residuals:

$$\min_{\theta_0, \theta_1} \sum_{i=1}^{n} \left(y_i - (\theta_1 x_i + \theta_0)\right)^2$$

This leads to the well-known formulas for the least squares estimates:

$$\theta_1 = \frac{n \sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n \sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2}$$

$$\theta_0 = \frac{1}{n} \sum_{i=1}^{n} y_i - \theta_1 \frac{1}{n} \sum_{i=1}^{n} x_i$$

While linear least squares has a closed-form solution, nonlinear models require iterative optimization techniques. They can take various mathematical forms, such as exponential, power-law, polynomial, logistic, or custom functions specific to the problem domain. The general approach to nonlinear least-squares fitting involves iterative optimization algorithms like Gauss-Newton, Levenberg-Marquardt, or gradient descent to find the optimal parameters that minimize the sum of squared residuals. In Figure 2.16, linear and nonlinear least squares curve fitting examples are illustrated.

In this thesis, we employ nonlinear least squares curve fitting to estimate the parameters of the bi-elementary path model based on observed trajectory data.

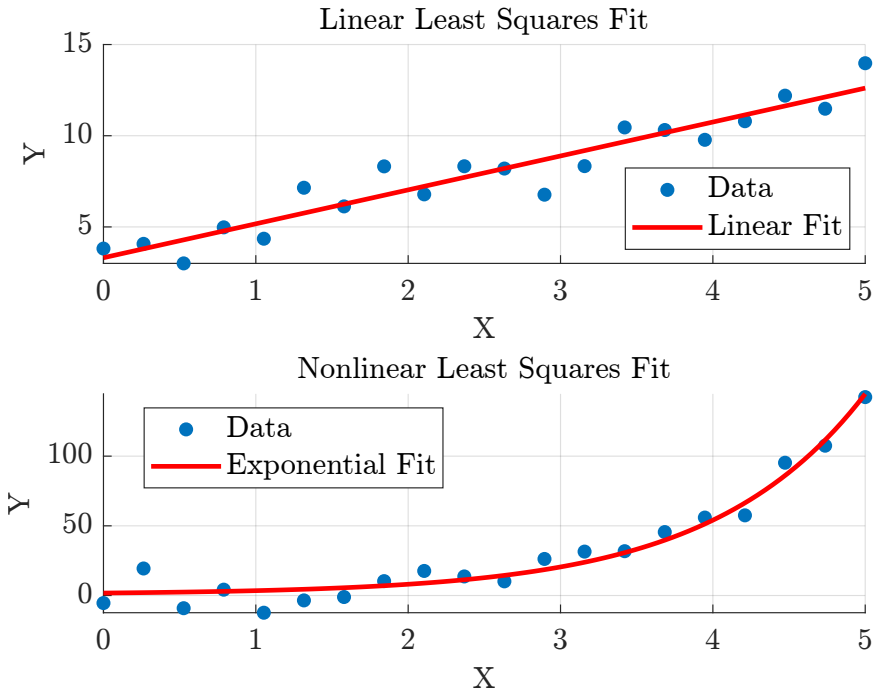

Figure 2.16: Linear and Nonlinear Least Squares Curve Fittings

CHAPTER 3

METHODOLOGY

For autonomous vehicles, it is important to predict possible lane change maneuvers of the surrounding vehicles in order to plan safer and smoother trajectories.

To make this prediction, the method used in this thesis study is based on fitting the historical position information of the surrounding vehicles to the template lane change trajectories. These template lane change trajectories are represented by bi-elementary paths as discussed in Section 2.1.3. Since bi-elementary paths can be uniquely characterized by only four parameters, these parameters can be estimated using least squares fitting in a fast and efficient way. Therefore, this implementation can also be adapted to real-time applications.

In the following sections, first the necessary measurements are discussed and it is shown that the measured positions of the lane changing vehicles become similar to a bi-elementary path with suitable parameters as time progresses. Then, the detailed steps of the prediction algorithm are presented. Afterward, the transformation needed to convert the predicted trajectories from spatial coordinates to temporal coordinates is explained.

3.1 Lane Change Trajectories and Bi-elementary Paths

In this thesis study, trajectory prediction focuses solely on lane change maneuvers. The main goal is to generate the best possible trajectory that accurately represents the vehicle's future movement during a lane change maneuver. To achieve this, a precise mathematical model for lane change trajectories is essential.

In a typical lane change maneuver, the vehicle moves smoothly. Initially, the vehicle's steering wheel is turned towards the target lane. It it is then held steady as the car transitions, followed by a counter-steer to align the vehicle within the new lane, completing the maneuver. Throughout this process, the curvature of the trajectory changes smoothly.

As discussed in Section 2.1.1, clothoids, characterized by their linearly varying curvature, offer a suitable basis for modeling such trajectories. By combining clothoid segments appropriately, we construct lane change trajectory templates known as bi-elementary paths. A detailed explanation of this construction is provided in Section 2.1.3.

3.1.1 Trajectory Measurements

To predict trajectories, we require historical position, velocity, and heading angle data of the surrounding vehicle. Position information serves as the input for curve fitting, while velocity data is essential for converting predicted trajectories from spatial (arc-length based) to temporal (time-based) coordinates. The heading angle is used to infere the vehicle's intended maneuver.

These data are assumed to be measured by the onboard sensors of the host vehicle. The workflow for collecting these measurements is as follows:

- Initially, there are no measurements.
- As soon as the host vehicle detects a surrounding vehicle, it begins accumulating position measurements over time. Velocity and heading measurements are stored only for the current time instant.

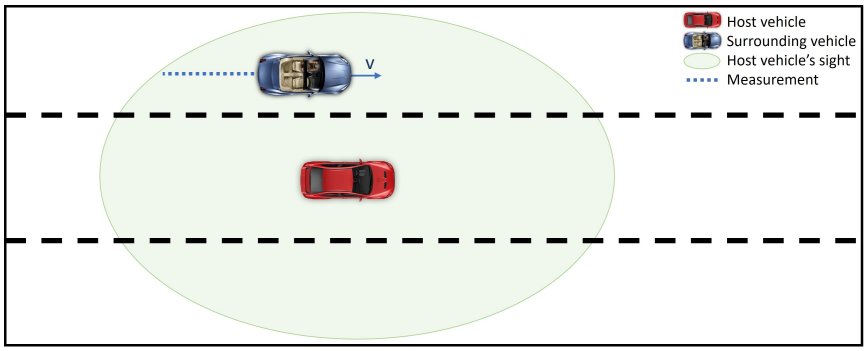

Figure 3.1: Starting accumulating measurements.

- If enough measurements are collected, the older ones are discarded, and the newer ones are added.

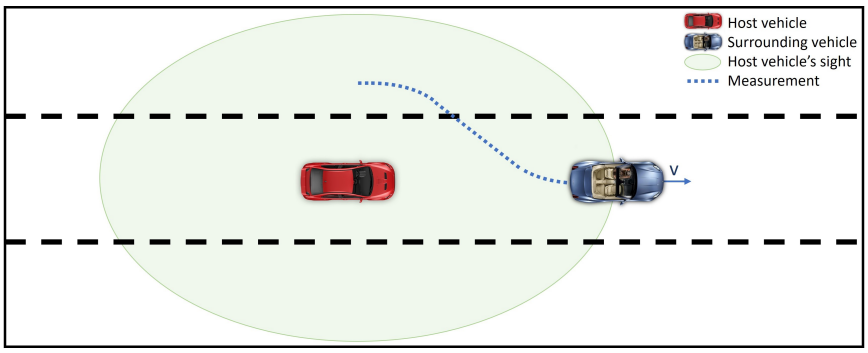

Figure 3.2: Accumulating measurements until the vehicle disappears.

- When the surrounding vehicle disappears from the sight of the host vehicle, measurement collection stops.

Conceptually, these data are measured and represented in the body frame of the host vehicle. To align with our prediction algorithm, they are then transformed into the Frenet frame, as explained in Section 2.2.

Although we assume these measurements are obtained from onboard sensors, we utilize the NGSIM dataset for evaluation purposes. This dataset directly provides position and velocity data in the Frenet frame, sampled at 10 Hz. Figures 3.3 and 3.4 illustrate example trajectories from the dataset.

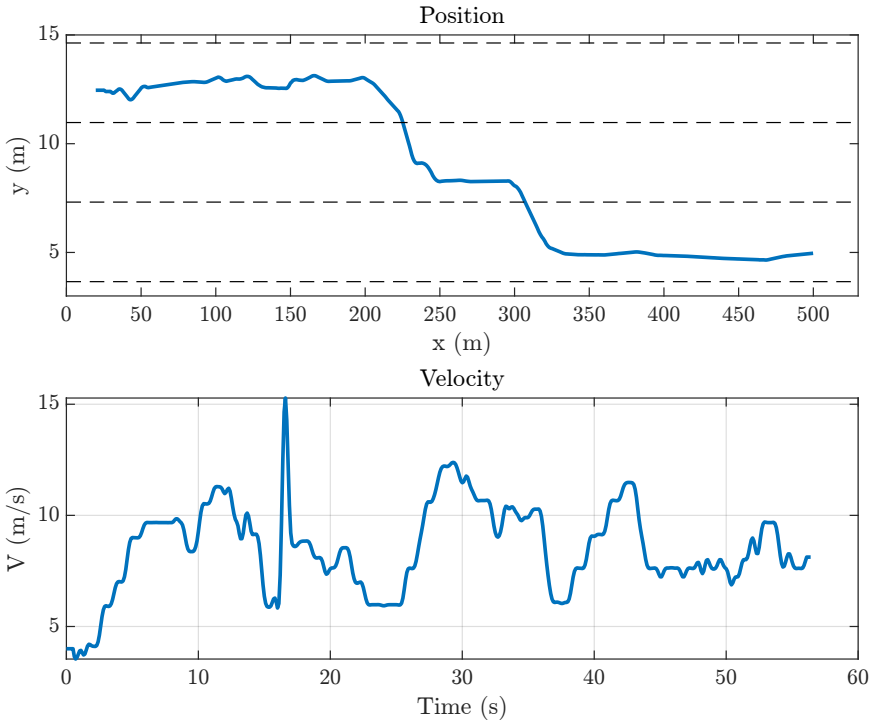

Figure 3.3: Vehicle ID: 159 in I-80 Highway, Position and Velocity Data

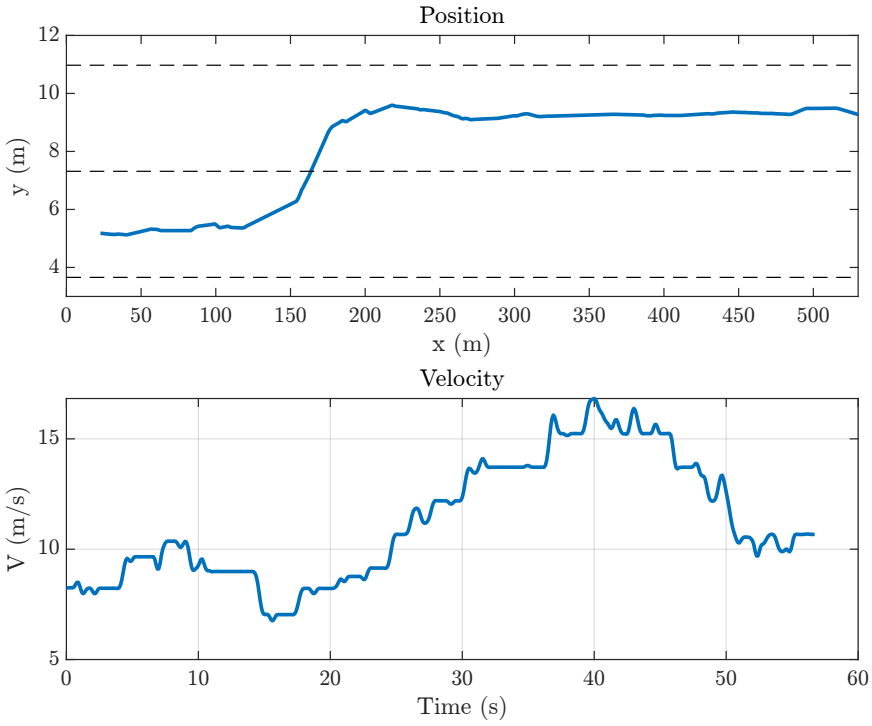

Figure 3.4: Vehicle ID: 76 in US101 Highway, Position and Velocity Data

3.1.2 Illustration of Trajectory Behaviour with Time

In this subsection, it will be shown that as time progresses and measurements are accumulated, the measured position data become increasingly similar to bi-elementary paths.

For illustration, we consider a segment of the position data for Vehicle ID 159 from the NGSIM I-80 dataset.

At time t = 1, the measured data is a straight line and it is not possible to fit a lane changing bi-elementary path to this data. Therefore a bi-elementary path with ΔY= 0 (essentially a line segment) is the only mathematical model that can represent this data, as shown in Figure 3.5.

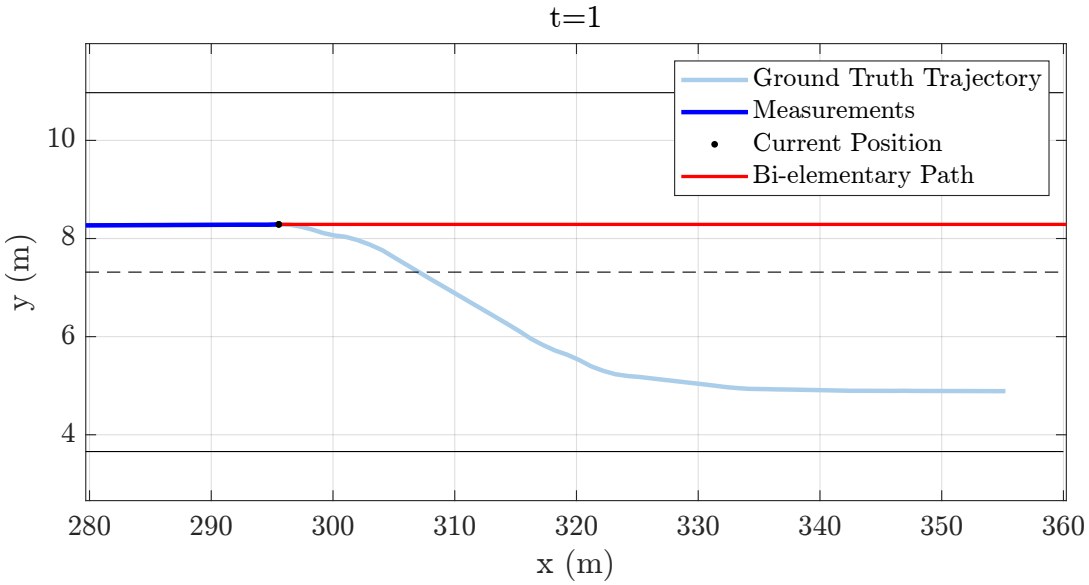

Figure 3.5: Measurements and Trajectories for t=1

At time t = 3, more comprehensive measurements have been accumulated. These measurements now include a part of the lane changing behavior of the vehicle. Therefore, bi-elementary paths can be generated from this data to represent this maneuver. In Figure 3.6 illustrates some of these candidate bi-elementary paths, manually generated and overlaid in red on the measured data points.

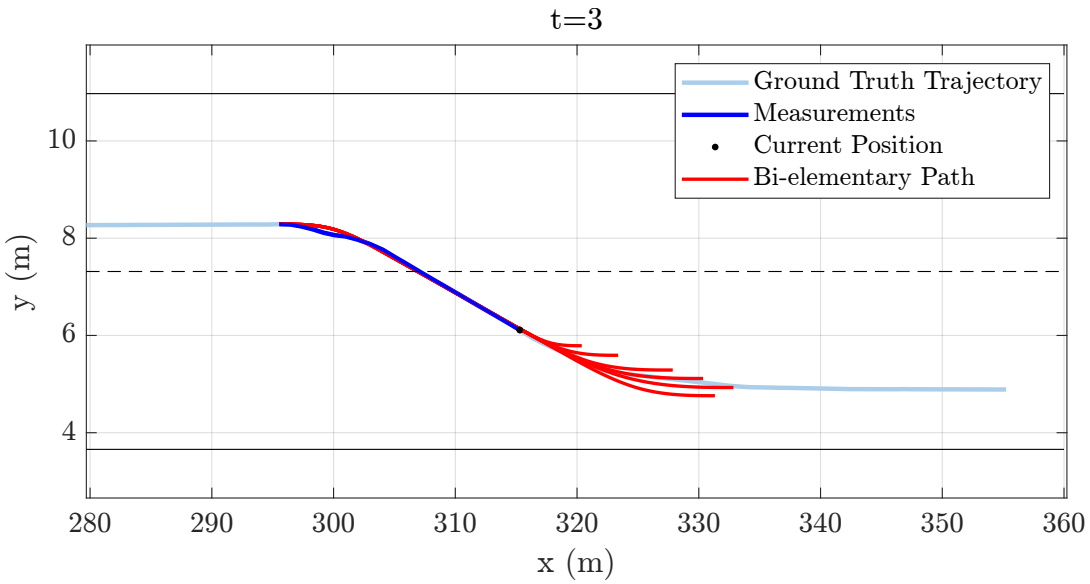

Figure 3.6: Measurements and Trajectories for t=3

As time progress, the measurements nearly corresponds to the entire lane change maneuver as shown in Figure 3.7. Consequently, the possible bi-elementary paths that can represent this data become closer to the ground-truth trajectory.

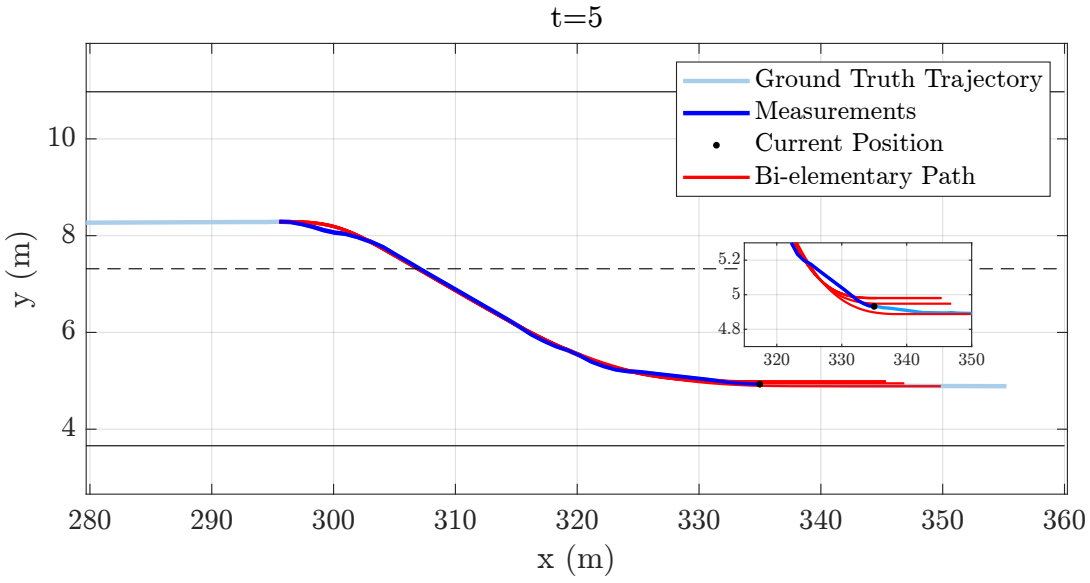

Figure 3.7: Measurements and Trajectories for t=5

To fit an optimal bi-elementary path, least squares curve fitting is applied to the measurement data. This process is explained in detail in Section 3.2

3.2 Lane Change Trajectory Prediction using Least Squares Curve Fitting

To predict future lane change trajectories of surrounding vehicles, we apply least squares curve fitting to measured position data. This process estimates the bi-elementary path parameters that best represent the future trajectory. These parameters are ΔY, S, λ and γ as explained in Section 2.1.3. The least squares curve fitting function used for the implementation in this thesis is the MATLAB built-in 'lsqcurvefit()' function which utilizes 'trust-region-reflective' algorithm .

Since the bi-elementary path model represents lane change trajectories in the Frenet frame, the measured position data must be transformed into the Frenet frame for compatibility with the model. As explained in Section 2.2, body frame measurements can be transformed into the Frenet frame assuming that the road model is known. Hence, this method is applicable for both straight and curved road trajectories when a suitable transformation is applied to the measurement data.

Another property of bi-elementary paths is that they represent trajectories as x and y positions parameterized by arc length, without any explicit time information. Since "prediction" inherently involves future time, these arc length-based coordinates need to be transformed into temporal coordinates, associating each coordinate point with a corresponding time value. This transformation process will be further explained in Section 3.2.2.

3.2.1 Algorithmic Steps of the Prediction

In the prediction algorithm, the first step is extracting the suitable fitting data from the vehicle's position measurements. The fitting data must be in the form that it can be represented by a bi-elementary path. For a bi-elementary path, zero tangent angle only occurs at the beginning and end of the path if ΔY is nonzero. This implies that data containing straight sections is not suitable for fitting a bi-elementary path. In Figure 3.8, examples for both suitable and unsuitable fitting data for the same trajectory can be seen. Since the first data contains an initial straight section, it cannot be well represented by a bi-elementary path.

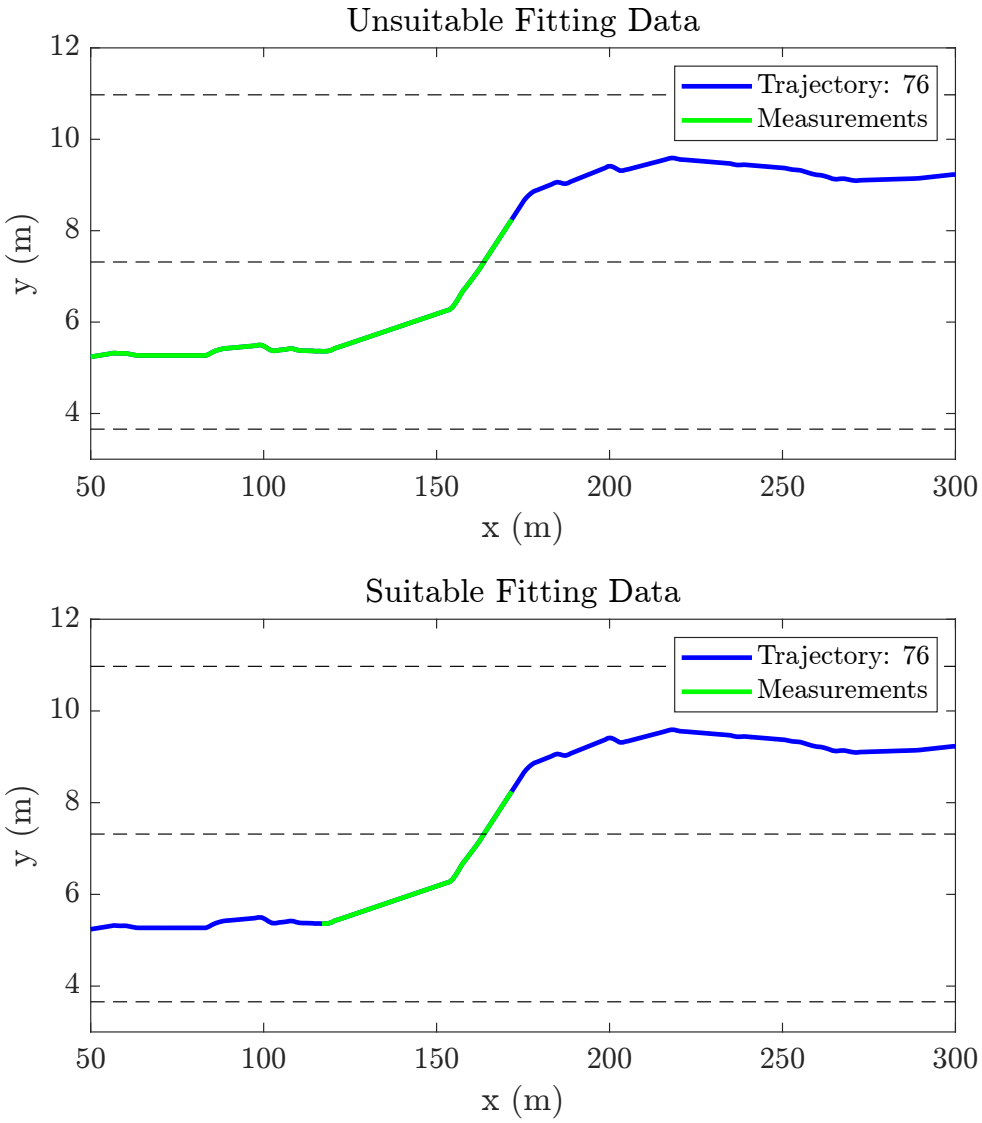

Figure 3.8: Suitable and unsuitable fitting data for straight sections

Another property of a bi-elementary path is that it can only represent a single lane change trajectory. Therefore, fitting data that includes measurements from multiple lane changes cannot be modeled by this approach.

As illustrated in Figure 3.9, for scenarios with multiple lane changes, the fitting data should be reset after every lane change and extracted from the measurements according to the aforementioned considerations.

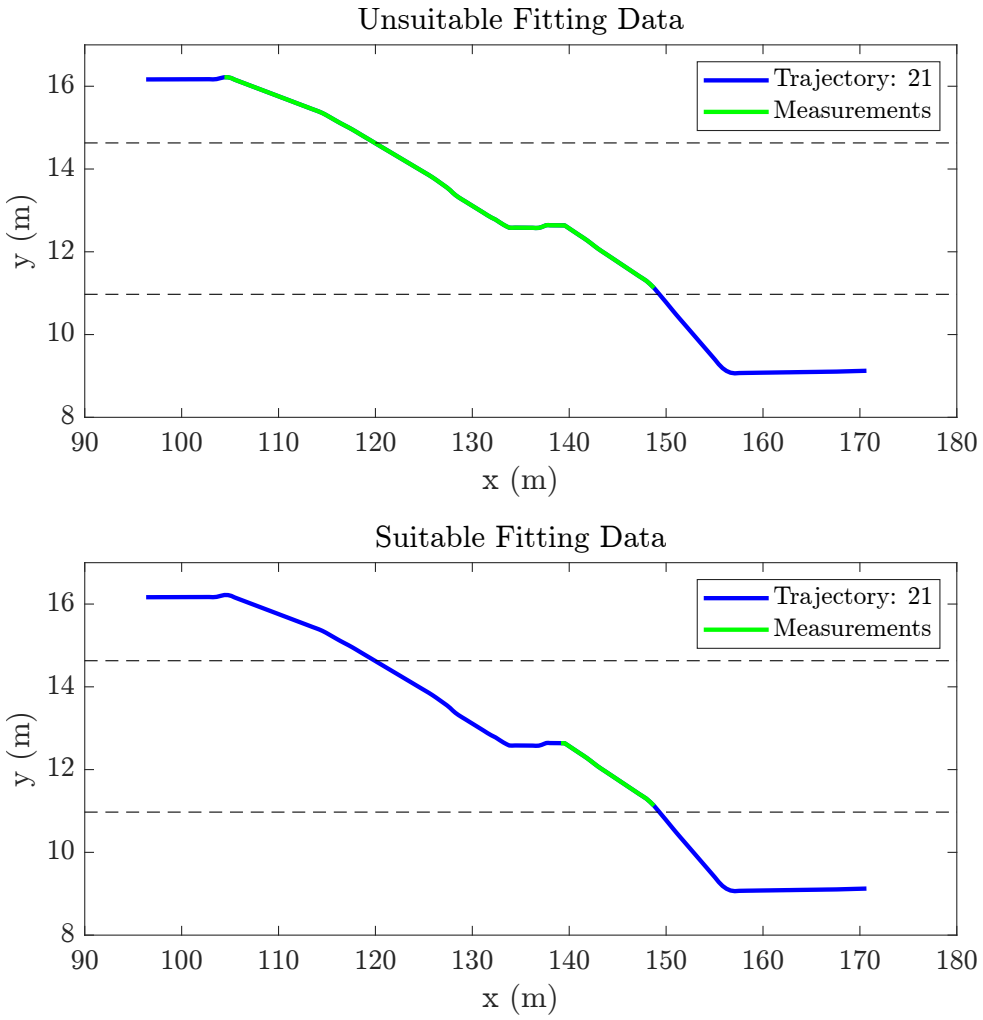

Figure 3.9: Suitable and unsuitable fitting data for a double lane change scenario

To select appropriate data points for curve fitting, the vehicle's heading angle is calculated at each time step. If the heading angle falls below a predefined threshold, the previous measurements are excluded from the fitting data set. This ensures that the fitting process focuses on the lane change maneuver itself, excluding irrelevant data points that cannot be represented by a bi-elementary path model.

While it is conceptually assumed that heading angle information is obtained through direct measurement, the NGSIM dataset does not provide this data. Therefore, the heading angle must be calculated from the available position data, as detailed in Equation 3.1.

$$\Phi(i) = \arctan\left(\frac{P_y(i) - P_y(i-1)}{P_x(i) - P_x(i-1)}\right) \tag{3.1}$$

The parameters in Equation (3.1) are defined as follows:

- $\Phi(i)$ is the heading angle of the vehicle in the Frenet frame
- $P_x(i)$ and $P_y(i)$ are the x and y position measurements in the current time step
- $P_x(i-1)$ and $P_y(i-1)$ are the x and y position measurements in the previous time step

The heading angle $\Phi(i)$ is compared to a predefined threshold angle Φ_{th}, which is set close to zero. If $\Phi(i) < \Phi_{th}$, it is inferred that the vehicle is moving straight, and the fitting model must be reset. The time instant of this reset is stored as $i_{reset} = i$. Additionally, the predicted trajectory when $\Phi(i) < \Phi_{th}$ is considered to be a straight line parallel to the road. Consequently, the curve fitting algorithm is bypassed under this condition.

When $\Phi(i) \geq \Phi_{th}$, the fitting data is extracted as the measurements following the last reset point. This can be mathematically expressed as:

$$fittingData = P(i_{reset} : i) \tag{3.2}$$

where $P(i)$ represents the position measurements at time step i. Equation (3.2) defines the fitting data used for the curve fitting, excluding unsuitable straight-line sections before the actual lane change.

Following the extraction of fitting data, the least-squares curve fitting algorithm requires the definition of appropriate lower and upper bounds, as well as initial conditions, for the parameters to be estimated.

The set of parameters is defined in a state vector as $\mathbf{x} = [\Delta Y, S, \lambda, \gamma]$, where:

- ΔY represents the lateral displacement of the vehicle during the lane change maneuver.
- S denotes the path length.
- λ and γ are additional parameters characterizing the bi-elementary path model which are explained in Section 2.1.3.

To establish appropriate bounds for ΔY parameter, first the lateral deviation of the vehicle from last reset is calculated as

$$dy(i) = P_y(i) - P_y(i_{reset}) \tag{3.3}$$

For the upper bound and the initial condition, the distance to the center of the target lane needs to be calculated. For this calculation, the necessary parameters to be introduced are LW and LO which are the lane width and the lane offset respectively.

The lane width is the distance between the two consecutive lanes, which is typically constant and 3.6 meters for the highways used in NGSIM dataset.

The lane offset is the distance of the vehicle to the right lane boundary. It is calculated as:

$$LO = P_y(i_{reset}) \mod LW \tag{3.4}$$

The lane offset value is compared with a predefined threshold, LO_{th}. Based on this comparison, the target lanes are determined. For instance, if the vehicle is very close to the right lane boundary, i.e., $LO < LO_{th}$, the target lane for a left lane change is the current lane of the vehicle, as illustrated in Figure 3.10.

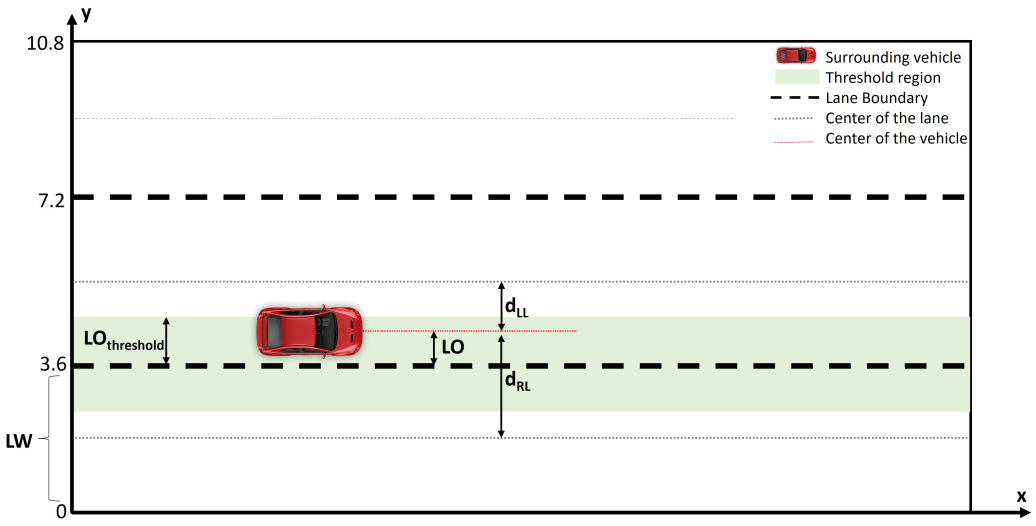

Figure 3.10: Target lanes for the vehicle in close proximity to the right boundary

Conversely, if the vehicle's lateral offset from the right lane boundary, denoted by LO, is greater than or equal to the predefined threshold LO_{th}, as in Figure 3.11, this

implies the vehicle is not in close proximity to the right edge. In such scenarios, the target left lane for potential lane change is the neighboring lane, as expected.

The same principles apply symmetrically for a right lane change.

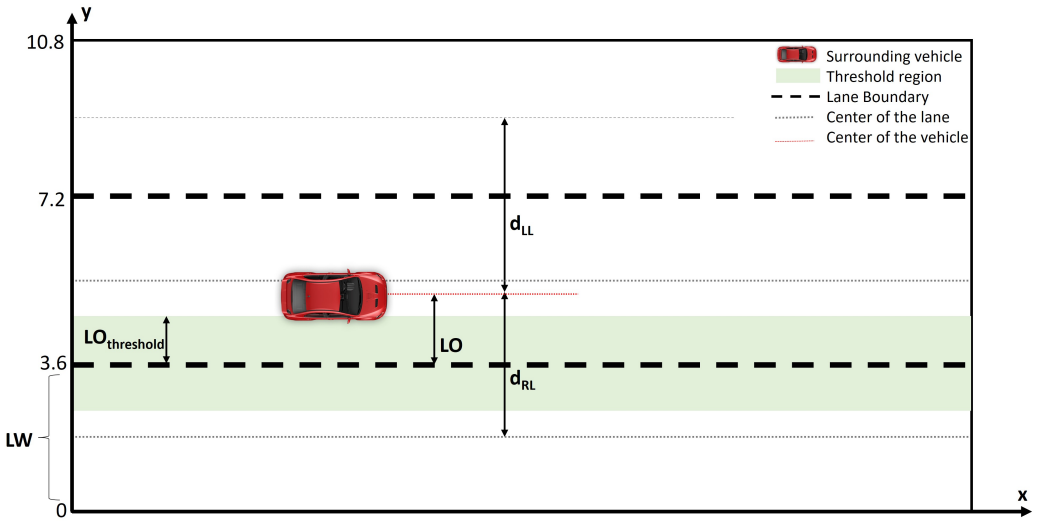

Figure 3.11: Target lanes for the vehicle not in close proximity to the right boundary

In Figures 3.10 and 3.11, the distance to the left lane and the distance to the right lane are denoted as d_{LL} and d_{RL}, respectively. The calculations for them are as follows:

$$d_{LL} = \begin{cases} \left(\left\lceil \frac{P_y(i_{reset})}{LW} \right\rceil + \frac{1}{2} \right) \cdot LW - P_y(i_{reset}) & \text{if } LO \geq LO_{th} \\ \left(\left\lfloor \frac{P_y(i_{reset})}{LW} \right\rfloor + \frac{1}{2} \right) \cdot LW - P_y(i_{reset}) & \text{if } LO < LO_{th} \end{cases} \tag{3.5}$$

$$d_{RL} = \begin{cases} P_y(i_{reset}) - \left(\left\lfloor \frac{P_y(i_{reset})}{LW} \right\rfloor - \frac{1}{2} \right) \cdot LW & \text{if } LO \leq LW - LO_{th} \\ P_y(i_{reset}) - \left(\left\lceil \frac{P_y(i_{reset})}{LW} \right\rceil - \frac{1}{2} \right) \cdot LW & \text{if } LO > LW - LO_{th} \end{cases} \tag{3.6}$$

To determine the direction of the lane change, the sign of the vehicle's lateral deviation, dy, is analyzed. A positive value indicates a lane change to the left, while a negative value indicates a lane change to the right. This directional information,

along with the calculated distance to the target lane, is then used to establish appropriate lower and upper bounds, as well as an initial condition, for the ΔY parameter as follows:

If $dy > 0$

$$\begin{gathered} dy < \Delta Y < d_{LL} + m_{\Delta Y} \\ \Delta Y(0) = d_{LL} \end{gathered} \tag{3.7}$$

If $dy < 0$

$$\begin{gathered} -d_{RL} - m_{\Delta Y} < \Delta Y < dy \\ \Delta Y(0) = -d_{RL} \end{gathered} \tag{3.8}$$

The minimum allowable lateral displacement for the lane change maneuver, reflected by the lower bound for the ΔY parameter, is set to the vehicle's current lateral deviation from its position at the maneuver reset point, dy. It's important to acknowledge that the lower bound doesn't necessarily encompass a full lane change. This is because the vehicle might potentially abort the maneuver and return to its original lane, which can be also represented by a bi-elementary path. The curve fitting process needs to be able to capture such scenarios as well, which is why the lower bound for ΔY is established accordingly.

The upper bound for the ΔY parameter is established to ensure sufficient lateral displacement during the lane change maneuver. This bound is set to a value exceeding the calculated distance to the target lane by a predefined safety margin $m_{\Delta Y}$. This margin accounts for the cases where the final destination of the vehicle might not be exactly the center of the target lane.

The initial condition for ΔY is chosen to be the calculated distance to the target lane.

For the other parameters, the established boundaries and initial conditions are as follows:

$$\begin{gathered} S_{data} < S < 200 \\ S(0) = 60 \end{gathered} \tag{3.9}$$

where S_{data} is the length of the fitting data.

$$\begin{gathered} 0 < \lambda < 1 \\ \lambda(0) = 0.5 \end{gathered} \tag{3.10}$$

$$
\begin{aligned}
0 < \gamma < 1 \\
\gamma(0) = 0.8
\end{aligned}
\tag{3.11}
$$

For λ and γ, the boundaries are selected based on their well-defined regions, as expressed in Equation 2.20.

The optimal values for these parameters within the defined ranges are estimated applying least-squares curve fitting.

For illustration, a left lane change trajectory from the NGSIM I-80 dataset is used for curve fitting and parameter estimation. Let say at the reset point, at which the measurements for fitting data are started to collected, time is defined as zero ($t = 0$). The trajectory and measurement data points at time $t = 1$ s can be observed in Figure 3.12.

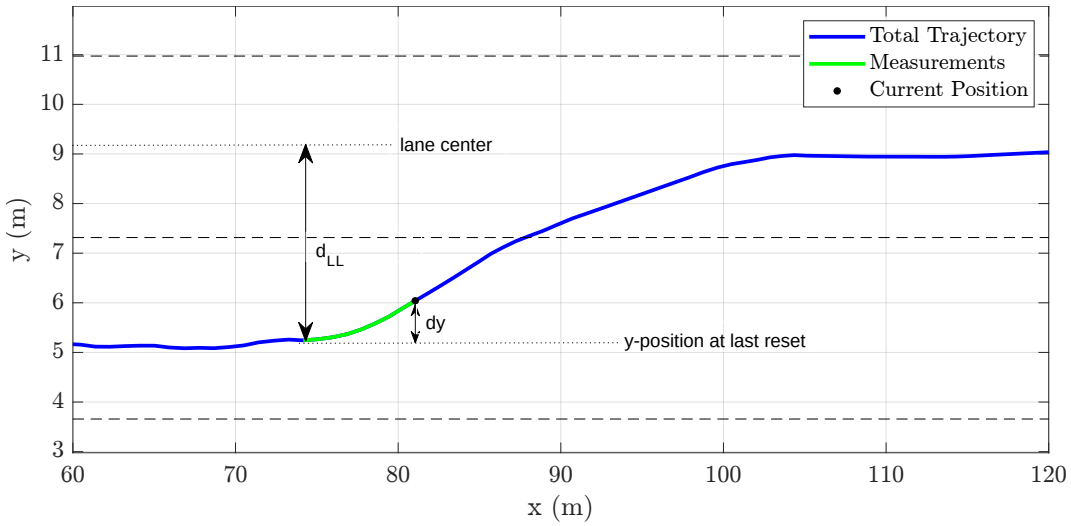

Figure 3.12: Left lane change trajectory at t=1 s

It is found that the lateral displacement, dy is 0.79 m. As this value is positive, it is inferred that the target lane for the maneuver is the left lane and the distance to that lane, d_{LL} is calculated as 3.9 m. Additionally, the length of the measurement data, S_{data} is found to be 6.94 m. Let $m_{\Delta Y}$ be 0.2 m.

Based on this prior information, the parameter boundaries and the initial conditions are established as:

$$
\begin{aligned}
0.79 < \Delta Y < 4.1 \quad &, \quad \Delta Y(0) = 3.9 \text{ m} \\
6.94 < S < 200 \quad &, \quad S(0) = 60 \text{ m} \\
0 < \lambda < 1 \quad &, \quad \lambda(0) = 0.5 \\
0 < \gamma < 1 \quad &, \quad \gamma(0) = 0.8
\end{aligned}
\tag{3.12}
$$

Using this setting, least squares curve fitting is applied for the measurement data shown in Figure 3.12. The estimated bi-elementary path parameters are as follows:

$$
\begin{aligned}
\Delta Y_{est} &= 4.02 \text{ m} \\
S_{est} &= 35.55 \text{ m} \\
\lambda_{est} &= 0.19 \\
\gamma_{est} &= 0.68
\end{aligned}
\tag{3.13}
$$

The generated bi-elementary path with these parameters is plotted on top of the vehicle trajectory as 'Predicted Trajectory' in Figure 3.13. The x and y positions of the bi-elementary path are computed for the arc-lengths: $s = (0 : dx : S_{est})$, where dx is the arc-length increment and S_{est} is the estimated total arc-length.

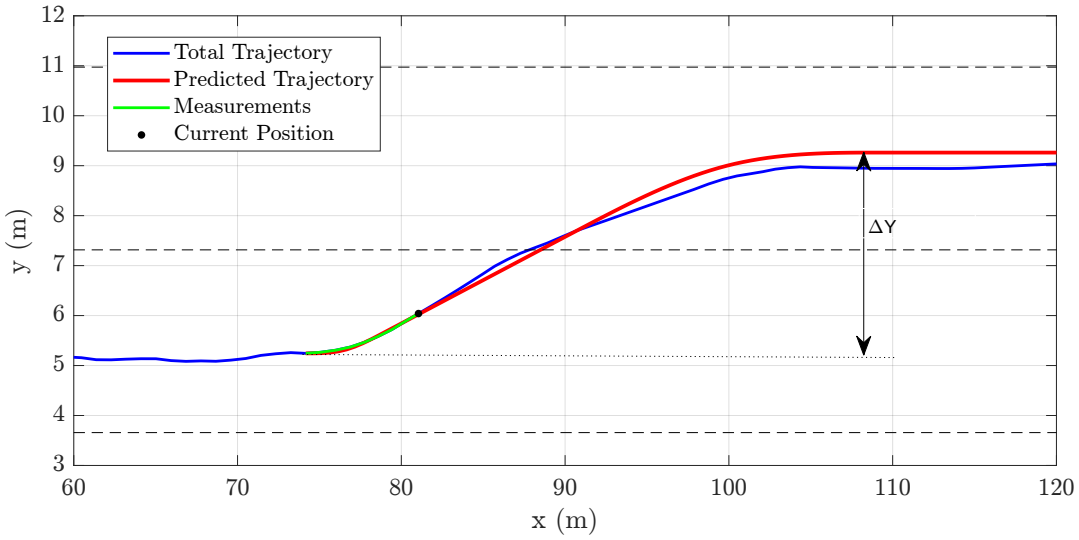

Figure 3.13: Left lane change trajectory and the prediction at t=1 s

Note that the generated bi-elementary path is extended with a line segment at the end. Since the last tangent angle and the curvature is zero for bi-elementary paths, adding a line segment does not cause any discontinuity, and this extension is intended to allow for predictions over longer horizons.

To evaluate the performance of the prediction, the position error between the generated bi-elementary path and the ground truth trajectory is calculated for points following the current position. Figure 3.14 illustrates the x and y position errors with respect to the path length, with the path length defined as zero at the current position.

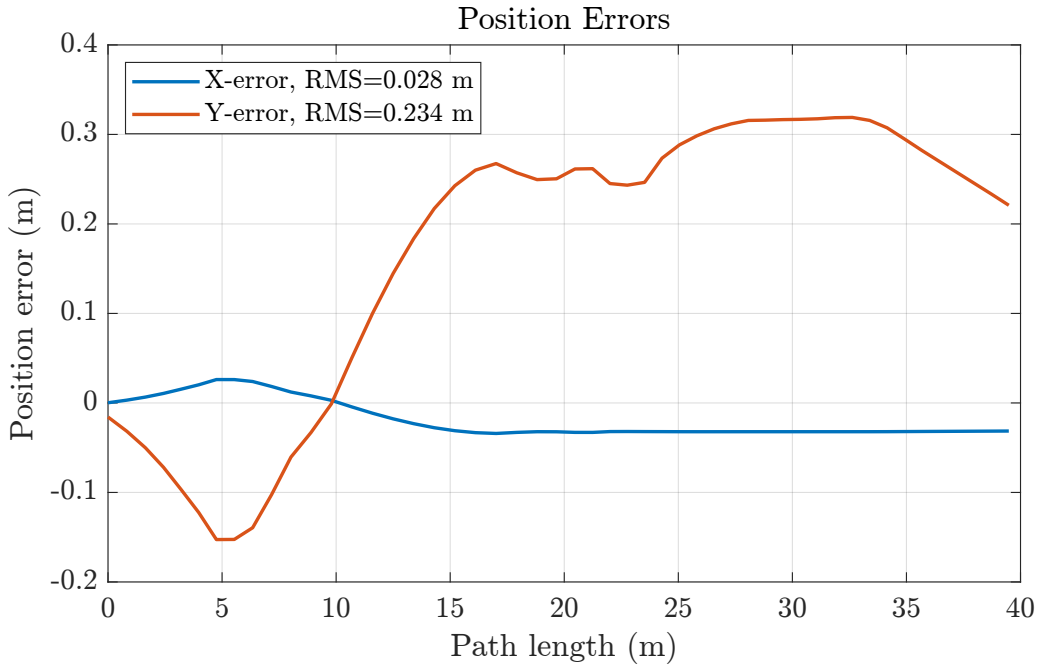

Figure 3.14: Predicted position errors at t=1 s for the left lane change trajectory

For the same trajectory, the curve fitting for a future time at $t = 3$ s is illustrated next. As seen in Figure 3.15, the measurement data is larger with $S_{data} = 18.54$ m and it contains majority of the lane-changing maneuver. Consequently, dy is larger, being 2.66 m, which will make the boundaries for ΔY more confined. As the starting point of the measurements remains same as the previous one, d_{LL} stays the same. Let $m_{\Delta Y}$ be 0.2 m

The parameter boundaries and the initial conditions are established as:

$$\begin{aligned} 2.66 < \Delta Y < 4.1 \quad &, \quad \Delta Y(0) = 3.9 \text{ m} \\ 18.54 < S < 200 \quad &, \quad S(0) = 60 \text{ m} \\ 0 < \lambda < 1 \quad &, \quad \lambda(0) = 0.5 \\ 0 < \gamma < 1 \quad &, \quad \gamma(0) = 0.8 \end{aligned} \tag{3.14}$$

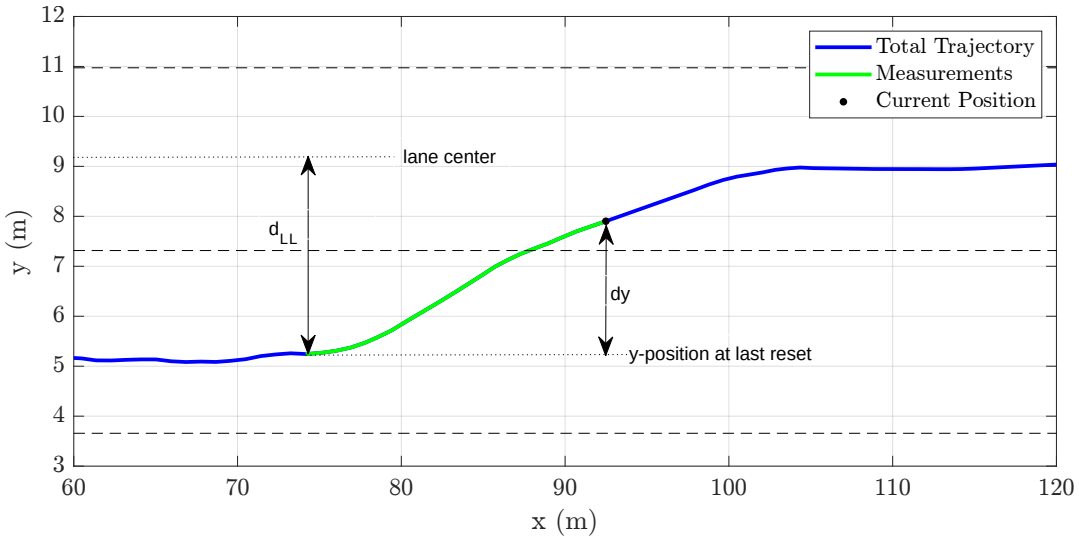

Figure 3.15: Left lane change trajectory at t=3 s

After applying the curve fitting, the parameters are estimated as follows:

$$\begin{aligned} \Delta Y_{est} &= 3.75 \text{ m} \\ S_{est} &= 40.46 \text{ m} \\ \lambda_{est} &= 0.12 \\ \gamma_{est} &= 1 \end{aligned} \tag{3.15}$$

The generated bi-elementary path with these parameters can be seen in Figure 3.16.

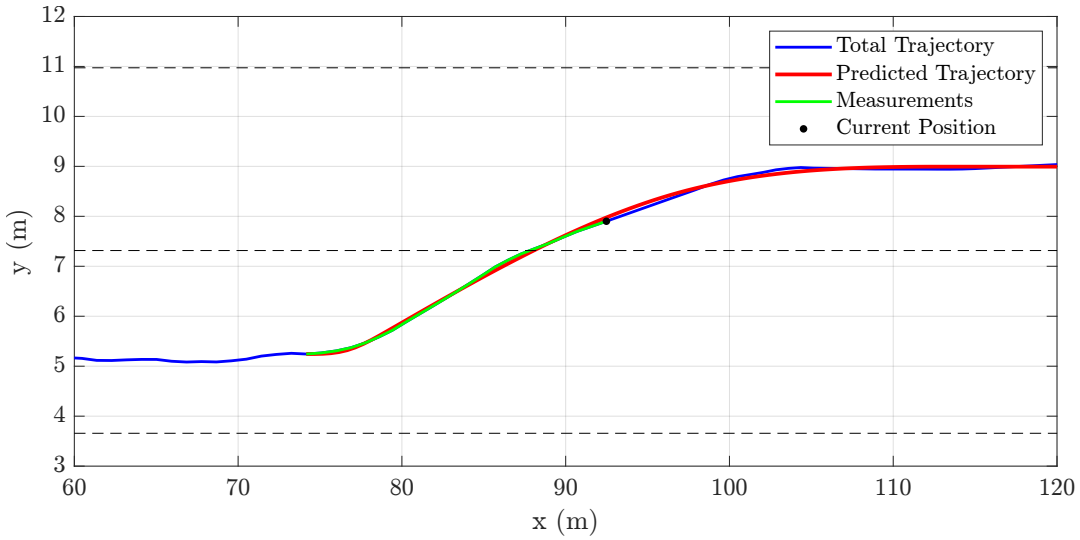

Figure 3.16: Left lane change trajectory and the prediction at t=3 s

The position errors, along with the (the root mean square) RMS results, are shown in Figure 3.14.

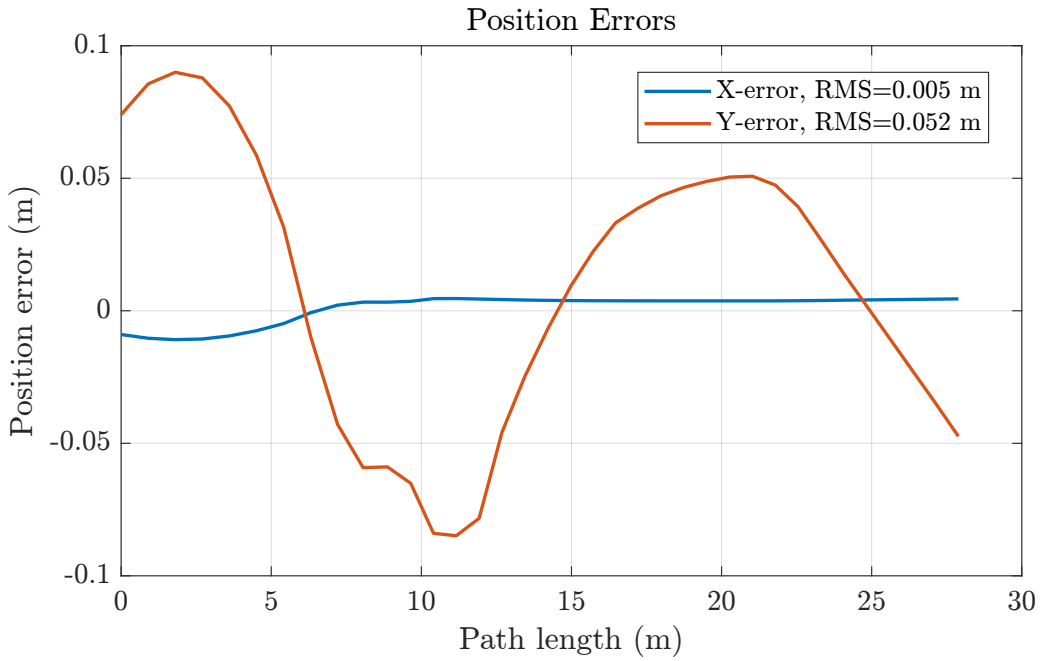

Figure 3.17: Predicted position errors at t=3 s for the left lane change trajectory

Given that the measurements taken at t = 3 s encompass a more comprehensive portion of the lane-changing maneuver compared to those at t = 1 s, the prediction performance improves notably. Figures 3.14 and 3.17 illustrate this enhancement: the

root mean square (RMS) of the x position error decreases from 0.028 m to 0.005 m, and the RMS of the y position error similarly drops from 0.234 m to 0.052 m.

For the next example, a right lane change trajectory from the NGSIM I-80 dataset is examined. At the reset point, time is defined as zero. In Figure 3.18, the acquired measurement data at time $t = 1$ s is illustrated.

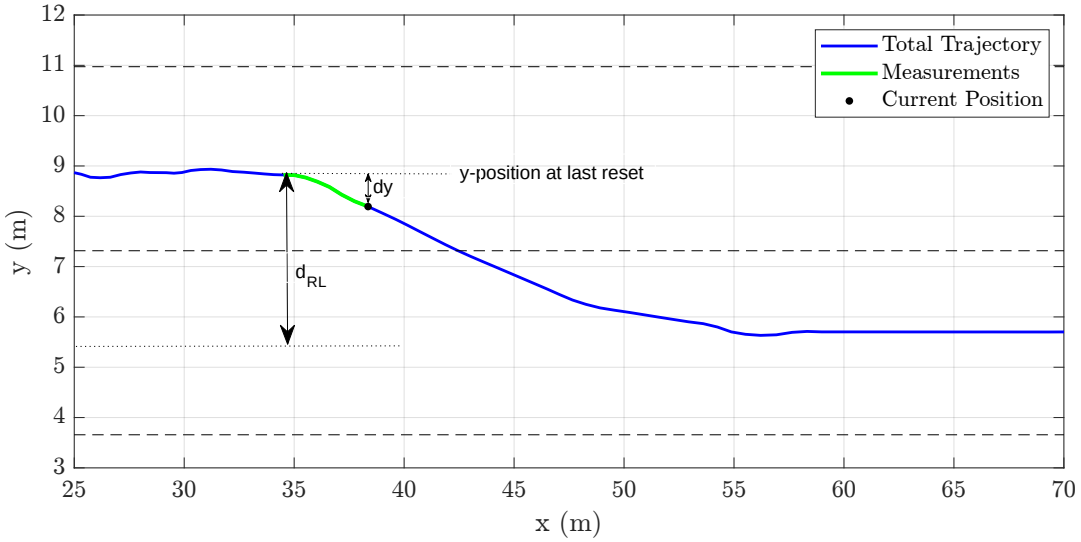

Figure 3.18: Right lane change trajectory at t=1 s

At this point, the lateral deviation dy is found to be -0.63 m, indicating a right lane change maneuver. The lateral distance to the center of the right lane, d_{RL}, is calculated to be 3.34 m. The length of the measurement data, S_{data}, to be used in fitting is determined to be 3.91 m. Let $m_{\Delta Y}$ be 0.2 m

Based on this prior information, the parameter boundaries and the initial conditions are established as:

$$\begin{aligned} -3.54 < \Delta Y < -0.63 \quad &, \quad \Delta Y(0) = -3.34 \text{ m} \\ 3.91 < S < 200 \quad &, \quad S(0) = 60 \text{ m} \\ 0 < \lambda < 1 \quad &, \quad \lambda(0) = 0.5 \\ 0 < \gamma < 1 \quad &, \quad \gamma(0) = 0.8 \end{aligned} \tag{3.16}$$

After applying the least squares curve fitting, the estimated bi-elementary path pa-

rameters are found as below:

$$\begin{aligned} \Delta Y_{est} &= -2.71 \text{ m} \\ S_{est} &= 17.07 \text{ m} \\ \lambda_{est} &= 0.21 \\ \gamma_{est} &= 0.52 \end{aligned} \tag{3.17}$$

The generated bi-elementary path with these parameters is illustrated in Figure 3.19.

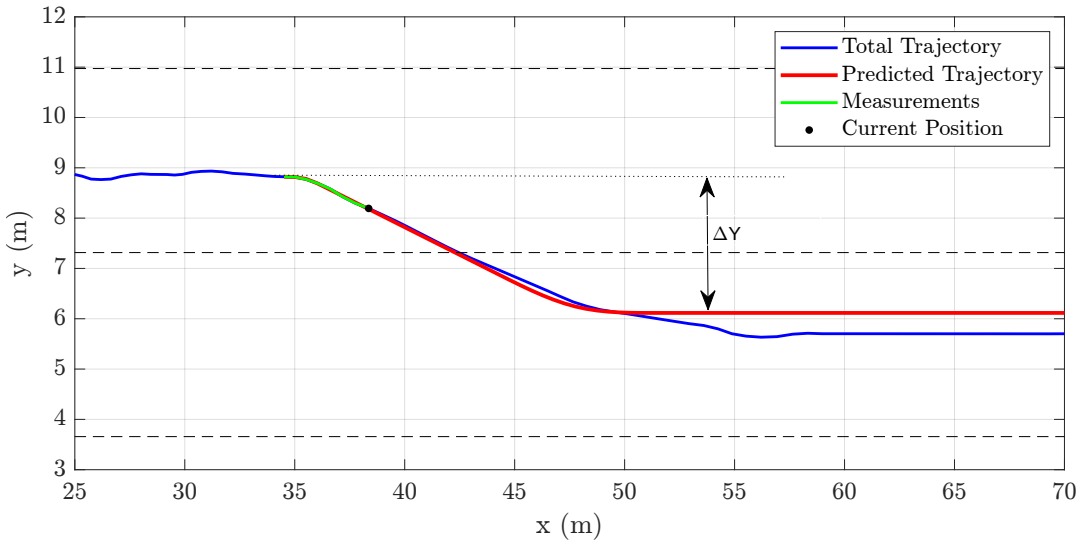

Figure 3.19: Right lane change trajectory and the prediction at t=1 s

Figure 3.20 illustrates the x and y position errors along with the corresponding RMS results for the measurement data at t=1 s.

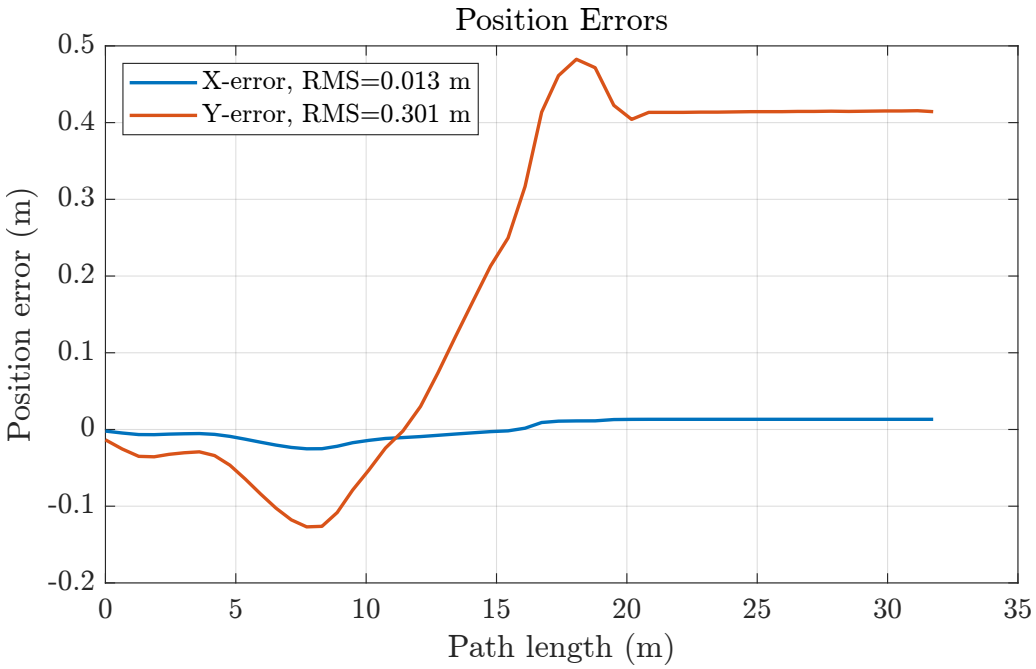

Figure 3.20: Predicted position errors at t=1 s for the right lane change trajectory

For the same right lane trajectory, the current position of the vehicle and the measurements taken at time $t = 6$ s can be seen in Figure 3.21.

At this point the lateral deviation and the length of the measurement data increase as expected. The calculations for them are dy is -2.69 m and S_{data} is 15.32 m. Let $m_{\Delta Y}$ be 0.2 m. Using these values along with the pre-calculated d_{RL} for the same trajectory, the boundaries and the initial conditions for the bi-elementary path parameters to be estimated are as follows:

$$\begin{aligned} -3.54 < \Delta Y < -2.69 \quad &, \quad \Delta Y(0) = -3.34 \text{ m} \\ 15.32 < S < 200 \quad &, \quad S(0) = 60 \text{ m} \\ 0 < \lambda < 1 \quad &, \quad \lambda(0) = 0.5 \\ 0 < \gamma < 1 \quad &, \quad \gamma(0) = 0.8 \end{aligned} \tag{3.18}$$

The bi-elementary path parameters after applying the least squares cruve fitting are found as below:

$$\begin{aligned} \Delta Y_{est} &= -3.18 \text{ m} \\ S_{est} &= 26.71 \text{ m} \\ \lambda_{est} &= 0.08 \\ \gamma_{est} &= 0.85 \end{aligned} \tag{3.19}$$

The generated bi-elementary path with these parameters can be seen in Figure 3.21 as 'Predicted Trajectory'.

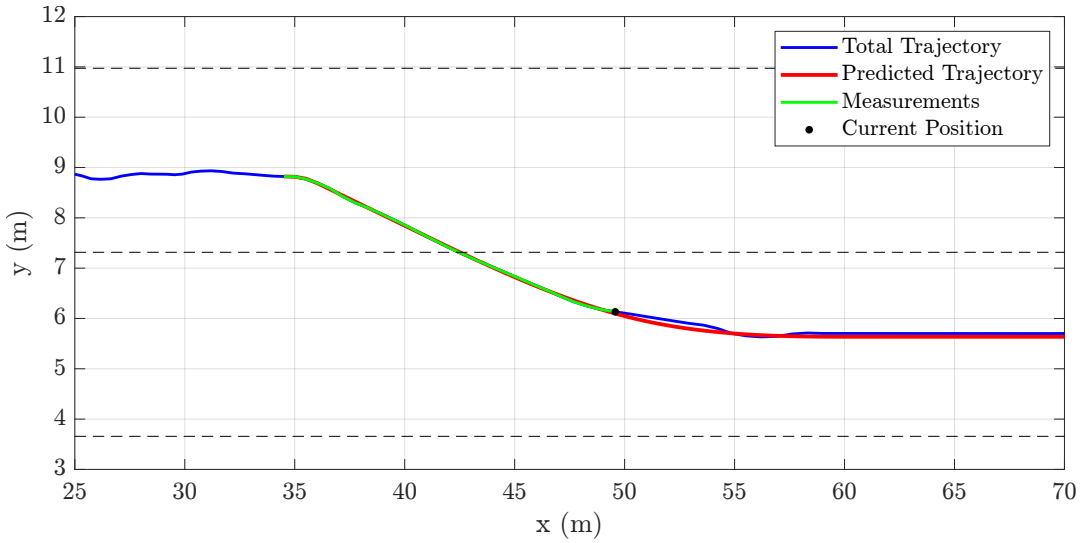

Figure 3.21: Right lane change trajectory and the prediction at t=6 s

Figure 3.14 illustrates the x and y position errors along with the corresponding RMS results for the measurement data at t=6 s.

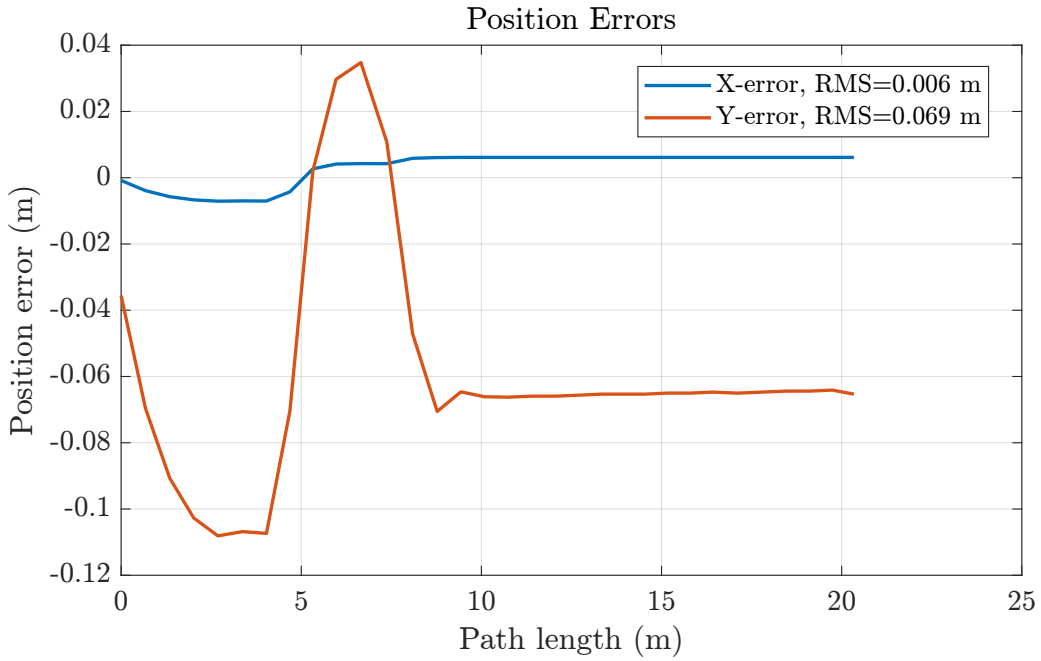

Figure 3.22: Predicted position errors at t=6 s for the right lane change trajectory

Similar to the left lane change example, larger and more comprehensive measurement data also improves the prediction results for this right lane trajectory. Figures 3.20 and 3.22 illustrate that the RMS of the x position error decreases from 0.013 m to 0.006 m, and the RMS of the y position error decreases from 0.301 m to 0.069 m.

In the algorithm, after estimating the bi-elementary path parameters, each coordinate point along the path needs to be assigned a corresponding time value to make the prediction for the defined future time. This transformation from spatial coordinates to temporal coordinates is detailed in Section 3.2.2.

The pseudo-code for the prediction algorithm can be found in Algorithm 1.

Algorithm 1 Lane Change Prediction Algorithm

Input: Measured trajectory (position, velocity (v), acceleration (a) and heading angle (Φ)) in Frenet frame

Output: Predicted trajectory of the vehicle

for each point along the trajectory **do**
  if $\Phi < \Phi_{th}$ **then**
    $predictedTrajectory$ is a straight line
    reset the model
  else
    $fittingData \leftarrow$ measurement after last reset
    calculate the distance to the objective lane center
    set the initial conditions, IC, lower bounds, LB, and the upper bounds, UB, for the parameters
    $[\Delta Y_{est}, S_{est}, \lambda_{est}, \gamma_{est}] =$ lsqcurvefit(bielementaryModel, $fittingData, IC, LB, UB$) {estimate the bi-elementary path parameters}
    $predictedTrajectory$ = bielementaryModel($\Delta Y_{est}, S_{est}, \lambda_{est}, \gamma_{est}$)
  end if
  $predictedTrajectoryTime$ = space2time($\Delta Y_{est}, S_{est}, \lambda_{est}, \gamma_{est}$, v, a) {transform the prediction to temporal coordinates}
end for

3.2.2 Transformation from Spatial Coordinates to Temporal Coordinates

While the bi-elementary path model offers a robust framework for generating vehicle trajectories consistent with observed behavior, it is necessary to include the dimension of time for practical predictions. This section explains the methodology used to map spatial coordinates along the bi-elementary path to their corresponding temporal coordinates, thus enabling accurate forecasting of vehicle positions at specific future time points.

Since the bi-elementary path model generates x and y coordinates as a function of arc length (s), the transformation to temporal coordinates requires finding the corresponding arc length for each specified time value.

Let t_{pre} denote the prediction time, defined as the time elapsed since the vehicle's current position, and let t_{final} denote the last time point within the prediction time horizon. We are interested in finding the arc-length values corresponding to the time points within the prediction horizon, given by $t_{pre} = (0 : dt : t_{final})$, where dt is the time increment (0.1 s for the NGSIM dataset).

To realize this transformation, it is necessary to model the vehicle's velocity within the prediction horizon. Several assumptions can be employed for this purpose, including the constant velocity (CV), constant acceleration (CA), and partial acceleration (PA) models.

- Constant Velocity (CV): This model assumes that the vehicle maintains a constant velocity throughout the prediction horizon.
- Constant Acceleration (CA): The CA model assumes a constant acceleration, leading to a linear increase or decrease in velocity over time.
- Partial Acceleration (PA): This model combines aspects of both CV and CA models. It assumes a constant acceleration for a certain duration, after which the velocity becomes constant.

In this thesis, we explore the performance of each model under various conditions to determine their suitability for accurate trajectory prediction.

Let v and a represent the measured velocity and acceleration of the vehicle at time $t_{pre} = 0$, respectively. The predicted velocity v_{pre} at any time within the prediction horizon can be formulated as follows:

$$v_{pre}(t_{pre}) = \begin{cases} v & \text{if} \quad t_{pre} = 0 \\ v_{pre}(t_{pre} - dt) + a \cdot dt & \text{if} \quad 0 < t_{pre} \leq t_{acc} \\ v_{pre}(t_{acc}) & \text{if} \quad t_{acc} < t_{pre} \leq t_{final} \end{cases} \tag{3.20}$$

where:

t_{acc}	: Duration of the constant acceleration phase.
$v_{pre}(t_{pre})$	: Predicted velocity at time t_{pre}.
$v_{pre}(t_{pre} - dt)$	: Predicted velocity at the previous time step.
dt	: Time increment.
v	: Measured velocity.
a	: Measured acceleration.

In Equation (3.20);

- The first case ($t_{pre} = 0$) represents the initial condition where the predicted velocity is simply the measured velocity.
- The second case ($0 < t_{pre} < t_{acc}$) models the period of constant acceleration. The predicted velocity at the current time step is calculated by adding the acceleration multiplied by the time increment, dt, to the predicted velocity from the previous time step.
- The third case ($t_{acc} \leq t_{pre} \leq t_{final}$) represents the period after the constant acceleration phase, where the velocity remains constant at the value it reached at the end of the acceleration phase.

It is important to note that in Equation (3.20);

- Setting $t_{acc} = 0$ results in the constant velocity (CV) model.
- Setting $t_{acc} = t_{final}$ corresponds to the constant acceleration (CA) model.

- Intermediate t_{acc} values imply partial acceleration (PA) model.

Once the velocity model is determined, the corresponding arc-length (s) values along the bi-elementary path for each time instant within the prediction horizon can be calculated as follows:

$$s(t_{pre}) = \begin{cases} S_{data} & \text{if} \quad t_{pre} = 0 \\ s(t_{pre} - dt) + v_{pre}(t_{pre} - dt) \cdot dt + \frac{1}{2} \cdot a \cdot dt^2 & if \quad 0 < t_{pre} \leq t_{acc} \\ s(t_{pre} - dt) + v_{pre}(t_{pre} - dt) \cdot dt & \text{if} \quad t_{acc} < t_{pre} \leq t_{final} \end{cases} \tag{3.21}$$

where:

$s(t_{pre})$: Arc-length at time t_{pre}
$s(t_{pre} - dt)$: Arc-length at the previous time instant
S_{data} : Total length of the measured data.

At $t_{pre} = 0$, the corresponding arc-length of the bi-elementary path is not zero because the starting point of the bi-elementary path coincides with the initial point of the measurement data, which is defined as the reset point. Therefore, the arc-length at $t_{pre} = 0$ is equivalent to the total length of the measurement data, denoted as S_{data}.

For $t_{pre} > 0$, the arc-length at each subsequent time step is determined by adding the incremental distance traveled during that time interval to the arc-length at the previous time step. The calculation of this incremental distance depends on the chosen velocity model.

- If $t_{pre} \leq t_{acc}$, the constant acceleration (CA) model is applied.
- If $t_{pre} > t_{acc}$, the constant velocity (CV) model is applied.

If the final arc-length value within the prediction horizon, $s(t_{final})$, is larger than the estimated arc-length of the bi-elementary path after fitting, S_{est}, the bi-elementary path is extended by a straight line segment. The length of this extension is equal to the difference of these values: $s(t_{final}) - S_{est}$. This extension does not introduce

any discontinuity, as the bi-elementary path concludes with zero curvature and zero tangent angle. The purpose is to enable trajectory predictions over extended horizons.

The pseudo-code illustrating the process of transforming spatial coordinates to temporal coordinates is presented in Algorithm 2.

Algorithm 2 Mapping to Temporal Coordinates

Input: v, a, S_{data}, $x_{est} = [\Delta Y_{est}, S_{est}, \lambda_{est}, \gamma_{est}]$

Output: Predicted trajectory with a function of time

for $i = 0$ to t_{final}/dt **do**
  $t_{pre} \leftarrow i \cdot dt$
  if $t_{pre} = 0$ **then**
    $v_{pre}(i) \leftarrow v$
    $s(i) \leftarrow S_{data}$
  else if $t_{pre} \leq t_{acc}$ **then**
    $v_{pre}(i) \leftarrow v_{pre}(i-1) + a \cdot dt$
    $s(i) \leftarrow s(i-1) + v_{pre}(i-1) \cdot dt + 0.5 \cdot a \cdot dt^2$
  else
    $v_{pre}(i) \leftarrow v_{pre}(i-1)$
    $s(i) \leftarrow s(i-1) + v_{pre}(i-1) \cdot dt$
  end if
  if $s(i) > S_{est}$ **then**
    $lenLine \leftarrow s(i) - S_{est}$
  else
    $lenLine \leftarrow 0$
  end if
  $predictedTrajectory(i)$ = extendedBielementaryModel($s(i), x_{est}, lenLine$)
end for

To illustrate the application of the transformation, consider the previous trajectory prediction example shown in Figure 3.13. At time $t = 1$ s, (corresponds to $t_{pre} = 0$), the vehicle's velocity and acceleration are measured as 8.09 m/s and -0.42 m/s^2, respectively. S_{data} is calculated as 6.94 m. For a prediction horizon of 5 seconds ($t_{final} = 5$ s) and a time increment of 0.1 s ($dt = 0.1$ s), the predicted velocities within the 5-second horizon for the aforementioned velocity models are illustrated in

Figure 3.23.

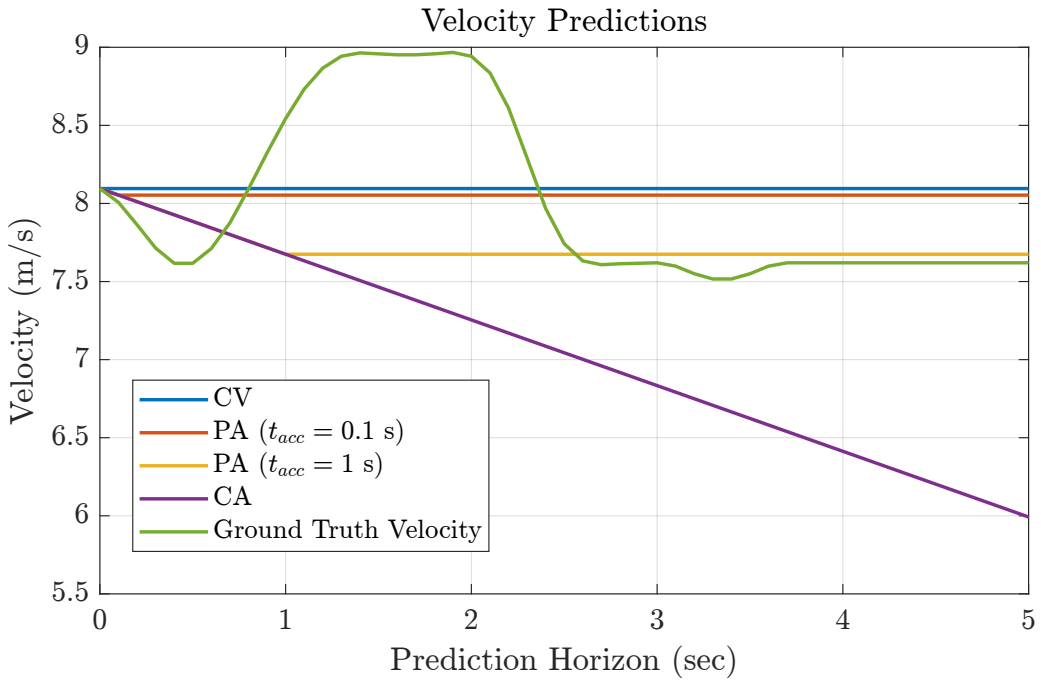

Figure 3.23: Predicted velocities for different velocity models for the left lane change example at t=1 s

Using the velocity models and Equation (3.21), the bi-elementary path coordinates are transformed into temporal coordinates, resulting in the time-based position predictions shown in Figure 3.24.

Figure 3.23 reveals that the constant acceleration (CA) model may not be suitable for this scenario. The measured acceleration is negative, which, under a CA assumption, leads to a linearly decreasing velocity prediction. However, the ground truth velocity exhibits a non-monotonic behavior, decreasing only briefly before increasing again. This discrepancy is further highlighted in Figure 3.24, where the RMSE calculations for position predictions demonstrate a higher x-error for the CA model compared to other models. Similarly for the PA_1 (partial acceleration with $t_{acc} = 1s$) model, although the velocity prediction improves after approximately $t_{pre} = 3$ s, the predictions before this time is inaccurate, leading to poorer prediction performance compared to CV (constant velocity) and $PA_{0.1}$ (acceleration measurement is used only for the initial time instant) models.

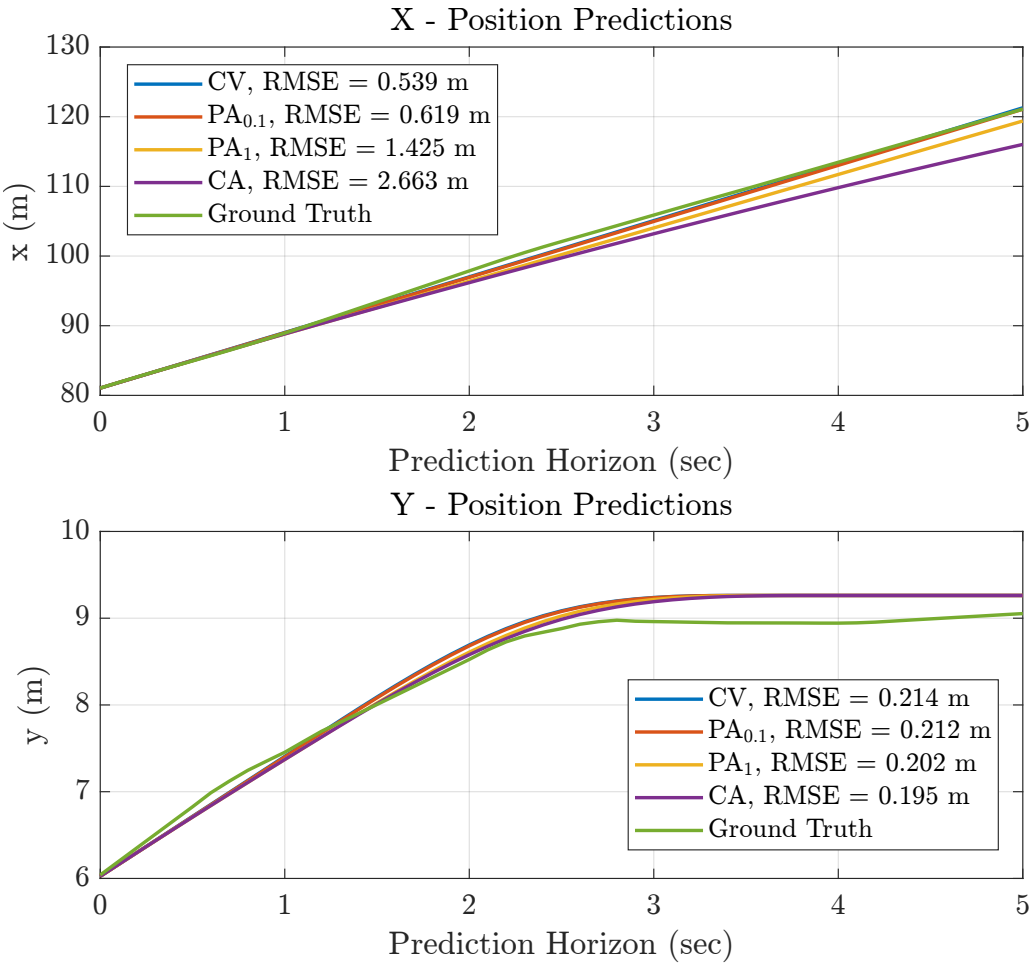

Figure 3.24: Predicted positions and RMS errors for the left lane change trajectory at t=1 s

Another example examines the previous trajectory prediction for the right lane change maneuver illustrated in Figure 3.19. At the initial time instant, the vehicle's velocity and acceleration are measured as 5.99 m/s and 2.82 m/s^2, respectively. The length of the measurement data, S_{data}, is 3.91 m. The predicted velocities within a 5-second horizon for the aforementioned velocity models are shown in Figure 3.25.

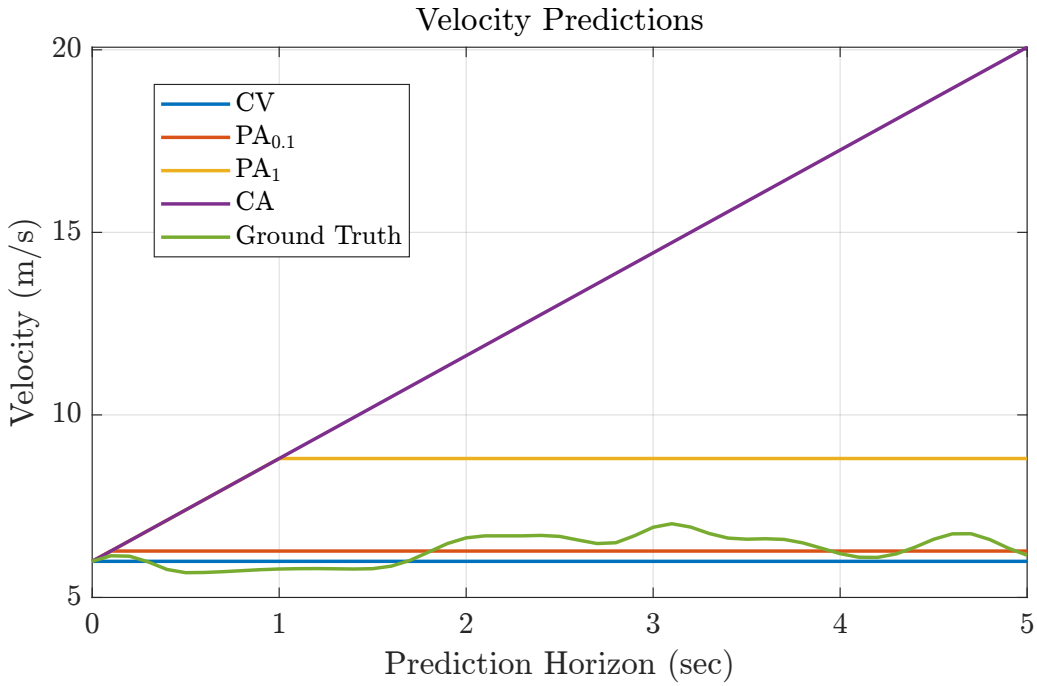

Figure 3.25: Predicted velocities for different velocity models for the right lane change example at t=1 s

In this scenario, the acceleration measurement has a larger magnitude compared to the previous example. Therefore, the difference between the velocity predictions generated by various velocity models become more pronounced. Given that the ground truth velocity remains relatively stable within the prediction horizon, the CA model leads to least accurate position prediction compared to other models as expected, as shown in 3.26. The position error for the PA_1 model is also larger compared to $PA_{0.1}$ and CV models since the duration of the measured acceleration is much shorter than 1 seconds. Among the examined velocity models, the $PA_{0.1}$ model yields the most accurate position prediction for this scenario.

Figures 3.24 and 3.26 illustrate that the choice of velocity model primarily affects the x-position error, while the y-position error is predominantly affected by the fitting

process. Additionally, it can be observed that the y-position predictions converge to the same value for all models because the estimated lateral displacement of the bi-elementary path, ΔY_{est}, is independent of the chosen model. However, the velocity model does affect the time it takes to reach that convergent y-position.

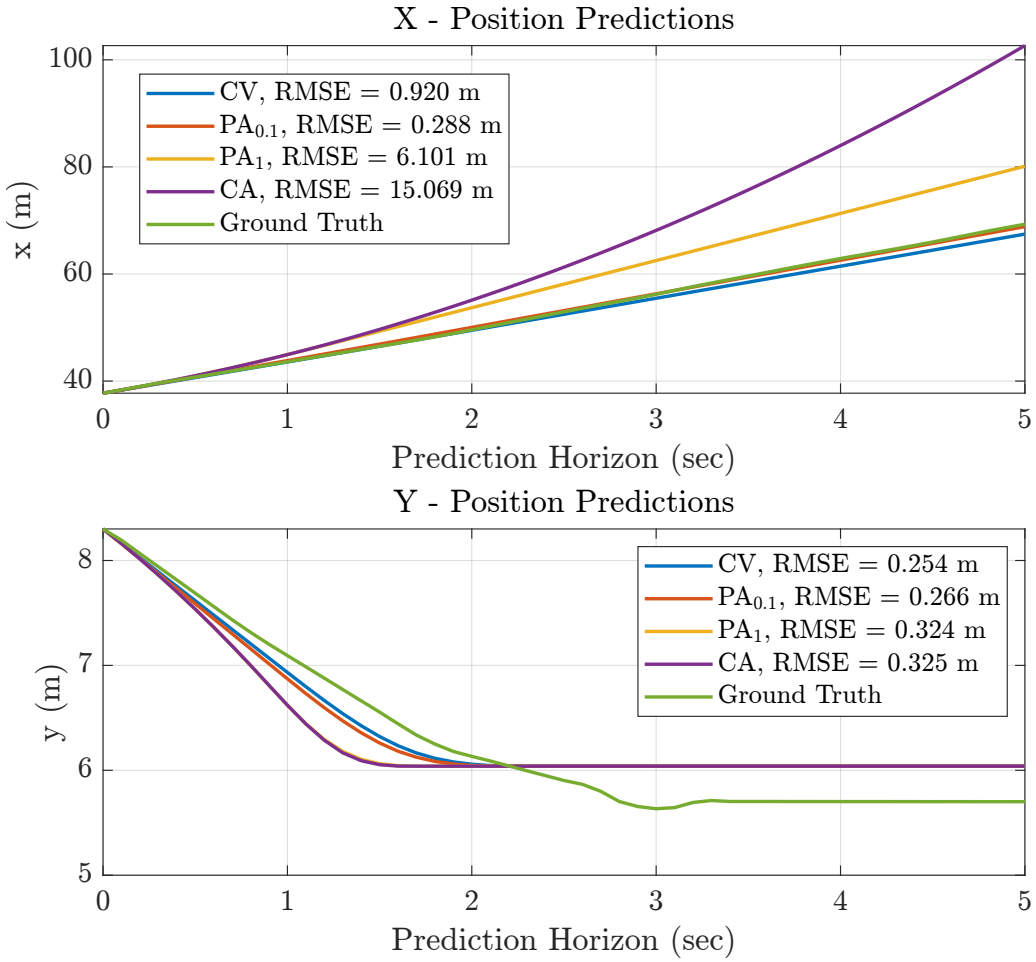

Figure 3.26: Predicted positions and RMS errors for the right lane change trajectory at t=1 s

Based on the two examples examined, using a CA model for long duration ($t_{acc} \geq 1$), appears to result in inaccurate predictions, as the vehicles' velocities tend to remain constant during lane change maneuvers. Hence, employing a CV assumption or using acceleration information only for the initial phase of the maneuver ($PA_{0.1}$) might be a more suitable approach overall. However, this statement cannot be generalized based on only two examples and will be further explored and validated in Section 4.3.3, where the $PA_{0.3}$ model will be shown to yield the most favorable performance.

CHAPTER 4

EVALUATION OF THE METHOD

In this chapter, the objective is to assess the accuracy, robustness, and real-time applicability of the proposed method under a variety of conditions. To achieve this, a comprehensive evaluation approach is adopted, utilizing both synthetically generated data and real-world driving data from the Next Generation Simulation (NGSIM) dataset. The synthetic data allows for assessment of estimation accuracy of bi-elementary path parameter estimation through curve fitting and overall trajectory prediction performance in a controlled environment. The NGSIM dataset, with its diverse real-world driving scenarios, enables us to assess the method's effectiveness under realistic conditions using all the evaluation metrics described in Section 4.1.

The remainder of this chapter is organized as follows: Section 4.1 introduces the evaluation metrics used to assess the performance of the lane change prediction method. Section 4.2 details the evaluation using synthetic trajectories. Section 4.3 focuses on the evaluation with the NGSIM dataset. Finally, Section 4.4 discusses the key findings, highlights the strengths and weaknesses of the method, and outlines potential directions for future research.

4.1 Evaluation Metrics

In order to assess the performance of the proposed lane-change prediction method, various evaluation metrics will be employed. By analyzing these metrics across different datasets, scenarios, and parameter settings, we will gain a comprehensive understanding of the strengths and limitations of the proposed lane-change prediction method. The metrics are:

1. **Parameter Estimation Error:** The accuracy of the bi-elementary path model's parameter estimation is crucial for reliable lane-change prediction. We will quantify this accuracy using the following error metrics:

   - **Absolute Error (AE):** The absolute difference between the estimated parameter value $\hat{p}$ and its true value p, i.e., $(\text{AE} = |\hat{p} - p|)$.
   - **Mean Absolute Error (RE):** The average of the absolute errors of the parameter over all estimations: $(\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|p_i - \hat{p}_i|)$.

     where:

     N : Number of estimations

     p_i : True value of the parameter for the i-th estimation

     $\hat{p}_i$: Estimated value of the parameter for the i-th estimation

2. **Root Mean Square Error (RMSE):** The accuracy of lane-change trajectory predictions is evaluated using the root mean square error (RMSE), which measures the average deviation between the predicted and ground truth trajectory points. We will use two forms of RMSE:

   - **RMSE in Spatial Coordinates:** This measures the Euclidean distance between predicted and ground truth points along the trajectory's arc length, capturing geometric accuracy.

     $$\text{RMSE-S} = \sqrt{\frac{1}{N}\frac{1}{S}\sum_{n=1}^{N}\sum_{s=1}^{S}(\hat{P}(s) - P(s))^2} \tag{4.1}$$

     where:

     S : Length of the prediction distance

     N : Total number of trajectories

     $\hat{P}(s)$: Predicted position at arc-length s

     $P(s)$: Ground truth position at arc-length s

   - **RMSE in Temporal Coordinates:** This measures the deviation between predicted and ground truth positions at specific time instances, accounting for the velocity profile and timing of the lane change.

     $$\text{RMSE-T} = \sqrt{\frac{1}{N}\frac{1}{T}\sum_{n=1}^{N}\sum_{t=1}^{T}(\hat{P}(t) - P(t))^2} \tag{4.2}$$

where:

T : Prediction time

N : Total number of trajectories

$\hat{P}(t)$: Predicted position at time t

$P(t)$: Ground truth position at time t

3. **Max Error:** Max error is the largest absolute deviation observed between predicted and ground truth values within a given metric. It provides insights into the worst-case prediction accuracy and helps identify potential outliers or scenarios where the model struggles.

4. **F1 Score:** To assess the effectiveness of the lane change prediction model in classifying true lane change events while minimizing false alarms, we utilize the F1 score. This metric serves as a balanced measure of the model's precision and recall.

   Precision quantifies the proportion of correctly predicted lane changes out of all positive predictions whereas recall measures the proportion of correctly predicted lane changes out of all actual lane changes:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4.3}$$

   where:

   TP (True Positive) : Number of lane changes predicted correctly

   FP (False Positive) : Number of lane keepings predicted as lane change

   FN (False Negative) : Number of lane changes predicted as lane keeping

   The F1 score is computed as the harmonic mean of precision and recall:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \times 100\% \tag{4.4}$$

   By combining precision and recall, the F1 score offers a comprehensive evaluation of the model's ability to identify true lane changes (recall) while minimizing incorrect predictions (precision). A high F1 score indicates a good balance between precision and recall.

5. **Computation Time:** Computation time is the time taken by the model to process input data and generate lane-change predictions. This metric is crucial for real-time applications where quick decision-making is essential. It is important to note that the computation time reported in this study was measured on a computer equipped with an i7-12700H CPU and 16 GB RAM.

4.2 Evaluation with Synthetic Trajectories

To assess the accuracy and robustness of the proposed lane-change prediction method in a controlled environment, synthetic trajectories were generated using the bi-elementary path model. This allows for the evaluation of the least-squares curve fitting algorithm by examining the accuracy of the estimated bi-elementary path parameters. This approach is necessary because it is not possible to directly assess the parameter estimation performance with real-world datasets due to the unavailability of ground truth parameters. However, with synthetic trajectories, we can directly compare the estimated parameters to the known ground truth values. Along with parameter estimation accuracy, trajectory prediction performance in spatial coordinates is evaluated, while time-based prediction will be presented in detail in the real-dataset evaluation section.

To generate the trajectories, a range of reasonable values was defined for each bi-elementary path parameter $(S, \Delta Y, \lambda, \gamma)$ as shown in Equation (4.5). This range was chosen to encompass a variety of lane change maneuvers with varying characteristics.

$$\begin{gathered} S \in (20, 100) \\ \Delta Y \in (-3.9, -3.4) \cup (3.4, 3.9) \\ \lambda \in (0.2, 0.8) \\ \gamma \in (0.2, 1) \end{gathered} \tag{4.5}$$

For each trajectory, parameter values were randomly sampled from these defined ranges using a uniform distribution, ensuring diversity in the generated maneuvers. The bi-elementary path model was then used to generate 100 trajectories based on the sampled parameters, with 50 representing left lane changes and 50 representing right lane changes, which are illustrated in Figure 4.1 and 4.2. Each trajectory consists of a sequence of position coordinates, expressed as a function of arc-length, repre-

senting the vehicle's movement throughout the maneuver. The true parameter values used for generating each trajectory were recorded as the ground truth for subsequent evaluation.

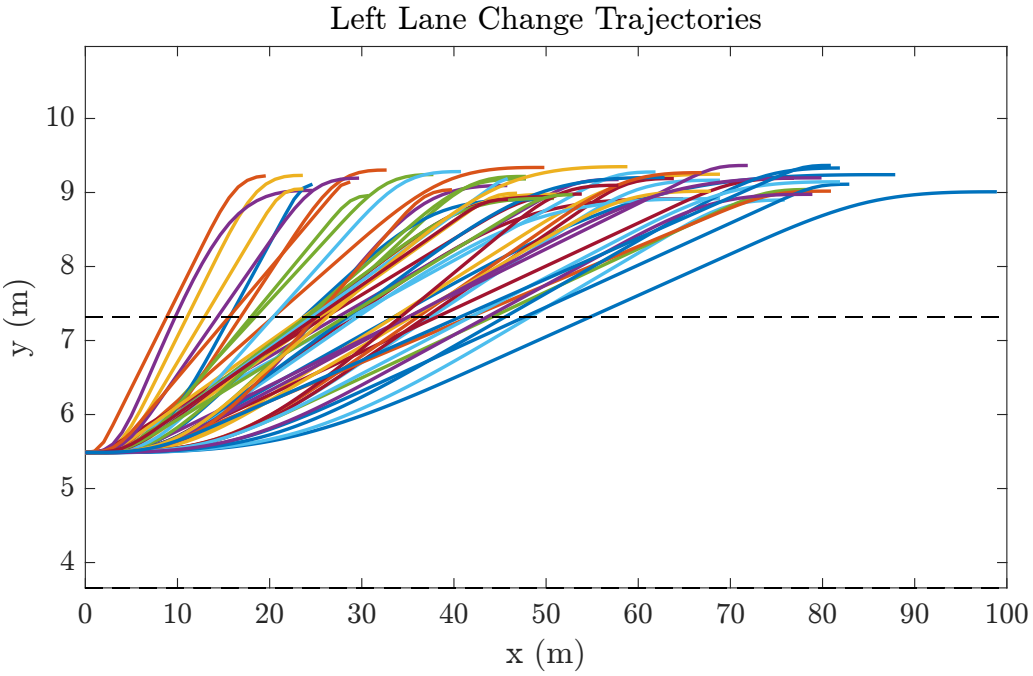

Figure 4.1: Generated left lane trajectories

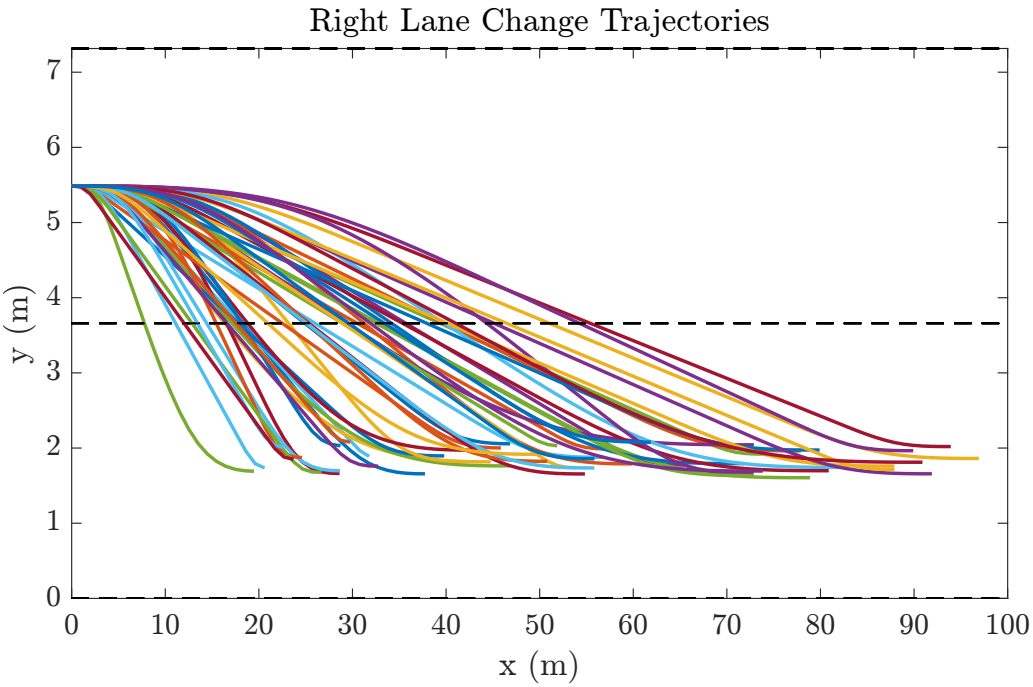

Figure 4.2: Generated right lane trajectories

Additionally, for each trajectory four current position points selected at equal arc-length intervals. For each of these points, the accumulated position data up to that point was used as measurement data for curve fitting. The main purpose of applying predictions at different positions within the same trajectory is to demonstrate that as the amount of accumulated data increases, the bi-elementary path characteristics are better represented, leading to improved curve fitting accuracy.

For these trajectories, a constant velocity of $v = 10$ m/s is assumed. The sampling period is defined as $dt = 0.1$ s, which is consistent with the real dataset. Therefore, the distance between each point in the trajectories, dx, is 1 m, as calculated by $dx = v \cdot dt$.

To simulate real-world sensor inaccuracies and assess the method's robustness to measurement errors, Gaussian noise with varying levels was added to the measurement data. The standard deviation of the noise is denoted as w.

To illustrate the process, a single trajectory is first evaluated, characterized by the following bi-elementary path parameters: $S = 40$ m, $\Delta Y = 3.7$ m, $\lambda = 0.5$, and $\gamma = 0.9$ (denoted as $B(40, 3.7, 0.5, 0.9)$ using bi-elementary path notation).

Given the total arc-length of the bi-elementary path, $S = 40$ m, the lengths of the measurement data corresponding to each current position point are $S_{\text{data}} = 10, 20, 30,$ and 40 m.

First, for the zero measurement noise case, $w = 0$, least-squares curve fitting is applied at each five position. Table 4.1 presents the estimated parameters (S_{est}, ΔY_{est}, λ_{est}, γ_{est}) and their absolute errors (AE).

Table 4.1: Parameter Estimation Errors for a Synthetic Left Lane Change Trajectory with $w = 0$

Position	$S(40\text{ m})$		$\Delta Y(3.7\text{ m})$		$\lambda(0.5)$		$\gamma(0.9)$	
	S_{est}	AE	ΔY_{est}	AE	λ_{est}	AE	γ_{est}	AE
1	20.766	19.234	1.554	2.146	0.818	0.318	1	0.09
2	37.796	2.204	3.807	0.107	0.594	0.094	0.802	0.098
3	39.996	0.004	3.699	0.001	0.5	0	0.9	0
4	40	0	3.7	0	0.5	0	0.9	0

To evaluate the performance of lane change trajectory prediction, the RMSE-S in x and y coordinates and maximum error are calculated within the arc-length range of 0 to S. If the estimated arc-length S_{est} is smaller than the ground truth S, then the predicted trajectory is extended by a straight line segment to compensate the length difference. The resulting errors are presented in Table 4.2, and the predicted trajectories are visualized in Figure 4.3.

Table 4.2: Prediction Errors for a Synthetic Trajectory with $w = 0$

Position	RMSE-S_x (m)	RMSE-S_y (m)	Max Error (m)
1	0.0960	1.1667	2.1521
2	0.0096	0.0655	0.1452
3	0	0.0001	0.0001
4	0	0	0

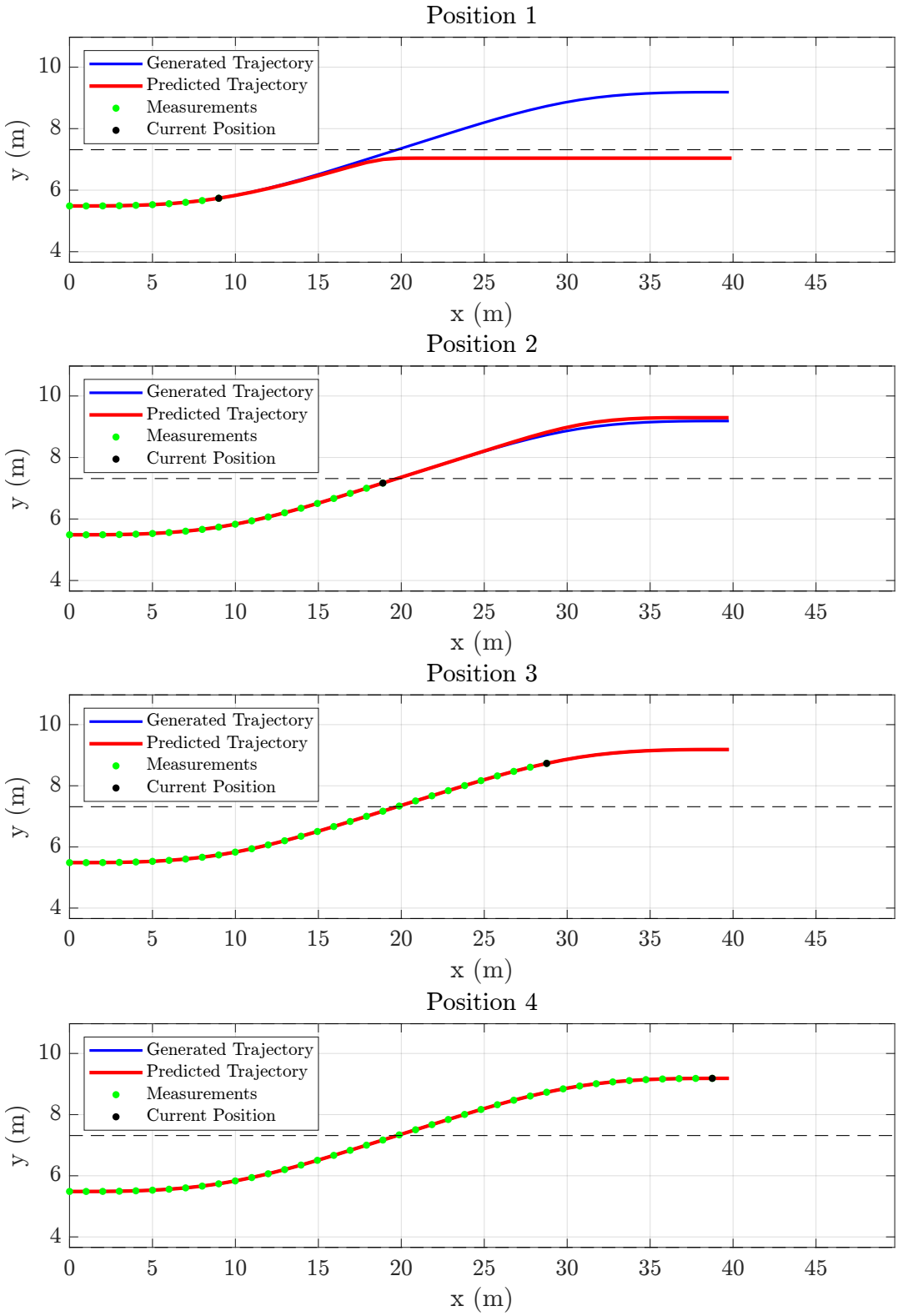

Figure 4.3: Predictions for a Synthetic Trajectory with $w = 0$

For the same trajectory, the measurement noise with $w = 10$ cm is introduced and the same evaluation process in conducted. The parameter estimation results are shown in Table 4.3

Table 4.3: Parameter Estimation Errors for a Synthetic Trajectory with $w = 10$ cm

Position	$S(40\text{ m})$		$\Delta Y(3.7\text{ m})$		$\lambda(0.5)$		$\gamma(0.9)$	
	S_{est}	AE	ΔY_{est}	AE	λ_{est}	AE	γ_{est}	AE
1	20.535	19.465	0.268	3.432	0.372	0.128	0.519	0.381
2	30.584	9.416	3.807	0.108	0.810	0.310	0.945	0.045
3	38.324	1.676	3.508	0.192	0.523	0.023	1.000	0.100
4	39.423	0.577	3.727	0.027	0.530	0.030	1.000	0.100

The trajectory prediction errors are presented in Table 4.4, and the predicted trajectories are illustrated in Figure 4.4.

Table 4.4: Prediction Errors for a Synthetic Trajectory with $w = 10$ cm

Position	RMSE-S_x (m)	RMSE-S_y (m)	Max Error (m)
1	0.1591	2.0837	3.4413
2	0.0722	0.2476	0.6101
3	0.0047	0.0809	0.1925
4	0.0128	0.0538	0.0995

For this single trajectory, Tables 4.1 and 4.3 illustrate that parameter estimation accuracy improves as more input data is accumulated for fitting. At the final position (position 4), where the input data encompasses the entire trajectory, the parameters are perfectly estimated in the absence of measurement noise. Introducing measurement noise increases the errors, but the overall relationships between parameters and prediction accuracy remain consistent. A more detailed analysis will be conducted after evaluating all generated trajectories.

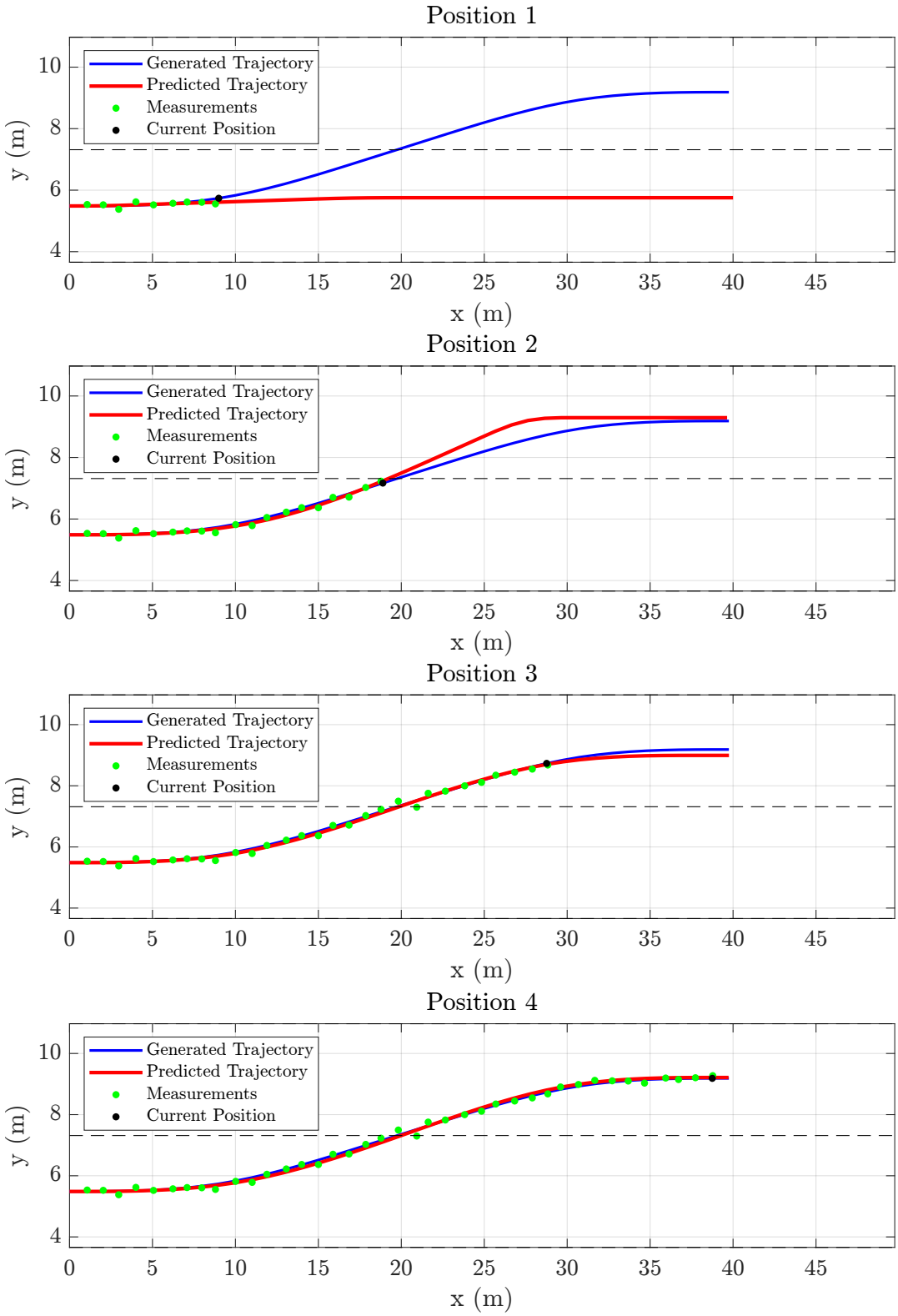

Figure 4.4: Predictions for a Synthetic Trajectory with $w = 10$ cm

To evaluate overall performance, all 100 generated trajectories were assessed. For each trajectory, 50 realizations of measurement noise were applied, resulting in a total of 5000 Monte Carlo simulations. We computed the MAE (Mean Absolute Error) for each parameter and the trajectory prediction errors (RMSE-S) at each of the 4 positions. The measurement noise level is varied to observe the sensitivity of performance to noise. Table 4.5 presents overall results.

Table 4.5: Overall Estimation Errors for Synthetic Trajectories

w (cm)	Position	Mean Absolute Error				RMSE-S		Max Error (m)
		S (m)	ΔY (m)	λ	γ	x (m)	y (m)	
0	1	27.873	1.939	0.211	0.218	0.059	1.082	3.078
	2	5.743	0.144	0.160	0.108	0.005	0.064	0.405
	3	2.849	0.121	0.081	0.056	0.003	0.047	0.364
	4	0.522	0.012	0.016	0.012	0.001	0.006	0.103
5	1	28.867	1.891	0.250	0.279	0.114	1.231	3.739
	2	7.247	0.145	0.194	0.144	0.023	0.1125	1.476
	3	4.083	0.144	0.095	0.075	0.005	0.061	0.482
	4	2.428	0.051	0.063	0.052	0.002	0.017	0.264
10	1	30.818	2.056	0.297	0.304	0.136	1.339	3.818
	2	7.523	0.159	0.198	0.157	0.028	0.166	2.662
	3	4.351	0.143	0.114	0.089	0.008	0.069	0.598
	4	3.347	0.072	0.082	0.075	0.004	0.029	0.319
20	1	29.476	2.005	0.311	0.332	0.161	1.368	3.897
	2	8.294	0.139	0.197	0.190	0.046	0.249	2.393
	3	4.287	0.133	0.117	0.109	0.012	0.086	0.747
	4	4.247	0.102	0.113	0.115	0.007	0.052	0.384
50	1	31.778	1.891	0.356	0.345	0.226	1.446	3.985
	2	10.498	0.189	0.247	0.250	0.083	0.394	3.902
	3	5.416	0.165	0.157	0.172	0.031	0.161	1.796
	4	4.966	0.134	0.151	0.181	0.018	0.116	0.676

As anticipated, in Table 4.5, we observe a clear degradation in the accuracy of parameter estimation as the measurement noise (w) increases. This is reflected in the increasing Mean Absolute Error (MAE) values for all four bi-elementary path parameters ($S, \Delta Y, \lambda, \gamma$). This trend aligns with expectations, as noisier input data inherently makes it harder to accurately extract the underlying parameters of the bi-elementary path model. Similarly, the RMSE-S values for both x and y positions also exhibit an upward trend with increasing noise levels, indicating that the predicted trajectories deviate further from the ground truth as the noise increases, impacting both the longitudinal and lateral accuracy of the predictions. The increase in maximum error further emphasizes the negative effect of noise on the worst-case prediction scenarios.

Analyzing the performance across different positions along the trajectory reveals that the initial phase of the lane-change maneuver (Position 1) is particularly challenging to predict. Both MAE and RMSE-S values are notably higher at this stage compared to later positions. This is likely due to the limited data available at this early stage, making it difficult to distinguish a lane change intention from regular lane-keeping behavior and to reliably estimate the bi-elementary path parameters. As more data points are accumulated for fitting, the model's performance improves significantly at Positions 2 and 3, as justified by the decrease in both MAE and RMSE-S. At Position 4, where the entire trajectory is captured, the errors are generally the lowest, showing the model's ability to accurately predict the final stages of the lane-change maneuver.

Examining specific parameter estimation, we observe that the MAE for S (total arc length) is considerably higher than for the other parameters, particularly at Position 1. This is consistent with the fact that S has a wider range of possible values and thus greater uncertainty in its estimation, especially with limited initial data. In contrast, the relatively low MAE for ΔY can be attributed to the tighter bounds used in the estimation process, derived from the known lane width and target lane information. This constraint results in more accurate estimates for ΔY, which directly translates to better predictions of the lateral movement during the lane change. The consistent relationship between the accurate estimation of λ and γ and the improved RMSE-S performance emphasizes the importance of capturing the shape of the bi-elementary path for achieving precise predictions.

The lower RMSE-S observed in the x-position compared to the y-position aligns with the predominantly straight nature of bi-elementary paths, where lateral displacements are relatively small compared to longitudinal movements. Since RMSE-S is computed using points of equal arc-length, which is primarily influenced by longitudinal movement, it inherently serves as a good approximation of the x-coordinate, leading to smaller errors in x-position errors. In contrast, the y-position errors are more directly linked to the accuracy of estimating the lateral displacement parameter, ΔY.

4.3 Evaluation with NGSIM Dataset

This section evaluates the lane-change prediction method using real-world driving data from the NGSIM dataset, collected from various highway segments, including the US-101 and I-80 highways. The NGSIM dataset's diverse real-world driving scenarios provide a valuable benchmark for assessing the performance of our method under realistic conditions, capturing a wide range of vehicle behaviors, traffic densities, and environmental factors.

To prepare the NGSIM data for evaluation, we extracted relevant lane-change trajectories based on instances where the vehicle's lane ID changes. Each trajectory has a duration of 20 seconds, which has been observed to be sufficient to capture the entire lane change maneuver in the dataset. Additionally, we filtered out trajectories with missing or inconsistent data points to maintain data integrity. The total number of extracted trajectories is 500. Some of these are illustrated in Figure 4.5.

For the initial evaluations, we will utilize a constant velocity assumption to transform the predicted coordinates into temporal coordinates as explained in Section 3.2.2. A comprehensive comparison of different velocity models will be presented later in this section.

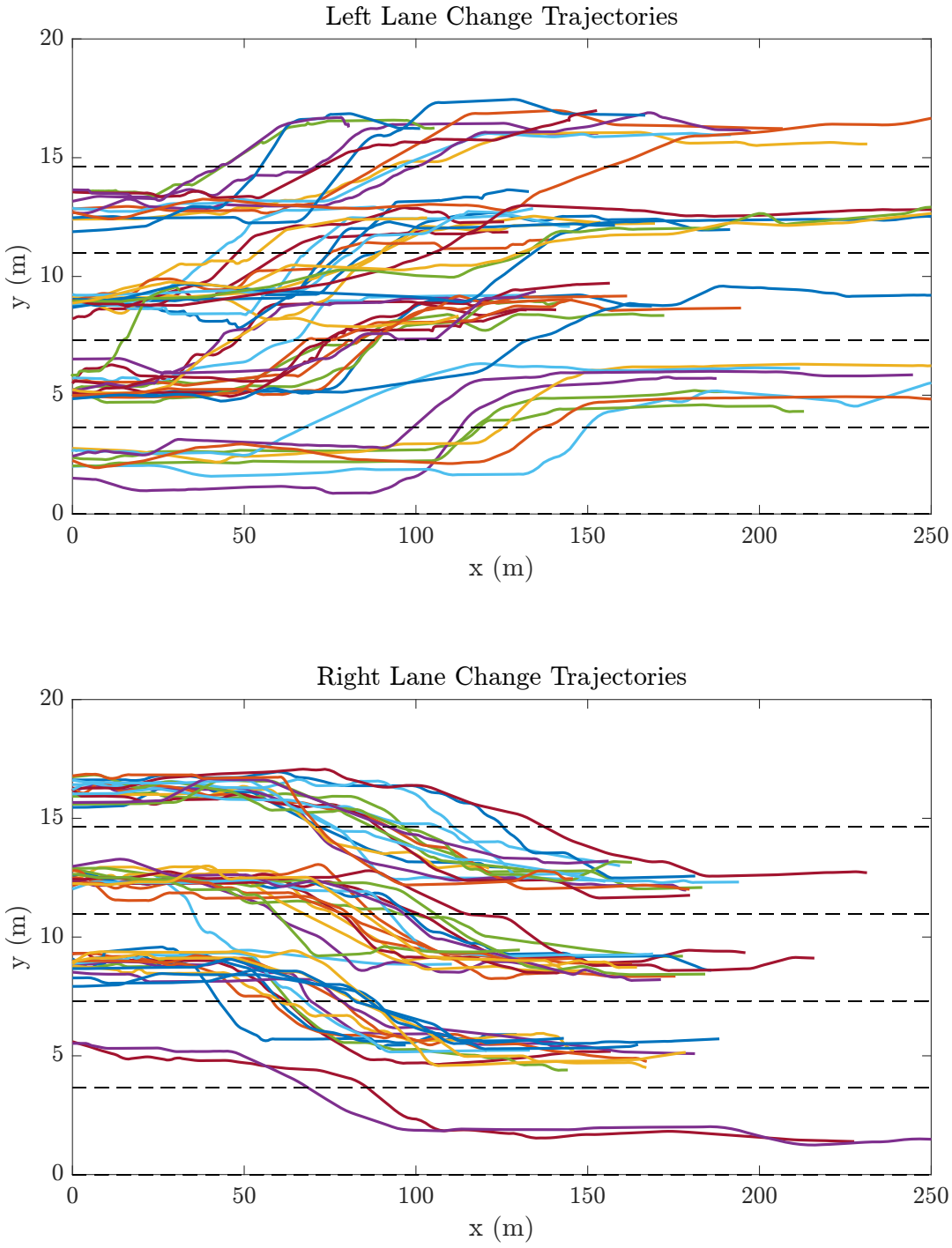

Figure 4.5: NGSIM Dataset Trajectories

4.3.1 Parameter Tuning

The performance of the proposed lane-change prediction method is influenced by several design parameters, including the heading threshold (Φ_{th}), lane offset threshold (LO_{th}), and lateral displacement ($m_{\Delta Y}$), as described in Section 3.2.1. We employ a parameter tuning process to optimize these values and achieve the best possible prediction accuracy on the NGSIM dataset.

The descriptions and impact of these parameters are:

- **Heading Threshold (Φ_{th}):** This threshold determines when a vehicle is considered to be deviating from a straight path, indicating the start of a lane change. A lower Φ_{th} may lead to earlier detection of lane changes, potentially increasing false positives, while a higher Φ_{th} may delay the detection of lane changes, potentially impacting the accuracy of early predictions.
- **Lane Offset Threshold (LO_{th}):** This threshold helps determine the target lane for a potential lane change based on the vehicle's proximity to the lane boundaries. An incorrectly set LO_{th} may result in the algorithm misinterpreting the vehicle's intended lane, leading to inaccurate estimation of the lateral displacement (ΔY) and subsequent trajectory predictions.
- **Lateral Displacement Margin ($m_{\Delta Y}$):** This parameter provides a safety margin added to the calculated distance to the target lane when setting the upper bound for the lateral displacement (ΔY) during the lane change, accounting for potential variations in the final lane position.

These design parameters are varied within a reasonable range. Table 4.6, 4.7 and 4.8 present the results of parameter tuning for the lane-change prediction method, focusing on the parameters: heading threshold , lane offset threshold , and lateral displacement margin, respectively. The performance is assessed for 5-second prediction horizon using two metrics: RMSE-T (Root Mean Square Error in Temporal Coordinates) for x and y positions, and F1 score for lane change detection accuracy.

Table 4.6: Parameter comparison for heading threshold Φ_{th}

Φ_{th} (°)	RMSE-T		F1 Score (%)
	x (m)	y (m)	
0.5	3.641	0.842	79.067
1	3.641	**0.837**	**80.915**
1.5	3.641	**0.837**	79.962
2	3.641	0.859	79.754
2.5	3.641	0.872	78.613
3	3.641	0.886	75.075
5	3.641	0.978	61.653

From Table 4.6, it can be deduced that the RMSE-T in the x-direction remains consistently at 3.641 m across all tested Φ_{th} values. This indicates that the heading threshold has negligible impact on the longitudinal prediction accuracy. The primary factor influencing longitudinal prediction in this case is likely the velocity model, which is kept constant in this analysis.

The RMSE-T in the y-direction (lateral error) shows a slight increase as Φ_{th} increases beyond 1°. This suggests that a higher heading threshold might lead to a delayed detection of lane changes, causing the model to miss the initial lateral movements and resulting in slightly larger lateral prediction errors. The optimal value for minimizing lateral error appears to be around $\Phi_{th} = 1°$.

The F1 score, which balances precision and recall in lane change detection, peaks at $\Phi_{th} = 1°$ and then steadily declines with increasing Φ_{th}. This trend highlights the trade-off between early detection and false positives. A low Φ_{th} might trigger lane change predictions prematurely, leading to false positives and reducing precision. On the other hand, a high Φ_{th} delays detection, potentially missing true lane changes and impacting recall. The optimal Φ_{th} of 1° appears to strike the best balance, maximizing the F1 score.

Table 4.7: Parameter comparison for lane offset threshold LO_{th}

LO_{th} (m)	RMSE-T		F1 Score (%)
	x (m)	y (m)	
0	3.641	0.894	78.291
0.5	3.641	0.853	80.881
0.75	3.641	0.841	**81.021**
1	3.641	**0.837**	80.915
1.25	3.641	0.844	79.421
1.5	3.641	0.863	71.447
2	3.641	0.977	32.843

In Table 4.7, similar to the previous table, the RMSE-T in the x-direction remains constant across all LO_{th} values, further emphasizing the minimal influence of this parameter on longitudinal prediction accuracy.

The RMSE-T in the y-direction shows a U-shaped pattern, with the lowest value at LO_{th} = 1 m. This suggests that an appropriate lane offset threshold is crucial for accurate lateral prediction. If LO_{th} is too low or too high, the model might misinterpret the target lane, leading to incorrect estimation and increased lateral errors.

The F1 score reaches its peak at LO_{th} = 0.75 m and then decreases as LO_{th} deviates from this value. This implies the importance of LO_{th} in correctly identifying the target lane for accurate lane change detection. A misidentified target lane would result in both false positives and false negatives, negatively affecting the F1 score.

Table 4.8: Parameter comparison for lateral displacement margin $m_{\Delta Y}$

$m_{\Delta Y}$ (m)	RMSE-T		F1 Score (%)
	x (m)	y (m)	
0.1	3.641	0.844	80.322
0.2	3.641	**0.837**	80.915
0.3	3.641	0.841	80.785
0.4	3.641	0.850	80.844
0.5	3.641	0.861	80.807
1	3.641	0.938	**81.092**
2	3.641	1.081	80.279

In Table 4.8, again, the RMSE-T in the x-direction shows no significant change with varying $m_{\Delta Y}$, confirming its limited role in longitudinal prediction.

The lateral RMSE-T increases steadily with increasing $m_{\Delta Y}$. This suggests that a larger margin for lateral displacement might lead to overestimation of ΔY, causing the model to predict wider lane changes than actually occur, thereby increasing the lateral error. The optimal $m_{\Delta Y}$ appears to be around 0.2 m, achieving the lowest lateral RMSE-T.

The F1 score exhibits a more complex behavior for this parameter. While the margin does influence the upper bound on the estimated final lateral position, it's important to recognize that the the occurrence of a lane change and the identification of the target lane has already been made based on the heading and lane offset thresholds. Consequently, the F1 score, which primarily reflects the accuracy of lane change detection, remains relatively unaffected by variations in the lateral displacement margin.

Based on the parameter tuning results, the selected values for the parameters are:

$$\begin{aligned} \Phi_{th} &= 1^\circ \\ \text{LO}_{th} &= 1 \text{ m} \\ \text{m}_{\Delta Y} &= 0.2 \text{ m} \end{aligned} \tag{4.6}$$

4.3.2 Comparison with Straight-Line Model

To benchmark the performance of the proposed lane change prediction method, a comparison is made against a simpler straight-line model. This baseline model assumes that the vehicle maintains its current velocity and heading angle throughout the prediction horizon, resulting in a linear trajectory prediction. The effectiveness of both methods will be evaluated using various metrics across different prediction horizons.

These metrics are RMSE-T (x and y), F1 score, max error and computation time. We will compare the performance of both models at various prediction horizons, ranging from 1 second to 5 seconds. Additionally, we will provide visualizations comparing the predicted trajectories from both methods to illustrate the performance differences.

To gain preliminary insights, we will first examine a specific right lane change trajectory at different time instants, where the starting point of the trajectory is defined as time t=0. Figure 4.6 to 4.9 illustrate the bi-elementary path and straight-line predictions for a 5-second horizon, along with the ground truth trajectory. The RMSE-T results for x and y coordinates are also provided on the figures. In the figures each interval between dots on the trajectories corresponds to 1 second.

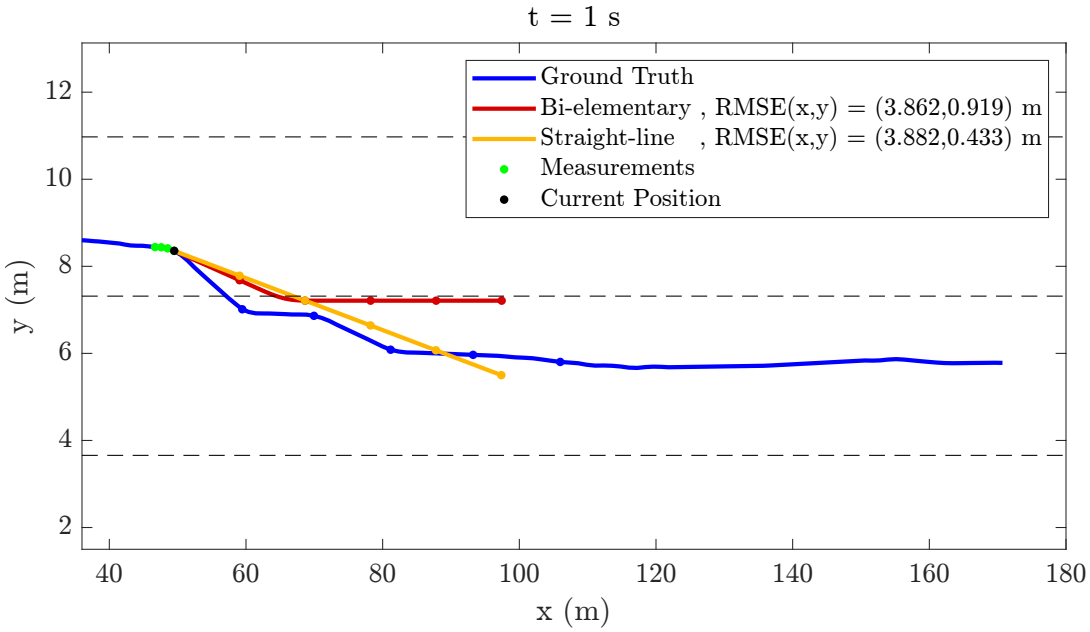

Figure 4.6: Predictions for a right lane change trajectory at t = 1s

As shown in Figure 4.6, at t = 1 s, the lane change maneuver has not begun yet, and the available measurements are insufficient to characterize it. Consequently, the bi-elementary path prediction is inaccurate, leading to a high lateral error. In this specific instance, the straight-line prediction outperforms the bi-elementary path model for the 5-second horizon.

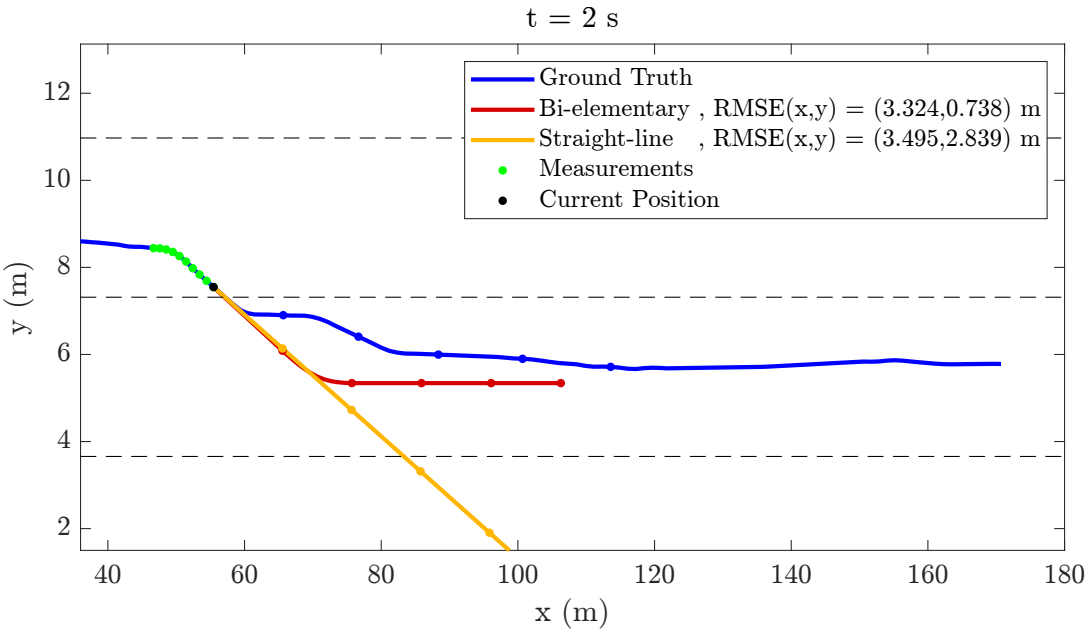

Figure 4.7: Predictions for a right lane change trajectory at t = 2s

In Figure 4.7, at t = 2 s, the increased measurement data allows for a more accurate bi-elementary path prediction, estimating the final lateral position close to the right lane center. However, the ground truth trajectory exhibits some mid-maneuver behavior that a single bi-elementary path cannot fully capture, leading to a larger estimation error, particularly prediction for 2-second later (t_{pre} = 2 s). In contrast, the straight-line model, with its constant heading assumption, results in a significantly larger lateral error compared to the bi-elementary path model after t_{pre} = 2 s.

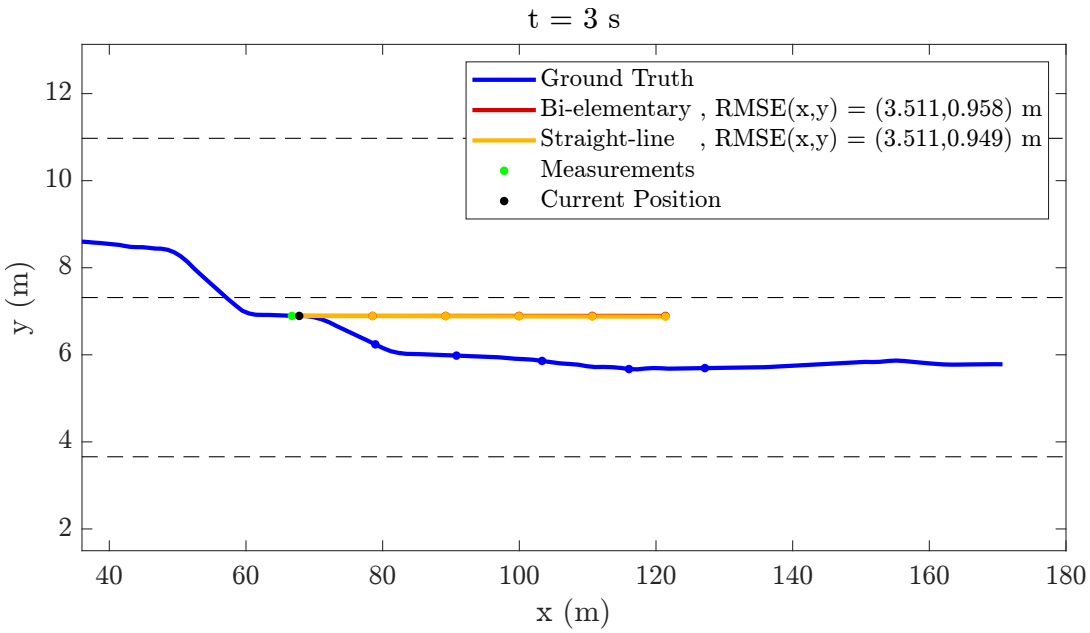

Figure 4.8: Predictions for a right lane change trajectory at t = 3s

In Figure 4.8, at t = 3 s, the vehicle straightens its path in close proximity to the lane markings. As the heading angle falls below the defined threshold at this point, the prediction model is reset. Consequently, the bi-elementary path model generates the same prediction as the straight-line model.

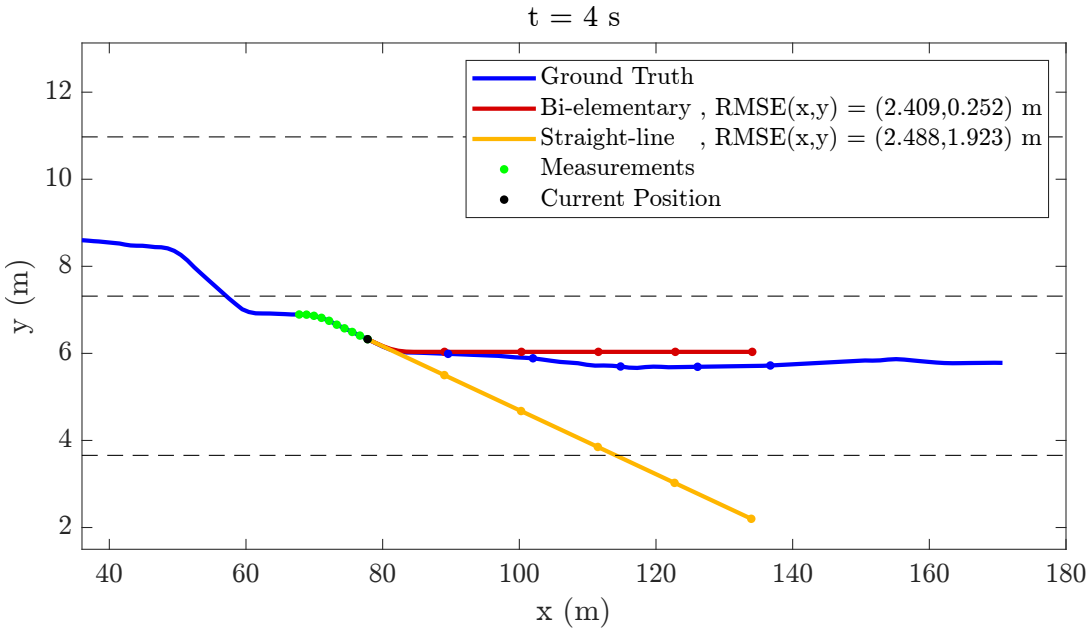

Figure 4.9: Predictions for a right lane change trajectory at t = 4s

In Figure 4.9, at t = 4 s, the measurements once again capture a portion of the lane-changing behavior. In this instance, as the vehicle's proximity to the left lane marking at the reset point is less than the lane offset threshold, the target right lane is identified as the center of the vehicle's current lane, which aligns with the actual scenario. The bi-elementary path model demonstrates superior performance compared to the straight-line prediction in terms of y-position accuracy for this case.

It can be observed that the x-coordinate error is nearly the same for both methods in all cases. This is because the x-error is primarily caused by the velocity assumption. Since a constant velocity model is employed for both methods, this behavior is expected.

Following the illustrative example, we now conduct a comprehensive evaluation of both the bi-elementary path model and the straight-line model using the entire trajectory set. This will involve calculating the performance metrics (RMSE-T, F1 score, max error, and computation time) for each model across different prediction horizons.

Table 4.9 summarizes the performance of the proposed bi-elementary path model and the baseline straight-line model in predicting lane change trajectories on the NGSIM dataset.

Table 4.9: Performance comparison of the bi-elementary path and straight-line models for lane change trajectory prediction on the NGSIM dataset across different prediction horizons

Prediction Horizon (s)	Method	RMSE-T		Max Error		F1 Score (%)	Computation Time (ms)
		x (m)	y (m)	x (m)	y (m)		
1	Bi-elementary path	0.363	0.277	13.051	2.946	80.827	9
1	Straight-line	0.364	0.256	13.117	2.946	85.972	0.1
2	Bi-elementary path	0.942	0.511	23.247	3.684	78.841	9
2	Straight-line	0.947	0.580	23.368	6.081	84.090	0.1
3	Bi-elementary path	1.684	0.668	32.679	4.221	80.554	9
3	Straight-line	1.691	0.925	32.844	9.263	74.957	0.1
4	Bi-elementary path	2.588	0.769	43.231	4.142	80.822	9
4	Straight-line	2.599	1.292	43.428	12.461	63.889	0.1
5	Bi-elementary Path	3.641	0.837	53.848	5.372	80.915	9
5	Straight-line	3.654	1.681	54.068	15.668	53.040	0.1

The results is Table 4.9 can be analyzed for each performance metric as follows:

- **RMSE-T (x):** The longitudinal (x) prediction error (RMSE-T) is nearly identical for both methods across all prediction horizons. This is expected as both methods are using the same constant velocity assumption for predicting longitudinal movement.

- **RMSE-T (y):** The lateral (y) prediction error shows a clear advantage for the bi-elementary path model, especially as the prediction horizon increases. This highlights the bi-elementary path model's ability to capture the curved nature of lane changes, leading to more accurate lateral predictions compared to the straight-line assumption. Figure 4.10, illustrates the RMSE-T (x,y) against prediction horizon for both methods.

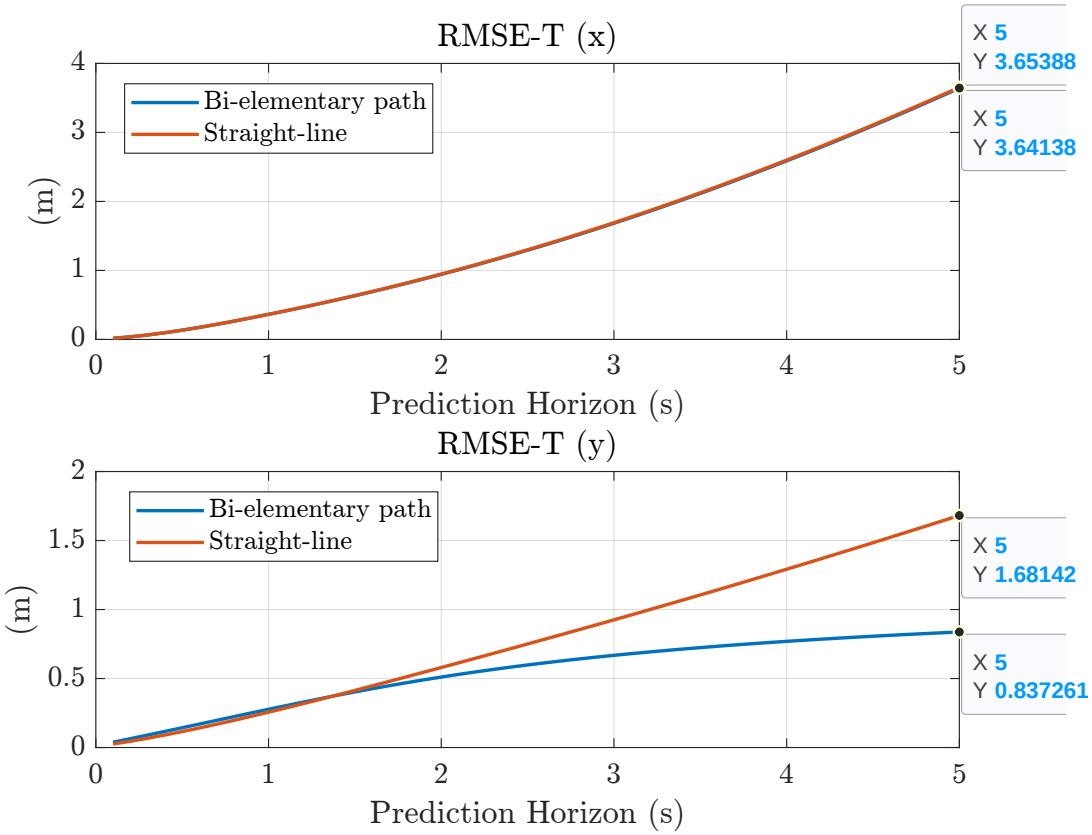

Figure 4.10: RMSE-T plots for bi-elementary path and straight-line models against prediction horizon

- **Max Error:** The maximum error, both in x and y, is consistently lower for the bi-elementary path model, showcasing its ability to handle more challenging scenarios and avoid large prediction deviations.
- **F1 Score:** The bi-elementary path model demonstrates a superior F1 score for longer prediction horizons. This indicates its better capability in balancing precision and recall for lane change detection compared to the straight-line model, which tends to produce more false positives or false negatives as the prediction horizon extends.
- **Computation Time:** While the straight-line model is significantly faster due to its simplicity, the bi-elementary Path model's computation time is also relatively low, suggesting its feasibility for real-time applications.

Further analysis based on prediction horizons reveals the following:

- **1-second Horizon:** The straight-line model performs slightly better in terms of lateral error and F1 score at this short horizon. This might be because, in the very early stages of a lane change, the vehicle's trajectory is still close to a straight line, and the bi-elementary path model might be over-fitting to the limited data.

- **2-5 second Horizons:** The bi-elementary path model consistently outperforms the straight-line model in lateral error, max error, and F1 score as the prediction horizon increases. This emphasizes its effectiveness in capturing the non-linear dynamics of lane changes over longer durations. The straight-line model's performance decreases notably, especially in lateral prediction, due to its inability to account for the vehicle's curved trajectory.

Overall, the comparative analysis with the straight-line model shows the superiority of the bi-elementary path model for lane change trajectory prediction, especially for longer prediction horizons. Its ability to accurately capture the curved dynamics of lane changes translates to improved performance in lateral error, maximum error, and F1 score. While the straight-line model offers computational advantages, the bi-elementary path model has a reasonable computation time, with 9 ms, making it a promising candidate for real-time applications in autonomous driving systems.

4.3.3 Evaluation of Velocity Models

As detailed in Section 3.2.2, the transformation of predicted bi-elementary path coordinates from spatial to temporal coordinates requires modeling the vehicle's velocity within the prediction horizon. This subsection aims to compare the performance of various velocity models, namely Constant Velocity (CV), Constant Acceleration (CA), and Partial Acceleration (PA), to determine the most suitable model for accurate lane change trajectory prediction.

We utilize the same NGSIM dataset trajectories as in previous evaluations. The lane change prediction method is applied with each velocity model (CV, CA, and PA with different t_{acc} values), and the resulting predictions are evaluated using RMSE-T and max error, across different prediction horizons (1, 3 and 5 seconds). Table 4.10

presents the results.

Table 4.10: Performance comparison of different velocity models for lane change trajectory prediction on the NGSIM dataset

Prediction Horizon (s)	Velocity Model	RMSE-T		Max Error	
		x (m)	y (m)	x (m)	y (m)
1	CV	0.363	0.277	7.010	2.946
1	$PA_{0.1}$	0.328	0.277	6.686	2.946
1	$PA_{0.2}$	**0.314**	0.278	6.396	2.946
1	$PA_{0.3}$	0.315	0.278	6.140	2.946
1	$PA_{0.4}$	0.324	0.278	5.918	2.946
1	$PA_{0.5}$	0.337	0.278	5.730	2.946
1	PA_{1}	0.369	0.279	**5.303**	2.946
1	CA	0.369	0.279	5.303	2.946
3	CV	1.684	0.668	21.817	4.222
3	$PA_{0.1}$	1.613	0.668	20.810	4.222
3	$PA_{0.2}$	**1.581**	0.668	20.005	4.222
3	$PA_{0.3}$	1.585	0.668	20.005	4.222
3	$PA_{0.4}$	1.619	0.668	20.005	4.222
3	$PA_{0.5}$	1.678	0.668	20.005	4.222
3	PA_{1}	2.126	0.668	20.005	4.222
3	CA	2.954	0.669	26.455	4.222
5	CV	3.641	0.837	35.108	5.373
5	$PA_{0.1}$	3.539	0.837	**34.749**	5.373
5	$PA_{0.2}$	3.487	0.837	34.811	5.373
5	$PA_{0.3}$	**3.483**	**0.836**	34.872	5.373
5	$PA_{0.4}$	3.523	**0.836**	34.932	5.373
5	$PA_{0.5}$	3.601	**0.836**	34.990	5.373
5	PA_{1}	4.361	0.837	40.815	5.373
5	CA	7.819	0.839	67.914	5.373

Focusing on the RMSE-T, for longitudinal (x) error, the Constant Acceleration (CA) model consistently exhibits the highest x-error, especially at longer horizons, suggesting that its assumption of constant acceleration doesn't adequately capture the behaviour of real-world lane-change dynamics. In contrast, the Partial Acceleration (PA) models, particularly those with shorter t_{acc} values (0.1s to 0.3s), tend to outperform both CV and CA models, especially for longer prediction horizons. This indicates that incorporating some degree of acceleration modeling, while limiting its duration, can enhance longitudinal prediction accuracy.

The lateral (y) prediction error remains relatively stable across different velocity models and prediction horizons. This indicates that the lateral accuracy is primarily determined by the bi-elementary path curve fitting itself, rather than the specific velocity model employed. However, a slight increase in y-error is noticeable for the CA model at longer horizons, likely due to the effect of inaccurate longitudinal predictions on the overall trajectory estimation. The RMSE-T results for x and y positions against prediction horizon up to 5 seconds are also illustrated in Figure 4.11.

The maximum error results further emphasizes the limitations of the CA model. It consistently exhibits significantly higher maximum x-errors, especially at longer prediction horizons, revealing its susceptibility to large prediction deviations in challenging scenarios. The max y-error, similar to the RMSE-T (y), remains relatively consistent across different velocity models.

Based on this comprehensive evaluation, the Partial Acceleration (PA) model with an acceleration duration 0.3 s appears to be the most suitable choice for lane change trajectory prediction within the context of this method and using this dataset. It consistently outperforms both the CV and CA models in longitudinal prediction accuracy while maintaining comparable lateral accuracy. The PA model's ability to account for the initial acceleration or deceleration phase, without overemphasizing its impact, aligns well with the observed dynamics of real-world lane change maneuvers in the NGSIM dataset.

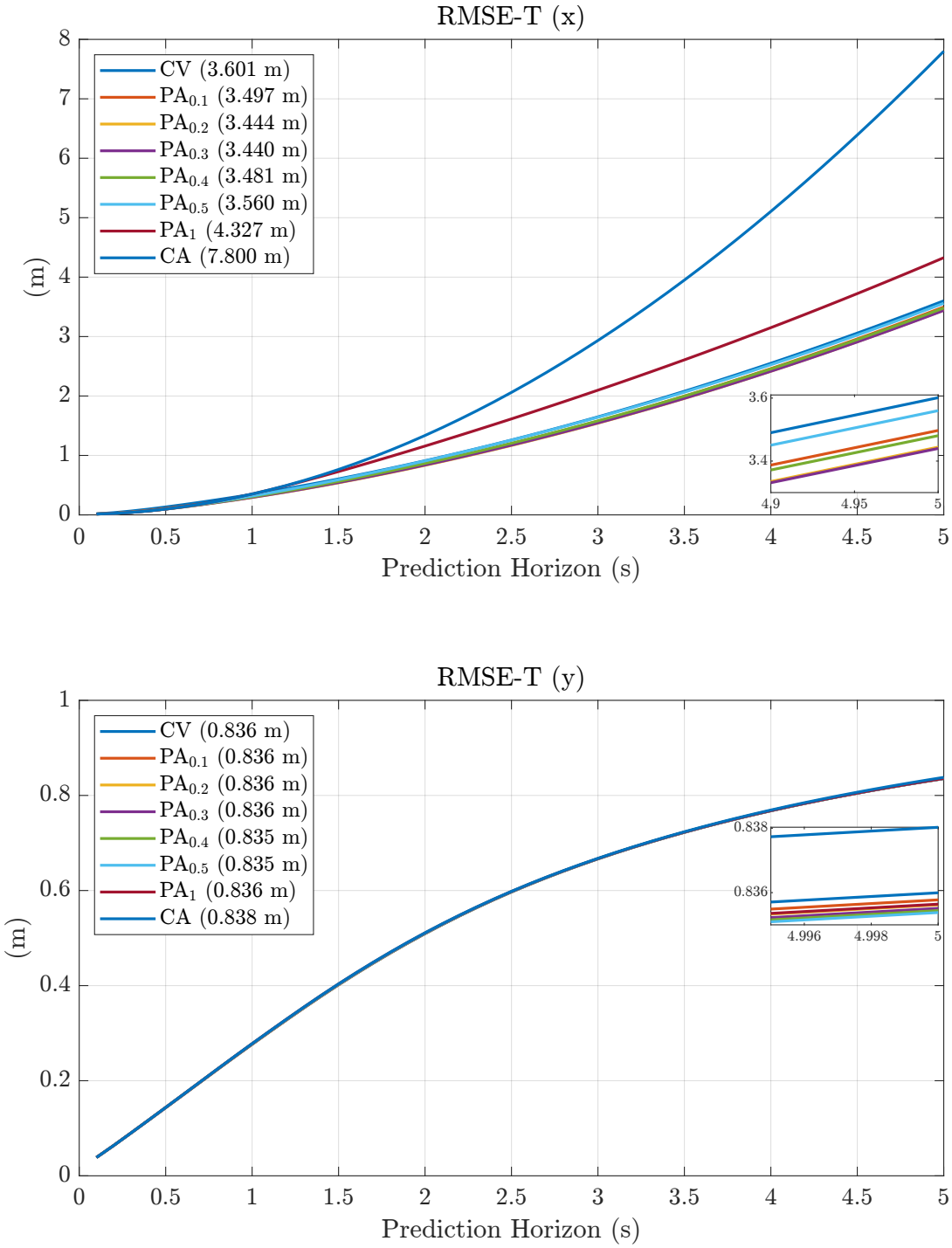

Figure 4.11: RMSE-T plots for different velocity models

4.3.4 Comparison with the Existing Studies

In this section, we compare the performance of our proposed method against several state-of-the-art trajectory prediction approaches, including:

- **IMM-KF:** Employs an Interacting Multiple Model Kalman Filter with intention-based models for vehicle motion prediction, ensuring non-colliding predictions through an optimization-based projection [24].

- **CS-LSTM:** Utilizes an LSTM encoder-decoder model with convolutional social pooling to learn and predict multi-modal trajectories based on maneuver classes [39].

- **TS-GAN:** Leverages a multi-vehicle collaborative learning model with spatio-temporal tensor fusion and a generative adversarial network to capture the multi-modal nature of vehicle behavior and interactions within the traffic context [46].

- **GISNet:** Employs a graph-based information sharing network to facilitate information exchange between the target vehicle and its neighbors, while also encoding historical trajectory data [44].

- **GRIP:** Models interactions between nearby objects using a graph representation, utilizing graph convolutional blocks for feature extraction and an encoder-decoder LSTM model for trajectory prediction [45].

Table 4.11 presents a comparative analysis, presenting the RMSE (Root Mean Square Error of the L2 distance) for each method at prediction horizons from 1s to 5s. Additionally, the table includes computation times for our proposed method and three reference methods. All methods were stated to be evaluated on the NGSIM dataset, ensuring a fair comparison.

Table 4.11: Comparison with the existing trajectory prediction methods on the NGSIM dataset

Method	RMSE (m)					Computation Time (ms)
	1s	2s	3s	4s	5s	
IMM-KF	0.58	1.36	2.28	3.37	4.55	3
CS-LSTM	0.62	1.29	2.13	3.20	4.52	35
TS-GAN	0.60	1.24	1.95	2.78	3.72	-
GISNet	0.33	0.83	1.42	2.14	3.23	-
GRIP	0.37	0.86	1.45	2.21	3.16	6
Proposed	0.42	1.01	1.72	2.57	3.58	9

As evident from the table, the proposed method consistently demonstrates competitive performance in terms of RMSE across all prediction horizons. While methods like GISNet and GRIP exhibit slightly lower RMSE values, it is crucial to consider the computational efficiency aspect.

Most studies in the literature do not explicitly report computation time for their prediction methods. However, one study indicates that predicting trajectories for 1000 objects takes around 35 s and 6 s for two different LSTM-based methods, namely CS-LSTM and GRIP, using an i7 CPU and an NVIDIA Titan GPU [45]. While inferring per-object computation time by dividing by 1000 may not be entirely accurate due to potential parallel processing and optimizations, we present these values (35 ms and 6 ms) in the table for reference.

In contrast, our proposed method achieves an average computation time of 9 ms per trajectory on a standard CPU, demonstrating comparable efficiency to GRIP even without GPU acceleration. Also, the simplicity of our method, relying on curve fitting to a 4-parameter function without the need for complex training, GPU acceleration, further emphasizes its computational advantage. This streamlined approach, lacking computationally intensive components like deep neural networks, enables rapid processing, crucial for real-time autonomous vehicle decision-making.

In summary, the proposed method not only achieves competitive accuracy but also demonstrates promising computational efficiency, comparable to state-of-the-art learning-based methods like GRIP. Its ability to produce accurate lane change predictions rapidly, even on standard hardware, makes it a strong candidate for integration into real-world autonomous driving systems, where accuracy and real-time performance are important.

4.4 Discussion

This chapter systematically evaluated the performance of our proposed lane-change trajectory prediction method. We employed both synthetically generated trajectories and real-world driving data from the NGSIM dataset to assess accuracy, robustness, and real-time applicability.

The use of synthetic trajectories allowed us to directly examine the accuracy of bi-elementary path parameter estimation, revealing a clear improvement as more input data became available. The introduction of measurement noise highlighted the method's robustness, although accuracy understandably decreased with increasing noise levels.

Evaluation on the NGSIM dataset further validated the method's effectiveness in real-world scenarios. Parameter tuning demonstrated the influence of heading threshold, lane offset threshold, and lateral displacement margin on prediction accuracy and lane change detection. The comparison with a straight-line model underscored the superior performance of the bi-elementary path model, especially for longer prediction horizons, due to its ability to capture the curved dynamics of lane changes.

The assessment of various velocity models revealed that the partial acceleration (PA) model with a short acceleration phase is most effective. This is likely because it captures the initial acceleration or deceleration that often occurs at the beginning of a lane change, while still recognizing that the majority of the maneuver is typically performed at a relatively constant velocity.

Finally, a comparison with state-of-the-art methods demonstrated the proposed method's

competitive accuracy and computational efficiency. It achieved performance comparable to GRIP while requiring only standard CPU processing, highlighting its potential for real-time implementation in autonomous vehicles. The method's inherent simplicity, resulting from its reliance on curve fitting to a 4-parameter function, further strengthens its computational advantage.

CHAPTER 5

CONCLUSION

This thesis has presented a novel model-based approach for lane change trajectory prediction, using the strengths of the bi-elementary path model and data-driven optimization through least-squares curve fitting. The core of our method lies in the representation of lane changes using the geometrically intuitive bi-elementary path model. This model, composed of combined clothoid segments efficiently approximated by arc-splines, provides a flexible and compact framework that captures the smooth and continuous changes in curvature characteristic of lane change maneuvers. To accurately predict the diverse lane change scenarios encountered in real-world driving, we employ a least-squares curve fitting technique to estimate the parameters of the bi-elementary path model. By relying on a model with a minimal parameter set and optimized curve fitting, our method achieves a balance of improved accuracy and computational efficiency.

Through comprehensive evaluation on both synthetically generated trajectories and real-world driving data from the NGSIM dataset, we have demonstrated the effectiveness and robustness of our method. The use of synthetic data allowed us to directly examine the accuracy of bi-elementary path parameter estimation, revealing how the model's performance improves as more input data becomes available, even in the presence of measurement noise. Evaluation on the NGSIM dataset further validated the method's real-world applicability, highlighting its adaptability to the diverse driving behaviors encountered on highways. Our method outperformed a baseline straight-line model, particularly in terms of lateral prediction accuracy. Furthermore, comparisons with state-of-the-art methods revealed competitive performance of our method. Our exploration of various velocity models also emphasizes

the importance of considering initial acceleration/deceleration phases for achieving precise time-based predictions.

The findings of this thesis have important implications for the development of safer and more efficient autonomous driving systems. By adapting the bi-elementary path model from its original context in path control to the domain of lane change prediction, we have demonstrated its effectiveness and potential for wider applications. Our method's interpretability, based on its reliance on a well-defined mathematical model, and computational efficiency, made possible by its compact representation, sets it apart from many existing approaches. The ability to generate accurate and real-time predictions, even on standard hardware without the need for extensive training data or specialized computational resources, positions our method as a convincing solution for real-time deployment in autonomous vehicles. This approach offers a valuable framework for developing future trajectory prediction methods that balance accuracy, interpretability, and computational efficiency for autonomous systems.

While our method has demonstrated promising results, there are several directions for future research. Firstly, the integration of our method with efficient maneuver-based methods, could lead to improved detection of lane change initiation points, further enhancing prediction accuracy. Secondly, exploring more accurate and complex velocity models could improve the transformation from spatial to temporal coordinates, leading to more precise time-based predictions. Additionally, it is crucial to assess the dynamic feasibility of the predicted trajectories to ensure they meet realistic vehicle dynamics constraints. Furthermore, developing a measure for the uncertainty associated with the predicted trajectories would be valuable for decision-making and risk assessment in autonomous driving systems. Finally, extending our approach to other complex maneuvers, such as merging and overtaking, would significantly broaden its impact on autonomous vehicle technology.

In conclusion, this thesis has presented a novel and effective approach for lane change trajectory prediction. The demonstrated accuracy, computational efficiency, and interpretability of our method make it a valuable contribution to the ongoing pursuit of fully autonomous driving, promising safer and more intelligent vehicle behavior on our roads.

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, no. 14, 2022.

[2] I. O. Olayode, B. Du, A. Severino, T. Campisi, and F. J. Alex, "Systematic literature review on the applications, impacts, and public perceptions of autonomous vehicles in road transportation system," *Journal of Traffic and Transportation Engineering (English Edition)*, vol. 10, no. 6, pp. 1037–1060, 2023.

[3] R. Hussain and S. Zeadally, "Autonomous cars: Research results, issues, and future challenges," *IEEE Communications Surveys Tutorials*, vol. 21, no. 2, pp. 1275–1313, 2019.

[4] Liu, Zongwei, Jiang, Hao, Tan, Hong, and Zhao, Fuquan, "An overview of the latest progress and core challenge of autonomous vehicle technologies," *MATEC Web Conf.*, vol. 308, p. 06002, 2020.

[5] X. Song, H. Gao, T. Ding, Y. Gu, J. Liu, and K. Tian, "A review of the motion planning and control methods for automated vehicles," *Sensors*, vol. 23, no. 13, 2023.

[6] F. Rosique, P. Navarro, C. Fernández, and A. Padilla, "A systematic review of perception system and simulators for autonomous vehicles research," *Sensors (Basel, Switzerland)*, vol. 19, 2019.

[7] J. Leu, Y. Wang, M. Tomizuka, and S. Di Cairano, "Autonomous vehicle parking in dynamic environments: An integrated system with prediction and motion planning," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 10890–10897, 2022.

[8] Y. Jeong, "Self-adaptive motion prediction-based proactive motion planning for

autonomous driving in urban environments," *IEEE Access*, vol. 9, pp. 105612–105626, 2021.

[9] M. A. Arman and C. Tampère, "Empirical study of lane-changing maneuvers in a weaving area based on reconstructed trajectories of floating car data," *Transportation Research Record: Journal of the Transportation Research Board*, 2023.

[10] A. Qayyum, B. D. Baets, S. V. Ackere, F. Witlox, G. D. Tré, and N. van de Weghe, "When driving becomes risky: Micro-scale variants of the lane-changing maneuver in highway traffic," *Traffic Injury Prevention*, vol. 24, pp. 583 – 591, 2023.

[11] O. Prasolenko and V. Chumachenko, "Regularities of the traffic lane change by the driver when interacting with car-obstacle," *Transport technologies*, 2023.

[12] S. Dixit, S. Fallah, U. Montanaro, M. Dianati, A. Stevens, F. Mccullough, and A. Mouzakitis, "Trajectory planning and tracking for autonomous overtaking: State-of-the-art and future prospects," *Annual Reviews in Control*, vol. 45, pp. 76–86, 2018.

[13] Y. Huang, J. Du, Z. Yang, Z. Zhou, L. Zhang, and H. Chen, "A survey on trajectory-prediction methods for autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 3, pp. 652–674, 2022.

[14] S. Lefevre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *Robomech Journal*, vol. 1, 07 2014.

[15] S. Ammoun and F. Nashashibi, "Real time trajectory prediction for collision risk estimation between vehicles," in *2009 IEEE 5th International Conference on Intelligent Computer Communication and Processing*, pp. 417–422, 2009.

[16] R. Schubert, E. Richter, and G. Wanielik, "Comparison and evaluation of advanced motion models for vehicle tracking," in *2008 11th International Conference on Information Fusion*, pp. 1–6, 2008.

[17] P. Lytrivis, G. Thomaidis, and A. Amditis, "Cooperative path prediction in vehicular environments," in *2008 11th International IEEE Conference on Intelligent Transportation Systems*, pp. 803–808, 2008.

[18] A. Barth and U. Franke, "Where will the oncoming vehicle be the next second?," in *2008 IEEE Intelligent Vehicles Symposium*, pp. 1068–1073, 2008.

[19] C.-F. Lin, A. Ulsoy, and D. LeBlanc, "Vehicle dynamics and external disturbance estimation for vehicle path prediction," *IEEE Transactions on Control Systems Technology*, vol. 8, no. 3, pp. 508–518, 2000.

[20] C. M. Kang, S. J. Jeon, S.-H. Lee, and C. C. Chung, "Parametric trajectory prediction of surrounding vehicles," in *2017 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, pp. 26–31, 2017.

[21] J. Do, K. Han, and S. B. Choi, "Lane change–intention inference and trajectory prediction of surrounding vehicles on highways," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 7, pp. 3813–3825, 2023.

[22] T. Batz, K. Watson, and J. Beyerer, "Recognition of dangerous situations within a cooperative group of vehicles," in *2009 IEEE Intelligent Vehicles Symposium*, pp. 907–912, 2009.

[23] M. Deng, S. Li, X. Jiang, and X. Li, "Vehicle trajectory prediction method based on "current" statistical model and cubature Kalman filter," *Electronics*, vol. 12, p. 2464, 05 2023.

[24] V. Lefkopoulos, M. Menner, A. Domahidi, and M. N. Zeilinger, "Interaction-aware motion prediction for autonomous driving: A multiple model Kalman filtering scheme," *IEEE Robotics and Automation Letters*, vol. 6, no. 1, pp. 80–87, 2021.

[25] R. Toledo-Moreo and M. Zamora-Izquierdo, "IMM-based lane-change prediction in highways with low-cost GPS/INS," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, pp. 180–185, 2009.

[26] H. M. Mandalia and M. D. D. Salvucci, "Using support vector machines for lane-change detection," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 49, no. 22, pp. 1965–1969, 2005.

[27] S. Gambs, M.-O. Killijian, and M. Nunez del Prado Cortez, "Next place prediction using mobility Markov chains," *Proceedings of the First Workshop on Measurement, Privacy, and Mobility*, 01 2012.

[28] H. Berndt, J. Emmert, and K. Dietmayer, "Continuous driver intention recognition with hidden Markov models," in *2008 11th International IEEE Conference on Intelligent Transportation Systems*, pp. 1189–1194, 2008.

[29] S. Qiao, D. Shen, X. Wang, N. Han, and W. Zhu, "A self-adaptive parameter selection trajectory prediction approach via hidden Markov models," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 284–296, 2015.

[30] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. Cambridge, MA, USA: MIT Press, 2009.

[31] Y. Jiang, B. Zhu, S. Yang, J. Zhao, and W. Deng, "Vehicle trajectory prediction considering driver uncertainty and vehicle dynamics based on dynamic Bayesian network," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 2, pp. 689–703, 2023.

[32] J. Li, B. Dai, X. Li, X. Xu, and D. Liu, "A dynamic Bayesian network for vehicle maneuver prediction in highway driving scenarios: Framework and verification," *Electronics*, vol. 8, p. 40, 01 2019.

[33] J. Liu, X. Mao, Y. Fang, D. Zhu, and M. Q. Meng, "A survey on deep-learning approaches for vehicle trajectory prediction in autonomous driving," *CoRR*, vol. abs/2110.10436, 2021.

[34] S. Jia, F. Hui, C. Wei, X. Zhao, and J. Liu, "Lane-changing behavior prediction based on game theory and deep learning," *Journal of Advanced Transportation*, 2021.

[35] D. fan Xie, Z. Fang, B. Jia, and Z. He, "A data-driven lane-changing model based on deep learning," *Transportation Research Part C: Emerging Technologies*, 2019.

[36] Y. yuan Ren, L. Zhao, X. lian Zheng, and X. Li, "A method for predicting diverse lane-changing trajectories of surrounding vehicles based on early detection of lane change," *IEEE Access*, vol. 10, pp. 17451–17472, 2022.

[37] Z. Li, X. Huang, T. Mu, and J. Wang, "Attention-based lane change and crash risk prediction model in highways," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, pp. 22909–22922, 2022.

[38] W. Ding and S. Shen, "Online vehicle trajectory prediction using policy anticipation network and optimization-based context reasoning," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 9610–9616, 2019.

[39] N. Deo and M. M. Trivedi, "Convolutional social pooling for vehicle trajectory prediction," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1549–15498, 2018.

[40] G. Xie, A. Shangguan, R. Fei, W. Ji, W. Ma, and X. Hei, "Motion trajectory prediction based on a CNN-LSTM sequential model," *Science China Information Sciences*, vol. 63, 11 2020.

[41] A. Mohamed, K. Qian, M. Elhoseiny, and C. Claudel, "Social-STGCNN: A social spatio-temporal graph convolutional neural network for human trajectory prediction," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (Los Alamitos, CA, USA), pp. 14412–14420, IEEE Computer Society, jun 2020.

[42] T. Phan-Minh, E. Grigore, F. A. Boulton, O. Beijbom, and E. M. Wolff, "CoverNet: Multimodal behavior prediction using trajectory sets," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (Los Alamitos, CA, USA), pp. 14062–14071, IEEE Computer Society, jun 2020.

[43] Y. Yoon and K. Yi, "Trajectory prediction using graph-based deep learning for longitudinal control of autonomous vehicles: A proactive approach for autonomous driving in urban dynamic traffic environments," *IEEE Vehicular Technology Magazine*, vol. 17, pp. 18–27, 2022.

[44] Z. Zhao, H. Fang, Z. Jin, and Q. Qiu, "GISNet:graph-based information sharing network for vehicle trajectory prediction," in *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7, 2020.

[45] X. Li, X. Ying, and M. C. Chuah, "GRIP: Graph-based interaction-aware trajec-

tory prediction," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 3960–3966, 2019.

[46] Y. Wang, S. Zhao, R. Zhang, X. Cheng, and L. Yang, "Multi-vehicle collaborative learning for trajectory prediction with spatio-temporal tensor fusion," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 1, pp. 236–248, 2022.

[47] A. H. Mohammed, *Optimal control problems for safe and efficient lane changes of self-driving vehicles*. Phd thesis, Cankaya University, 2017.

[48] U.S. Department Of Transportation Federal Highway Administration, "Next generation simulation (NGSIM) vehicle trajectories and supporting data," 2017.