MID-AMERICA TRANSPORTATION CENTER

Report # MATC-UNL: 004-23

Final Report
WBS: 25-1121-0005-004-23

Virtual Barriers for Mitigating and Preventing Run-off-Road Crashes - Phase III

Cody Stolle, PhD
Research Assistant Professor
Department of Mechanical and Materials Engineering
Midwest Roadside Safety Facility
University of Nebraska-Lincoln

Ricardo Jacome, BSME
Graduate Research Assistant, MwRSF

Ronald K. Faller, PhD, PE
MwRSF Director, Research Professor
Department of Civil and Environmental Engineering

Michael Sweigard, MSME
Former Graduate Research Assistant, MwRSF

2021

A Cooperative Research Project sponsored by
U.S. Department of Transportation- Office of the Assistant Secretary for Research and Technology

Virtual Barriers for Mitigating and Preventing Run-off-Road Crashes – Phase III

Cody Stolle, Ph.D.
Midwest Roadside Safety Facility (MwRSF)
Research Assistant Professor, Mechanical and Materials Engineering

Ronald K. Faller, Ph.D., P.E.
MwRSF Director
Research Professor, Civil Engineering

Ricardo Jacome, B.S.M.E.
Graduate Research Assistant, MwRSF

Michael Sweigard, M.S.M.E.
Former Graduate Research Assistant, MwRSF

A Report on Research Sponsored by

Mid-America Transportation Center

University of Nebraska–Lincoln

Final Report

TECHNICAL REPORT DOCUMENTATION PAGE

<table>
<tr><td colspan="2">1. Report No.
25-1121-0005-004-23</td><td>2.</td><td colspan="2">3. Recipient's Accession No.</td></tr>
<tr><td colspan="3" rowspan="2">4. Title and Subtitle
Virtual Barriers for Mitigating and Preventing Run-off-Road Crashes – Phase III</td><td colspan="2">5. Report Date
Final Report
February 2021</td></tr>
<tr><td colspan="2">6.</td></tr>
<tr><td colspan="3">7. Author(s)
Jacome, R.O., Sweigard, M., Stolle, C.S., and Faller, R.K.</td><td colspan="2">8. Performing Organization Report No.
25-1121-0005-004-23</td></tr>
<tr><td colspan="3" rowspan="2">9. Performing Organization Name and Address
Midwest Roadside Safety Facility (MwRSF)
Nebraska Transportation Center
University of Nebraska-Lincoln

Main Office:
Prem S. Paul Research Center at Whittier School
Room 130, 2200 Vine Street
Lincoln, Nebraska 68583-0853

Outdoor Test Site:
4630 NW 36th Street
Lincoln, Nebraska 68524</td><td colspan="2">10. Project/Task/Work Unit No.</td></tr>
<tr><td colspan="2">11. Contract (C) or Grant (G) No.
69A3551747107</td></tr>
<tr><td colspan="3" rowspan="2">12. Sponsoring Organization Name and Address
Office of the Assistant Secretary for Research and Technology
1200 New Jersey Ave., SE
Washington, D.C. 20590</td><td colspan="2">13. Type of Report and Period Covered
Final Report: 2019-2021</td></tr>
<tr><td colspan="2">14. Sponsoring Agency Code
TRB RiP # 91994-60</td></tr>
<tr><td colspan="5">15. Supplementary Notes
Prepared in cooperation with U.S. Department of Transportation, Federal Highway Administration.</td></tr>
<tr><td colspan="5">16. Abstract
This research study describes progress made during a multi-year evaluation of concepts to prevent vehicles from departing the roadway. The objective of the Year 3 effort was to conclude the development of a dynamically-concise map framework for guiding connected and automated vehicles equipped with necessary hardware to identify the vehicle's current position and heading, and react to current and upcoming lane geometry to ensure continued safe travel.

The method is separated into three main modules denoted as: Local Path Generation, Local Positioning, and Vehicle Guidance/Warning. The Local Path Generation module explored techniques to wirelessly convey road data to a vehicle while requiring a very low quantity and frequency of data exchange between the vehicle and the environment. The guidance information is collected to develop a local road database and referenced locally, geospatially, and relative to other adjacent road segments. As well, the vehicle instantaneous position is identified using the Local Positioning module, in which the coordinates of the vehicle can be quickly related in terms of position, speed, and orientation with respect to the roadway with minimal lag. The Vehicle Guidance System module is the reaction system which compares data from Local Path Generation and Local Positioning modules to determine if the risk of roadside departure exceeds an unacceptable level of risk, and responds by notifying the driver and/or performing safety maneuvers to control the vehicle path. Feasibility and application of these modules and concepts were explored and further research recommendations were provided for the third and final year of MATC funding.</td></tr>
<tr><td colspan="3">17. Document Analysis/Descriptors
Highway Safety, Run off road crashes, Path Prediction, Vehicle Controls, Inter-Vehicle Communication, Traffic Management</td><td colspan="2">18. Availability Statement
No restrictions. Document available from: National Technical Information Services, Springfield, Virginia 22161</td></tr>
<tr><td>19. Security Class (this report)
Unclassified</td><td>20. Security Class (this page)
Unclassified</td><td>21. No. of Pages
105</td><td colspan="2">22. Price</td></tr>
</table>

SI* (MODERN METRIC) CONVERSION FACTORS

APPROXIMATE CONVERSIONS TO SI UNITS

Symbol	When You Know	Multiply By	To Find	Symbol
		LENGTH		
in.	inches	25.4	millimeters	mm
ft	feet	0.305	meters	m
yd	yards	0.914	meters	m
mi	miles	1.61	kilometers	km
		AREA		
in^2	square inches	645.2	square millimeters	mm^2
ft^2	square feet	0.093	square meters	m^2
yd^2	square yard	0.836	square meters	m^2
ac	acres	0.405	hectares	ha
mi^2	square miles	2.59	square kilometers	km^2
		VOLUME		
fl oz	fluid ounces	29.57	milliliters	mL
gal	gallons	3.785	liters	L
ft^3	cubic feet	0.028	cubic meters	m^3
yd^3	cubic yards	0.765	cubic meters	m^3
		1 NOTE: volumes greater than 1,000 L shall be shown in m^3		
		MASS		
oz	ounces	28.35	grams	g
lb	pounds	0.454	kilograms	kg
T	short ton (2,000 lb)	0.907	megagrams (or “metric ton”)	Mg (or "t")
		TEMPERATURE (exact degrees)		
°F	Fahrenheit	5(F-32)/9 or (F-32)/1.8	Celsius	°C
		ILLUMINATION		
fc	foot-candles	10.76	lux	lx
fl	foot-Lamberts	3.426	candela per square meter	cd/m^2
		FORCE & PRESSURE or STRESS		
lbf	poundforce	4.45	newtons	N
lbf/in^2	poundforce per square inch	6.89	kilopascals	kPa

APPROXIMATE CONVERSIONS FROM SI UNITS

Symbol	When You Know	Multiply By	To Find	Symbol
		LENGTH		
mm	millimeters	0.039	inches	in.
m	meters	3.28	feet	ft
m	meters	1.09	yards	yd
km	kilometers	0.621	miles	mi
		AREA		
mm^2	square millimeters	0.0016	square inches	in^2
m^2	square meters	10.764	square feet	ft^2
m^2	square meters	1.195	square yard	yd^2
ha	hectares	2.47	acres	ac
km^2	square kilometers	0.386	square miles	mi^2
		VOLUME		
mL	milliliter	0.034	fluid ounces	fl oz
L	liters	0.264	gallons	gal
m^3	cubic meters	35.314	cubic feet	ft^3
m^3	cubic meters	1.307	cubic yards	yd^3
		MASS		
g	grams	0.035	ounces	oz
kg	kilograms	2.202	pounds	lb
Mg (or "t")	megagrams (or “metric ton”)	1.103	short ton (2,000 lb)	T
		TEMPERATURE (exact degrees)		
°C	Celsius	1.8C+32	Fahrenheit	°F
		ILLUMINATION		
lx	lux	0.0929	foot-candles	fc
cd/m^2	candela per square meter	0.2919	foot-Lamberts	fl
		FORCE & PRESSURE or STRESS		
N	newtons	0.225	poundforce	lbf
kPa	kilopascals	0.145	poundforce per square inch	lbf/in^2

*SI is the symbol for the International System of Units. Appropriate rounding should be made to comply with Section 4 of ASTM E380.

Table of Contents

List of Figures

List of Tables

Acknowledgements

The authors wish to acknowledge several sources that made a contribution to this project: (1) Mid-America Transportation Center.; (2) Richard Stepp at Florida DOT David Boruff at Indiana DOT; Erik Emerson at Wisconsin DOT, and Phil Tenhulzen at Nebraska DOT for serving as project monitors; and (3) Jason Marks at National Instruments, Jon Barad at Velodyne Lidar, Amanda Hamm at Virginia DOT and head of the Connected Vehicles Pooled Fund, and Fahad Shuja at the Ontario Good Roads Association and Municipal Alliance for Connected and Automated Vehicles in Ontario for providing data and providing feedback on research content. Acknowledgement is also given to the following individuals who contributed to the completion of this research project.

Midwest Roadside Safety Facility

J.D. Reid, Ph.D., Professor
J.D. Rasmussen, Ph.D., P.E., Former Research Associate Professor
J.C. Holloway, M.S.C.E., E.I.T., Test Site Manager
K.A. Lechtenberg, M.S.M.E., E.I.T., Research Engineer
R.W. Bielenberg, M.S.M.E., E.I.T., Research Engineer
S.K. Rosenbaugh, M.S.C.E., E.I.T., Research Engineer
A.T. Russell, B.S.B.A., Testing and Maintenance Technician II
S.M. Tighe, Construction and Testing Technician I
D.S. Charroin, Construction and Testing Technician I
M.A. Rasmussen, Former Construction and Testing Technician I
E.W. Krier, Construction and Testing Technician II
M.T. Ramel, Construction and Testing Technician I
R.M. Novak, Construction and Testing Technician I
J.E. Kohtz, B.S.M.E., Former CAD Technician
E.L. Urbank, B.A., Research Communication Specialist
Undergraduate and Graduate Research Assistants

Disclaimer Statement

This report was completed with funding from the Mid-American Transportation Center. The contents of this report reflect the views and opinions of the authors who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the Mid-American Transportation Center. This report does not constitute a standard, specification, regulation, product endorsement, or an endorsement of manufacturers.

Abstract

This research study describes progress made during a multi-year evaluation of concepts to prevent vehicles from departing the roadway. The objective of the Year 3 effort was to conclude the development of a dynamically-concise map framework for guiding connected and automated vehicles equipped with necessary hardware to identify the vehicle's current position and heading, and react to current and upcoming lane geometry to ensure continued safe travel.

The method is separated into three main modules denoted as: Local Path Generation, Local Positioning, and Vehicle Guidance/Warning. The Local Path Generation module explored techniques to wirelessly convey road data to a vehicle while requiring a very low quantity and frequency of data exchange between the vehicle and the environment. The guidance information is collected to develop a local road database and referenced locally, geospatially, and relative to other adjacent road segments. As well, the vehicle instantaneous position is identified using the Local Positioning module, in which the coordinates of the vehicle can be quickly related in terms of position, speed, and orientation with respect to the roadway with minimal lag. The Vehicle Guidance System module is the reaction system which compares data from Local Path Generation and Local Positioning modules to determine if the risk of roadside departure exceeds an unacceptable level of risk, and responds by notifying the driver and/or performing safety maneuvers to control the vehicle path. Feasibility and application of these modules and concepts were explored and further research recommendations were provided for the third and final year of MATC funding.

Chapter 1 Introduction

1.1 Problem Statement

The Federal Highway Administration (FHWA) reported that approximately 53% of fatal crashes (18,779) between 2014 and 2016 were related to roadside departures or lane departures [1]. According to the National Highway Traffic Safety Administration (NHTSA) *Fatal Accident Reporting System* (FARS) statistics [2] and the Insurance Institute for Highway Safety (IIHS) and Highway Loss Data Institute (HLDI) annual compilations [3], approximately 10,000 fatal run-off-road (ROR) crashes occur each year involving roadside fixed objects. As a result, ROR fixed-object fatal crashes account for approximately 1/3 of all fatal crashes. Cross-median crashes are among the most deadly type of ROR crash, in which a vehicle exits the travel-way and crosses a median, striking an opposing vehicle from the opposite travel direction.

Non-fatal ROR crashes are also concerning and economically devastating. Research conducted at the University of Nebraska-Lincoln identified approximately 440,000 crashes involving only roadside trees and utility poles in a five-year period spanning twelve geographically dissimilar states [4]. It was determined that the United States experiences an estimated $13 to 17 billion in direct (emergency medical services, first responders, cleanup, infrastructure repairs) and indirect costs (traffic congestion, loss of workdays and taxable income, incapacitation, lawsuits) related only to non-fatal roadside tree and utility pole crashes every year. The crashes reviewed indicated that many ROR crashes exhibit similar attributes:

- Drift-Off Road: vehicle slowly departs roadway (typically at a small angle of departure and straight-line trajectory). This condition is most commonly associated with drowsy or impaired drivers, or drivers with medical episodes.

- Overcorrection: the vehicle experiences a path change (drift out of lane, lane change, avoidance maneuver), then the driver overcompensates and over-steers while attempting to guide the vehicle back to the desired lane. This roadside departure type commonly results in spinout and skidding.
- Failure to Negotiate Curve: vehicle veers to the outside of a curve. Condition is frequently associated with high travel speeds or poor pavement friction (e.g., ice).
- Avoidance Maneuver: vehicle performs evasive maneuver to avoid crashing into an object, person, or animal in lane. This roadside departure condition is commonly associated with higher travel speeds (e.g., freeway), and is abrupt and panicked.

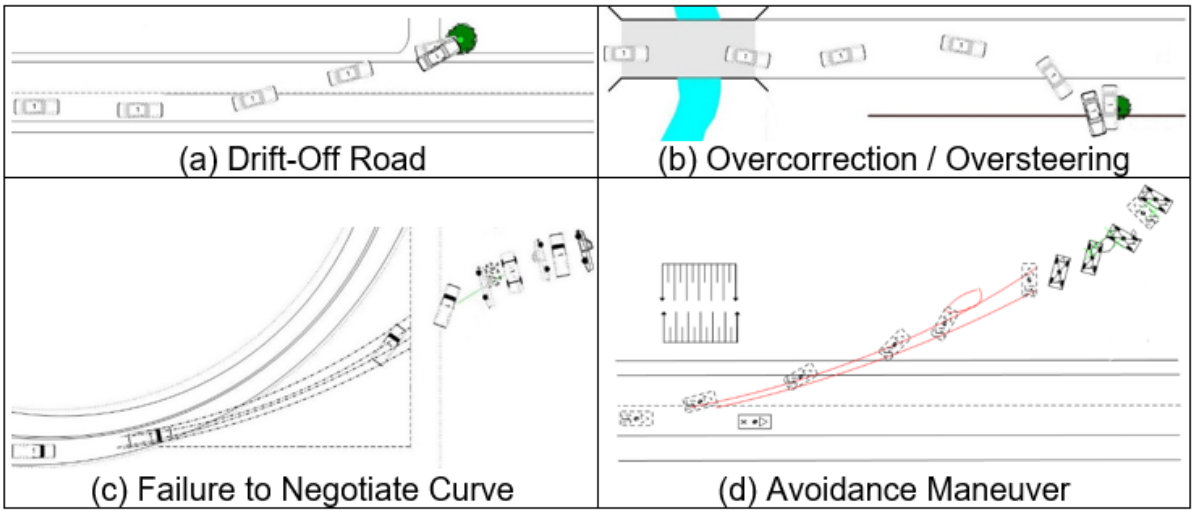

Figure 1.1 Examples of ROR Crashes (images take from NHTSA's NASS CDS)

1.2 Current Technology Review

New technology is installed in modern vehicles to help reduce the frequency of ROR excursions. Advanced driver-assistance systems (ADAS) assist the driver by identifying the geometry of the road using lane markings to help keep the driver on the road [5-7].

The use of ADAS technology has been augmented through wireless communications. Leading to different schemes such as Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I). These vehicles with wireless capacities are denoted as Connected and Autonomous Vehicles (CAVs). In recent years, V2I communications with CAVs have been used to provide traffic flow data to vehicles that comes from a station monitoring current state of traffic of a given area. As of today, this technology is still in development and being tested by Departments of Transportation (DOTs) and research agencies.

According to SAE Recommended Practice J306 2018, CAVs can be classified into different levels of autonomy [8]. These levels are based on the amount of Dynamic Driving Tasks (DDTs) a vehicle can perform. DDTs are classified into Longitudinal DDTs and Lateral DDTs:

- Longitudinal DDT: Controlling the brake and throttle to achieve the desired forward vehicle travel behavior.
- Lateral DDT: Maintaining a desired vehicle travel direction by modulating braking and steering that range from simple guidance on the road and evolve to avoidance for emergency maneuvering.

DDTs can be performed by either a human driver or an autonomous controller in the vehicle. For levels of autonomy 0-2, a human driver performs all DDTs. Levels 3 and 4 provide some complete vehicle automation wherein a controller takes over certain DDTs subject to

criteria (minimum speed, clear visibility, etc.), and level 5 consists of a vehicle performing all DDTs. Currently, no level 5 autonomy vehicle exists on the current market [9, 10].

1.2.1 Sensors and Limitations

The level of autonomy depends entirely on the sensors a vehicle has. A simple overview of the main sensors in a vehicle is shown in Figure 1.2, with more sensor specification found in previous work [11]. Sensors primarily serve two functions: localization and environment recognition. Environment recognition can be in the form of light sensors, or ultrasound sensors. Localization sensors are usually in the form of Global Positioning Systems. These sensors are prone to sources of errors while driving. Environment recognition sensors rely on the amount of brightness, paint quality in lane markings, and background objects that can blend into vision fields. For positioning, GPS has presented a degree of error around 1.82 m (~6 ft.), which is enough to misplace a vehicle in a different lane [12, 13].

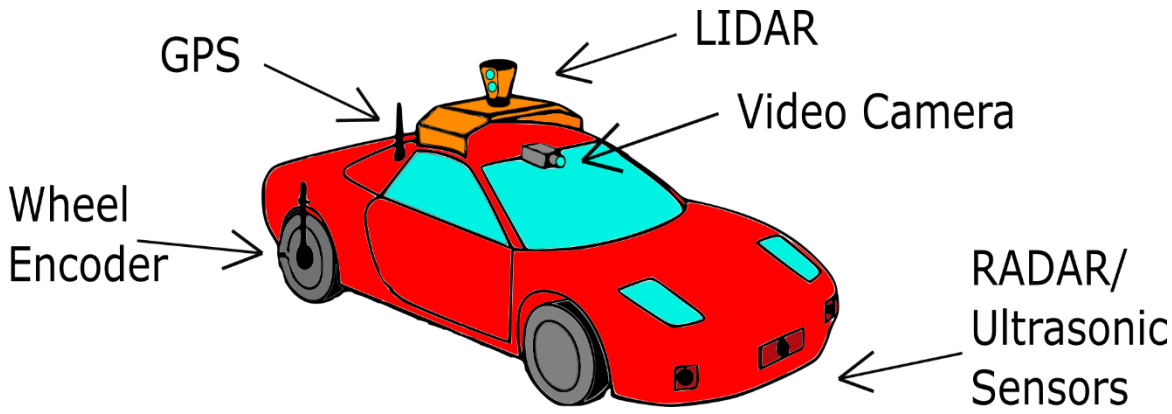

Figure 1.2 Sensors Example

Current research focuses on developing different categories of vehicle lane-keeping technology, mainly addressing errors in sensor-based data acquisition. These features have

demonstrated excellent performance under controlled environments, but in-practice roadways differ significantly from the as-tested environments [14]. For example, poor roadway quality or deviations between old and new roadside paint lines can create recognition problems for optic systems. Adverse weather events, such as rain, snow, ice, and fog may obscure lane edges and prevent proper recognition of visual systems. Many northern states utilize salt mixtures to prevent or manage ice formation on roads during winter months, which can lead to pavement bleaching and affect lane edge identification.

GPS-assisted systems to date have not shown sufficient accuracy to maintain vehicles in a lane and prevent them from encroaching into adjacent lanes, which could cause a crash or instability. Options such as Differential GPS and Real-Time Kinematic systems provide higher accuracy but implementation costs are unfeasible for driving vehicles. Currently, connected vehicle technology has opened up as the avenue to solve the aforementioned problems [15].

1.2.2 Connected Vehicles

In recent years, CAVs have opened up many research fields such as the Variable Speed Limit (VSL), in which speed limits depend on factors that optimize traffic flow [16]. These efforts involve the use of Roadside Units (RSUs) which along with cellular stations, broadcast information to vehicles as shown in Figure 1.3. Implementation efforts include WYDOT, which has developed V2I projects to investigate the relationship on RSUs with trucks or cargo vehicles [17]. Similarly, VDOT has developed their own RSU infrastructure, and cloud databases to store information such as construction sites or speed limits [18]. The culminations of these groups is to offer a working prototype for full-scale deployment communications in between RSUs and CAVs [19].

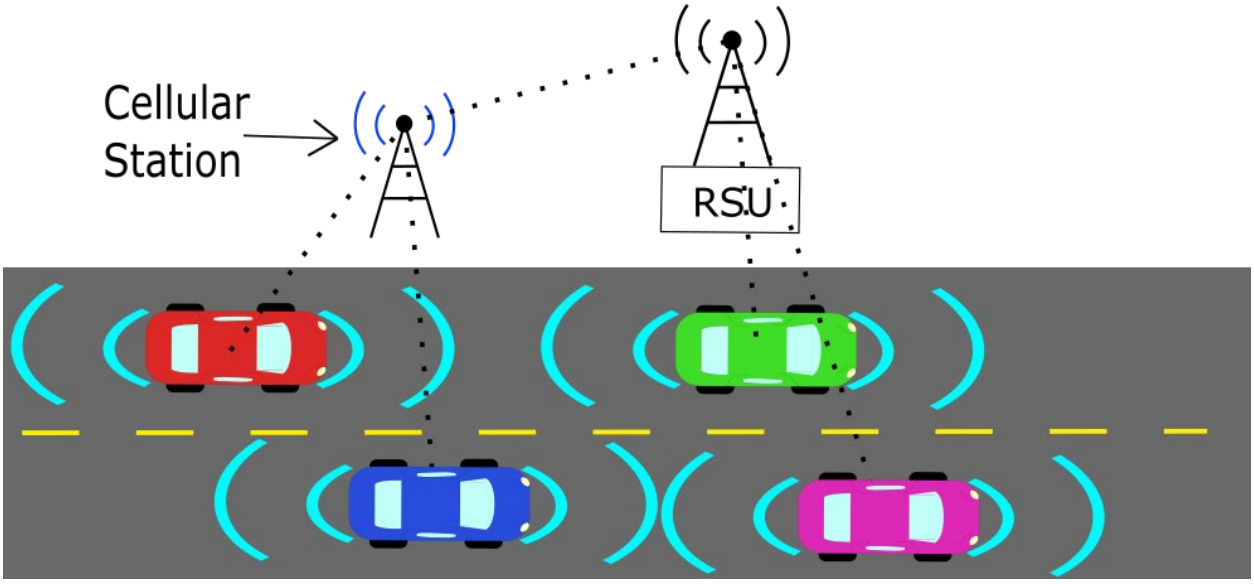

Figure 1.3 Connected Vehicles Example

Traditionally, infrastructure technology is designed to provide human drivers with input to navigate roads. Sensor recognition technology aims to augment the human capacity of environment recognition for higher safety. However, V2I technology fosters the possibility of providing road-characteristics recognition that is rendered as useless to conventional drivers. These road-characteristics are the mathematical curvature and street designs that are inherent to the road itself.

1.3 Proposed System Solution

A new alternative was offered and denoted as the MATC Smart Barrier. This proposed system is independent of the vehicle sensors and is not affected by some of the same environmental disturbances that can adversely affect the operation of current ADAS systems. This system is not intended to replace existing ADAS or CAVs. Instead, it is designed in conjunction with previously established CAV technology. A previous study has outlined the steps necessary to develop the entire MATC Smart Barrier system [20]. A brief summary will be

given and the remainder of this work will focus on elaborating one sub-part of the MATC Smart Barrier.

The MATC Smart Barrier is divided in three main modules:

- Road Module: Mathematical description of road centerlines for vehicles to reference while on-driving scenarios.
- Communication Module: Transmitting road module outputs to vehicles through V2I communications and localization of vehicles with respect to RSUs.
- Vehicle Module: Controlling the vehicle to maneuvering according to the output obtained from the communication module.

The three modules are depicted in Figure 1.4 below, where a red dotted line denotes the moment in which the vehicle has obtained the road centerline reference. The black dotted line denotes the road reference the vehicle can use, and the angle θ denotes the amount of steering needed to compensate the turn and go back to the road reference. The communication and vehicles modules have been discussed and elaborated in previous studies [21-23]. This work will focus only on the theoretical and mathematical development of the Road Module.

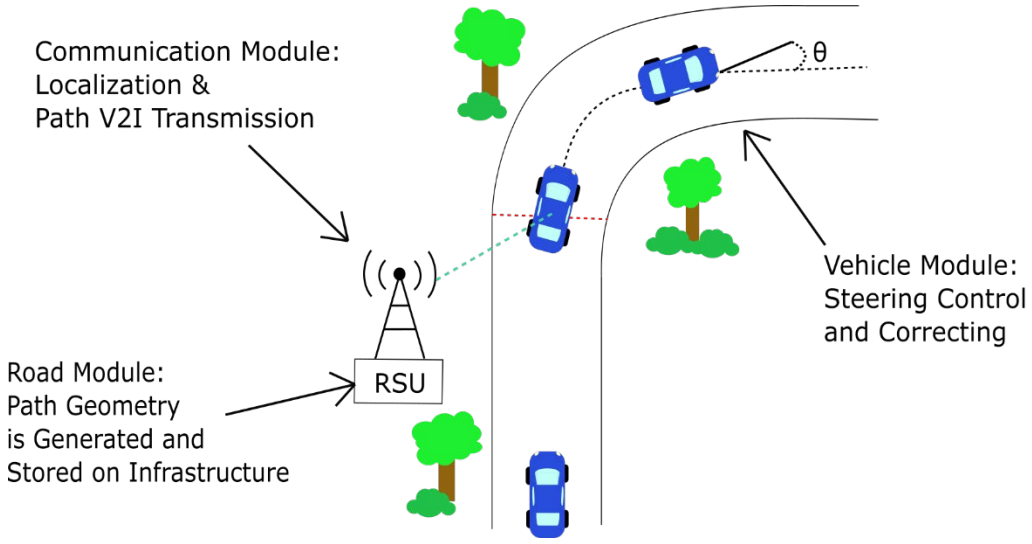

Figure 1.4 MATC Smart Barrier Concept with Modules [20]

The objective of this work is to establish the mathematical foundation of a road reference centerline. This road reference can be stored in RSUs so that CAVs have a backup for navigation when other systems become unreliable.

The presented study has the potential to be implemented in a distributed model of vehicle fleets, but is not limited solely to passenger vehicles. Examples of other vehicle types, which could utilize the target path formulation for positional error estimation and corrections, include agricultural vehicles, transport vehicles (e.g., autonomous trucks), unmanned aerial systems, or mobile robots.

The following scheme is proposed for an implementation of the discrete road decomposition, as shown in Figure 1.5. The first step involves collection of road data through any convenient means: GPS data, surveying, or aerial scanning. The following step (denoted in Figure 1.5 with a dotted box) involves how input data is processed. This step will constitute the entirety of this work. This road data contains a representation of the road centerlines, which can be exported in different formats. These road centerlines are decomposed, stored in a road target-

path matrix, and transmitted wirelessly to a vehicle in motion. The infrastructure may also assist with precise vehicle localization to improve error estimation, allowing the vehicle onboard systems to have excellent real-time observation of potential deviations from the target path. Finally, a controller is developed to consider the heading based on the discrete road decomposition and navigate safely through the road.

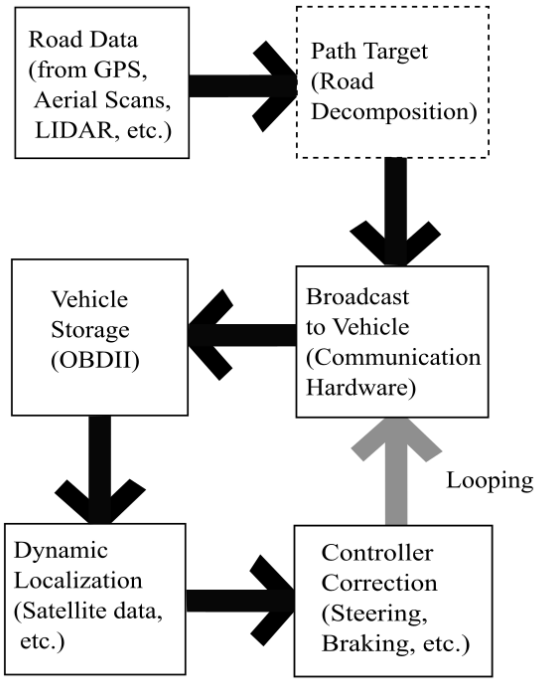

Figure 1.5 Implementation Scheme for Road Curvature Decomposition [24]

Because the system does not rely on local ad hoc determination of lane boundaries and does not utilize machine vision or de facto external tracking systems, the system is well positioned to provide guidance for autonomous vehicles even in adverse weather conditions, poor visibility, and for temporary road or lane closures. The dynamic road network relay to autonomous vehicles may allow for alternative route selection in the event of congestion or crash events, and external guidance information such as tire-pavement friction reductions reported by

other vehicles or estimated from weather reports may be broadcast to the vehicle in targeted geospatial areas. As such, this technique for vehicle guidance systems could be complimentary to existing lane keeping and ADAS systems for crash avoidance or mitigation.

In conclusion, this work will focus on only the mathematical foundation of a road reference centerline. Different data sources will be explored and compared after the mathematical formulations are defined. The next chapters will explore the current road design practices, and the underlying mathematical tools needed to develop the methods proposed in this work.

Chapter 2 Current Road Design Practices

To formulate a mathematical basis of a road reference module, a review and understanding of road design is first needed. This chapter explains the current methods for highway road design, and its limitations.

In the United States, the prevailing standards for road design come from The American Association of State Highway and Transportation Officials, referred as the Green Book. This book offers an extensive review of road design considerations that comply with vehicle dynamic behavior [25]. In this chapter, road designs will be considered in two main aspects: alignment and superelevation/friction.

2.1 Alignment

In road design, an alignment is defined by a series of points, lines and/or curves. These points are obtained from high accuracy GPS coordinates, each line and curve must connect exactly on a shared point or be coincident. When curves are formed, a radius of curvature is assigned to it. This is defined, as the radius of an imaginary (prescribed) circle that would fit per section of the road as shown in Figure 2.1.

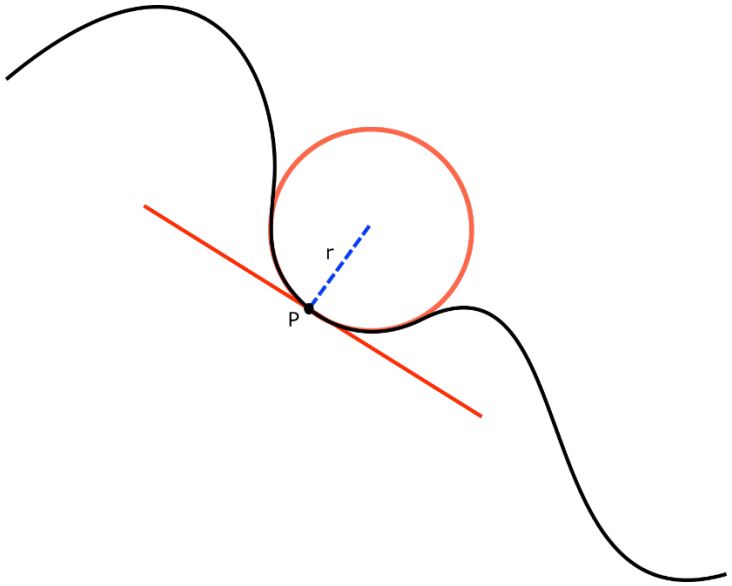

Figure 2.1 Radius of Curvature Example for Arbitrary Curve

Alignment is classified as either horizontal or vertical. Horizontal alignment refers to the lines or curves viewed topologically, similar to conventional maps. Vertical alignment refers to the changes in elevation of the roadway, resulting in a vehicle's change in pitch angle. An example of both is shown in Figure 2.2. Horizontal can be thought as the top view of a street and vertical as the orthogonal profile view of the same street.

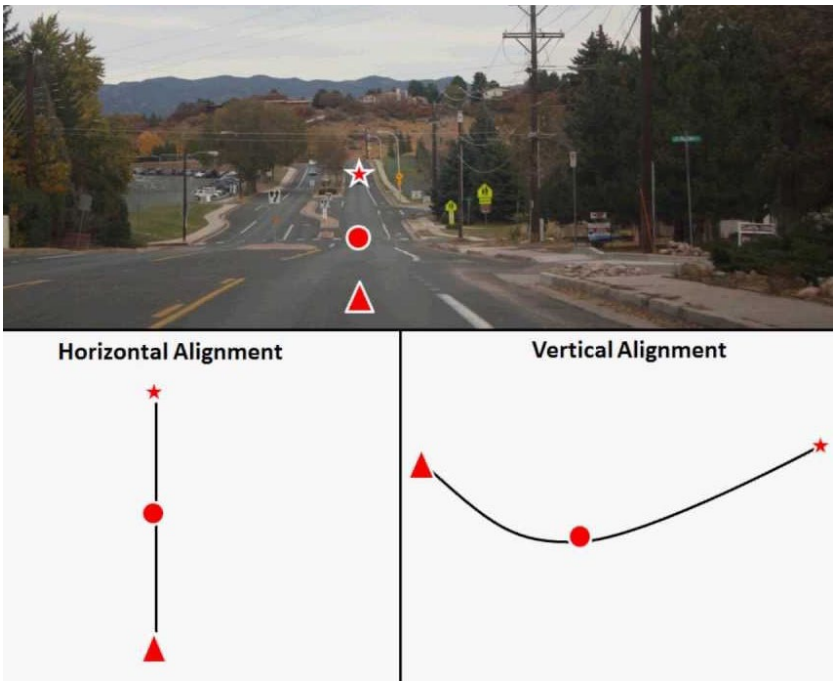

Figure 2.2 Horizontal and Vertical Alignment for a given street

Horizontal alignment affects the operational characteristics of roadways such as vehicle operating speeds, sight distances, and highway capacity. Highway road design is influenced by many factors including terrain, traffic volume, environmental factors, or right-of-way availability. Thus, road design engineers must design horizontal alignment curves to provide a safe, functional roadway facility that provides adequate sight distances within economic constraints.

There exist multiple types of horizontal alignment designs, as shown in Figure 2.3 with their respective names. In general, most designs are made with simple or spiral curves. The radius of curvature plays a crucial role in horizontal alignment design since it is the primary source of transition for vehicles in between curves.

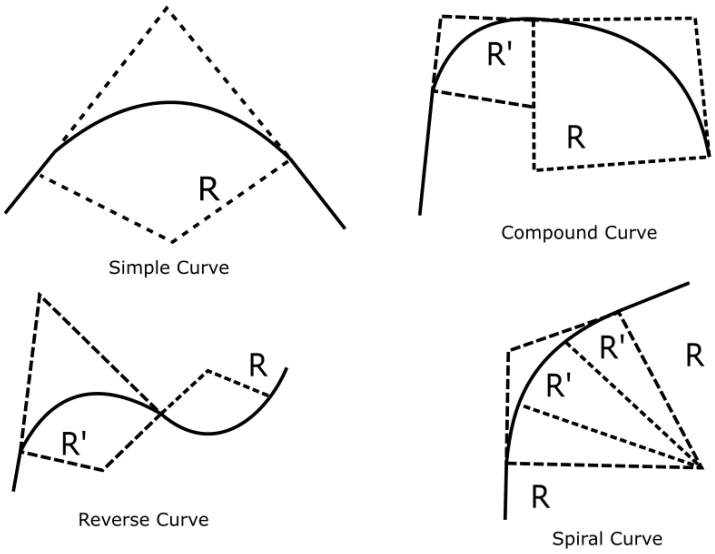

Figure 2.3 Example Horizontal Curves

The most general, and often most complicated, type of horizontal alignment is called a spiral curve. Spiral curves are made of five main components, known as initial tangent, spiral curve transition, circular curve, spiral curve transition, and final tangent. An example of a spiral curve is shown in Figure 2.4 below [26].

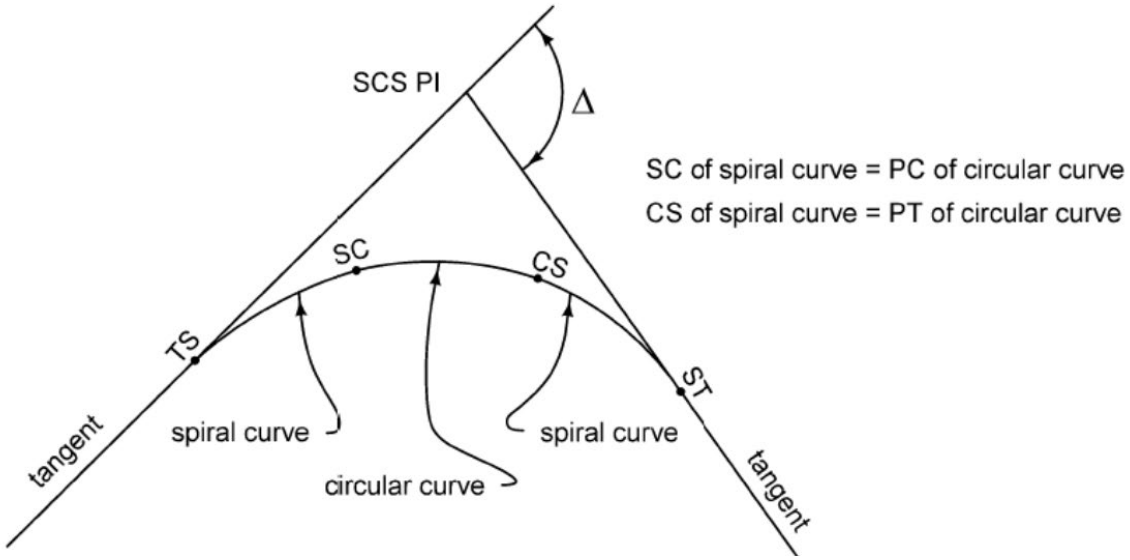

Figure 2.4 Spiral Curve Section Layout

In road design, spiral curves are designed using formulas provided by the DOT to design discrete length segments and its attributes. A sample half-spiral curve is shown in Figure 2.5 [27]. Mathematically, the formulas provided are not an analytical representation of the curve, but independent formulas to estimate appropriate waypoints, coordinates, or segment lengths of the curve. These formulas are used because they work in conjunction with current software of road design, but are not an appropriate reference for vehicles to have. The primary disadvantage is to store all different road design formulas and have to look up the appropriate set as different streets are encountered. Analytical models of the road horizontal alignment need to be represented for creating an appropriate road reference for autonomous vehicles. These analytical models will be derived from differential geometry and will be discussed in detail in Chapter 3. A more detailed example of the equations used in standard road horizontal alignment design is provided in Appendix A.1.

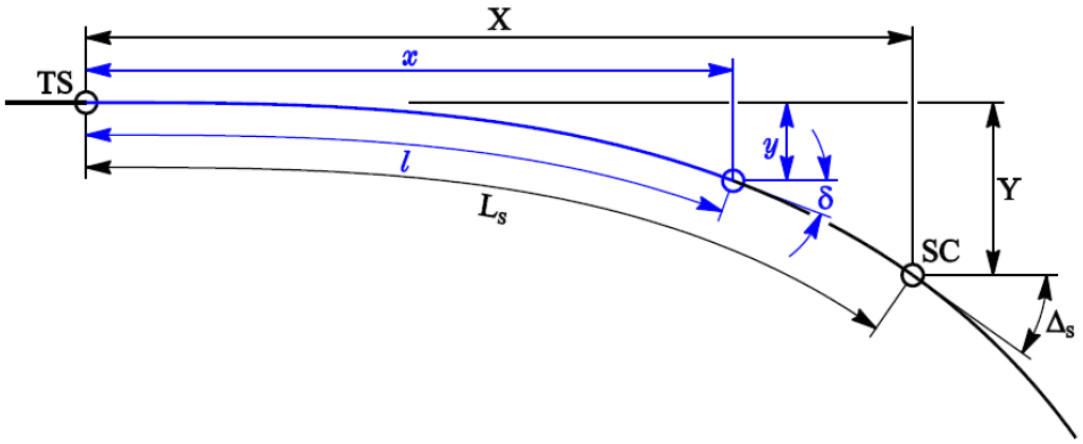

Figure 2.5 Half-Spiral Curve with Design Parameters

2.2 Superelevation/Friction

When traveling on a horizontal curve, if the road is not a straight line, centrifugal forces act on vehicles trying to pull them outward. At low speeds or large radius curves, these effects are neglected. However, at high speeds or small radius curves, the effects of centrifugal increase. An excessive amount of these forces may cause lateral movement of the turning vehicle and it may become unstable causing the vehicle to deviate from the road [28, 29].

To prevent these instabilities, superelevation and road friction are used. Superelevation is the banking of the roadway such that the outside edge of pavement is higher than the inside edge. This reduces the amount of centrifugal forces by creating a component to this force that is balanced with the vehicle's weight. The use of superelevation allows a vehicle to travel through a curve safely at a higher speed that would otherwise be impossible [25, 28]. Side friction developed between the tires and the road surface also acts to counterbalance the outward pull on the vehicle. Side friction is reduced when water, ice, or snow is present or when tires become excessively worn. When side friction is considerably low, driving should change accordingly to avoid instabilities.

Analysis on superelevation and side friction is performed by point particle dynamics as per the Green Book. The dynamics are formulated using Newton's Second Law of motion. A Free Body Diagram of the front vehicle is shown in Figure 2.6.

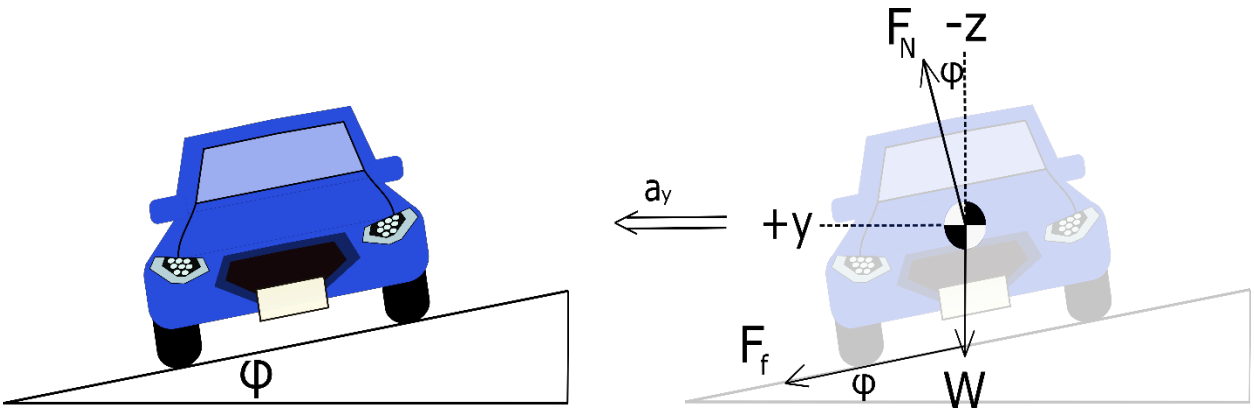

Figure 2.6 Vehicle FBD in Road with Superelevation

By performing a force analysis on the front and top view of a vehicle, the following formula is obtained [25]:

$$\frac{v^2}{g\rho} = \frac{\mu + 0.01e}{1 - 0.01\mu e} \tag{2.1}$$

v = Vehicle velocity (m/s)
e = Superelevation (as a percentage)
g = Gravitational acceleration (9.81 m/s^2)
μ = Coefficient of side road friction
ρ = Radius of curvature (m)

The Green Book graphs different combination cases of superelevation, friction, velocity, and radius of curvature as shown in Figure 2.7. Similarly, the Green Book provides tabulations

for interpolation of values as it is decided by the road design engineer. In road design, it is desired that roads contain a range of suitable parameters for a large classification of vehicles. This permits that small car vehicles and trailers can maneuver and exist in the same road with similar driving parameters.

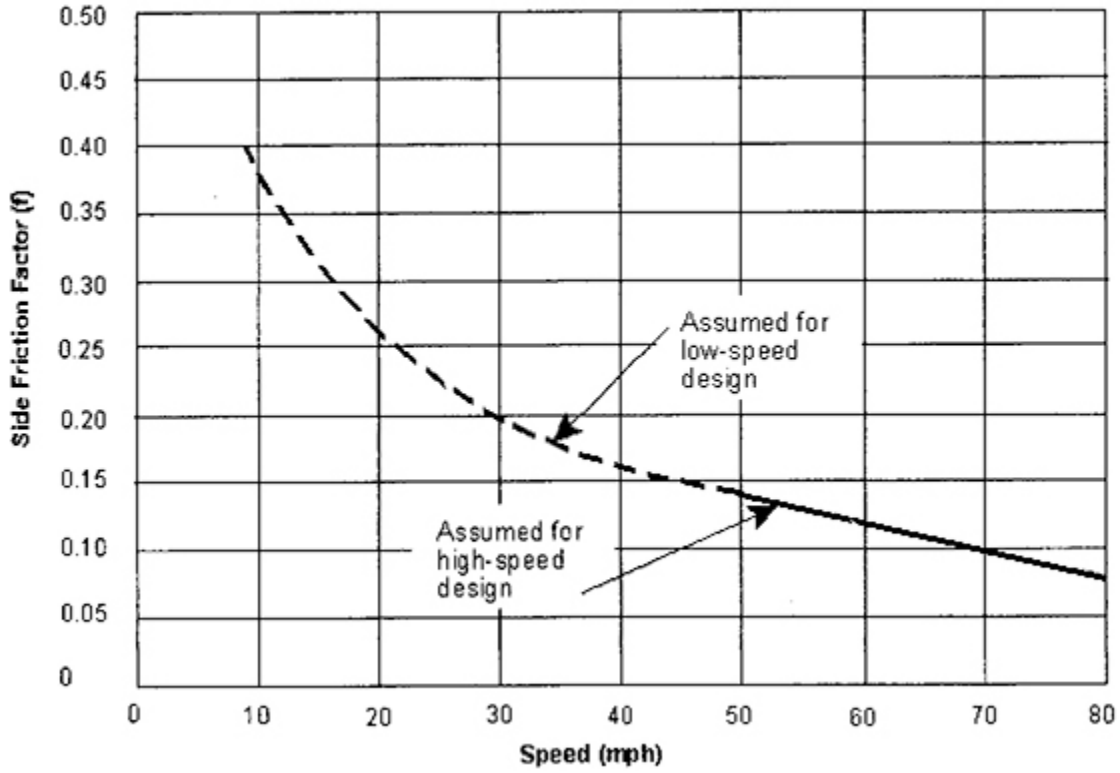

Figure 2.7 Side Friction against Speed [25]

A limitation of the Green Book includes that point particle dynamics does not take into consideration many dynamic factors that are dependent on vehicle characteristics such as track width or length.

In practice, the data to create horizontal alignment curves is based from high accuracy GPS data, and a line creation tool. For this work, the road reference will be based on a numerical

and analytical representation. The analytical portion will be analyzed next based on the differential geometry study of space curves.

2.3 Vehicle Dynamics Considerations

The Green Book analysis is limited to primarily road parameters and point-particle dynamic analysis. During high-speed and low-speed cornering, a more thorough analysis of vehicle dynamics is necessary to develop a relationship between road data and vehicle data.

The model that will be assumed is the Bicycle Model as shown in Figure 2.8 [28]. This model assumes that the behavior of the right and left tires is symmetric across the Center-of-Gravity (CG) longitudinal axis. Note in Figure 2.8, subscript denotes either rear or front, and superscripts denote either longitudinal or transverse directions.

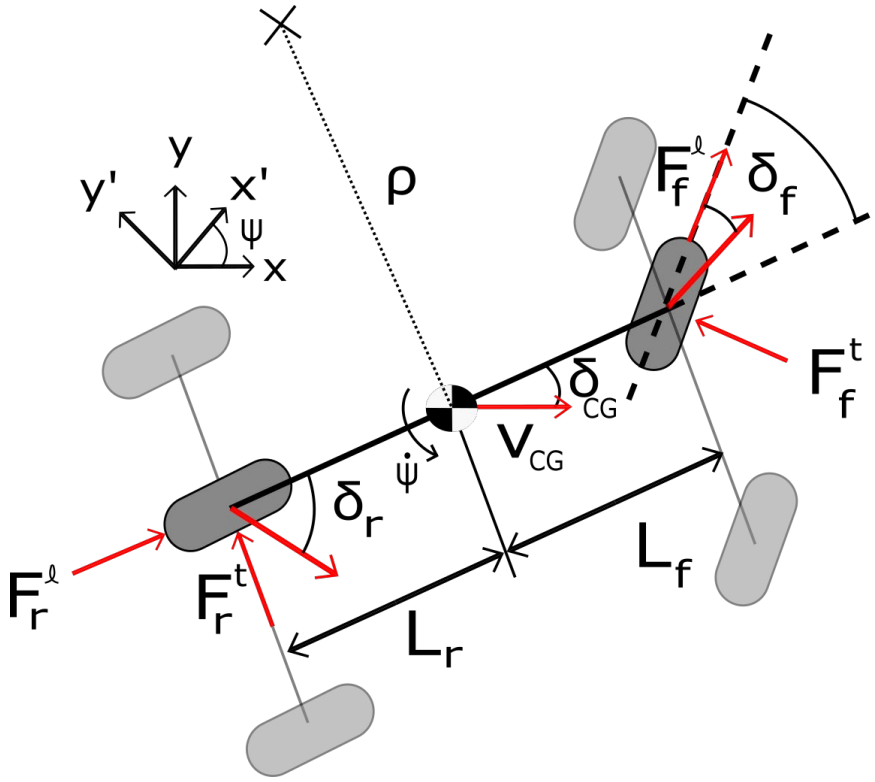

Figure 2.8 Bicycle Kinematic Model

Using the Bicycle Model, geometric considerations and Newton's Second Law of motion can be used to obtain the famous Ackerman Steering formula [28, 29]:

$$\delta_f = (57.3L + \eta v^2)\rho^{-1} \tag{2.2}$$

Where:
$\delta_f =$ Front wheel directional angle (deg)
$\rho =$ Radius of curvature (m)
$v_{CG} =$ CG Vehicle velocity (m/s)
$L =$ Total sum of rear length L_r and front length L_f (m)
$\eta =$ Understeer gradient (deg-s^2/m)

This formula relates the wheel steering angle (or Ackerman's angle) to the vehicle's current parameters, offering more information of vehicle behavior at curves. Wheel steering angle controls the change of the heading vehicle angle. Thus, proper maneuvering maintains the wheel steering angle under a range that does not cause the current vehicle angle to go on yaw instabilities.

Since $\rho \in$ R and often appears in the denominator, to avoid division by zero, the inverse function known as curvature is used such that $\kappa = \rho^{-1}$. From this point forward, only curvature κ will be used since curvature is unique for every vehicle when traversing any arbitrary road. Furthermore, all roads have already pre-determined standard curvatures that were designed for a distribution of vehicles as discussed before.

Chapter 3 Differential Geometry Overview

As described in Chapter 1, CAVs require a mathematical model to reference a given road section. However, the techniques for road designs, do not offer analytical models to represent the roads a vehicle can use.

The objective of this chapter is to develop a mathematical foundation that will be used in later chapters to represent roads in an analytical form that vehicles can reference during autonomous navigation. Fundamental principles come from classical and differential geometry. The current chapter focuses on the necessary theory behind differential geometry for the road reference. A basic understanding of geometry is assumed, preliminary geometry formulae will be presented in this chapter and serve as an establishment of the notations used throughout this work.

x_1, x_2, x_3 : Cartesian coordinates in 3-dimensional Euclidean space $\in \mathbb{R}^3$
Bolding: Vectors in space $\in \mathbb{R}^3$, i.e. $\boldsymbol{x}$ represents the components x_1, x_2, x_3
s: Arc length of curve
τ: Torsion of a curve
ρ: Radius of curvature
κ: Curvature of a curve
$\boldsymbol{N}$: Unit principal normal vector of a curve
$\boldsymbol{T}$: Unit tangent vector
$\boldsymbol{B}$: Unit binormal vector
$\|\boldsymbol{a}\| = \sqrt{\boldsymbol{a} \cdot \boldsymbol{a}}$: Norm of vector
$\boldsymbol{a} \times \boldsymbol{b} = \|\boldsymbol{a}\| \, \|\boldsymbol{b}\| \sin(\theta)$: Geometric cross product of vectors
$\boldsymbol{a} \cdot \boldsymbol{b} = \|\boldsymbol{a}\| \, \|\boldsymbol{b}\| \cos(\theta)$: Geometric dot product of vectors

In Chapter 2, radius of curvature was explained as a main characteristic that road designers use. However, for representation of analytical curves, the inverse function is used such that curvature κ is presented as:

$$\kappa(s) = \frac{1}{\rho(s)}$$

The curvature is defined as the rate of change of the tangent of the arc length. Let ρ represent the radius of a perfect circle, ds be its differential arc length, and $d\theta$ be the angle in between any axis (i.e. horizontal axis) and ρ. As it is shown in Figure 3.1, assuming small angles, the curvature of a curve can be denoted as:

$$\sin d\theta = \frac{ds}{\rho} \rightarrow d\theta = ds\kappa \rightarrow \kappa = \frac{d\theta}{ds}$$

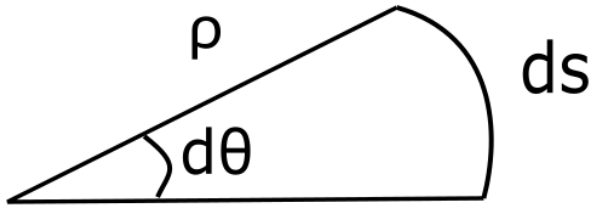

Figure 3.1 Radius of Curvature with respect to a Segment Section

3.1 Clothoids

Let a curve be represented by the form $x = x(s)$ where the arbitrary parameter s can represent arc length. By looking at the representation $x = x(s)$, there is a dependence on the choice of coordinate system in space. In this work, it is needed to represent a curve by invariant terms such that it is independent of coordinate systems. Earliest investigations of the invariance of curve representation come from Leonard Euler [30].

Invariance is defined as an intrinsic property that a curve has that is independent of the coordinate system chosen. In Euclidean 3D space, the distance represented with the norm of two points is the simplest invariant quantity that can be found. Such that if $\boldsymbol{d = y - x}$**:**

$$\|\boldsymbol{d}\| = \sqrt{\sum_{i=1}^{3}(y_i - x_i)^2} = \sqrt{\sum_{i=1}^{3}(d_i)^2} = \sqrt{(\boldsymbol{d}\cdot\boldsymbol{d})}$$

Previous studies in the theory of curves have developed the existence and uniqueness of an invariant representation of curves [30, 31]. The theory shows that arc length, curvature, and torsion are invariant functions that can describe any general curve. By making the arc length the independent variable of curvature and torsion, the following two equations are found:

$$\kappa = \kappa(s) \qquad \& \qquad \tau = \tau(s)$$

These are called the *intrinsic* or *natural equations* of the curve, such that the natural equations represent an invariant form for any general curve. Curvature can be pictured as the amount of deviation of a curve from a straight line, and torsion as the deviation of a curve from being constrained to a plane. Thus, torsion will always be zero for two-dimensional space curves.

If $\theta(s)$ is the angle to the tangent vector t(s) to the curve from a horizontal axis, then, $t(s) = [\cos(\theta)\ \sin(\theta)]$. As it was previously determined, curvature can be expressed as:

$$\kappa = \frac{d\theta}{ds} \tag{3.1}$$

Considering the case of a 2D curve, a representation can be found such that the natural equations are:

$$\kappa = \frac{s}{c^2} \qquad \& \qquad \tau = 0 \tag{3.2}$$

Where c represents an arbitrary constant. This set of natural equations are known as a Cornu Spiral. Recalling Equation (2.1) and $\theta(s)$, the angle can be found as follows:

$$\kappa = \frac{d\theta}{ds} \rightarrow d\theta = \kappa ds \rightarrow \theta = \int \kappa ds + \theta_0$$

Assuming zero initial conditions, substituting the natural equations (3.2), and applying a change of variable:

$$\theta = \int_0^s \frac{1}{c^2} \sigma d\sigma \rightarrow \theta = \frac{s^2}{2c^2} \tag{3.3}$$

By recalling the tangent of a curve C, $dx = cos\big(\theta(s)\big)ds$ and $dy = \sin\big(\theta(s)\big)\, ds$, and using Equation (3.3), the following equations known as Cosine/Sine Fresnel Integrals are shown:

$$x = \int_0^s \cos\left(\theta(\sigma)\right) d\sigma = \int_0^s \cos\left(\frac{\sigma^2}{2c^2}\right) d\sigma$$
$$y = \int_0^s \sin\left(\theta(\sigma)\right) d\sigma = \int_0^s \sin\left(\frac{\sigma^2}{2c^2}\right) d\sigma$$

These integrals are computationally expensive to evaluate as they cannot be evaluated in terms of elementary functions. Examples of the Fresnel Integrals are shown in Figure 3.2 and Figure 3.3.

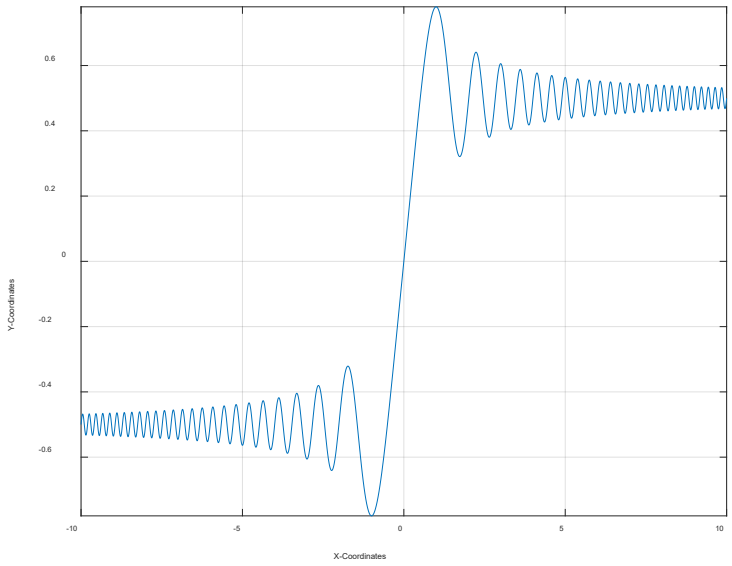

Figure 3.2 Cosine Fresnel Integral

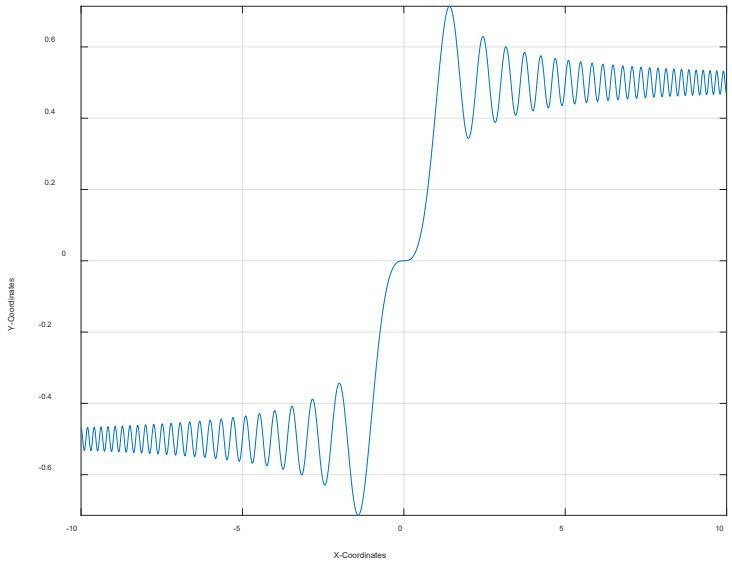

Figure 3.3 Sine Fresnel Integral

When these two are plotted together, a clothoid is formed. An example clothoid is shown in Figure 3.4. Clothoids are often used as the main resource on curve design because clothoids

have a linear curvature profile. So that, when travelled at a constant speed, the curvature varies in proportion to time to provide a smooth ride. An example of the top section of the clothoid is shown along with its curvature profile on Figure 3.5 in which the linear relationship is noticeable.

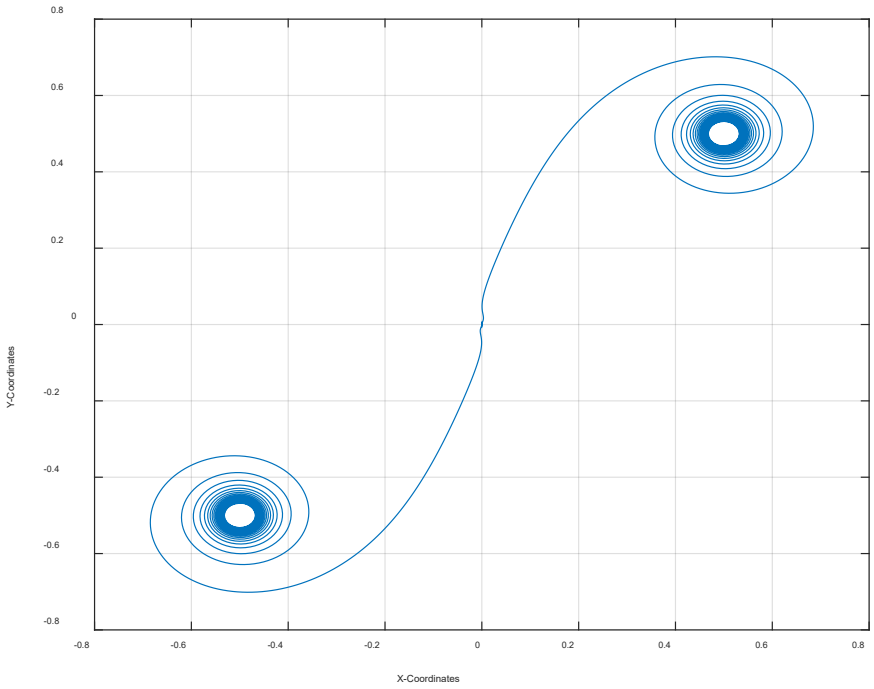

Figure 3.4 Clothoid Example

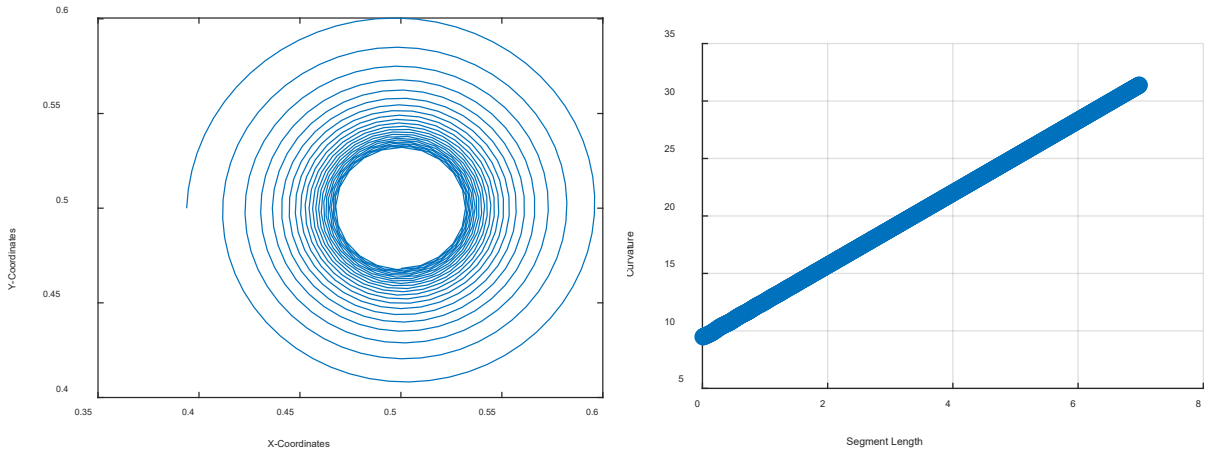

Figure 3.5 Clothoid Section (left) and Clothoid Curvature (right)

3.2 Serret-Frenet Coordinates

This subsection will elaborate on the formulation of the Serret-Frenet formulas and their canonical representations. At this point, it will be assumed that all vectors are functions of segment length s, unless otherwise stated. Considering a curve C be defined by a vector in three-dimensional space with $\boldsymbol{r} = [r_1, r_2, r_3]$. For C, the unit tangent vector can be defined as the unit vector of the derivatives of $\boldsymbol{r}$:

$$\boldsymbol{T} = \frac{\boldsymbol{r}'}{\|\boldsymbol{r}'\|} = \frac{d\boldsymbol{r}}{ds}$$

As it was defined previously, the curvature can be related for small angles through (3) to obtain:

$$\kappa = \left\| \frac{d\boldsymbol{T}}{ds} \right\|$$

The principal normal will be defined as:

$$\boldsymbol{N} = \frac{\frac{d\boldsymbol{T}}{ds}}{\left\|\frac{d\boldsymbol{T}}{ds}\right\|} = \frac{1}{\kappa}\frac{d\boldsymbol{T}}{ds} \rightarrow \frac{d\boldsymbol{T}}{ds} = \kappa\boldsymbol{N} \tag{3.4}$$

Noting that $\|\boldsymbol{T}\| = \boldsymbol{T} \cdot \boldsymbol{T} = 1$, it follows that $\left(\frac{d\boldsymbol{T}}{ds}\boldsymbol{T} + \boldsymbol{T}\frac{d\boldsymbol{T}}{ds}\right) = 0$, this indicates that for the dot product of two unit tangent vectors to be zero, the principal normal has to be zero as well, proving that both unit tangent and principal normal are perpendicular.

Lastly, the binormal product is defined by the cross product of the unit tangent and principal normal:

$$\boldsymbol{B} = \boldsymbol{T} \times \boldsymbol{N}$$

The Serret-Frenet triad (**T**, **N, B**) can now be illustrated with their corresponding plane names in Figure 3.6. Where their respective plane names are T-N: Osculating Plane, N-B: Normal Plane, and B-T: Rectifying Plane.

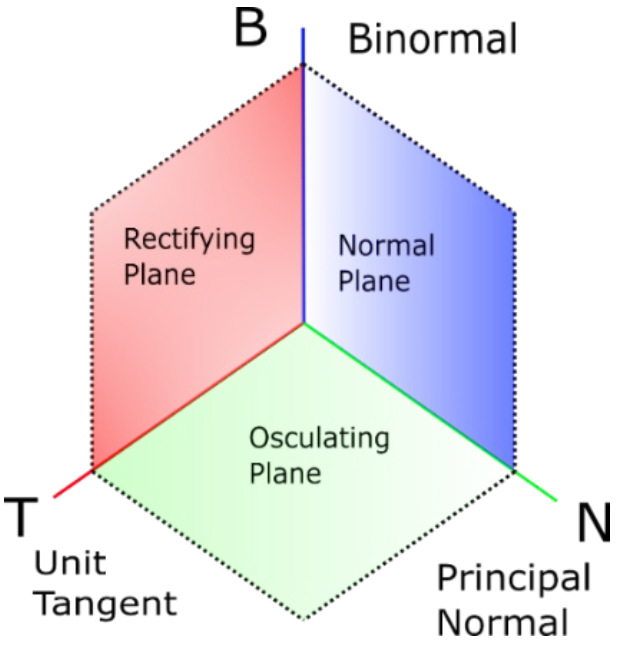

Figure 3.6 Labeled Serret-Frenet Triad

Continuing on the derivation of the Serret-Frenet formulas, similar to the unit tangent vector, the principal normal can be derived as follows:

$$\|\boldsymbol{N}\| = \boldsymbol{N} \cdot \boldsymbol{N} = 1 \overset{\frac{d}{ds}}{\rightarrow} \left(\frac{d\boldsymbol{N}}{ds}\boldsymbol{N} + \boldsymbol{N}\frac{d\boldsymbol{N}}{ds}\right) = 0$$

Implying that the unit principal normal has to be perpendicular to its derivative with respect to s. To find this derivative, first assume that $\frac{d\boldsymbol{N}}{ds}$ is a superposition of the three unit vectors (**T**, **N**, and **B**) with arbitrary constants such that:

$$\frac{d\boldsymbol{N}}{ds} = \alpha\boldsymbol{T} + \tau\boldsymbol{B} + \gamma\boldsymbol{N}$$

Noting that $\frac{d\boldsymbol{N}}{ds} \perp N$, the vector $\frac{d\boldsymbol{N}}{ds}$ is now constrained to the rectifying plane:

$$\frac{d\boldsymbol{N}}{ds} = \alpha\boldsymbol{T} + \tau\boldsymbol{B} \tag{3.5}$$

For some arbitrary constants α and τ. Since the binormal vector is defined to be perpendicular to the osculating plane, the constant τ can be defined as a deviation from the osculating plane. This will be denoted as torsion. Deriving the cross product used earlier:

$$\frac{d\boldsymbol{B}}{ds} = \frac{d(\boldsymbol{T} \times \boldsymbol{N})}{ds} = \frac{d\boldsymbol{T}}{ds} \times \boldsymbol{N} + \boldsymbol{T} \times \frac{d\boldsymbol{N}}{ds}$$

$$\frac{d\boldsymbol{B}}{ds} = \kappa\boldsymbol{N} \times \boldsymbol{N} + \boldsymbol{T} \times (\alpha\boldsymbol{T} + \tau\boldsymbol{B}) = \boldsymbol{T} \times \tau\boldsymbol{B} = -\tau\boldsymbol{N} \tag{3.6}$$

This approach can be repeated with the principal normal to find α:

$$\boldsymbol{N} = \boldsymbol{B} \times \boldsymbol{T}$$

$$\frac{d\boldsymbol{N}}{ds} = \frac{d(\boldsymbol{B} \times \boldsymbol{T})}{ds} = \frac{d\boldsymbol{B}}{ds} \times \boldsymbol{T} + \boldsymbol{B} \times \frac{d\boldsymbol{T}}{ds}$$

$$\frac{d\boldsymbol{N}}{ds} = -\tau\boldsymbol{N} \times \boldsymbol{T} + \boldsymbol{B} \times \kappa\boldsymbol{N} = \tau\boldsymbol{B} - \kappa\boldsymbol{T} \tag{3.7}$$

By comparison of (9) with (7), it is noticed that $\alpha = -\kappa$. Formulas (6), (8), and (9) together are called the Serret-Frenet Formulas:

$$\begin{aligned} \frac{d\boldsymbol{T}}{ds} &= \kappa\boldsymbol{N} \\ \frac{d\boldsymbol{N}}{ds} &= -\kappa\boldsymbol{T} + \tau\boldsymbol{B} \\ \frac{d\boldsymbol{B}}{ds} &= -\tau\boldsymbol{N} \end{aligned}$$

3.3 Acceleration in Serret-Frenet Coordinates

Recall from the chain rule that the following is true when deriving with respect to segment length:

$$\frac{d}{ds} = \frac{dt}{ds}\frac{d}{dt} = \left(\frac{1}{\frac{ds}{dt}}\right)\frac{d}{dt} = \frac{1}{\dot{s}}\frac{d}{dt} \rightarrow \frac{d}{dt} = \dot{s}\frac{d}{ds} \tag{3.8}$$

This equation denotes that, for unity constant velocity, deriving with respect to time or segment length is the same. Equation (3.8) will be used throughout the following formulations rather consistently. Using a curve with the vector representation $\boldsymbol{r} = [r_1, r_2, r_3]$ the following time derivative can be found:

$$\boldsymbol{v} = \frac{d\boldsymbol{r}}{dt} = \dot{s}\,\frac{d\boldsymbol{r}}{ds} = \dot{s}\boldsymbol{T}$$

$$\frac{d^2\boldsymbol{r}}{dt^2} = \frac{d}{dt}(\dot{s}\boldsymbol{T}) = \frac{d}{dt}(\dot{s})\boldsymbol{T} + \frac{d\boldsymbol{T}}{dt}\dot{s} = \ddot{s}\,\boldsymbol{T} + \dot{s}\left(\frac{d\boldsymbol{T}}{ds}\frac{ds}{dt}\right) = \ddot{s}\,\boldsymbol{T} + \dot{s}^2\left(\frac{d\boldsymbol{T}}{ds}\right) = \ddot{s}\,\boldsymbol{T} + \dot{s}^2(\kappa\boldsymbol{N})$$

$$\boldsymbol{a} = \frac{d^2\boldsymbol{r}}{dt^2} = \ddot{s}\,\boldsymbol{T} + \dot{s}^2\kappa\,\boldsymbol{N} \tag{3.9}$$

This formula denotes the classical acceleration of a particle on an arbitrary curve in Serret-Frenet coordinates. By construction, the acceleration vector in Equation (3.9) will never contain a component in the binormal direction. This construction can only be valid for a planar curve assumption [30]. When considering three-dimensional or space curves, the assumption is no longer valid [33].

For most practical purposes, analysis of space curves simplifies the motion of particles to be plane. This is if a curve can be represented through a vector representation where $\boldsymbol{r} \neq \boldsymbol{r}(\kappa, \tau)$ and $\tau = 0$. This assumption becomes invalid when the desired trajectory is a space curve such that the vector $\boldsymbol{r} = \boldsymbol{r}(\kappa, \tau)$.

To implement the general case of acceleration in a space curve, a canonical representation $\boldsymbol{r} = \boldsymbol{r}(\kappa, \tau)$ will be used. The primary advantage is that the resulting vector will be invariant under any coordinate transformation. Starting from Taylor's approximation:

$$\boldsymbol{r}(s) = r(0) + \sum_{n=1}^{3} \frac{s^n}{n!} \left(\frac{d^n r(0)}{ds^n} \right) + o(s^3)$$

Recalling the first derivative with respect to segment length:

$$\frac{d\boldsymbol{r}}{ds} = \boldsymbol{T}$$

The second derivative:

$$\frac{d^2\boldsymbol{r}}{ds^2} = \frac{dT}{ds} = \kappa \boldsymbol{N}$$

Finally the third:

$$\frac{d^3x}{ds^3} = \frac{d}{ds}(\kappa \boldsymbol{N}) = \frac{d\kappa}{ds}\boldsymbol{N} + \kappa \frac{d\boldsymbol{N}}{ds} = \frac{d\kappa}{ds}\boldsymbol{N} - \kappa^2 \boldsymbol{T} + \kappa\tau \boldsymbol{B}$$

Assuming zero initial condition for unit vectors:

$$\boldsymbol{T}(0) = (1,0,0) \quad \boldsymbol{N}(0) = (0,1,0) \quad \boldsymbol{B}(0) = (0,0,1)$$

Applying Taylor's approximation, we can approximate any curve in terms of its invariants forms:

$$\boldsymbol{r}(s) = r(0) + \frac{s^1}{1!}\left(\frac{d^1 r(0)}{ds^1}\right) + \frac{s^2}{2!}\left(\frac{d^2 r(0)}{ds^2}\right) + \frac{s^3}{3!}\left(\frac{d^3 r(0)}{ds^3}\right) + o(s^3)$$

Substituting derivatives:

$$\boldsymbol{r}(s) = r(0) + s\,\boldsymbol{T} + \frac{s^2}{2}(\kappa \boldsymbol{N}) + \frac{s^3}{6}\left(\frac{d\kappa}{ds}\boldsymbol{N} - \kappa^2 \boldsymbol{T} + \kappa\tau \boldsymbol{B}\right) + o(s^3)$$

$$\boldsymbol{r}(s) = \left(s - \frac{\kappa^2 s^3}{6}\right)\boldsymbol{T} + \left(\frac{s^2}{2}\kappa + \frac{d\kappa}{ds}\frac{s^3}{6}\right)\boldsymbol{N} + \left(\frac{\kappa\tau s^3}{6}\right)\boldsymbol{B} + o(s^3)$$

This leads to a canonical representation of a curve:

$$r_1(s) = s - \frac{\kappa^2}{6}s^3 + o(s^3)$$
$$r_2(s) = \frac{\kappa}{2}s^2 + \frac{d\kappa}{ds}\frac{1}{6}s^3 + o(s^3)$$
$$r_3(s) = \frac{\kappa\tau}{6}s^3 + o(s^3)$$

This canonical representation $\boldsymbol{r} = \boldsymbol{r}(\kappa, \tau)$ now contains the necessary parameters to be invariant while being general to the space curve case. However, the acceleration of the particle needs to be found such as in equation (3.9). The necessary steps make use of equation (3.8) denoting the difference between deriving with respect to time and segment length. Assuming only first term approximations, the curve representation is:

$$\boldsymbol{a} = \frac{d^2}{dt^2}(s\boldsymbol{T}) + \frac{d^2}{dt^2}\left(\frac{\kappa}{2}s^2\boldsymbol{N}\right) + \frac{d^2}{dt^2}\left(\frac{\kappa\tau}{6}s^3\boldsymbol{B}\right)$$

The procedure will be available in Appendix A.1, but the resulting acceleration components are found to be:

$$\frac{d^2}{dt^2}(s\boldsymbol{T}) = \ (\ (1 - s\ddot{s}\kappa^2)\ddot{s}\boldsymbol{T} + 2\dot{s}^2\kappa\boldsymbol{N} + \tau\boldsymbol{B}\ )$$

$$\frac{d^2}{dt^2}\left(\frac{\kappa}{2}s^2\boldsymbol{N}\right) = \ m\left(\left((\ddot{\kappa}\dot{s}\kappa + \dot{\kappa})\left(-\frac{1}{2}s^2\right) + (\dot{s}^2\kappa^2 + \kappa\dot{s})(-s)\right)\boldsymbol{T}\right.$$
$$+\left((\ddot{\kappa} - \kappa^3\dot{s} - \kappa\tau^2\,\dot{s})\frac{1}{2}s^2 + 2s\dot{\kappa}\dot{s} + \kappa\dot{s}^2 + \kappa s\ddot{s}\right)\boldsymbol{N}$$
$$\left.+\left(\left(\frac{\ddot{\kappa}}{2}s\dot{s} + \kappa\dot{s}^2 + \frac{\dot{\kappa}}{2}s + \kappa\dot{s}\right)s\tau + \frac{\kappa}{2}s^2\dot{\tau}\right)\boldsymbol{B}\right)$$

$$\frac{d^2}{dt^2}\left(\frac{\kappa\tau}{6}s^3\boldsymbol{B}\right) =$$
$$\left(\left(\frac{1}{6}\,(\ddot{\kappa}\tau + 2\dot{\kappa}\dot{\tau} + \kappa\ddot{\tau})s^3 + \left((\dot{\kappa}\tau + \kappa\dot{\tau})\dot{s} + \frac{\kappa\tau}{2}\ddot{s}\right)s^2 + \kappa\tau s\dot{s}^2\right)\boldsymbol{B}\right.$$
$$\left.-\left(\frac{1}{6}(\dot{\kappa}\tau^2\dot{s} + 2\kappa\tau\dot{\tau}\dot{s} + \kappa\tau^2\ddot{s} - \kappa\tau^2\dot{s}^2 + \dot{\kappa}\tau^2\dot{s} + \kappa\dot{\tau}\dot{s}\tau)s^3 + \kappa\tau^2\dot{s}^2s^2\right)\boldsymbol{N}\right)$$

Comparing the acceleration of the invariant form compared to (11), the level of mathematical complexity and numerical computations increases drastically.

3.4 Bertrand (Parallel) Curves

When two curves share a common principal normal at any of their points, they are called Bertrand curves. If C is a plane (2D) curve, it is always possible to find a curve C' such that C and C' are Bertrand curves. By definition of involute, all curves orthogonal to the tangents of C are located in the plane E [30]. Thus, if E is the plane evolute of C, all involutes C' of E have the same principal normal as C such that these Bertrand curves are parallel with a constant distance separating them.

The relationship between curve C and C' can be expressed by a linear operation with a shared principal tangent such that if $C: r(s)$ and $C': r'(s)$. The following equation can be used:

$$r' = r + d\boldsymbol{N} \tag{3.10}$$

Where:

d = Constant distance in between curves

$\boldsymbol{N}$ = Principal normal function

To verify the existence of a curve C' such that C and C' are parallel, the curvature and torsion of C have to satisfy the following linear relationship with constant coefficients:

$$a_1\kappa + a_2\tau = 1$$

If the previous relation is verified, the parallel curve C' will have the following description where $d = a_1$:

$$r' = r + a_1\boldsymbol{N}$$

In the special case of 2D curves, $\tau(s) = 0$ such that:

$$a_1 = \frac{1}{\kappa(s)}$$

Which can be plugged into Equation (3.10) to obtain the following parallel curve relationship:

$$r'(s) = r(s) + \frac{\boldsymbol{N}(s)}{\kappa(s)}$$

Thus, as long as the curvature of curve C is found, a curve C' can be found to be proportional to the curvature of C and its corresponding principal normal.

The design formulas used in street design as discussed in Chapter 2 employ many independent equations to avoid the analytical construction of these curves.

Chapter 4 MDC Geometric Formulation

This chapter will focus on the geometric formulation of curvature assuming the Serret-Frenet coordinates as established earlier on Chapter 3.

4.1 Vehicle Dynamics and Road Design

Principles of vehicle dynamics were used in order to generate a road mapping technique, which would automatically resolve limitations on vehicle stability and control. It was noted that all vehicle-road interactions are governed by the force generated at the wheels, and all vehicle controls are dictated by the direction and magnitude of friction force [28, 29]. Using Newton's second Law, those forces can be related to the fundamental kinematic constraints of path motion.

A Frenet-Serret reference frame is used and assumed that the vehicle navigates on a 2D Euclidean Space as shown in Figure 4.1.

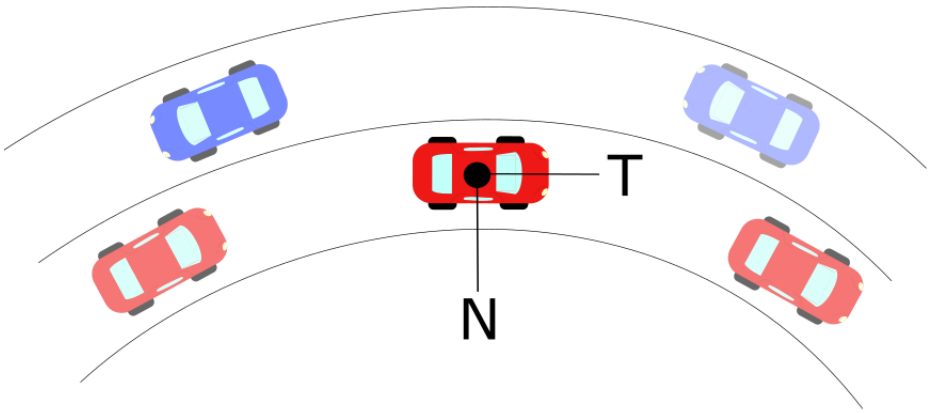

Figure 4.1 Two-Dimensional Serret-Frenet Coordinates Example in Vehicle's Center of Mass

The net acceleration acting on the vehicle at an instant in time is described by the time variance of the path. These limits are related to the acceleration of a vehicle under circular motion, which is denoted from Equation (3.9) again as:

$$\boldsymbol{a} = \dot{v}\,\boldsymbol{T} + \kappa v^2\,\boldsymbol{N}$$

Where:

$\boldsymbol{a}$ = Total acceleration of vehicle (m/s^2)
v = Tangential velocity of vehicle (m/s)
κ = Curvature at an instantaneous point (m^{-1})
$\boldsymbol{N}$ =Normal unit vector
$\boldsymbol{T}$ = Tangential unit vector

Longitudinal accelerations are produced by a net longitudinal force, which either increases or decreases vehicle speed. Lateral accelerations in the normal direction (perpendicular to the velocity vector) do not affect speed and instead turn the vehicle's trajectory. Lateral forces are generated during turns and from road cross-section geometry (superelevation, banks, crowning). Curvature, κ, which is the reciprocal of the radius of curvature, is related to the instantaneous rate of change of the tangential unit vector T with respect to time or distance traveled [31, 32, 34].

The AASHTO standardized road designs provide control for tire-pavement friction using equations for superelevation, crowning, and turn radius based on reasonable limits of vehicle performance in a variety of weather and visibility conditions and driver and occupant comfort [25]. These road design parameters are based on numerous historical studies of driver tolerance for lateral accelerations [35]. Speed limits are controlled on roadways based on measured reductions in friction during wet travel conditions [36]. Hence, the Frenet-Serret coordinates are highly compatible with onboard vehicle systems and prevailing road geometrical design.

Moreover, determining target vehicle path geometries using Frenet-Serret formulation is highly conducive for autonomous vehicle control systems. For example, accelerometers measure acceleration along principal axes; rate transducers record vehicle angular rates of change; gyroscopes identify instantaneous vehicle inclinations; wheel sensors and GPS are useful for

estimating current speeds; and steering wheel rotational sensors can detect wheel steer angles. Wheel steer angles can be related to the instantaneous curvature using Ackerman estimates and corrections for understeer gradient [28, 29]. For vehicles whose orientation closely follows the roadway tangent vector T, the lateral acceleration can provide an additional evaluation of the instantaneous curvature of the vehicle.

A target path was developed with geometric constraints using Frenet-Serret coordinates in local space and then those coordinates were mapped to the surface of the earth using transformation matrices based on GPS coordinates. Thus, curvature can be expressed in a vector form that has a direction parallel to the Normal Unit Vector shown in Figure 1.2. Similarly, a vector perpendicular to the curvature direction will provide a velocity tangent-vector approximation at that point. This velocity vector provides a heading angle to the desired trajectory that is needed to follow a road path.

4.2 Spatial Curvature Formulation

The instantaneous curvature of a geospatial point (deemed "A") was obtained using the spatial coordinates of adjacent points. The technique may be scaled with smaller or larger segmentation, leading to a cheaper computational cost [37].

Let a scalene triangle with corners A, B, and C have a circumscribed circle of radius ρ in Euclidean 2D space as shown in Figure 4.2. The vertices of the triangle are connected using vectors AB, AC, and BC.

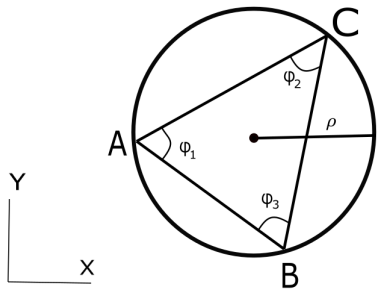

Figure 4.2 Circumscribed Circle in Scalene Triangle

A vector AD, equal to the cross product in between the vectors AB and AC, will be normal to the plane defined by the intersection of AB and AC. The magnitude for vector AD may be identified based on the cross product:

$$\|D\| = \|AB \times AC\| = \|AB\| \, \|AC\| \, \sin \phi_1$$

Let a vector E be the cross product of AD with the vector AB defining this new vector in the direction of e, as shown in red in Figure 4.3. The magnitude of vector E is defined as:

$$\|E\| = \|D \times AB\| = \|AB\|^2 \|AC\| \, \sin \phi_1$$

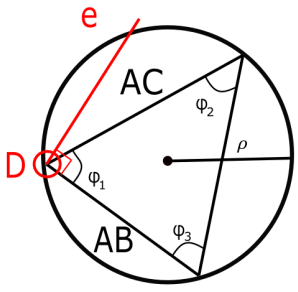

Figure 4.3 First Unit Vector Direction on Triangle

Similarly, let a vector F be the cross product of AD with the vector AC, defining this new vector in the direction of f shown in blue in Figure 4.4. The magnitude of vector f is defined as:

$$\|F\| = \|D \times AC\| = \|AB\|\|AC\|^2 \sin\phi_1$$

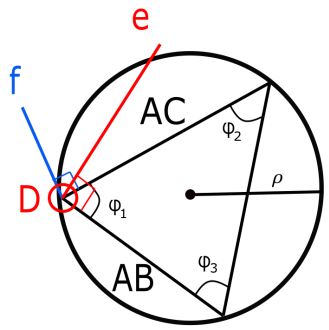

Figure 4.4 First- and Second-Unit Vectors on Triangle

The unit vectors of $\boldsymbol{e}$ and $\boldsymbol{f}$ are defined by the following:

$$\boldsymbol{e} = \frac{E}{\|AB\|^2\|AC\| \sin\phi_1}$$

$$\boldsymbol{f} = \frac{F}{\|AB\|\|AC\|^2 \sin\phi_1}$$

The midsection of any triangle's side intersects with each other at a point P corresponding to the center of the circle inscribing points A, B, and C. These intersecting lines denote two triangles with the same angle ϕ_1 in between the unit vectors and their corresponding midsections, as shown in Figure 4.5.

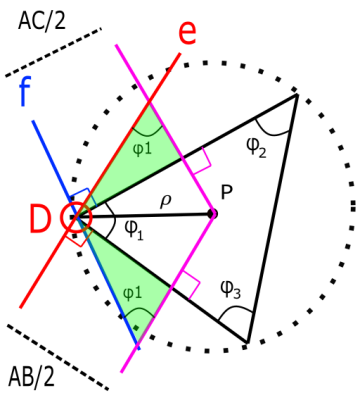

Figure 4.5 Radius of Curvature obtained from Geometric Relationships

From these triangles, the radial vector AP may be described using components along unit vectors $\boldsymbol{e}$ and $\boldsymbol{f}$. As a result, vector AP is described as:

$$AP_1 = \frac{AC}{2\sin\phi_1}\, e = \frac{E}{2\|AB\|^2 \sin^2\phi_1}$$

$$AP_2 = \frac{-AB}{2\sin\phi_1}\, f = \frac{-F}{2\|AC\|^2 \sin^2\phi_1}$$

From our previous definition of the vector AD, it is possible to simplify further:

$$AP_1 = \frac{\|AC\|^2 E}{2\|D\|^2}$$

$$AP_2 = \frac{-\|AB\|^2 F}{2\|D\|^2}$$

With these components, it is possible to obtain the magnitude as follows:

$$AP = \frac{\|AC\|^2 E}{2\|D\|^2} - \frac{\|AB\|^2 F}{2\|D\|^2}$$

$$\rho = \frac{\|AC\|^2 E - \|AB\|^2 F}{2\|D\|^2}$$

Using previous definitions of E and F:

$$\rho = \frac{\|AC\|^2\|AD \times AB\| - \|AB\|^2\|AD \times AC\|}{2\|D\|^2}$$

Using a previous definition of AD, it is possible to obtain the radius of the prescribed circle in terms of only the difference in between points A, B, and C:

$$\rho = \frac{\|AC\|^2\|(AB \times AC) \times AB\| - \|AB\|^2\|(AB \times AC) \times AC\|}{2\|(AB \times AC)\|^2} \tag{4.1}$$

Noting that $\kappa = {}^{1}/_{\rho}$, it is possible to calculate curvature as:

$$\kappa = \frac{2\|(AB \times AC)\|^2}{\|AC\|^2\|(AB \times AC) \times AB\| - \|AB\|^2\|(AB \times AC) \times AC\|} \tag{4.2}$$

This process can be executed for small and large spacing between consecutive points along a curve, as shown in Figure 4.6. As a result, both finely and coarsely sampled data may be utilized to generate geospatial curvature maps.

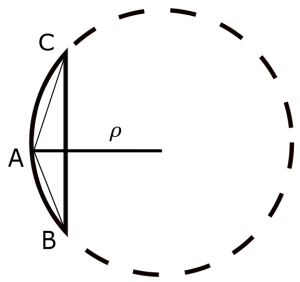

Figure 4.6 Scalene Triangle in Small Arc-Segment

Curvature is related to the second-order differential of position and has strong association with the limits on lateral acceleration. Given that the second order differential equations are used

to condition non-holonomic boundary conditions, determination of instantaneous curvature provides half of the data needed to fully define smooth, continuous, and deterministic target path formulations.

In addition to the second order curvature-based non-holonomic constraint, a 1st-order boundary condition was also identified based on the roadway tangent vector. The roadway tangent vector, T, is determined using an orthogonal phase shift of the curvature vector, such that:

$$T = \frac{(AD \times \kappa)}{\|(AD \times \kappa)\|}$$

Imposing local curvature and tangent coordinate vectors at each geospatial road data-point ensures that the target road path will be consistent with Frenet-Serret formulation. Then, roadway data checking can be performed to ensure that the curvature and point data is consistent with road design parameters and useful for identifying potential errors or skew datasets.

4.3 Segment Length Estimation

Between each consecutive set of geospatial points, the optimized roadway target path length, *s*, may be known to assist with relating vehicle current position to an equivalent position along the roadway target path. This error calculation is essential for determining if the vehicle's trajectory angle, speed, and current position combination put the vehicle at risk of departing the lane or roadway.

The length of each segment of the road path may be identified using the fundamental determination of arc length to radius based on included angle. The arc-length s of a curve is defined as the length traveled by the angle θ along a constant radius ρ:

$$\theta = \frac{s}{\rho}$$

Recalling curvature is the inverse of the radius of curvature, it follows that:

$$\theta = \kappa s$$

A differential form may be used to relate the change in angle to the segment length, *s*:

$$ds = \frac{d\theta}{\kappa}$$

By separation of variables and integration:

$$\int ds = \int \frac{d\theta}{\kappa}$$

Finally, the segment lengths can be determined through numerical integration of the curvature and angle changes:

$$\Delta s = \int \frac{d\theta}{\kappa} \tag{4.3}$$

Chapter 5 MDC Geometric Formulation Implementation

5.1 Road Curvature Decomposition

The curvature formulation shown in Equation (4.2) and road segment calculations shown in Equation (4.3) may be applied to a discrete point cloud collected from a road geometry to determine the instantaneous curvature for every point, except at terminal ends of roadways. An example of the determination of curvature is shown in Figure 5.1.

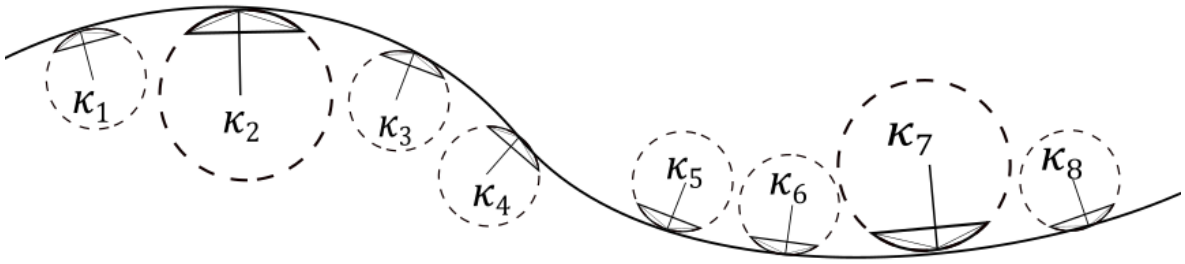

Figure 5.1 Road with Discrete Curvature Sections

When roadway curvature is tangent, the curvature evaluates to zero and is stable; in contrast, the instantaneous radius of curvature of a tangent road is infinite for a 2D Cartesian map, or may be related to the earth radius in 3D maps. An example of the radius vector map is shown in Figure 5.2.

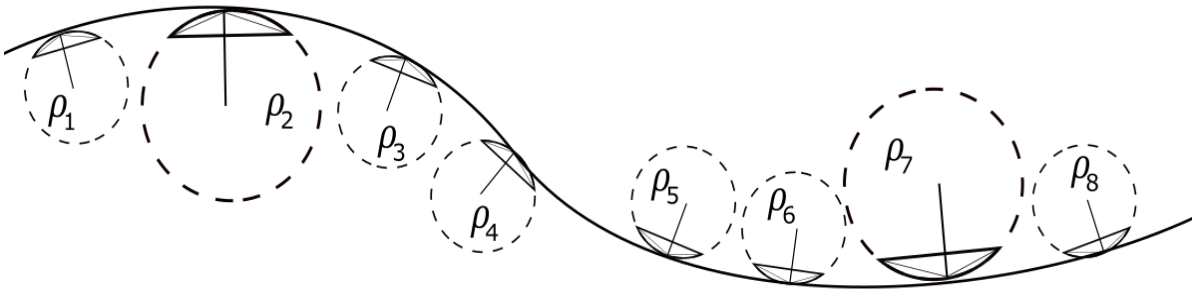

Figure 5.2 Road with Discrete Radius of Curvature Sections

With the use of different technologies such as Aerial Photography, LIDAR scanning, GPS collection, or Road Surveying, it is possible to obtain a geospatial map of roadway centerlines or roadway lane edges (or limits of travelway for rural, unmarked roads). This data may be processed to identify the instantaneous curvature and heading angle of road points, and the segment length connecting consecutive points on the roadway. Continuous mathematical curve formulations may then be used to connect the geospatial point data in accordance with the curvature and heading angle calculations calculated previously. An example of this technique would be a parametric polynomial representation for X and Y road coordinates. An example of the road curvature decomposition scheme is represented in Figure 5.3.

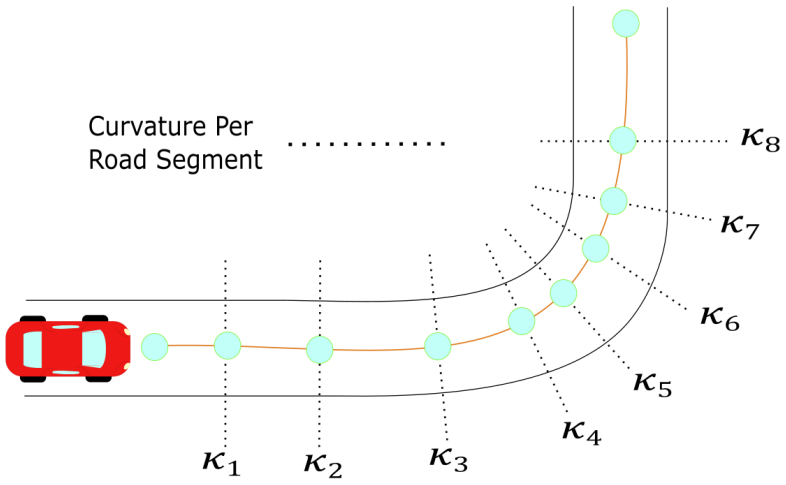

Figure 5.3 Road Curvature Decomposition Example

Using the discrete curvatures calculated for each point, a piecewise-linear, continuous curvature model was developed. Road profiles may be deterministically modeled using relationships for curvature points, heading angles, and roadway segment lengths. Application of this technique to the creation of target paths was deemed the "Midwest Discrete Curvature" (MDC) method. The efficacy of this method in estimating road profiles was evaluated in the following sections.

5.2 Implementation

Typical highway roads are designed based on AASHTO guidelines to provide a natural, easy-to-follow path for drivers, such that the lateral accelerations increase and decrease gradually as the vehicle begins and ends curved road segments [25]. The continuity of the road curvature and adaptability for road tangents using the MDC method were compared by calculating the curvature throughout a road segment constructed consistently with AASHTO design guidelines, then compared using real-world data from satellite photography and point selection as well as GPS data. All source code material can be found in Appendix A.2.

5.2.1 Roadway Decomposition: AASHTO Base Model

This model strictly used AASHTO guidelines to design an ideal highway road for a vehicle traversing at constant 60 mph. The curve consisted of 5 different sections that can be classified as: straight section, entrance transition, constant radius curve, exit transition, and straight section. The road path constructed in accordance with AASHTO Green Book design guidelines is shown schematically in Figure 5.4.

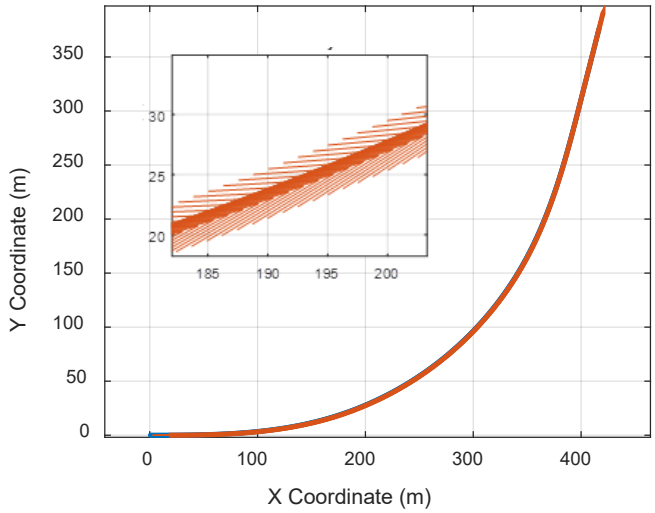

Figure 5.4 Discretized AASHTO Base Model: Road with Velocity Vectors

Applying the MDC approach to this curve, curvature vectors were plotted with respect to the road segments, as shown Figure 5.5. The curvature magnitude and orthogonal heading angle were plotted with respect to road segments to obtain a base curvature profile, as shown in Figure 5.6 and Figure 5.7.

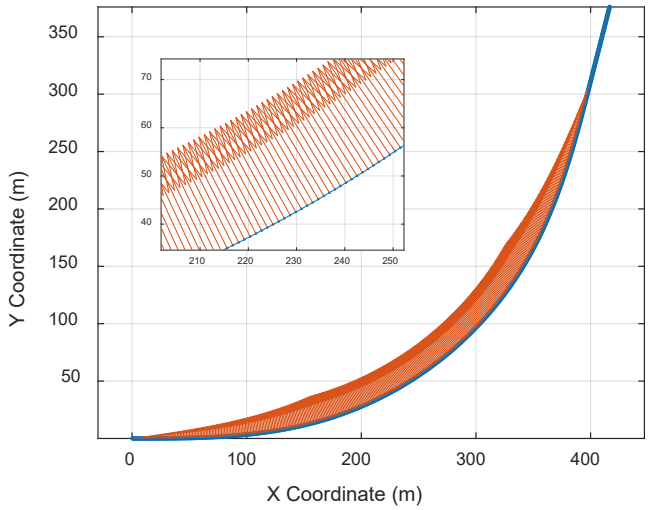

Figure 5.5 Discretized AASHTO Base Model: Road with Curvature Vectors

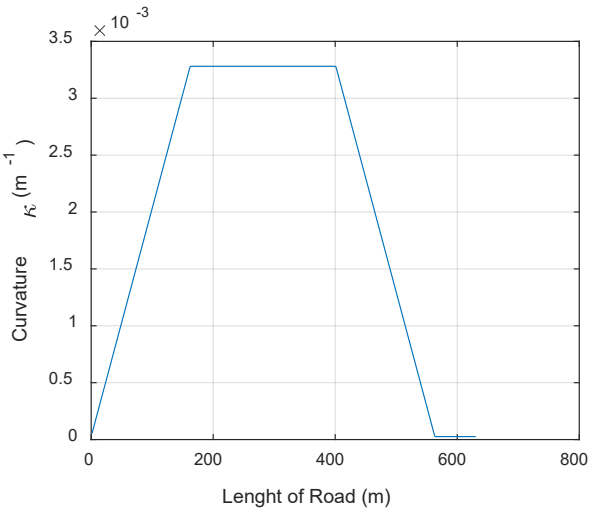

Figure 5.6 Discretized AASHTO Base Model: Curvature κ vs. Cumulative Curve Length

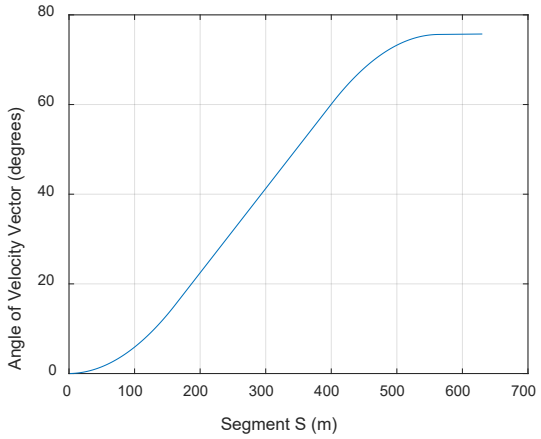

Figure 5.7 Discretized AASHTO Base Model: Orthogonal Phase Shift of Curvature Vector to Calculate Tangent Vectors

5.2.2 Roadway Decomposition: Google Earth Images

Next, the efficacy of the continuous curvature model was evaluated using finely-discretized data collected from aerial photography of a real road segment with a design speed of 60 mph. The road segment is a part of Interstate 80 (I-80), which connects Lincoln and Omaha in Nebraska, as shown in Figure 5.8. The points were picked as close as possible to resemble the road centerline of the highway. The road profile and resulting vectors from applying the discrete geometry approach are shown in Figure 5.9. It is noticeable that the fine discretization of the road points led to some inconsistencies between consecutive tangent vectors. The curvature magnitude with respect to length was also plotted in Figure 5.10, and it was observed that magnitude deviations also increased considerably compared to the AASHTO Green Book theoretical road design model. However, these inconsistencies are strongly related to very short segment lengths relative to curve radii. By using longer segment lengths or averages spanning multiple longitudinal points, results are considerably smoother.

Figure 5.8 Google Earth: I-80 Road Example

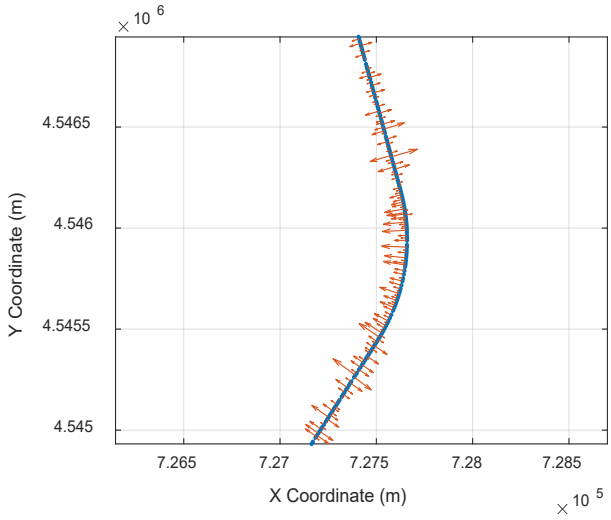

Figure 5.9 Google Earth Model: Road with Curvature Vectors

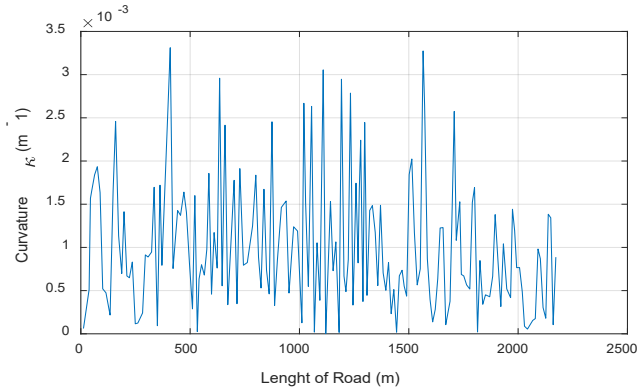

Figure 5.10 Google Earth Model: Curvature κ vs. Cumulative Curve Length

Although curvature magnitudes varied significantly due to short segment lengths, the velocity vector angles were observed to be smooth overall along the segments, as shown in Figure 5.11.

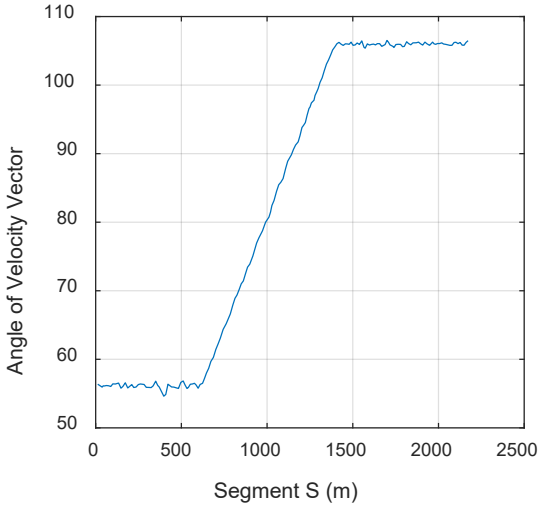

Figure 5.11 Google Earth Model: Orthogonal Phase Shift Approach

Despite noise in the segmented curvature and heading angle discretization, the resulting plot of estimated lane centerline matched the road profile with excellent accuracy. An overplot of the calculated road profile based on the MDC method is shown in Figure 5.12.

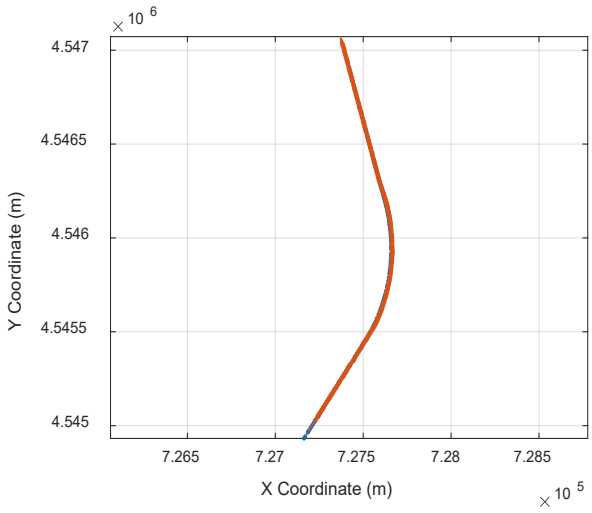

Figure 5.12 Google Earth Model: Road with Tangent Vectors

5.2.3 Roadway Decomposition: GPS Model

The last evaluation of the value of the MDC method utilized GPS data collected while driving along a road with a speed limit of 60 mph. The data was collected with a VC4000 Unit produced by Vericom Computers, Inc., at a frequency rate of 10 Hz. It should be noted that using single-trip GPS data with L1-rated accuracy provides a nominal error estimate of 1.981 m per data point [12], and therefore was the least accurate and smallest dataset evaluated. Nonetheless, applying the MDC method to identify the lane centerline coordinates demonstrated the power of this method in modeling road geometries, as shown in Figure 5.13, Figure 5.14, and Figure 5.15. It is expected that with larger datasets from multiple vehicle trips, highly precise lane centerline data may be identified even when using GPS without differential accuracy estimates.

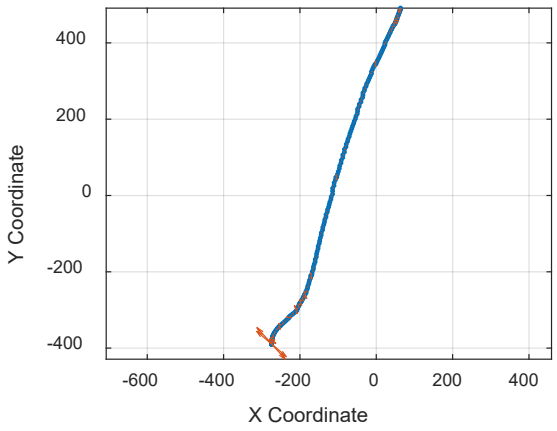

Figure 5.13 GPS Model: Road with Curvature Vectors

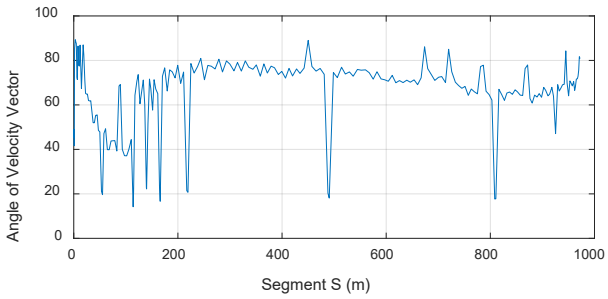

Figure 5.14 GPS Data: Tangent Vector Evaluation using Orthogonal Phase Shift Approach

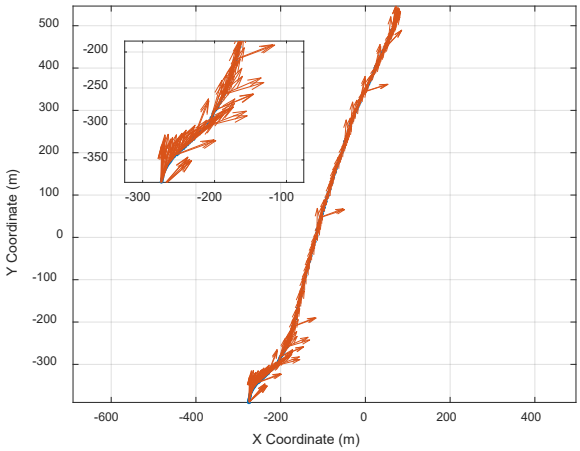

Figure 5.15 GPS Model: Road Construction with Tangent Vectors

5.3 Smoothing Techniques

On trajectory generation, many techniques focused on interpolation restrict the motion of vehicles to maintain a certain level of commodity and stability [38, 39]. Given that the presented approach obtains a heading angle based on discrete data sets for later storage, smoothing techniques may be required to sustain a better approximation of road centerlines while offering a different option to store road decompositions.

Many methods, such as Lagrange's interpolation and Linear Regressions, are used in fitting data for analysis [40]. However, for this application, a local regression with weighted linear least squares was selected to maintain a minimum contribution of the data outliers. This avoids misrepresentation of data often provided by GPS sampling, shown in Figure 5.14 as peaks.

The data was given a span of 15% for outlier acceptance, and the resulting smoothed GPS data is shown in Figure 5.16, in which outliers are not part of the desired prescribed angle values. Smoothed data was plotted and compared to the original GPS model in Figure 5.17. It is noticeable how the data is easily smoothed from a GPS receiver. However, there is an initial swerving behavior on the heading data, which could be due to initialization of the GPS.

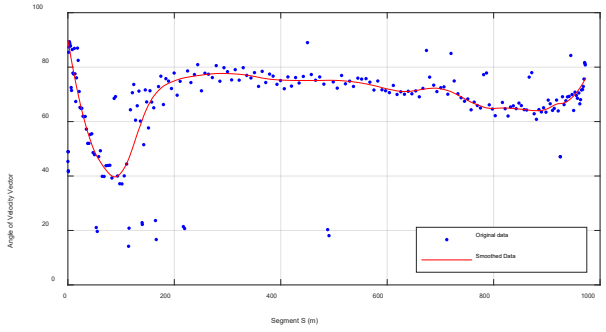

Figure 5.16 GPS Data: Smoothed Tangent Vector Data

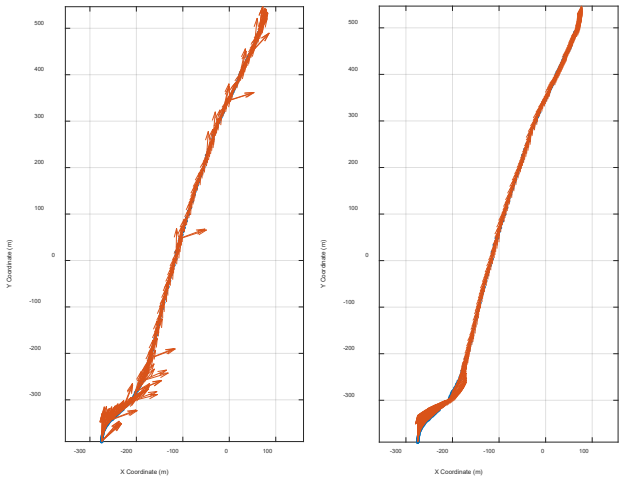

Figure 5.17 GPS Model: Original Data (Left) and Smoothed Data (Right)

The algorithm presented has shown that smoothing data from the heading vector directions is considerably more efficient than smoothing the road curvature magnitude per se. Thus offering a smoothing technique that could potentially improve data for vehicle reference.

5.4 MDC Evaluation on Lane Keeping Assist System

A Lane Keeping Assist with Model Predictive Control (MPC) based on MATLAB was used to simulate the response of a bicycle model vehicle to an input curvature obtained from Chapter 4 [41]. The curvature will come from the same Google Earth Model explored earlier in section 5.2.2. Figure 5.18 previews the road with its tangent heading angles on the left, and its curve magnitude representation on the right.

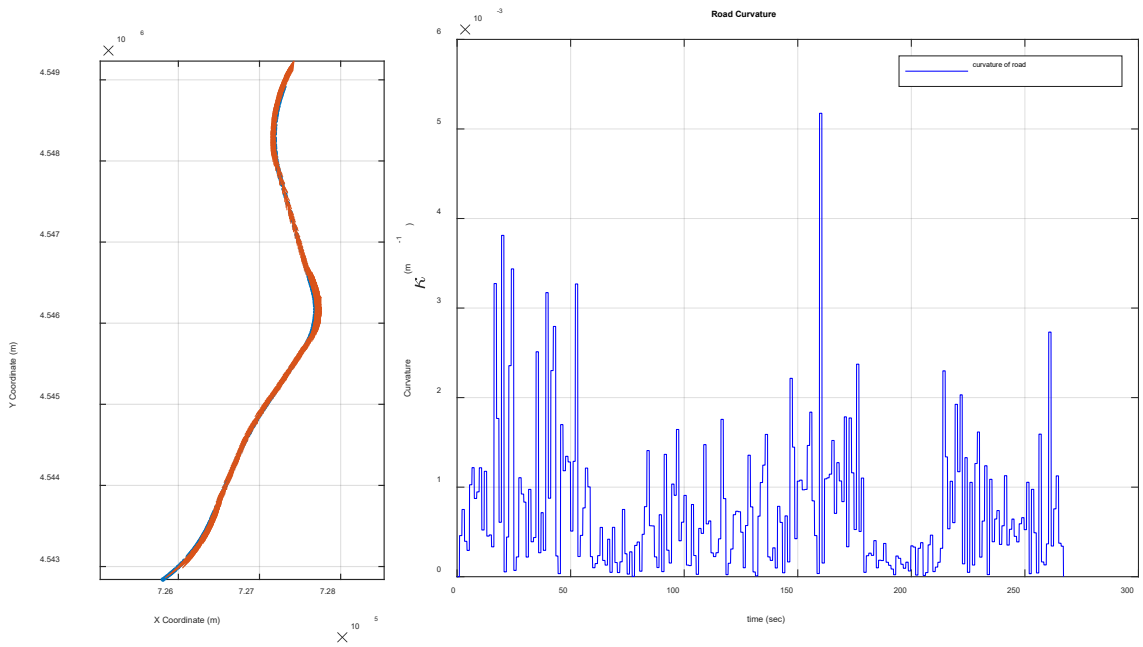

Figure 5.18 Google Earth Model: Heading Angle (Left) and Curvature Magnitude (Right)

Two different run models were explored with an input velocity of 25 m/s (~55 mph), for traveling a distance of 6.7 km lasting about 270 seconds (~ 5 min); the only differing input was their prediction horizon. Prediction Horizon defines the amount of road curvature preview available to maneuver it. This is analogous to the amount of eyesight a person has when viewing a road while driving it. In conventional ADAS, the prediction horizon is limited because of the limited range sensors have. To represent this in the first model, a prediction horizon of 10 steps was used. The results of the model are shown in Figure 5.19, in which the outputs include lateral deviation, relative yaw angle, and steering angle.

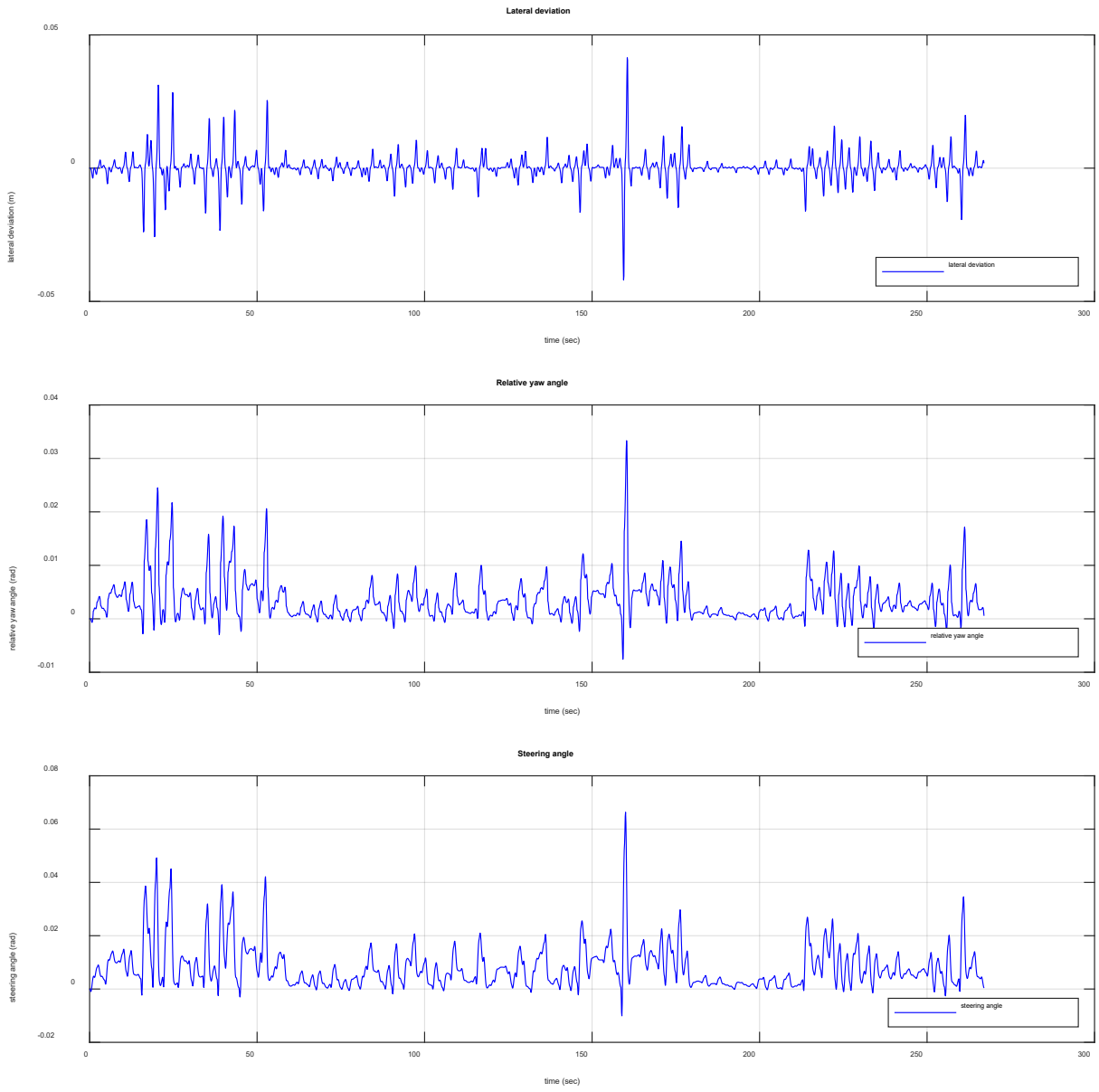

Figure 5.19 Lane Keeping Assist Results with Prediction Horizon of 10 Steps

The MDC explored on Chapter 4 works under the assumption that large road sections can be transmitted to the vehicle, thus amplifying the prediction horizon to any desirable quantity independent of sensor range. A tenfold amplification was used for the second test giving 100 steps with the results shown in Figure 5.20. Comparing both Figure 5.19 and Figure 5.20, the

relative yaw angle and steering angle inputs of the Lane Keeping Assist is relatively similar. However, the lateral deviation decreases considerably for the case of a higher prediction horizon. This shows how the availability of increasing horizon with the MDC method, in fact increases Lane Keeping Assist performance.

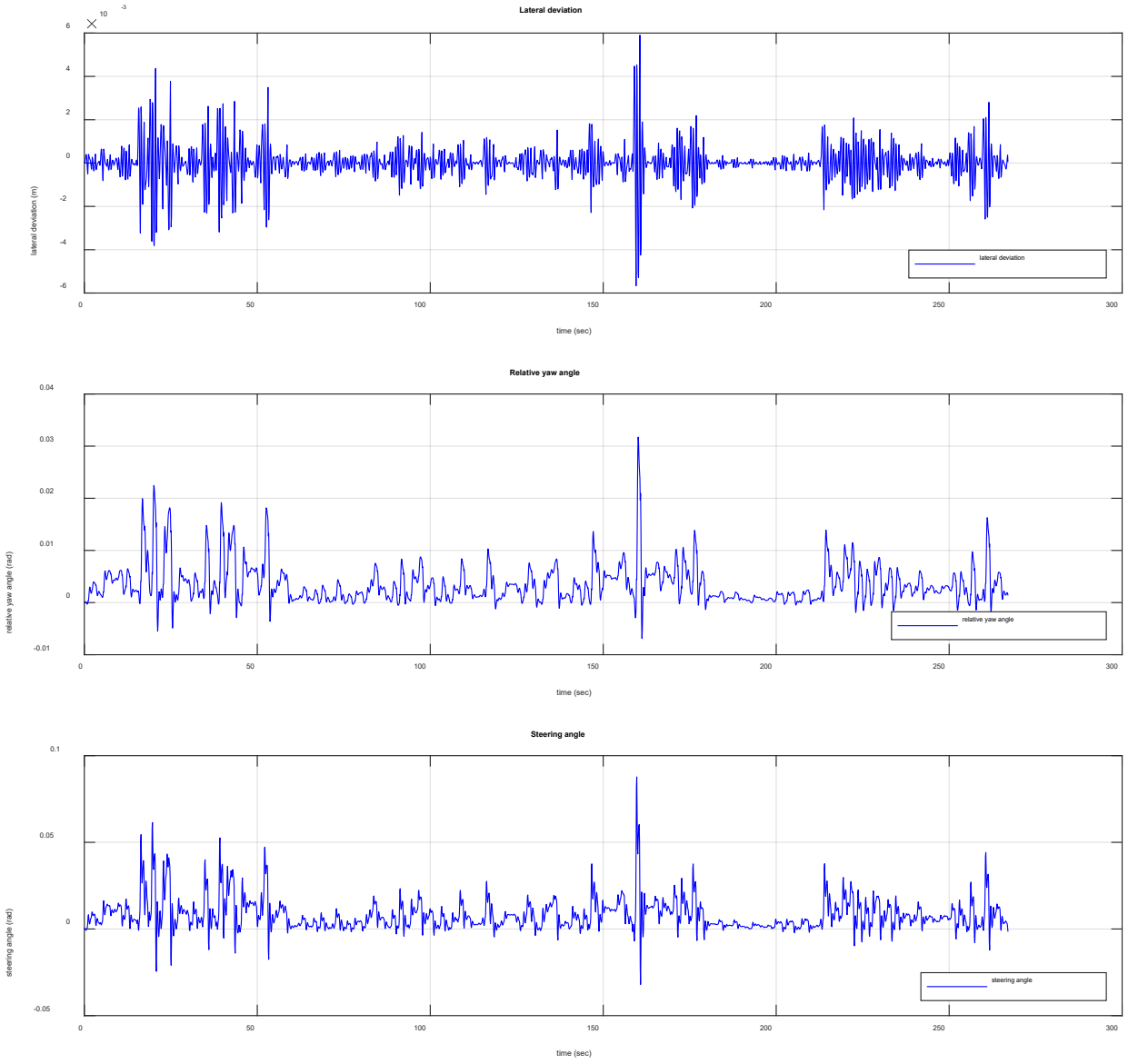

Figure 5.20 Lane Keeping Assist Results with Prediction Horizon of 100 Steps

Chapter 6 MDCR Optimization Math

As it was demonstrated in Chapter 4, the tangent to the curvature vectors can be used as reference for vehicle heading angle at an instantaneous point in time [24]. Such that heading tangent angle for vehicles can be obtained from a curvature profile. For this reason, being able to provide vehicles with pre-determined curvatures poses a new guiding factor for autonomous vehicle navigation such that vehicles will have a baseline road profile to navigate in the absence of reliable sensor information. This chapter will develop a reference scheme based on the MDC technique developed earlier and will add a component distinctive of road design standards.

6.1 Preliminaries

In previous chapters, road curvature profiles have been explored as an efficient method to obtain road tangent vectors coming from multiple sources such as GPS data and Google Earth coordinates [24]. In conjunction with V2I technology, road profiles can be stored and referenced on autonomous vehicle navigation. In this section, the necessary research will be presented to move on with the proposed reference scheme.

6.1.1 Baseline Static Reference

This baseline refers to a reference that is always available to be sent for vehicles. It does not change when conditions on the road can indeed change. In general, given discrete road coordinate data, reference tangent angles for vehicles can be provided. This reference is purely geometric, which often limits the real-time implementations. An example formed with GPS data is shown in Figure 6.1 and Figure 6.2. This method offers a sensor independent reference path for vehicles to follow under conditions of low sensor resolution. However, the baseline static reference does not account road nor dynamic factors such as friction or vehicle velocity in which heading angles might need to be modified to comply with driving conditions.

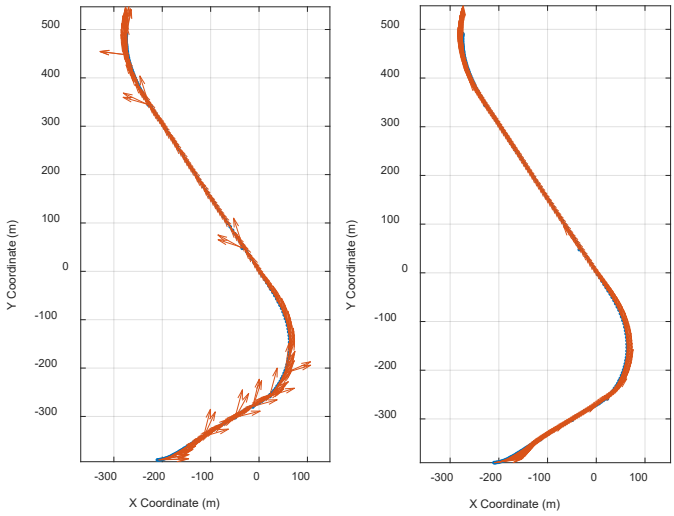

Figure 6.1 Baseline Static Reference GPS Data Example with Unsmoothed Data (left image) and Smoothed Data (right image).

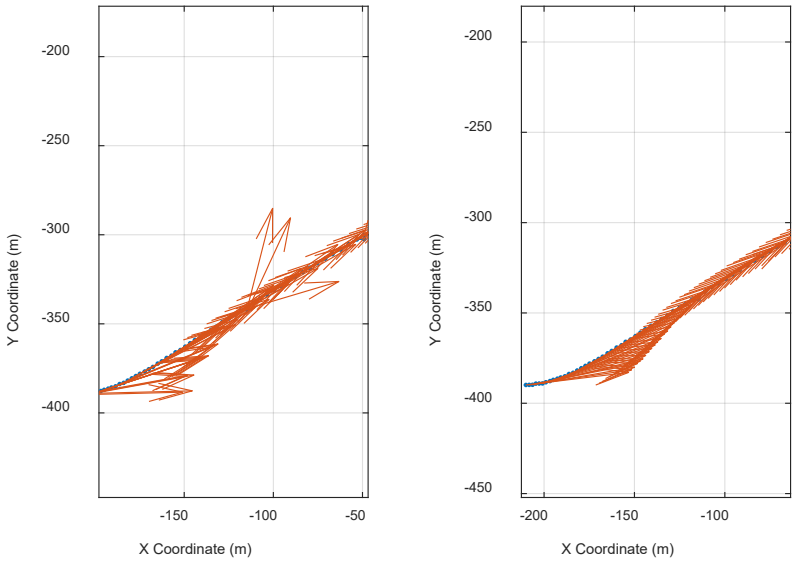

Figure 6.2 Baseline Static Reference Zoomed in GPS Data Example with Unsmoothed Data (left image) and Smoothed Data (right image).

6.1.2 Semi-Dynamic Road Reference

This road reference is the first proposed, which considers changes in road friction due to weather conditions. For example, when rain is present, the coefficient of rolling friction in between the tire and roads does change. To address this, the road reference will be adjusted and stored on RSUs for availability. Recalling equation (2.1), friction plays a role with the instantaneous curvature and current vehicle velocity. This paper will detail out how to create a semi-dynamic road reference based on particle dynamics (independent of vehicle specific parameters), to improve the baseline static reference. The obtained semi-dynamic road reference will serve as the speed a vehicle should traverse a curve under arbitrary friction changes, along with the already developed baseline static reference.

6.1.3 Dynamic Road Reference

This second road reference considers all previous models along with vehicle specific parameters such as understeer gradients, track width, and vehicle length. Current RSU research has shown that it is possible to exchange data wirelessly such that it will be assumed for this model. Thus, this dynamic road reference assumes that the RSU can obtain such information from a vehicle. With a model combining equations (2.1) and (2.2), the dynamic road reference takes an input from vehicles along with the previous models and creates a reference velocity path and heading angle for safe travel.

6.2 Mathematical Road Curvature Models

Proceeding with the proposed Semi-Dynamic and Dynamic Road References, a mathematical curvature model needs to be established in accordance to the trajectory requirements of autonomous vehicles. To construct the road reference models, it is first

necessary to understand the curvature information obtained from section 2.3.1 and establish a connection in between this baseline and the subsequent models.

Trajectory generation models in autonomous vehicle research often take the quintic polynomial approximation [42, 43]. These quintic models often formulate the curvature or the position in 2-dimensional Euclidean space of a car such that:

$$\kappa(s) = a_0 + a_1 s + a_2 s^2 + a_3 s^3 + a_4 s^4 + a_5 s^5$$

Studies of horizontal curves as shown in Figure 1.1 indicate that most designs are made of curvatures with constant, and/or linear slopes. Differing from driving considerations, quintic polynomials do not offer the curvature characteristics for which a road is designed. To create a road reference that relates road to vehicle parameters, the following mathematical curvature models are proposed:

Piecewise linear model 1:

$$\kappa_1 = \left(\frac{x_5}{x_2 - x_1}\right)(s - x_1)[\varphi(s - x_1) - \varphi(s - x_2)] + x_5[\varphi(s - x_2) - \varphi(s - x_3)] + \left(\left(\frac{x_5}{x_4 - x_3}\right)(-s + x_3) + x_5\right)[\varphi(s - x_3) - \varphi(s - x_4)] \quad \text{M.1}$$

Piecewise linear model 2:

$$\kappa_2 = \left(\frac{x_4}{x_2 - x_1}\right)(s - x_1)[\varphi(s - x_1) - \varphi(s - x_2)] + x_4[\varphi(s - x_2) - \varphi(s - x_3)] \quad \text{M.2}$$

Where:

$\varphi(s-a)$ = Unit Step Function with a shift of $a \in \mathbb{R}$
x_i = Values that determine the shape of the curvature function with $i \in [1,5]$

Both models were mathematically designed to be continuous for all s but are only intended to model a single road section of $s \in [s_{initial}, s_{end}]$ which should contain only one turning motion. M.1 offers the flexibility of having its x_i parameters to be easily identified as basic geometric properties of a trapezoid. M.2 was designed to be the simplest model available that can represent roads with minimum use of x_i parameters. M.1 is illustrated in Figure 6.3 for the general model κ_1 along s with parameters x_i. M.2 can be easily visualized the same as M.1 but without the last downward slope section.

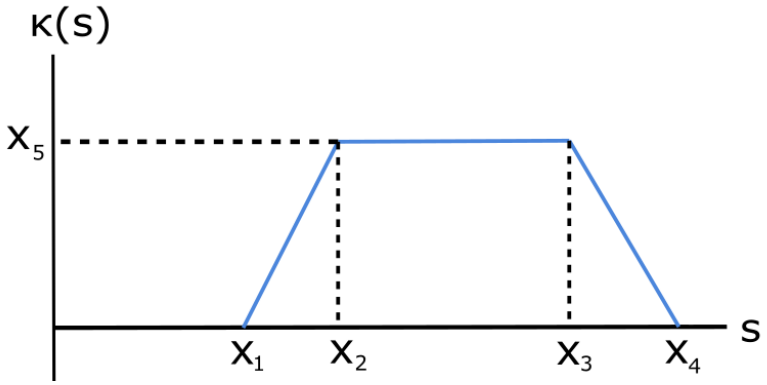

Figure 6.3 General Curvature M.1, with Design Values

A detailed elaboration of the trapezoid math is available in Appendix A.3.

6.3 Road Guidance Optimization Problem Formulation

All road reference models proposed need to shape the previous curvature models to discrete curvature data through least squares minimization. With the mathematical curvature

models established, two optimization algorithms are proposed to create the semi-dynamic road reference and dynamic road reference.

6.3.1 Least Squares Fitting

The models are subject to an unconstrained Least Squares Error - Minimization problem, denoted as Pr.1, such that:

$$\min_{x} \|\kappa_m(s) - \hat{\kappa}[s]\|^2 \qquad \text{Pr.1}$$

Where:
$\kappa_m(s) =$ Road curvature model m in terms of segment s and constants $x_1, \dots, x_n$
$\hat{\kappa}[s] =$ Discrete road sampled curvature data

This curve fitting minimization problem denoted as Pr.1 is used to test analytical curvature models M.1 and M.2. It is important to note that the models κ_m can be either linear or non-linear. Pr.1 will focus on representing models $\kappa_m(x_1, \dots x_n)$ as a representation of any generic road data input. The next two sections 6.3.2 and 6.3.3 address a second optimization problem denoted as Pr.2 that will depend on the objective function selection.

6.3.2 Semi-Dynamic Road Reference Numerical Optimization Routine

As a vehicle increases speed, the friction developed in between the road and the tires decreases. This decrease in available friction increases the likelihood of a vehicle departing from the road on a curve. Addition of superelevation on roads reduces the portion of forces due to longitudinal accelerations. When the coefficient of friction decreases due to weather factors such as the rain or snow, the available velocity for a vehicle to maintain stability decreases as well. The Green Book has taken into consideration analysis on friction levels on wet surfaces [25]. However, variability in tire conditions and road degradation considerably affect vehicle handling during cornering.

For these reasons, the semi-dynamic road reference model uses Equation (2.1) as an objective function to establish a relationship between current Green Book standards and instantaneous vehicle curvature. In general, there exists velocity ranges, and steering wheel angles such that extra constraints could be added to Pr.2 in the following manner:

$$\min_{y} \frac{y_1^2}{g}\kappa_m(s) - \frac{\mu + 0.01e}{1 - 0.01\mu e} \qquad \text{Pr.2}$$

$$subject\ to:$$

$$v_{min} < y_1 < v_{max} \quad \& \quad \delta_{min} < y_2 < \delta_{max} \qquad \text{C.1}$$

In practice, these constraints will be inactive for the most part. In fact, for this Semi-Dynamic routine, steering angle does not play a role in the optimization routine Pr.2. However, these can be defined by regulatory standards and are important to denote because dynamics' formulas do not account for speed limit regulations. In other words, going at a slow speed such as 45-mph on a 75-mph road will not violate dynamic stability requirements but might pose a risk for upcoming traffic. The objective function minimization at each road segment iteratively is presented with the pseudo-code in Figure 6.4.

- Load Road Data
- Calculate Discrete Curvature for Road Data $\kappa[s]$
- Select a Curvature Model M.X
  i.e. M.1
- Solve Least Squares Optimization of M.X with Curvature Data $\hat{\kappa}[s]$
  i.e. Pr. 1
- Obtain an Optimized Curvature $\kappa_{Opt}(s)$
- Define Road Parameters as appropriate
  i.e. friction

for Step = Initial Step: Total Steps in $\kappa_{Opt}(s)$

- Objective Function for Pr. 2
  i.e. Equation 1
- Define Constraints
  i.e. C.1 Constraints
- Apply Non-Linear Solver to find Optimal driving parameters
  $[v_{Opt}]$=Non-linear-Optimization-Solver(Eq.1,C.1)

end

Figure 6.4 Semi-Dynamic Road Reference Numerical Optimization Routine Pseudo-Code

6.3.3 Dynamic Road Reference Numerical Optimization Routine

Vehicle dynamic stability is related to curvature models through (2.1) and (2.2). For this routine, Pr. 2 will use Equation (2.2) as an objective function to optimize specific-vehicle performance under different road conditions. Where vehicle velocity $v = y_1$ and steering wheel angle $\delta = y_2$ for every segment on $\kappa_m(s)$ such that:

$$\min_{y} \; y_2 - (57.3L + \eta\, y_1^2)\, \kappa_m(s) \qquad \text{Pr.2}$$

$$subject\ to:$$

$$v_{min} < y_1 < v_{max} \quad \& \quad \delta_{min} < y_2 < \delta_{max} \qquad \text{C.1}$$

$$\frac{y_1^2}{g}\kappa_m(s) - \frac{\mu + 0.01e}{1 - 0.01\mu e} = 0 \qquad \text{C.2}$$

Pr. 2 is a non-linear constrained optimization problem in which the vehicle parameters (L, η) and road parameters (μ, e) are regarded as constants for any generic road/vehicle. In this second routine, the particle dynamics' equation is set as a constraint. Pr. 2 will find the optimized combination for both traveling velocity and wheel angle that uses the model $\kappa_m(s)$ as part of their process. It is important to note that Pr. 2 must be solved iteratively as the vector κ_m contains many values. The following pseudo-code was created to illustrate this proposed optimization routine.

- Load Road Data
- Calculate Discrete Curvature for Road Data $\kappa[s]$
- Select a Curvature Model M.X
  i.e. M.1
- Solve Least Squares Optimization of M.X with Curvature Data $\hat{\kappa}[s]$
  i.e. Pr. 1
- Obtain an Optimized Curvature $\kappa_{Opt}(s)$
- Define Vehicle/Road Parameters as appropriate
  i.e. friction, gravity

for Step = Initial Step: Total Steps in $\kappa_{Opt}(s)$

- Objective Function for Pr. 2
  i.e. Equation 2
- Define Constraints
  i.e. C.1 and C.2 Constraints
- Apply Non-Linear Solver to find Optimal driving parameters
  $[v_{Opt}]$=Non-linear-Optimization-Solver(Eq.2,C.1,C.2)

end

Figure 6.5 Dynamic Road Reference Numerical Optimization Routine Pseudo-Code.

Chapter 7 MDCR Optimization Implementation

The codes and datasets used in this section can be found in Appendix A.4. Chapter 7 uses all the proposed mathematical models and routines from subsections 6.1-6.3 and implements them accordingly.

7.1 Least Squares Fitting on M.1 and M.2

To exemplify Pr.1, a sample curvature data was generated that resembles a compound curve (shown in Figure 7.1) with Gaussian noise added. These models will be defined with starting points $x_i \forall\ i \in [1, n]$ where n defines the number of shape function values. It is crucial to note, that for convergence of Pr.1 with any model M.X, the initial value guesses must not be repeated and be incrementing values so that $x_j > x_{j-1}\ \forall\ j \in [2, n-1]$, excluding x_n because it is dependent on the maximum curvature, rather than segment length.

A sample optimized M.1 is shown in Figure 7.1 along with Table 7.1 having its corresponding coefficients. The behavior of M.1 under many datasets was tested, and it is noticeable how curves that lack a downward slope at the final segment length are still able to be modeled by M.1. The model can do that by providing a combination of x_3 and x_4 so that the section is approximately constant.

Table 7.1 Values obtained for M.1.

Value	Initial Guess	Optimized
x_1	0.9	0.4785
x_2	2.9	6.0857
x_3	6	4.7662
x_4	10	245.0074
x_5	$\max(\kappa)$	11.213

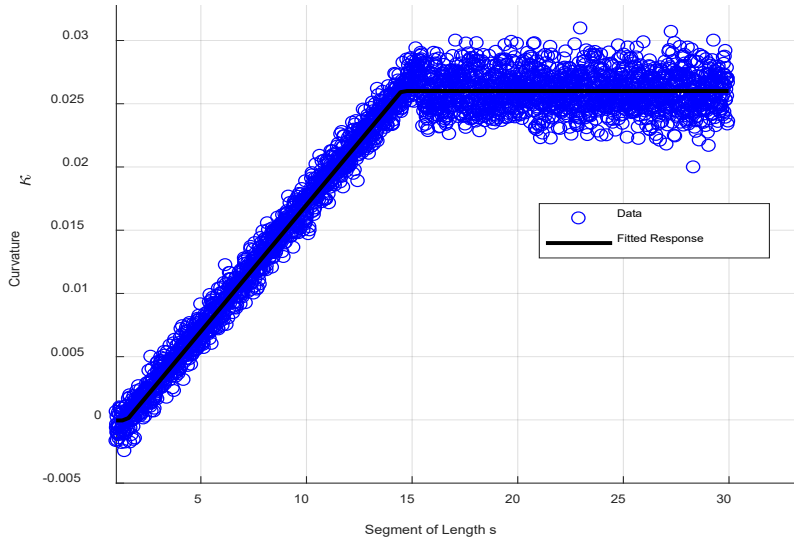

Figure 7.1 Optimization of Curvature M.1

A sample optimized M.2 is shown in Figure 7.2 along with Table 7.2 having its corresponding coefficients. The behavior of M.2 was not stable enough to fit the data properly. The reason being that x_2 and x_3 converge to relatively close values which cause this model to not fit the data. Thus, M.2 was not studied for the optimization routines shown in the next subsections.

Table 7.2 Values obtained for M.2.

Value	Initial Guess	Optimized
x_1	0.9	0.4689
x_2	2.9	5.9741
x_3	6	6.0000
x_4	$\max(\kappa)$	10.9656

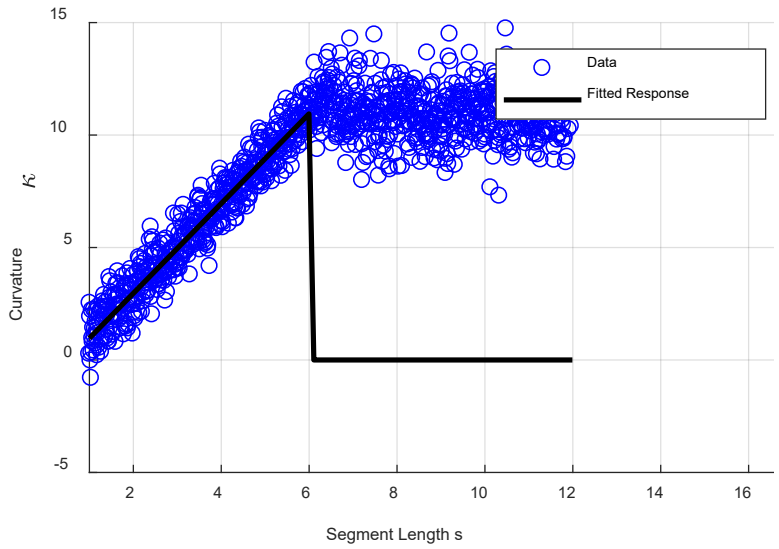

Figure 7.2 Optimization of Curvature M.2.

7.2 Selected Model M.1 on Datasets

Least Squares Fitting was applied to a mathematically designed curve that is compliant with the Green Book standards. Figure 7.3 shows the curve with its corresponding reference tangent angles and the obtained fitted curvature model. It is noticeable that the mathematical construction of this road will lead to a perfect fit with no noise. The primary purpose of this was to study the robustness of the routine to fit curvature models under different initial guesses. Since this curve is mathematically compliant to road designs, the curvature profile represents a transition spiral, followed by a constant radius curve, an exit transition spiral, and a last segment of straight line.

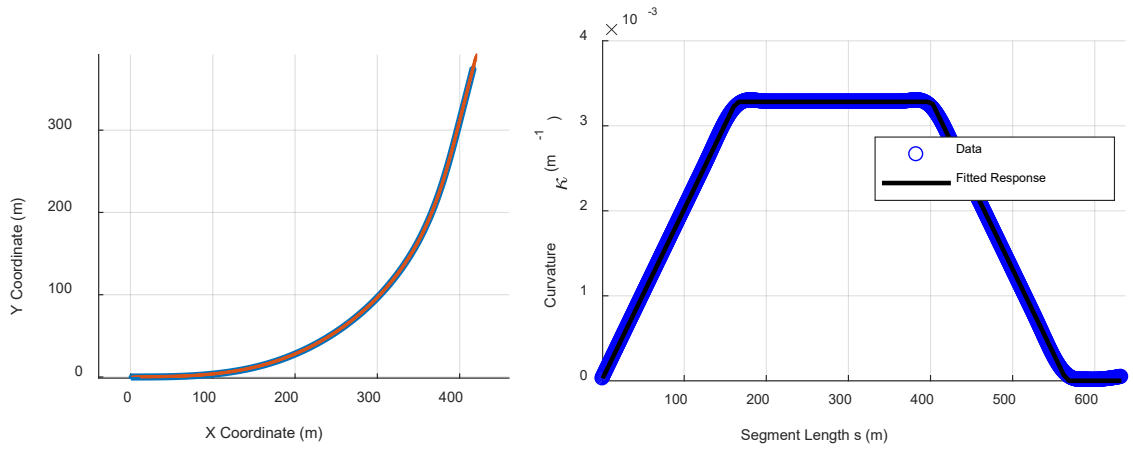

Figure 7.3 Green Book Road with Tangent Angles (left image) and Fitted Curvature Profile (right image)

With the model M.1 selected, the remainder of the routine uses appropriate data sets to obtain road representations for both semi-dynamic and dynamic routines. In general, there is a vast availability for road recognition that estimates road centerlines from satellite images, GPS data, or any road coordinate collection system [44, 45]. Based on maximum Green Book cornering limits, it is possible to sample a discrete number of points of about 12 points per km or about 1 every 1/12 of a km [25]. For this paper, a sample reference obtained from a Google Earth Satellite software is utilized that satisfies these requirements.

The sample-selected road comes from highway I-80 connecting the cities of Lincoln and Omaha in Nebraska (USA). Sample road is shown in Figure 7.4 (left) with its appropriate heading angles as calculated from previous studies [24]. The curvature magnitude per segment length is shown in the right of Figure 7.4. Changes in amplitude for this curvature trace come from how the road curves. In practice, straight roads will never achieve zero curvature, which led to values within a range of $\pm$1e-3 be considered as straight segments.

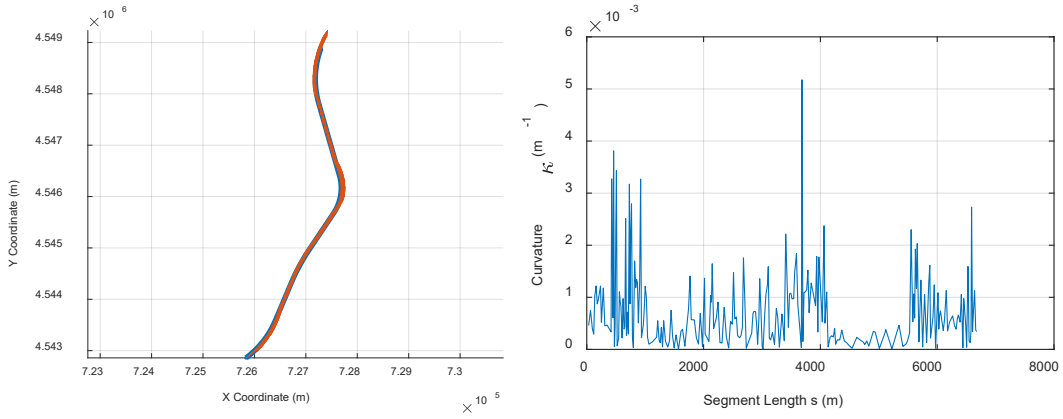

Figure 7.4 Google Earth Road with Tangent Angles (left image) and Fitted Curvature Profile (right image).

Typical highways are built in conjunction to the available landscape space for minimizing disruption of natural areas. This permits roads discretization on sequential curvature changes as shown in the right of Figure 7.5. Multiple curvature changes are to be found within long road sections requiring them to be separated into basic curvature segment components. In this dataset, the highest curvature section was found and analyzed. This section was selected as the highest average on curvature per rate of change in segment length. Alternative options include selecting initial guesses spanned on a user-selected segment length. The highest curvature road segment is shown in Figure 7.5 (left) and a smoothed data portion of its curvature profile in Figure 7.5 (right).

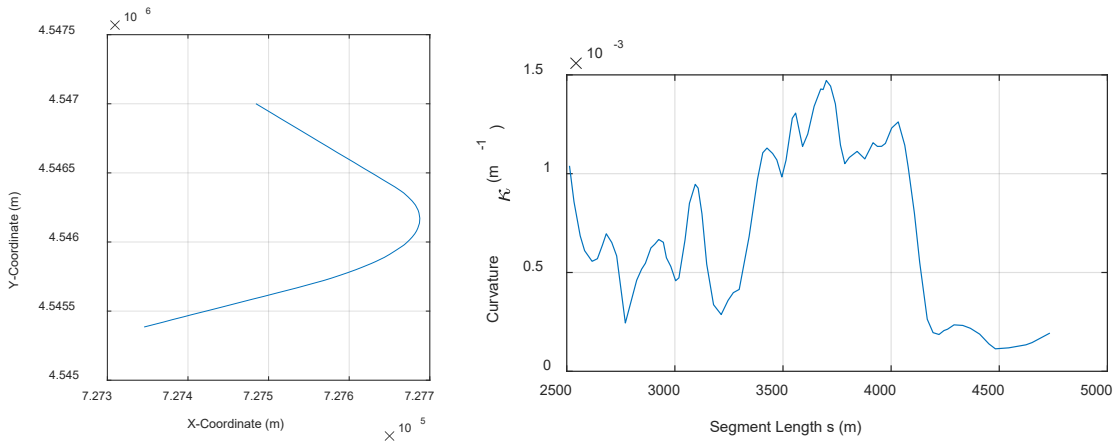

Figure 7.5 Reduced Google Earth Road (left image) and Corresponding Curvature Profile (right image)

With the curvature profile obtained, Least Squares Fitting yields the curvature model to be used in the two subsequent optimization routines. This fitted model is shown in Figure 7.6 with its corresponding initial guesses in Table 7.3. It is noted that the optimized values do not vary significantly. This indicates that the selection of initial guesses is crucial for the overall shape of the optimized fitted curvature model.

Table 7.3 Values obtained for Curvature Model

Value	Initial Guess	Optimized
x_1	3200	3176.83
x_2	3500	3503.55
x_3	4000	4017.41
x_4	4200	4204.49
x_5	max(κ)	1.245e-3

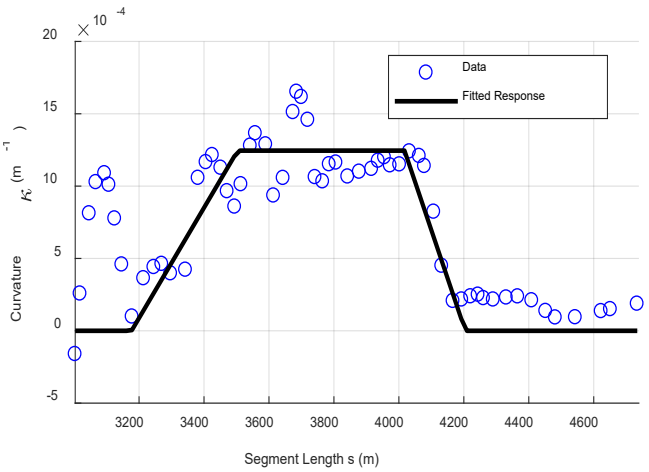

Figure 7.6 Fitted Response to Curvature on Road Section

7.3 Semi-Dynamic Road Reference Results

This optimization routine is based from Figure 6.4 and creates a reference velocity profile dependent on the curve's shape and the current road friction levels. Only road parameters influence what the reference speed of a vehicle should be. Table 7.4 offers a summary of the parameters given to the Semi-Dynamic Road Reference routine.

Two different road inputs were explored, a mathematically designed road based off the Green Book and a Google Earth data road. The results from the Pseudo-Code in Figure 6.4 were implemented in MATLAB and the results of both data samples are shown below in Figure 7.7. The road profile for the Green Book road is compliant to current street designs, such that the velocity required to go through it should be constant until it reaches the exit curve into a straight line. As expected for the Green Book model (Figure 7.7 top), a moderate level that considers the current road elevation and friction is first obtained followed by an increase on the exit of the curve. For the Google Earth data (Figure 7.7 bottom), the road starts with an initial velocity predefined by the previous road factors, such that when the vehicle enters the curve, a decrease is expected, followed by the same behavior as in the Green Book Model.

Table 7.4 Variables obtained for Curvature Model

Road Parameter	Quantity
e	6%
μ	0.3

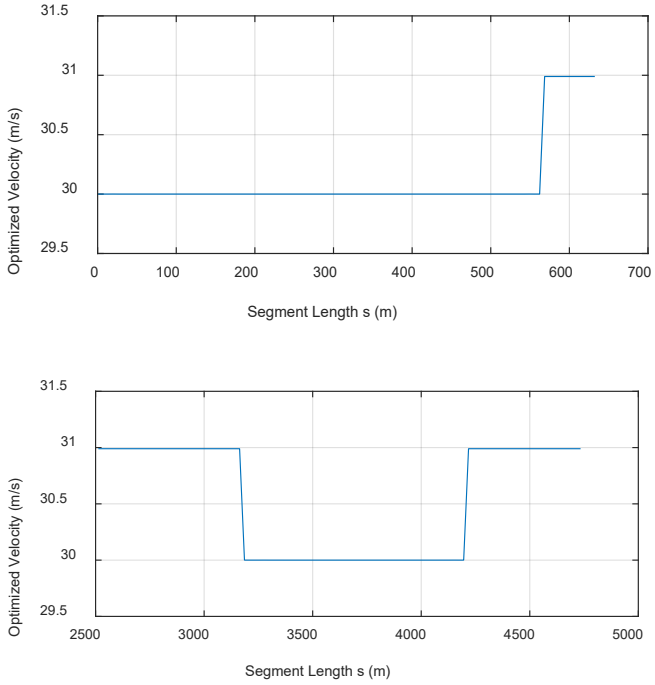

Figure 7.7 Dynamic Routine Optimized Velocity Profiles for Green Book road (top image) and Google Earth Model (bottom image)

7.4 Dynamic Road Reference Results

This routine is based on Figure 6.5, which optimizes the reference velocity based on vehicle-specific parameters and road parameters. The Green Book mathematically constructed road will be illustrated first with a sample optimization problem followed by the Google Earth model.

To illustrate the optimization problem to be solved iteratively, a single data point from $\kappa_1(s)$ is used to generate the contour plot in Figure 7.8, with ranges suitable for C.1. It is noted

that the minimizer lies somewhere in the line generated from the intersection of equation (2.1) and equation (2.2). Table 7.5 provides the road and vehicle parameters used throughout this optimization routine [46]. It is important to remark that the model is unbounded without the constraints that will lead to infeasible solutions.

Table 7.5 Input Parameters for M.1.

Parameter	Quantity	Unit
κ	0.0167	m^{-1}
$\overline{UG}$	1.95	degrees
L	2.5	m
g	9.81	m/s^2
e	6	%
μ	0.4	----

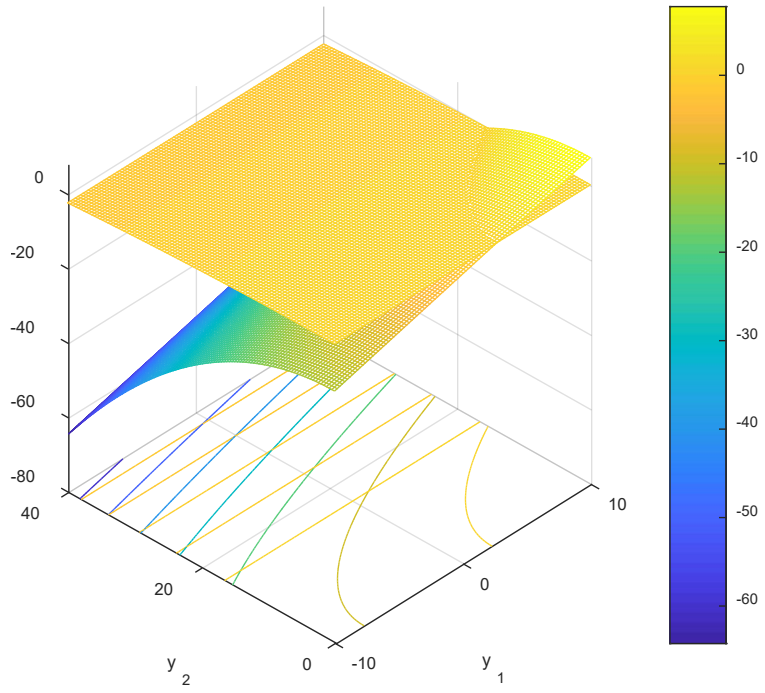

Figure 7.8 Contour Plots for M.1

This optimization routine obtains an angle of 3 degrees with a velocity of 16.63 m/s (36 mph), which is a compliant velocity for the input curvature of 0.0167 m^{-1}. To obtain a reference velocity and angle profiles, the routine needs to be implemented to a whole dataset of segment lengths and curvatures as opposed to a discrete curvature point. Section 7.3 input data, and parameters from Table 7.5 were used to solve Pr.1 and obtain an optimized velocity profile. This dataset was implemented with the Pseudo-Code shown in Figure 6.5, and its results are shown below in Figure 7.9.

A typical spiral curve contains a curvature profile characterized by piecewise functions such as M.1. Results for the Green Book optimized velocity (Figure 7.9 top) shows a closer resemblance to real-world driving scenarios such that smoother transitions are obtained by traversing the curve with a deceleration during the constant curvature segment. The last change reflects an adjustment of the spiral curve coming to a straight line such that optimal velocity follows standard velocity limits. However, in controller applications it would serve as an interpolating value to arrive for the safest speed while exiting the curve.

The results of the Google Earth model (Figure 7.9 bottom), shows the velocity expected for a wider curve radius as opposed to a short one. As the turn radius becomes bigger, optimal variables adjust an increase on velocity rather than a deceleration. The results of this model, again, can be used as a reference for controllers to interpolate appropriate exit maneuvering speeds and braking pressures to resemble the proper curve segment behavior.

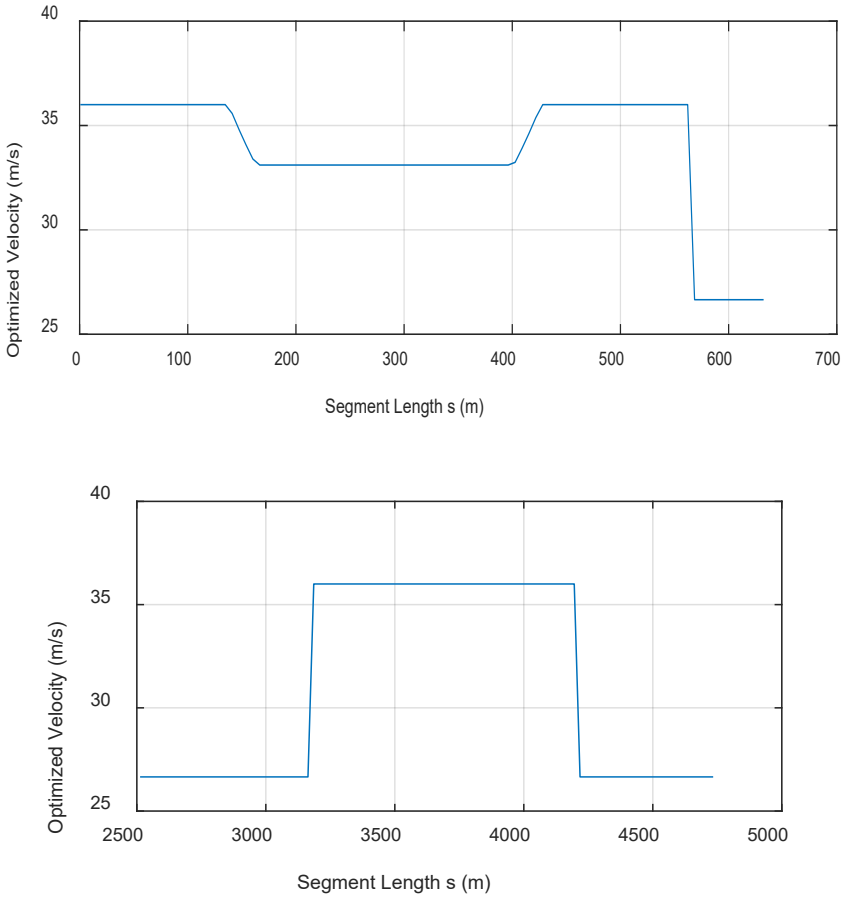

Figure 7.9 Dynamic Routine Optimized Velocity Profiles for Green Book Road (top image) and Google Earth Model (bottom image)

7.5 Chapter Conclusions

In this chapter, two optimization routines for generating guidance profiles were presented and analyzed. Initially, both routines utilize proposed curvature profiles obtained from roads and adjusted through least squares fitting. The first routine called Semi-Dynamic Road Reference utilizes road design standards to determine reference velocities in combination with heading angles from previous studies. The second routine called Dynamic road Reference uses the first routine in combination with vehicle dynamics to obtain a vehicle-specific reference velocity profile. Both routines were evaluated with two different datasets; an AASHTO mathematical

compliant profile, and a Google Earth extracted data profile. Differences were observed on the outcomes for both datasets, in which the initial conditions of the road shape are attributed to create different initial velocities. Furthermore, road total segment length was found to be a crucial factor on how the curvature profile should be modeled.

Possible ramifications from this study include the relationship on how the Google Earth empirical data compares to the AASHTO ideal model under similar road profiles such that road design models are derived from empirical data. Similarly, superposition of the mathematical models for more complex M.X models is yet to be explored.

Further implementation of this method could result in an upgrade for autonomous vehicle technology in which weather disruptions and poor road markings could stop being a problem for future vehicle generations. In conclusion, the presented optimization routines offer a reference profile for guidance in highway roads.

Chapter 8 Conclusions and Future Work

In this this report, a method for obtaining road reference paths was explained and detailed. This road reference was started from any general map source (aerial photography, GPS, scans, etc.) followed by a geometric formulation denoted as MDC. This geometric formulation was explored from multiple sources, and then determined smoothing techniques for its improvement on heading vehicle angle. The MDC method was evaluated with an MPC proving an improvement in lateral deviation. The next step followed obtaining an optimized reference using vehicle dynamics and AASHTO standards. The optimized algorithm was denoted as MDCR and offered a heading angle path along with suggested velocities that go according to vehicle and road specifications. The last chapter explored different map sources with the MDCR method to determine its feasibility from multiple sources. The general conclusions offer a reliable technique that can be applied to autonomous vehicles with V2I technology. Future work for this project will include creating an Application Programming Interface (API) for the implementation in real scale mapping. This API will be developed with the algorithms denoted from this Year 3 report and will be tested with DOT representatives that are willing to explore this avenue for CAV improvement.

References

1. FHWA. (2018). *Roadway Departure Safety*. Retrieved from USDOT: https://safety.fhwa.dot.gov/roadway_dept/

2. NHTSA. (2016). *Fatal Accident Reporting System (FARS) Encyclopedia*. Retrieved from National Highway Transportation Safety Administration: http://www-fars.nhtsa.dot.gov/Main/index.aspx

3. IIHS. (2018). *Roadway and environment: Fatality Facts*. Retrieved from Insurance Institute for Highway Safety, Highway Loss Data Institute: https://www.iihs.org/topics/fatality-statistics/detail/collisions-with-fixed-objects-and-animals

4. Stolle, C. S., Lechtenberg, K. A., Faller, R. K., & Yim, T. (2017). Initial Developments Supporting a Roadside Tree Removal Safety Campaign. *Journal of the International Conference of Roadside Safety*.

5. Andrews, S. (2012). Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) Communications and Cooperative Driving. *Eskandarian A. (eds) Handbook of Intelligent Vehicles*. doi:https://doi.org/10.1007/978-0-85729-085-4_46

6. Jurgen, R. K. (2012). *V2V/V2I Communications for Improved Road Safety and Efficiency.* SAE International.

7. Oisin, D., & O'Mahony, M. (2018). Assessing the Benefits of Installing V2I Communications on an Urban Orbital Motorway in a Medium-Sized European City. *97th Annual Meeting of the Transportation Research Board.* Washington D.C.

8. SAE. (2018). J3016 Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles. *SAE International*, 35.

9. NHTSA. (2018). *Automated Vehicles for Safety*. (USDOT National Highway Traffic Safety Administration) Retrieved 2020, from nhtsa.gov: https://www.nhtsa.gov/equipment/driver-assistance-technologies

10. NHTSA. (2019). *Driver Assistance Technologies*. Retrieved from nhtsa.gov: https://www.nhtsa.gov/equipment/driver-assistance-technologies

11. Stolle, C., Jacome, R., & Sweigard, M. (2018). *Virtual Barriers for Mitigating and Preventing Run-off Road Crashes, Phase I.* Transportation Research Board. doi:10.13014/K2TM78B5

12. DOD. (2020). *Global Positioning System Standard Positioning Service Performance Standard.* Department of Defense, Washington DC. Retrieved from https://www.gps.gov/systems/gps/performance/accuracy/

13. William J. Hughes Technical Center. (2017). *Global Positioning System (GPS) Standard Positioning Service (SPS) Performance Analysis Report.* Federal Aviation Administration.

14. Zang, S., Ding, M., Smith, D., Tyler , P., Rakotoarivelo, T., & Kaafar, M. A. (2019, March 12). The Impact of Adverse Weather Conditions on Autonomous Vehicles. *IEEE Vehicular Technology Magazine*. doi:10.1109/MVT.2019.2892497

15. Rahman, M. S., Abdel-Aty, M., Wang, L., & Lee, J. (2018). Understanding the Highway Safety Benefits of Different Approaches of Connected Vehicles in Reduced Visibility Conditions. *Journal of the Transportation Research Board*, 91-101. doi:10.1177/0361198118776113

16. Khondaker, B., & Kattan, L. (2015). Variable speed limit: A microscopic analysis in a connected. *Transportation Research Part C*. doi:https://doi.org/10.1016/J.TRC.2015.07.014

17. WYDOT. (2020, May 16). *WYDOT Connected Vehicle Pilot Program*. Retrieved from https://wydotcvp.wyoroad.info/

18. VDOT. (2018). *Smarter Roads VDOT*. Retrieved 2020, from https://smarterroads.org/login

19. USDOT. (2020, June 30). *USDOT Intelligent Transportation Systems Office - Connected Vehicle Pilot Deployment Program*. Retrieved from https://www.its.dot.gov/pilots/index.htm

20. Stolle, C., Jacome, R., & Sweigard, M. (2019). *Virtual Barriers for Mitigating and Preventing Run-off Road Crashes, Phase II.*

21. Sweigard, M. (2020). *Development of a Novel Vehicle Guidance System.* University of Nebraska-Lincoln.

22. Temel, S., Vuran, C. M., Rahman Lunar, M. M., Zhao, Z., Salam, A., Faller, R. K., & Stolle, C. (2018). Vehicle-to-Barrier Communication During Real-World Vehicle Crash Tests. *Computer Communications*. doi:10.1016/j.comcom.2018.05.009

23. Temel, S., Vuran, M. C., & Faller, R. K. (2016). A Primer on Vehicle-to-Barrier Communications:. *Vehicular Technology Conference*.

24. Jacome, R., Stolle, C., & Sweigard, M. (2020). Road Curvature Decomposition for Autonomous Guidance. *SAE Technical Paper 2020-01-1024*. doi:10.4271/2020-01-1024

25. AASHTO. (2011). *A Policy on Geometric Design of Highways and Streets (The Green Book)* (Sixth Edition ed.). Washington D.C.: American Association of State Highway and Transportation Officials.

26. IDOT. (2010). *Street Design Manual: Spiral Curves.* Iowa Department of Transportation.

27. WYDOT. (2011). *Horizontal Alignment and Superelevation.*

28. Gillespie, T. D. (1992). *Fundamentals of Vehicle Dynamics.* SAE International.

29. Pacejka, H. B. (2006). *Tyre and Vehicle Dynamics.* Butterworth-Heinemann.

30. Kreyszig, E. (1991). *Differential Geometry* (1st ed.). Dover Publications.

31. do Carmo, M. P. (1976). *Differential Geometry of Curves and Surfaces.* Englewood Cliffs, N.J: Prentice-Hall.

32. Pressley, A. N. (2010). *Elementary Differential Geometry.* London: Springer Undergraduate Mathematics Series. doi:10.1007/978-1-84882-891-9

33. Meriam, J. L., & Kraige, L. G. (2002). *Engineering Mechanics, Dynamics.* Wiley, John & Sons, Incorporated.

34. O'Reilly, O. M. (2010). *Engineering Dynamics: A Primer.* Springer Science & Business Media.

35. Morral, J. F., & Talarico, R. J. (1994). Side Friction Demanded and Margin of Safety on Horizontal Curves. *Journal of the Transportation Research Board*, 145-152.

36. Henry, J. J. (2000). *Evaluation of Pavement Friction Characteristics, a Synthesis of Highway Practice.* NCHRP Synthesis.

37. Mjaavatten, A. (2019, May 24). *Curvature of a 2D or 3D curve.* Retrieved from MATLAB Central File Exchange: (https://www.mathworks.com/matlabcentral/fileexchange/69452-curvature-of-a-2d-or-3d-curve)

38. Levien, R. L. (2009). *From Spiral to Spline: Optimal Techniques in Interactive Curve Design.* Berkeley.

39. Akima, H. (1970). A New Method of Interpolation and Smooth Curve Fitting based on Local Procedures. *ACM 17, no 4*, 589-602. doi:10.1145/321607.321609

40. Heath, M. T. (2018). *Scientific Computing: An Introductory Survey.* SIAM.

41. MathWorks. (2020). *Lane Keeping Assist System Using Model Predictive Control*. Retrieved from mathworks.com: https://www.mathworks.com/help/mpc/ug/lane-keeping-assist-system-using-model-predictive-control.html

42. Kelly, A., & Nagy, B. (2003). Reactive Nonholonomic Trajectory Generation via Parametric Optimal Control. *The International Journal of Robotics Research*, 583–602. doi:https://doi.org/10.1177/02783649030227008

43. Minh, V. T. (2013). Trajectory Generation for Autonomous Vehicles. *T. Březina (Ed.)*. doi:10.1007/978-3-319-02294-9_78

44. Rangel Costa, G. H., & Baldo, F. (2013). A method to automatically identify road centerlines from georeferenced smartphone data. *GEOINFO 2013, XIV Brazilian Symposium on GeoInformatics*.

45. Zhang, Y., Liu, J., Qian, X., Qiu, A., & Zhang, F. (2017). An Automatic Road Network Construction Method Using Massive GPS Trajectory Data. *International Journal of Geo-Information*. doi:10.3390/ijgi6120400

46. Dixit, N. R. (2009). *Evaluation of Vehicle Understeer Gradient Definitions.* Ohio.

Appendix A

A.1 Formulation Chapter 3: Acceleration in Canonical Form

Recalling the canonical representation of a curve from Chapter 3:

$$x(s) = \left(s - \frac{\kappa^2 s^3}{6}\right)\boldsymbol{T} + \left(\frac{s^2}{2}\kappa + \frac{d\kappa}{ds}\frac{s^3}{6}\right)\boldsymbol{N} + \left(\frac{\kappa\tau s^3}{6}\right)\boldsymbol{B} + o(s^3)$$

$$x_1(s) = s - \frac{\kappa^2}{6}s^3 + o(s^3)$$

$$x_2(s) = \frac{\kappa}{2}s^2 + \frac{d\kappa}{ds}\frac{1}{6}s^3 + o(s^3)$$

$$x_3(s) = \frac{\kappa\tau}{6}s^3 + o(s^3)$$

Assuming first order terms, the following representation is found:

$$\phi = s\boldsymbol{T} + \left(\frac{\kappa}{2}s^2\right)\boldsymbol{N} + \left(\frac{\kappa\tau}{6}s^3\right)\boldsymbol{B}$$

It is now desired to find:

$$\frac{d^2\phi}{dt^2} = \frac{d^2}{dt^2}\left(x_1(s) + x_2(s) + x_3(s)\right) = \frac{d^2}{dt^2}(s\boldsymbol{T}) + \frac{d^2}{dt^2}\left(\frac{\kappa}{2}s^2\boldsymbol{N}\right) + \frac{d^2}{dt^2}\left(\frac{\kappa\tau}{6}s^3\boldsymbol{B}\right)$$

At this point forward, bolding will be omitted to ease writing, and denoting that T, N, and B will be unit vectors. To ease the derivatives, the terms will be done separately.

First Term:

$$\frac{d^2}{dt^2}(sT) = \frac{d}{dt}\left(\frac{ds}{dt}T + s\frac{dT}{dt}\right)$$

$$\frac{d^2}{dt^2}(sT) = \frac{d}{dt}\left(\frac{ds}{dt}T + s\frac{dT}{dt}\right) = \left(\frac{d^2s}{dt^2}T + \frac{ds}{dt}\frac{dT}{dt} + \frac{ds}{dt}\frac{dT}{dt} + s\frac{d^2T}{dt^2}\right)$$

$$\frac{d^2}{dt^2}(sT) = \left(\ddot{s}\,T + 2\dot{s}\frac{dT}{dt} + s\frac{d^2T}{dt^2}\right) = \left(\ddot{s}\,T + 2\dot{s}\frac{dT}{ds}\frac{ds}{dt} + s\frac{d^2T}{ds^2}\frac{d^2s}{dt^2}\right)$$

$$\left(\ddot{s}\,T + 2\dot{s}^2\frac{dT}{ds} + s\ddot{s}\frac{d^2T}{ds^2}\right) = \left(\ddot{s}\,T + 2\dot{s}^2\kappa N + s\ddot{s}\frac{d}{ds}(\kappa N)\right)$$

$$= (\ddot{s}\,T + 2\dot{s}^2\kappa N + s\ddot{s}\kappa(-\kappa T + \tau B)) =$$

$$First\ Term:\ \frac{d^2}{dt^2}(sT) = \ (\ (1 - s\ddot{s}\kappa^2)\ddot{s}T\ + 2\dot{s}^2\kappa N + \tau B\ )$$

Second Term:

$$\frac{d^2}{dt^2}\left(\frac{\kappa}{2}s^2N\right) = \frac{d}{dt}\left(\frac{\dot{\kappa}}{2}s^2N + \kappa s\dot{s}N + \frac{\kappa}{2}s^2\frac{dN}{dt}\right) = \frac{d}{dt}\left(\frac{\dot{\kappa}}{2}s^2N + \kappa s\dot{s}N + \frac{\kappa}{2}s^2(-\kappa T + \tau B)\right)$$

$$= \frac{d}{dt}\left(\frac{\dot{\kappa}}{2}s^2N + \kappa s\dot{s}N - \frac{\kappa^2}{2}s^2T + \frac{\kappa}{2}s^2\tau B\right)$$

This will require taking the four terms separately:

$$\frac{d^2}{dt^2}\left(\frac{\kappa}{2}s^2N\right) = [1] + [2] + [3] + [4]$$

$$[1]\quad \frac{d}{dt}\left(\frac{\dot{\kappa}}{2}s^2\,N\right) = \left(\frac{\ddot{\kappa}}{2}s^2N + \dot{\kappa}s\dot{s}N + \frac{\dot{\kappa}}{2}s^2\frac{dN}{dt}\right) = \left(\frac{\ddot{\kappa}}{2}s^2N + \dot{\kappa}s\dot{s}N + \frac{\dot{\kappa}}{2}s^2\frac{dN}{ds}\frac{ds}{dt}\right) \rightarrow$$

$$\left(\frac{\ddot{\kappa}}{2}s^2N + \dot{\kappa}s\dot{s}N + \frac{\ddot{\kappa}}{2}s^2\dot{s}(-\kappa T + \tau B)\right) = \left(\frac{\ddot{\kappa}}{2}s^2N + \dot{\kappa}s\dot{s}N - \frac{\ddot{\kappa}}{2}s^2\dot{s}\kappa T + \frac{\ddot{\kappa}}{2}s^2\dot{s}\tau B)\right)$$

$$[1]\quad \frac{d}{dt}\left(\frac{\dot{\kappa}}{2}s^2N\right) = \left(-\frac{\ddot{\kappa}}{2}s^2\dot{s}\kappa T + \left(\frac{\ddot{\kappa}}{2}s + \dot{\kappa}\dot{s}\right)sN + \frac{\ddot{\kappa}}{2}s^2\dot{s}\tau B)\right)$$

$$[2]\quad \frac{d}{dt}(\kappa s\dot{s}N) = \left(\dot{\kappa}s\dot{s}N + \kappa\dot{s}^2N + \kappa s\ddot{s}N\ + \kappa s\dot{s}\frac{dN}{dt}\right)$$

$$= \left(\dot{\kappa}s\dot{s}N + \kappa\dot{s}^2N + \kappa s\ddot{s}N\ + \kappa s\dot{s}\frac{dN}{ds}\frac{ds}{dt}\right) \rightarrow$$

$$(\dot{\kappa}s\dot{s}N + \kappa\dot{s}^2N + \kappa s\ddot{s}N\ + \kappa s\dot{s}^2(-\kappa T + \tau B)\ ) \rightarrow$$

$$[2]\quad \frac{d}{dt}(\kappa s\dot{s}N)\ = (-s\dot{s}^2\kappa^2T\ + (\dot{\kappa}s\dot{s} + \kappa\dot{s}^2 + \kappa s\ddot{s})N\ + \kappa s\dot{s}^2\tau B\ )$$

$$[3]\quad \frac{d}{dt}\left(-\frac{\kappa^2}{2}s^2T\right) = -\left(\frac{\dot{\kappa}}{2}s^2T + \kappa s\dot{s}T + \frac{\kappa^2}{2}s^2\frac{dT}{dt}\right) = -\left(\frac{\dot{\kappa}}{2}s^2T + \kappa s\dot{s}T + \frac{\kappa^2}{2}s^2\frac{dT}{ds}\frac{ds}{dt}\right)$$

$$-\left(\frac{\dot{\kappa}}{2}s^2T + \kappa s\dot{s}T + \frac{\kappa^3}{2}s^2\dot{s}N\ \right)$$

$$[3]\quad \frac{d}{dt}\left(-\frac{\kappa^2}{2}s^2T\right) = -\left(\left(\frac{\dot{\kappa}}{2}s + \kappa\dot{s}\right)s\,T + \frac{\kappa^3}{2}s^2\dot{s}N\right)$$

$$[4]\quad \frac{d}{dt}\left(\frac{\kappa}{2}s^2\tau B\right) = \left(\frac{\dot{\kappa}}{2}s^2\tau B + \kappa\dot{s}s\tau B + \frac{\kappa}{2}s^2\dot{\tau}B + \frac{\kappa}{2}s^2\tau\,\frac{dB}{dt}\right)$$

$$= \left(\frac{\dot{\kappa}}{2}s^2\tau B + \kappa\dot{s}s\tau B + \frac{\kappa}{2}s^2\dot{\tau}B + \frac{\kappa}{2}s^2\tau\,\frac{dB}{ds}\frac{ds}{dt}\right)$$

$$= \left(\left(\frac{\dot{\kappa}}{2}s^2\tau + \kappa\dot{s}s\tau + \frac{\kappa}{2}s^2\dot{\tau}\right)B + \frac{\kappa}{2}s^2\tau\,\dot{s}(-\tau N)\right)$$

$$[4]\quad \frac{d}{dt}\left(\frac{\kappa}{2}s^2\tau B\right) = \left(\left(\frac{\dot{\kappa}}{2}s^2\tau + \kappa\dot{s}s\tau + \frac{\kappa}{2}s^2\dot{\tau}\right)B - \frac{\kappa}{2}s^2\tau^2\,\dot{s}N\right)$$

Combining all four terms:

$$\frac{d^2}{dt^2}\left(\frac{\kappa}{2}s^2N\right) = [1] + [2] + [3] + [4]$$

$$\textit{Second Term:}\quad \left(-\frac{\ddot{\kappa}}{2}s^2\dot{s}\kappa T + \left(\frac{\dot{\kappa}}{2}s + \kappa\dot{s}\right)sN + \frac{\ddot{\kappa}}{2}s^2\dot{s}\tau B)\right)$$
$$+ (-s\dot{s}^2\kappa^2T\ + (\dot{\kappa}s\dot{s} + \kappa\dot{s}^2 + \kappa s\ddot{s})N\ + \kappa s\dot{s}^2\tau B\ )$$
$$-\left(\left(\frac{\dot{\kappa}}{2}s + \kappa\dot{s}\right)s\,T + \frac{\kappa^3}{2}s^2\dot{s}N\right) + \left(\left(\frac{\dot{\kappa}}{2}s^2\tau + \kappa\dot{s}s\tau + \frac{\kappa}{2}s^2\dot{\tau}\right)B - \frac{\kappa}{2}s^2\tau^2\,\dot{s}N\right)$$

Simplifying:

$$\textit{Second Term:}\ \frac{d^2}{dt^2}\left(\frac{\kappa}{2}s^2N\right)$$
$$= \left(\left((\ddot{\kappa}\dot{s}\kappa + \dot{\kappa})\left(-\frac{1}{2}s^2\right) + (\dot{s}^2\kappa^2 + \kappa\dot{s})(-s)\right)T\right.$$
$$+\left((\ddot{\kappa} - \kappa^3\dot{s} - \kappa\tau^2\,\dot{s})\frac{1}{2}s^2 + 2s\dot{\kappa}\dot{s} + \kappa\dot{s}^2 + \kappa s\ddot{s}\right)N$$
$$\left.+\left(\left(\frac{\ddot{\kappa}}{2}s\dot{s} + \kappa\dot{s}^2 + \frac{\dot{\kappa}}{2}s + \kappa\dot{s}\right)s\tau + \frac{\kappa}{2}s^2\dot{\tau}\right)B\right)$$

Third Term:

$$\frac{d^2}{dt^2}\left(\frac{\kappa\tau}{6}s^3B\right) = \frac{d}{dt}\left(\frac{\dot{\kappa}\tau}{6}s^3B + \frac{\kappa\dot{\tau}}{6}s^3B + \frac{\kappa\tau}{6}3s^2\dot{s}B + \frac{\kappa\tau}{6}s^3\frac{dB}{dt}\right)$$
$$= \frac{d}{dt}\left(\frac{\dot{\kappa}\tau}{6}s^3B + \frac{\kappa\dot{\tau}}{6}s^3B + \frac{\kappa\tau}{2}s^2\dot{s}B + \frac{\kappa\tau}{6}s^3\frac{dB}{ds}\frac{ds}{dt}\right)$$

$$= \frac{d}{dt}\left(\frac{\dot{\kappa}\tau}{6}s^3B + \frac{\kappa\dot{\tau}}{6}s^3B + \frac{\kappa\tau}{2}s^2\dot{s}B + \frac{\kappa\tau}{6}s^3\dot{s}(-\tau N)\right)$$

This will require taking the four terms separately

$$\frac{d^2}{dt^2}\left(\frac{\kappa}{2}s^2N\right) = \{1\} + \{2\} + \{3\} + \{4\}$$

First term:

$$\{1\} \quad \frac{d}{dt}\left(\frac{\dot{\kappa}\tau}{6}s^3B\right) = \left(\frac{\ddot{\kappa}\tau}{6}s^3B + \frac{\dot{\kappa}\dot{\tau}}{6}s^3B + \frac{\dot{\kappa}\tau}{6}3s^2\dot{s}B + \frac{\dot{\kappa}\tau}{6}s^3\frac{dB}{dt}\right)$$

$$\left(\frac{\ddot{\kappa}\tau}{6}s^3B + \frac{\dot{\kappa}\dot{\tau}}{6}s^3B + \frac{\dot{\kappa}\tau}{2}s^2\dot{s}B + \frac{\dot{\kappa}\tau}{6}s^3\frac{dB}{ds}\frac{ds}{dt}\right)$$

$$= \left(\frac{\ddot{\kappa}\tau}{6}s^3B + \frac{\dot{\kappa}\dot{\tau}}{6}s^3B + \frac{\dot{\kappa}\tau}{2}s^2\dot{s}B + \frac{\dot{\kappa}\tau}{6}s^3(-\tau N)\dot{s}\right)$$

$$\{1\} \quad \frac{d}{dt}\left(\frac{\dot{\kappa}\tau}{6}s^3B\right) = \left(\frac{\ddot{\kappa}\tau}{6}s^3B + \frac{\dot{\kappa}\dot{\tau}}{6}s^3B + \frac{\dot{\kappa}\tau}{2}s^2\dot{s}B + \frac{\dot{\kappa}\tau}{6}s^3(-\tau N)\dot{s}\right)$$

$$\{2\} \quad \frac{d}{dt}\left(\frac{\kappa\dot{\tau}}{6}s^3B\right) = \left(\frac{\dot{\kappa}\dot{\tau}}{6}s^3B + \frac{\kappa\ddot{\tau}}{6}s^3B + \frac{\kappa\dot{\tau}}{6}3s^2\dot{s}B + \frac{\kappa\dot{\tau}}{6}s^3\frac{dB}{dt}\right)$$

$$\left(\frac{\dot{\kappa}\dot{\tau}}{6}s^3B + \frac{\kappa\ddot{\tau}}{6}s^3B + \frac{\kappa\dot{\tau}}{2}s^2\dot{s}B + \frac{\kappa\dot{\tau}}{6}s^3\frac{dB}{ds}\frac{ds}{dt}\right) = \left(\frac{\dot{\kappa}\dot{\tau}}{6}s^3B + \frac{\kappa\ddot{\tau}}{6}s^3B + \frac{\kappa\dot{\tau}}{2}s^2\dot{s}B + \frac{\kappa\dot{\tau}}{6}s^3(-\tau N)\dot{s}\right)$$

$$\{2\} \quad \frac{d}{dt}\left(\frac{\kappa\dot{\tau}}{6}s^3B\right) = \left(\frac{\dot{\kappa}\dot{\tau}}{6}s^3B + \frac{\kappa\ddot{\tau}}{6}s^3B + \frac{\kappa\dot{\tau}}{2}s^2\dot{s}B + \frac{\kappa\dot{\tau}}{6}s^3(-\tau N)\dot{s}\right)$$

$$\{3\} \quad \frac{d}{dt}\left(\frac{\kappa\tau}{2}s^2\dot{s}B\right) = \left(\frac{\dot{\kappa}\tau}{2}s^2\dot{s}B + \frac{\kappa\dot{\tau}}{2}s^2\dot{s}B + \frac{\kappa\tau}{2}2s\dot{s}^2B + \frac{\kappa\tau}{2}s^2\ddot{s}B + \frac{\kappa\tau}{2}s^2\dot{s}\frac{dB}{dt}\right)$$

$$\left(\frac{\dot{\kappa}\tau}{2}s^2\dot{s}B + \frac{\kappa\dot{\tau}}{2}s^2\dot{s}B + \kappa\tau s\dot{s}^2B + \frac{\kappa\tau}{2}s^2\ddot{s}B + \frac{\kappa\tau}{2}s^2\dot{s}\frac{dB}{ds}\frac{ds}{dt}\right)$$

$$= \left(\frac{\dot{\kappa}\tau}{2}s^2\dot{s}B + \frac{\kappa\dot{\tau}}{2}s^2\dot{s}B + \kappa\tau s\dot{s}^2B + \frac{\kappa\tau}{2}s^2\ddot{s}B + \frac{\kappa\tau}{2}s^2\dot{s}^2(-\tau N)\right)$$

$$\{3\} \quad \frac{d}{dt}\left(\frac{\kappa\tau}{2}s^2\dot{s}B\right) = \left(\frac{\dot{\kappa}\tau}{2}s^2\dot{s}B + \frac{\kappa\dot{\tau}}{2}s^2\dot{s}B + \kappa\tau s\dot{s}^2B + \frac{\kappa\tau}{2}s^2\ddot{s}B + \frac{\kappa\tau}{2}s^2\dot{s}^2(-\tau N)\right)$$

$$\{4\} \quad \frac{d}{dt}\left(\frac{\kappa\tau}{6}s^3\dot{s}(-\tau N)\right)$$

$$= -\left(\frac{\dot{\kappa}\tau^2}{6}s^3\dot{s}N + \frac{\kappa 2\tau\dot{\tau}}{6}s^3\dot{s}N + \frac{\kappa\tau^2}{6}3s^2\dot{s}^2N + \frac{\kappa\tau^2}{6}s^3\ddot{s}N + \frac{\kappa\tau^2}{6}s^3\dot{s}\frac{dN}{dt}\right)$$

$$= -\left(\frac{\dot{\kappa}\tau^2}{6}s^3\dot{s}N + \frac{\kappa\tau\dot{\tau}}{3}s^3\dot{s}N + \frac{\kappa\tau^2}{2}s^2\dot{s}^2N + \frac{\kappa\tau^2}{6}s^3\ddot{s}N + \frac{\kappa\tau^2}{6}s^3\dot{s}\frac{dN}{ds}\frac{ds}{dt}\right)$$

$$\{4\}\quad \frac{d}{dt}\left(\frac{\kappa\tau}{6}s^3\dot{s}(-\tau N)\right)$$
$$= -\left(\frac{\dot{\kappa}\tau^2}{6}s^3\dot{s}N + \frac{\kappa\tau\dot{\tau}}{3}s^3\dot{s}N + \frac{\kappa\tau^2}{2}s^2\dot{s}^2N + \frac{\kappa\tau^2}{6}s^3\ddot{s}N + \frac{\kappa\tau^2}{6}s^3\dot{s}^2(-\tau N)\right)$$

Combining Terms:

$$\frac{d^2}{dt^2}\left(\frac{\kappa}{2}s^2N\right) = \{1\} + \{2\} + \{3\} + \{4\}$$

$$Third\ Term\text{:}\quad \frac{d}{dt}\left(\frac{\dot{\kappa}\tau}{6}s^3B + \frac{\kappa\dot{\tau}}{6}s^3B + \frac{\kappa\tau}{2}s^2\dot{s}B + \frac{\kappa\tau}{6}s^3\dot{s}(-\tau N)\right) =$$
$$\left(\frac{\ddot{\kappa}\tau}{6}s^3B + \frac{\dot{\kappa}\dot{\tau}}{6}s^3B + \frac{\dot{\kappa}\tau}{2}s^2\dot{s}B + \frac{\dot{\kappa}\tau}{6}s^3(-\tau N)\dot{s}\right)$$
$$+\left(\frac{\dot{\kappa}\dot{\tau}}{6}s^3B + \frac{\kappa\ddot{\tau}}{6}s^3B + \frac{\kappa\dot{\tau}}{2}s^2\dot{s}B + \frac{\kappa\dot{\tau}}{6}s^3(-\tau N)\dot{s}\right)$$
$$+\left(\frac{\dot{\kappa}\tau}{2}s^2\dot{s}B + \frac{\kappa\dot{\tau}}{2}s^2\dot{s}B + \kappa\tau s\dot{s}^2B + \frac{\kappa\tau}{2}s^2\ddot{s}B + \frac{\kappa\tau}{2}s^2\dot{s}^2(-\tau N)\right)$$
$$-\left(\frac{\dot{\kappa}\tau^2}{6}s^3\dot{s}N + \frac{\kappa\tau\dot{\tau}}{3}s^3\dot{s}N + \frac{\kappa\tau^2}{2}s^2\dot{s}^2N + \frac{\kappa\tau^2}{6}s^3\ddot{s}N + \frac{\kappa\tau^2}{6}s^3\dot{s}^2(-\tau N)\right)$$

Simplifying:

$$\left(\frac{\ddot{\kappa}\tau}{6}s^3 + \frac{\dot{\kappa}\dot{\tau}}{6}s^3 + \frac{\dot{\kappa}\tau}{2}s^2\dot{s}\right)B - \frac{\dot{\kappa}\tau^2}{6}s^3\dot{s}\,N + \left(\frac{\dot{\kappa}\dot{\tau}}{6}s^3 + \frac{\kappa\ddot{\tau}}{6}s^3 + \frac{\kappa\dot{\tau}}{2}s^2\dot{s}\right)B - \frac{\kappa\dot{\tau}}{6}s^3\dot{s}\tau\,N$$
$$+\left(\frac{\dot{\kappa}\tau}{2}s^2\dot{s} + \frac{\kappa\dot{\tau}}{2}s^2\dot{s} + \kappa\tau s\dot{s}^2 + \frac{\kappa\tau}{2}s^2\ddot{s}\right)B - \frac{\kappa\tau^2}{2}s^2\dot{s}^2N$$
$$-\left(\frac{\dot{\kappa}\tau^2}{6}s^3\dot{s} + \frac{\kappa\tau\dot{\tau}}{3}s^3\dot{s} + \frac{\kappa\tau^2}{2}s^2\dot{s}^2 + \frac{\kappa\tau^2}{6}s^3\ddot{s} - \frac{\kappa\tau^2}{6}s^3\dot{s}^2\right)N$$

Grouping:

$$\left(\frac{\ddot{\kappa}\tau}{6}s^3 + \frac{\dot{\kappa}\dot{\tau}}{6}s^3 + \frac{\dot{\kappa}\tau}{2}s^2\dot{s}\right)B + \left(\frac{\dot{\kappa}\dot{\tau}}{6}s^3 + \frac{\kappa\ddot{\tau}}{6}s^3 + \frac{\kappa\dot{\tau}}{2}s^2\dot{s}\right)B$$
$$+\left(\frac{\dot{\kappa}\tau}{2}s^2\dot{s} + \frac{\kappa\dot{\tau}}{2}s^2\dot{s} + \kappa\tau s\dot{s}^2 + \frac{\kappa\tau}{2}s^2\ddot{s}\right)B - \frac{\dot{\kappa}\tau^2}{6}s^3\dot{s}\,N - \frac{\kappa\tau^2}{2}s^2\dot{s}^2N - \frac{\kappa\dot{\tau}}{6}s^3\dot{s}\tau\,N$$
$$-\left(\frac{\dot{\kappa}\tau^2}{6}s^3\dot{s} + \frac{\kappa\tau\dot{\tau}}{3}s^3\dot{s} + \frac{\kappa\tau^2}{2}s^2\dot{s}^2 + \frac{\kappa\tau^2}{6}s^3\ddot{s} - \frac{\kappa\tau^2}{6}s^3\dot{s}^2\right)N$$

$$Third\ Term\text{:}$$

$$\left(\left(\frac{\ddot{\kappa}\tau}{6}s^3+\frac{\dot{\kappa}\dot{\tau}}{6}s^3+\frac{\dot{\kappa}\tau}{2}s^2\dot{s}+\frac{\dot{\kappa}\dot{\tau}}{6}s^3+\frac{\kappa\ddot{\tau}}{6}s^3+\frac{\kappa\dot{\tau}}{2}s^2\dot{s}+\frac{\dot{\kappa}\tau}{2}s^2\dot{s}+\frac{\kappa\dot{\tau}}{2}s^2\dot{s}+\kappa\tau s\dot{s}^2+\frac{\kappa\tau}{2}s^2\ddot{s}\right)B\right.$$
$$-\left(\frac{\dot{\kappa}\tau^2}{6}s^3\dot{s}+\frac{\kappa\tau\dot{\tau}}{3}s^3\dot{s}+\frac{\kappa\tau^2}{2}s^2\dot{s}^2+\frac{\kappa\tau^2}{6}s^3\ddot{s}-\frac{\kappa\tau^2}{6}s^3\dot{s}^2++\frac{\dot{\kappa}\tau^2}{6}s^3\dot{s}\right.$$
$$\left.\left.+\frac{\kappa\tau^2}{2}s^2\dot{s}^2+\frac{\kappa\dot{\tau}}{6}s^3\dot{s}\tau\right)N\right)\rightarrow$$
$$\left(\left(\left(\frac{\ddot{\kappa}\tau}{6}+\frac{\dot{\kappa}\dot{\tau}}{6}+\frac{\dot{\kappa}\dot{\tau}}{6}+\frac{\kappa\ddot{\tau}}{6}\right)s^3+\left(\frac{\dot{\kappa}\tau}{2}\dot{s}+\frac{\kappa\dot{\tau}}{2}\dot{s}+\frac{\dot{\kappa}\tau}{2}\dot{s}+\frac{\kappa\dot{\tau}}{2}\dot{s}+\frac{\kappa\tau}{2}\ddot{s}\right)s^2+\kappa\tau s\dot{s}^2\right)B\right.$$
$$-\left(\left(\frac{\dot{\kappa}\tau^2}{6}\dot{s}+\frac{\kappa\tau\dot{\tau}}{3}\dot{s}+\frac{\kappa\tau^2}{6}\ddot{s}-\frac{\kappa\tau^2}{6}\dot{s}^2+\frac{\dot{\kappa}\tau^2}{6}\dot{s}+\frac{\kappa\dot{\tau}}{6}\dot{s}\tau\right)s^3\right.$$
$$\left.\left.+\left(\frac{\kappa\tau^2}{2}\dot{s}^2+\frac{\kappa\tau^2}{2}\dot{s}^2\right)s^2\right)N\right)$$

Simplifying further:

$$Third\ Term:\ \frac{d^2}{dt^2}\left(\frac{\kappa\tau}{6}s^3B\right)=$$
$$\left(\left(\frac{1}{6}(\ddot{\kappa}\tau+2\dot{\kappa}\dot{\tau}+\kappa\ddot{\tau})s^3+\left((\dot{\kappa}\tau+\kappa\dot{\tau})\dot{s}+\frac{\kappa\tau}{2}\ddot{s}\right)s^2+\kappa\tau s\dot{s}^2\right)B\right.$$
$$\left.-\left(\frac{1}{6}(\dot{\kappa}\tau^2\dot{s}+2\kappa\tau\dot{\tau}\dot{s}+\kappa\tau^2\ddot{s}-\kappa\tau^2\dot{s}^2+\dot{\kappa}\tau^2\dot{s}+\kappa\dot{\tau}\dot{s}\tau)s^3+\kappa\tau^2\dot{s}^2s^2\right)N\right)$$

Therefore, the final acceleration result is:

$$a=\frac{d^2}{dt^2}(sT)+\frac{d^2}{dt^2}\left(\frac{\kappa}{2}s^2N\right)+\frac{d^2}{dt^2}\left(\frac{\kappa\tau}{6}s^3B\right)=$$

$$
\begin{aligned}
a = (\ & (1 - s\ddot{s}\kappa^2)\ddot{s}T + 2\dot{s}^2\kappa N + \tau B\ ) \\
& + \left(\left((\ddot{\kappa}\dot{s}\kappa + \dot{\kappa}) \left(-\frac{1}{2}s^2 \right) + (\dot{s}^2\kappa^2 + \kappa\dot{s})(-s) \right) T \right. \\
& + \left((\ddot{\kappa} - \kappa^3\dot{s} - \kappa\tau^2\,\dot{s}) \frac{1}{2}s^2 + 2s\dot{\kappa}\dot{s} + \kappa\dot{s}^2 + \kappa s\ddot{s} \right) N \\
& + \left(\left(\frac{\ddot{\kappa}}{2}s\dot{s} + \kappa\dot{s}^2 + \frac{\dot{\kappa}}{2}s + \kappa\dot{s} \right) s\tau + \frac{\kappa}{2}s^2\dot{\tau} \right) B \Big) \\
& + \left(\left(\frac{1}{6}\,(\ddot{\kappa}\tau + 2\dot{\kappa}\dot{\tau} + \kappa\ddot{\tau})s^3 + \left((\dot{\kappa}\tau + \kappa\dot{\tau})\dot{s} + \frac{\kappa\tau}{2}\ddot{s} \right) s^2 + \kappa\tau s\dot{s}^2 \right) B \right. \\
& \left. - \left(\frac{1}{6}(\dot{\kappa}\tau^2\dot{s} + 2\kappa\tau\dot{\tau}\dot{s} + \kappa\tau^2\ddot{s} - \kappa\tau^2\dot{s}^2 + \dot{\kappa}\tau^2\dot{s} + \kappa\dot{\tau}\dot{s}\tau)s^3 + \kappa\tau^2\dot{s}^2 s^2 \right) N \right)
\end{aligned}
$$

A.2 Source Code Chapter 5: MDC Geometric Formulation Implementation

```
clear; close all; clc
%Google Earth Data
 load('GPS1Xft.mat');
load('GPS1Yft.mat');
 x2 = GPSX; y2 = GPSY;
 x2 = x2'*.3048; y2 = y2'*.3048;
%GPS DATA
% load('CVF9LatX.mat');
load('CVF9LongY.mat');
% x2 = LatX'; y2 = LongY';
%Ideal AASHTO
% load('MichXm.mat');
load('MichYm.mat');
% x2 = xm'; y2 = ym';
x2 = unique(x2,'stable');
y2 = unique(y2,'stable');
x2 = x2(1:numel(y2));X = [x2',y2'];
[L2,R2,K2] = curvature(X);
figure(1); plot(L2,R2); grid on;
title('Curvature radius \rho vs.
Cumulative curve length')
%The Radius of Curvature is High at
the Ends and Small at the middle
%Which is opposite for the
Curvature Kappa
xlabel('Length of Road');
ylabel('Radius \rho')
figure(2);
h = plot(x2,y2); grid on; axis
equal; set(h,'marker','.');
xlabel('X Coordinate'); ylabel('Y
Coordinate')
title('Road with Curvature
Vectors')
hold on
quiver(x2',y2',K2(:,1),K2(:,2));
hold off
% figure(3); hold on;
%
plot(x2,sqrt(K2(:,2).^2+K2(:,1).^2)
)
% xlabel('X-Coordinate');
ylabel('Curvature \kappa'); grid
on;
% title('Curvature \kappa vs X
Coordinate')
% figure(4); hold on;
%
plot(y2,sqrt(K2(:,2).^2+K2(:,1).^2)
)
% xlabel('Y-Coordinate');
ylabel('Curvature \kappa'); grid
on;
% title('Curvature \kappa vs Y
Coordinate')

KK = sqrt(K2(:,2).^2+K2(:,1).^2);
figure(5); scatter(L2,KK); grid on
xlabel('Lenght of Road');
ylabel('Curvature \kappa')
title('Curvature \kappa vs.
Cumulative curve length')
```

```
%  Smoothing Technique on
Curvature----------------
figure(1050)
x = L2; y = KK;
yy1 = smooth(x,y,0.15,'loess');
%Span of 15%
yy2 = smooth(x,y,0.15,'rloess');

%subplot(2,1,1)
plot(x,y,'b.',x,yy1,'r-');
ylim([0,.3]); grid on
legend('Original data','Smoothed
data using ''loess''',...
       'Location','best')
xlabel('Segment S (m)');
ylabel('Curvature K')
%subplot(2,1,2)
figure(1051)
plot(x,y,'b.',x,yy2,'r-'); grid on;
ylim([0,.3])
xlabel('Segment S (m)');
ylabel('Curvature K')
legend('Original data','Smoothed
Data',...
       'Location','best')

%-------
[Th,n] = DiscInteg(K2,L2);
figure(20)
plot(L2(1:n),Th*180/pi); grid on
xlabel('Segment S');
ylabel('Heading Angle \theta
degrees')
title('Numerically Integrated
Heading \theta')
%Note: This angle is in reference
to the completely horizontal line
% that was defined from the road
design perpesctive

figure(21)
O = atand(K2(:,2)./K2(:,1));
plot(L2,O)
xlabel('S'); ylabel('Angle from
Inverse Tan')
title('Angle from Inverse Tan')

figure(22)
[O1,O2] = direction(K2);
figure(21); plot(L2,O1)
xlabel('Segment S (m)');
ylabel('Angle from Inverse
Tangent')
title('Angle of Curvature
Direction'); grid on;
figure(22); plot(L2,O2)
xlabel('Segment S (m)');
ylabel('Angle of Velocity Vector')
title('Angle of Velocity
Direction');
grid on;
%  Smoothing Technique on Angles---
-------------
figure(1000)
x = L2; y = O2;
yy1 = smooth(x,y,0.15,'loess');
%Span of 15%
yy2 = smooth(x,y,0.15,'rloess');

%subplot(2,1,1)
plot(x,y,'b.',x,yy1,'r-')
legend('Original data','Smoothed
data using ''loess''',...
       'Location','SE')
xlabel('Segment S (m)');
ylabel('Angle of Velocity Vector')
%subplot(2,1,2)
figure(1001)
plot(x,y,'b.',x,yy2,'r-'); grid on
xlabel('Segment S (m)');
ylabel('Angle of Velocity Vector')
legend('Original data','Smoothed
Data',...
       'Location','SE')

% --------
e1 = cosd(O2); e2 = sind(O2);
figure(23)
subplot(1,2,1)
h1 = plot(x2,y2); grid on; axis
equal; set(h1,'marker','.');
hold on; quiver(x2',y2',e1,e2);
hold off
%title('Road with Velocity
Vectors')
xlabel('X Coordinate (m)');
ylabel('Y Coordinate (m)');


% ---- smoothed data quiver
subplot(1,2,2)
e1 = cosd(yy2); e2 = sind(yy2);
%figure(1002)
h1 = plot(x2,y2); grid on; axis
equal; set(h1,'marker','.');
hold on; quiver(x2',y2',e1,e2);
hold off
xlabel('X Coordinate (m)');
ylabel('Y Coordinate (m)');

%
figure(30)
```

```
subplot(2,2,1)
plot(L2,R2); grid on;
title('\rho vs. S')
xlabel S; ylabel('\rho')
subplot(2,2,2)
plot(x2,y2); grid on; hold on
xlabel X; ylabel Y
title('Road with Curvature
Vectors')
quiver(x2',y2',K2(:,1),K2(:,2));
hold off
subplot(2,2,3)
plot(L2,(sqrt(K2(:,2).^2+K2(:,1).^2
)))
grid on;
title('\kappa vs. S')
xlabel S; ylabel('\kappa')
subplot(2,2,4)
plot(L2(1:n),Th*180/pi); grid on
title('Integrated Heading \theta')
xlabel S; ylabel('\theta')
```

A.3 Formulation Chapter 6: Geometric Trapezoid

The formulation of the mathematical curvature model was based on the shape of a trapezoid. This trapezoid consists of three different functions: An increasing linear slope, a constant slope, and a decreasing linear slope. To achieve a continuous and differentiable function containing the aforementioned parts, unit step functions were utilized. A minimum of five different values were determined to generate a general trapezoid to be fit with data. The choice of the shape values for the trapezoid are shown in Figure A.1, in which x_1 through x_4 are values defined by the segment length and x_5 is defined by the curvature.

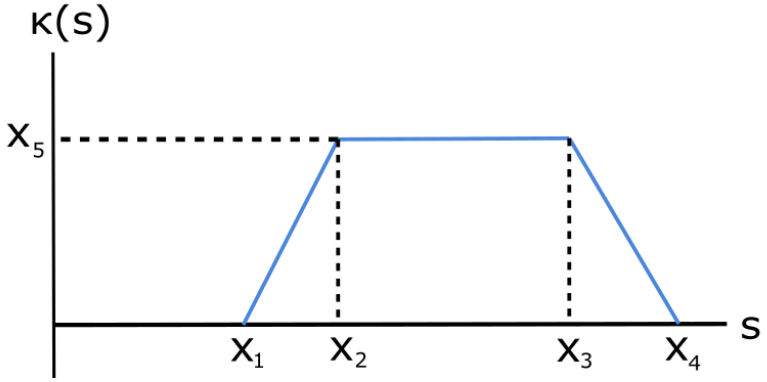

Figure A.1

To create this model, step functions were needed to encompass the continuity of the three desired functions. Thus, the range of the available step functions for a trapezoid are such that:

$$\kappa = A[\varphi(s - x_1) - \varphi(s - x_2)] + B[\varphi(s - x_2) - \varphi(s - x_3)] + C[\varphi(s - x_3) - \varphi(s - x_4)]$$

Eq XX

Where $\varphi(s - a) =$ Unit Step Function with a shift of $a \in \mathbb{R}$

It was needed to geometrically obtain the coefficients of A, B, and C such that all three together will constitute a continuous function. To find B, it was sufficient with the requirement of the segment being constant, such that $B = x_5$. With B determined, A and C needed to be two linear functions connected through the value x_5.

To obtain A, a linear increasing function is assumed. The formula for obtaining a linear function in terms of the allowed values is:

$$A = \left(\frac{x_5}{x_2 - x_1}\right) s - d$$

Where d is a constant defining the vertical intercept of the function below the horizontal axis. To describe the vertical intercept in terms of the allowed values, the angle θ in between the first linear function and the vertical created by x_2 had to be found. The angle θ, and the required values to find it are shown in Figure A.2.

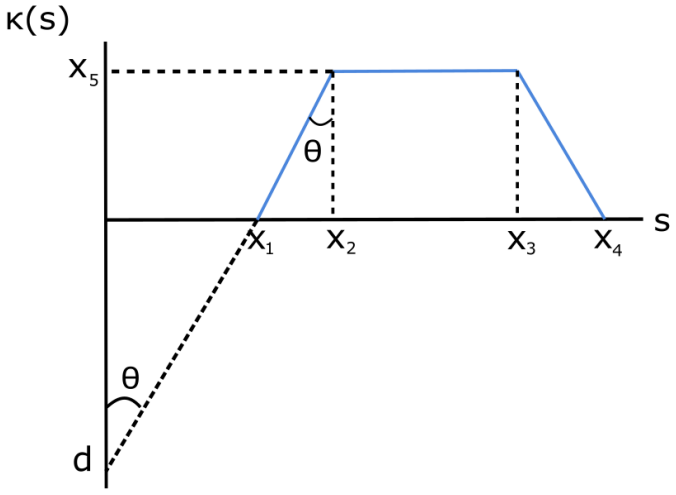

Figure A.2

By geometric relations, the angle is found to be:

$$\theta = \tan^{-1}\left(\frac{x_2 - x_1}{x_5}\right) = \tan^{-1}\left(\frac{x_1}{d}\right)$$

Which can be used to find the following:

$$d = \frac{x_1 x_5}{x_2 - x_1}$$

By plugging the definition of the vertical intercept in the definition of A, the following is found:

$$A = \left(\frac{x_5}{x_2 - x_1}\right)s - \frac{x_1 x_5}{x_2 - x_1} = \left(\frac{x_5}{x_2 - x_1}\right)(s - x_1)$$

To obtain B, a similar approach is used, in which C is modeled by a decreasing linear slope so that:

$$C = \left(\frac{x_5}{x_4 - x_3}\right)(-s) + e$$

Where e is a constant defining the vertical intercept of the function. To describe the vertical intercept in terms of the allowed values, the angle β in between the second linear function and the vertical created by x_3 had to be found. The angle β, and the required values to find it are shown in Figure A.3.

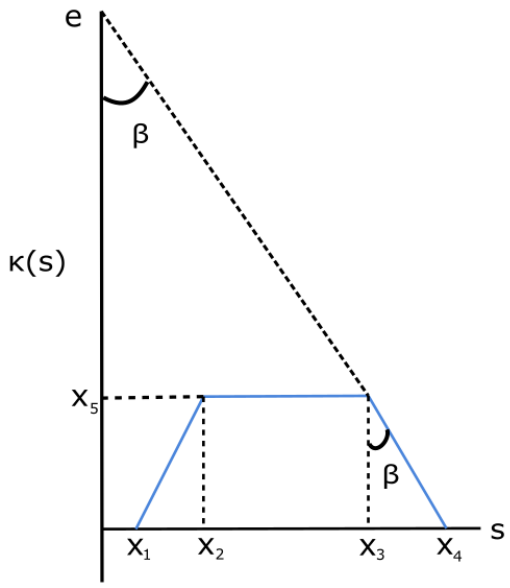

Figure A.3

By geometric relations, the angle is found to be:

$$\beta = \tan^{-1}\left(\frac{x_4 - x_3}{x_5}\right) = \tan^{-1}\left(\frac{x_3}{e - x_5}\right)$$

Which can be used to find the vertical intercept:

$$e = \frac{x_3 x_5}{x_4 - x_3} + x_5$$

By plugging the definition of the vertical intercept in the definition of C, the following is found:

$$C = \left(\frac{x_5}{x_4 - x_3}\right)(-s) + \frac{x_3 x_5}{x_4 - x_3} + x_5 = \left(\frac{x_5}{x_4 - x_3}\right)(-s + x_3) + x_5$$

By plugging the values obtained for A, B, and C, the following differentiable, continuous equation is found:

$$\kappa = \left(\frac{x_5}{x_2 - x_1}\right)(s - x_1)[\varphi(s - x_1) - \varphi(s - x_2)] + x_5[\varphi(s - x_2) - \varphi(s - x_3)] +$$

$$\left(\left(\frac{x_5}{x_4 - x_3}\right)(-s + x_3) + x_5\right)[\varphi(s - x_3) - \varphi(s - x_4)]$$

A.4 Source Code Chapter 7: MDCR Optimization Implementation

Curve Fitting Least Square Unstable Results

```
clc; clear all; close all
% From Section 5.1 on Paper
% Investigation on the stability of
the different M.X models proposed
% and their behavior under ideal
data with noise
s = 1:.01:12; n = numel(s)-1;
%y = [0 2 4.8 5.2 5 5.6];
y1 = 2.*s(1:n/2) - 1;   y2 =
11*ones(1,n/2);
y1o = awgn(y1,20,'measured'); y2o =
awgn(y2,20,'measured');

y = [y1o y2o];
x0 = [0.9 2.9 6 max(y)];
% Find the "minimized error".
fun1 = @(x,s) ((x(4)./(x(2)-
x(1))).*(s - x(1))).*(heaviside(s-
x(1)) - heaviside(s-x(2))) +...
     x(4).*(heaviside(s-x(2))-
heaviside(s-x(3)));
x = lsqcurvefit(fun1,x0,s(1:end-
1),y)
times = linspace(s(1),s(end-1));
hold on; plot(s(1:end-1),y,'bo')
plot(times,fun1(x,times),'k-
','linewidth',2)
legend('Data','Fitted Response');
title('Data and Fitted Curve')
xlim([times(1), times(end)+5])

x0 = [0.9 2.9 6 10 max(y)];
fun1 = @(x,s) ((x(5)./(x(2)-
x(1))).*(s - x(1))).*(heaviside(s-
x(1)) - heaviside(s-x(2))) +...
     x(5).*(heaviside(s-x(2))-
heaviside(s-x(3))) + ...
( ( x(5)./(x(4)-x(3))).*(-s+x(3))+
x(5) ).*(heaviside(s-x(3)) -
heaviside(s-x(4)));

x = lsqcurvefit(fun1,x0,s(1:end-
1),y)
times = linspace(s(1),s(end-1));
figure
hold on; plot(s(1:end-1),y,'bo')
plot(times,fun1(x,times),'k-
','linewidth',2)
legend('Data','Fitted Response');
title('Data and Fitted Curve')
xlim([times(1), times(end)+5])
```

Curve Fitting Least Square with Noise

```
clc; clear all; close all
% Analysis on the performance of
M.1 under different Gaussian Noise
Levels
% Behavior is good because even
though there should be
% a "drop" the function compensates
and does not generate a drop
%  Note: the function only behaves
properly when the data
% starts at 0 for y-axis
s = 1:.01:25;
%y = [0 2 4.8 5.2 5 5.6];
n = numel(s)-1;
y1 = 2.*s(1:n/2) - 1;
y2 = 23*ones(1,n/2);

for k = 1:9

y1o = awgn(y1,k,'measured');
y2o = awgn(y2,k,'measured');

y = [y1o y2o];
x0 = [0.9 2.9 4.89 7.85 11.50];

fun1 = @(x,s) ((x(5)./(x(2)-
x(1))).*(s - x(1))).*(heaviside(s-
x(1)) - heaviside(s-x(2))) +...
     x(5).*(heaviside(s-x(2))-
heaviside(s-x(3))) + ...
```

```
( ( x(5)./(x(4)-x(3))).*(-s+x(3))+
x(5) ).*(heaviside(s-x(3)) -
heaviside(s-x(4)));

x = lsqcurvefit(fun1,x0,s(1:end-
1),y)
times = linspace(s(1),s(end-1));
figure
hold on; plot(s(1:end-1),y,'bo')
plot(times,fun1(x,times),'k-
','linewidth',2)
xlim([times(1), times(end)+5])
legend('Data','Fitted
Response','location','best');
title('Data and Fitted Curve');
grid on

end
```

Semi-Dynamic Routine

```
%%%-------------------------------
--------------
% 6-25-2020
% This combines the research
performed in 2 papers:
% Jacome, R., Stolle, C. and
Sweigard, M.,
% "Road Curvature Decomposition for
Autonomous Guidance," SAE Technical
Paper
% 2020-01-1024, 2020,
doi:10.4271/2020-01-1024.
% And:
% Citation Needed::::::
% For this SemiDynamic Routine to
work
% (1) Select your Data i.e. GPS,
Google Earth, AASHTO, etc.
% (2) Select a range that occupies
one "curve segment" i.e.
% when does curvature change
considerably.
% (3) Select appropiate Initial
Conditions i.e. proportional to
% segment length.
% External files used: curvature.m
direction.m
%%%_-------------------------------
--------------
clear; close all; clc
%Google Earth Data
load('GPS1Xft.mat');
load('GPS1Yft.mat');
x2 = GPSX; y2 = GPSY;
x2 = x2'*.3048; y2 = y2'*.3048;
%GPS DATA
% load('CVF9LatX.mat');
load('CVF9LongY.mat');
% x2 = LatX'; y2 = LongY';
%Ideal AASHTO
% load('IdealXm.mat');
load('IdealYm.mat');
% x2 = xm'; y2 = ym';
x2 = unique(x2,'stable'); y2 =
unique(y2,'stable');
x2 = x2(1:numel(y2));
X = [x2',y2'];
[L,R,K] = curvature(X);
K(1,:) = []; K(end,:) = [];
L(1,:) = []; L(end,:) = [];
xlabel('Length of Road (m)');
ylabel('Radius \rho (m)')
figure(1);
x2(1) = []; x2(end) = [];
y2(1) = []; y2(end) = [];
h = plot(x2,y2); grid on; axis
equal; set(h,'marker','.');
xlabel('X Coordinate (m)');
ylabel('Y Coordinate (m)')
title('Road with Curvature
Vectors')
hold on
quiver(x2',y2',K(:,1),K(:,2)); hold
off
% ------------
y = sqrt(K(:,1).^2 + K(:,2).^2);
s = L;
%-------
[O1,O2] = direction(K);
e1 = cosd(O2); e2 = sind(O2);
figure(200);hold on; h1 =
plot(x2,y2); grid on; axis equal;
set(h1,'marker','.','Linewidth',3);
quiver(x2',y2',e1,e2); hold off
%title('Road with Velocity
Vectors')
xlabel('X Coordinate (m)');
ylabel('Y Coordinate (m)');
figure; plot(s,y)
%%% Road Section-----
ni = 1;
ne = numel(x2);
% ni = 120;
% ne = 180;
figure; plot(x2(ni:ne),y2(ni:ne));
grid on;
xlabel('X Coordinate (m)');
ylabel('Y Coordinate (m)');
figure; plot(s(ni:ne),y(ni:ne));
```

```
grid on;xlabel('Segment Length s
(m)'); ylabel('Curvature \kappa
(m^{-1})')

ySmoo =
smooth(s(ni:ne),y(ni:ne),0.15,'loes
s');
sSmoo = s(ni:ne);
figure; plot(sSmoo,ySmoo); grid on;
xlabel('Segment Length s (m)');
ylabel('Curvature \kappa (m^{-1})')
%---------------------------------
----------------------------
%From this point forward, I am
doing the Optimization Semi-Dynamic
Routine
% Initial Conditions, NEVER repeat
them.
%Ideal AASHTO IC.
%x0 = [ 100 200 300 400
max(ySmoo)];
%Google Earth IC.
  x0 = [3200 3500 4000 4200
max(ySmoo)];
%------
%x0 = [750 850 900 1000
max(ySmoo)];
%x0 = [.25*mean(sSmoo)
0.5*mean(sSmoo) 1.25*mean(sSmoo)
1.50*mean(sSmoo) max(sSmoo)];
%[sSmoo(1) 0.75*mean(sSmoo)
1.25*mean(sSmoo) sSmoo(end)
max(ySmoo)]
%x0 = [1.25*sSmoo(1) mean(sSmoo)
1.25*mean(sSmoo) .75*sSmoo(end)
max(ySmoo)]
%Note: Every single time, the x0
need ot be modified to achieve the
right
%form
% Curvature Model M.1
M1 = @(x,s) ((x(5)./(x(2)-
x(1))).*(s - x(1))).*(heaviside(s-
x(1)) - heaviside(s-x(2))) +...
     x(5).*(heaviside(s-x(2))-
heaviside(s-x(3))) + ...
( ( x(5)./(x(4)-x(3))).*(-s+x(3))+
x(5) ).*(heaviside(s-x(3)) -
heaviside(s-x(4)));
% Pr.1
fprintf('Pr. 1, Least Squares Min.
Has finalized');
options =
optimset('Display','off');
x =
lsqcurvefit(M1,x0,sSmoo,ySmoo,[],[]
,options)
snew =
linspace(sSmoo(1),sSmoo(end),100);
% <--- This defines the
% size of the "K_vector".
figure; hold on;
plot(sSmoo,ySmoo,'bo');
plot(snew,M1(x,snew),'k-
','linewidth',2);
xlim([snew(1), snew(end)+5]);
legend('Data','Fitted
Response','location','best');
title('Data and Fitted Curve');
grid on
xlabel('Segment Length s (m)');
ylabel('Curvature \kappa (m^{-1})')

% -------------------------
%Parameters
global K_temp e g mu
% Road Only
%e = 12; mu = 0.4;
e = 6; mu = 0.3;
% Both
g = 9.81; K_vector = M1(x,snew);
% -------------------------
%Iterative Optimization Routine for
Pr.2 given Optimized M.1
for i = 1:length(K_vector)
K_temp = K_vector(i);
% Objective Function Pr.2
fun = @(x) x(1)^2*K_temp/g - (mu +
0.01*e)/(1-0.01*mu*e) ;
%C.1 (Bounds)
lb = 30;
%ub = [30,35];
ub = 38;  % 60 < x2 < 80; mph
% There are no linear constraints,
so set those arguments to |[]|.
A = [];  b = []; % Linear In-
equality Constraints
Aeq = []; beq = [];  % Linear
Equality Constraints
%Initial Conditions
x0 = 1/4;
%Constraints as an annoynomous
function
Op(i,:) =
fmincon(fun,x0,A,b,Aeq,beq,lb,ub);
end
fprintf('Pr. 2 Has finalized \n');
figure; plot(snew,Op(:,1))
xlabel('Segment Length s (m)');
ylabel('Optimized Velocity (m/s)')
```

```
title('Segment Length vs Velocity
Optimized'); grid on
```

Optimization Contour

```
clear all; clc; close all
% Shown in Section 5.4 of Paper
% From (Dixit 2009)
% At Rho = 30 m -> U = 1.91
% At Rho = 60 m -> U = 1.95
y1 = linspace(-10,10); %Resonable
Angles
y2 = linspace(0,40); % Reasonable
Speed Ranges (m/s)
% 40 m/s ~ 90 mph ~ 144 km/hr
L = 2.5; % m
g = 9.81; % m/s^2
%AASHTO values
e = 6; mu = .4;
U  = 1.95; K = 1/60;
[X1,X2] = meshgrid(y1,y2);
Z = X1 - (53.7*L+U*X2.^2)*K;
figure; meshc(X1,X2,Z); hold on
Z2 = - X2.^2*K/g + (mu +
0.01*e)/(1-0.01*mu*e);
meshc(X1,X2,Z2); colorbar;
xlabel('y_1'); ylabel('y_2')
```

Dynamic Routine Optimization

```
%%%%------------------------------
--------------
% 6-25-2020
% This combines the research
performed in 2 papers:
% Jacome, R., Stolle, C. and
Sweigard, M.,
% "Road Curvature Decomposition for
Autonomous Guidance," SAE Technical
Paper
% 2020-01-1024, 2020,
doi:10.4271/2020-01-1024.
% And:
% Citation Needed::::::
% For this Dynamic Routine to work
% (1) Select your Data i.e. GPS,
Google Earth, AASHTO, etc.
% (2) Select a range that occupies
one "curve segment" i.e.
% when does curvature change
considerably.
% (3) Select appropiate Initial
Conditions i.e. proportional to
% segment length.
% External files used: curvature.m
direction.m
%%%_------------------------------
--------------
clear; close all; clc
%Google Earth Data
load('GPS1Xft.mat');
load('GPS1Yft.mat');
x2 = GPSX; y2 = GPSY;
x2 = x2'*.3048; y2 = y2'*.3048;
%GPS DATA
% load('CVF9LatX.mat');
load('CVF9LongY.mat');
% x2 = LatX'; y2 = LongY';
%Ideal AASHTO
% load('IdealXm.mat');
load('IdealYm.mat');
% x2 = xm'; y2 = ym';
x2 = unique(x2,'stable'); y2 =
unique(y2,'stable');
x2 = x2(1:numel(y2));
X = [x2',y2'];
[L,R,K] = curvature(X);
K(1,:) = []; K(end,:) = [];
L(1,:) = []; L(end,:) = [];
xlabel('Length of Road (m)');
ylabel('Radius \rho (m)')
figure(1);
x2(1) = []; x2(end) = [];
y2(1) = []; y2(end) = [];
h = plot(x2,y2); grid on; axis
equal; set(h,'marker','.');
xlabel('X Coordinate (m)');
ylabel('Y Coordinate (m)')
title('Road with Curvature
Vectors')
hold on
quiver(x2',y2',K(:,1),K(:,2)); hold
off
% ------------
y = sqrt(K(:,1).^2 + K(:,2).^2);
s = L;
%-------
[O1,O2] = direction(K);
e1 = cosd(O2); e2 = sind(O2);
figure(200);hold on; h1 =
plot(x2,y2); grid on; axis equal;
set(h1,'marker','.','Linewidth',3);
quiver(x2',y2',e1,e2); hold off
%title('Road with Velocity
Vectors')
xlabel('X Coordinate (m)');
ylabel('Y Coordinate (m)');
figure; plot(s,y); grid on;
```

```
xlabel('Segment Length s (m)');
ylabel('Curvature \kappa (m^{-1})')
% Segment Length Selections
% ni = 1;
% ne = numel(x2);
ni = 120;
ne = 180;
figure; plot(x2(ni:ne),y2(ni:ne));
grid on;
xlabel('X Coordinate (m)');
ylabel('Y Coordinate (m)');
figure; plot(s(ni:ne),y(ni:ne));
grid on;xlabel('Segment Length s
(m)'); ylabel('Curvature \kappa
(m^{-1})')

ySmoo =
smooth(s(ni:ne),y(ni:ne),0.15,'loes
s');
sSmoo = s(ni:ne);
figure; plot(sSmoo,ySmoo); grid on;
xlabel('Segment Length s (m)');
ylabel('Curvature \kappa (m^{-1})')
%---------------------------------
-----------------------------
%From this point forward, I am
doing the Optimization Dynamic
Routine
% Initial Conditions, NEVER repeat
them.
%Ideal AASHTO IC.
%x0 = [ 100 200 300 400
max(ySmoo)];
%Google Earth IC.
 x0 = [3200 3500 4000 4200
max(ySmoo)];
%x0 = [750 850 900 1000
max(ySmoo)];
%[sSmoo(1) 0.75*mean(sSmoo)
1.25*mean(sSmoo) sSmoo(end)
max(ySmoo)]
%x0 = [1.25*sSmoo(1) mean(sSmoo)
1.25*mean(sSmoo) .75*sSmoo(end)
max(ySmoo)]
%Note: Every single time, the x0
need ot be modified to achieve the
right
%form
% Curvature Model M.1
M1 = @(x,s) ((x(5)./(x(2)-
x(1))).*(s - x(1))).*(heaviside(s-
x(1)) - heaviside(s-x(2))) +...
     x(5).*(heaviside(s-x(2))-
heaviside(s-x(3))) + ...
( ( x(5)./(x(4)-x(3))).*(-s+x(3))+
x(5) ).*(heaviside(s-x(3)) -
heaviside(s-x(4)));
% Pr.1
fprintf('Pr. 1, Least Squares Min.
Has finalized \n');
options =
optimset('Display','off');


x =
lsqcurvefit(M1,x0,sSmoo,ySmoo,[],[]
,options)
snew =
linspace(sSmoo(1),sSmoo(end),100);
% <--- This defines the
% size of the "K_vector".
figure; hold on;
plot(sSmoo,ySmoo,'bo');
plot(snew,M1(x,snew),'k-
','linewidth',2);
xlim([snew(1), snew(end)+5]);
xlabel('Segment Length s (m)');
ylabel('Curvature \kappa (m^{-1})')
legend('Data','Fitted
Response','location','best');
title('Data and Fitted Curve');
grid on

% -------------------------
%Parameters
global K_temp e g mu U
% Vehicle Only
L = 2.5;  %U = 1.95;
U = 3;
% Road Only
%e = 12; mu = 0.4;
e = 6; mu = 0.3;
% Both
g = 9.81; K_vector = M1(x,snew);
% -------------------------
%Iterative Optimization Routine for
Pr.2 given Optimized M.1
for i = 1:length(K_vector)
K_temp = K_vector(i);
% Objective Function Pr.2
fun = @(x)  x(1) - (53.7*L +
U*x(2)^2/g)*K_temp;
%C.1 (Bounds)
lb = [-3,25]; % -3 < x1 < 3;
%ub = [30,35];
ub = [3,36];  % 60 < x2 < 80; mph
% There are no linear constraints,
so set those arguments to |[]|.
A = [];  b = []; % Linear In-
equality Constraints
Aeq = []; beq = [];  % Linear
Equality Constraints
%Initial Conditions
```

```
x0 = [1/4,1/2];
%Constraints as an annoynomous
function
nonlcon = @EqConstraint;
options =
optimoptions('fmincon','Display','o
ff');
Op(i,:) =
fmincon(fun,x0,A,b,Aeq,beq,lb,ub,no
nlcon,options);
end
fprintf('Pr. 2 Has finalized \n');
figure; plot(snew,Op(:,2))
%title('Segment Length vs Velocity
Optimized');
grid on
xlabel('Segment Length s (m)');
ylabel('Optimized Velocity (m/s)')

% Nonlinear Constaints (Not bounds)
function [c,ceq] = EqConstraint(x)
global K_temp e g mu
%Pr.2
% Nonlinear Inequality Constraints
c = x(2)^2*K_temp/g - (mu +
0.01*e)/(1-0.01*mu*e);
% Nonlinear Equality Constraints
ceq = [];
end
```

Dynamic Routine with Ideal Noise

```
clear; close all; clc
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%
% In this example, the noise is
added to the ideal mathematical
road
% However, no smoothing is
performed.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%
%GPS DATA
%load('CVF9LatX.mat');
load('CVF9LongY.mat');
%Ideal AASHTO
load('IdealXm.mat');
load('IdealYm.mat');
%x2 = LatX'; y2 = LongY';
x2 = xm'; y2 = ym';
x2 = unique(x2); y2 = unique(y2);
x2 = x2(1:numel(y2));
% Added White Noise, and now
everything "kinda" works, but not
really
x2 = awgn(x2,200,'measured');
y2 = awgn(y2,200,'measured');

X = [x2',y2'];
[L,R,K] = curvature(X);
K(1,:) = []; K(end,:) = []; L(1,:)
= []; L(end,:) = [];
x2(1) = []; x2(end) = []; y2(1) =
[]; y2(end) = [];
figure; scatter(x2,y2);
xlabel('X Coordinate (m)');
ylabel('Y Coordinate (m)')
title('Raw Road Data')
figure;
h = scatter(x2,y2); grid on; axis
equal; set(h,'marker','.');
xlabel('X Coordinate (m)');
ylabel('Y Coordinate (m)')
title('Road with Curvature
Vectors')
hold on

quiver(x2',y2',K(:,1),K(:,2)); hold
off
y = sqrt(K(:,1).^2 + K(:,2).^2);
s = L;
figure; plot(s,y)
% Initial Conditions, NEVER repeat
them.
x0 = [100 200 300 400 500];
% Curvature Model M.1
M1 = @(x,s) ((x(5)./(x(2)-
x(1))).*(s - x(1))).*(heaviside(s-
x(1)) - heaviside(s-x(2))) +...
     x(5).*(heaviside(s-x(2))-
heaviside(s-x(3))) + ...
( ( x(5)./(x(4)-x(3))).*(-s+x(3))+
x(5) ).*(heaviside(s-x(3)) -
heaviside(s-x(4)));
% Pr.1
fprintf('Pr. 1, Least Squares Min.
Has finalized\n');
options =
optimset('Display','off');
x =
lsqcurvefit(M1,x0,s(1:end),y,[],[],
options)
snew = linspace(s(1),s(end),100); %
<--- This defines the
% size of the "K_vector".
figure; hold on;
plot(s,y,'bo');
xlabel('S-Segment (m)'); ylabel
('Curvature(m^{-1})');
```

```
plot(snew,M1(x,snew),'k-
','linewidth',2);
xlim([snew(1), snew(end)+5]);
legend('Data','Fitted
Response','location','best');
title('Data and Fitted Curve');
grid on

% -------------------------
%Parameters
global K_temp e g mu U
% Vehicle Only
L = 2.5;  %U = 1.95;
U = 3;
% Road Only
e = 6; mu = 0.3;
% Both
g = 9.81; K_vector = M1(x,snew);
% -------------------------
%Iterative Optimization Routine for
Pr.2 given Optimized M.1
for i = 1:length(K_vector)
K_temp = K_vector(i);
% Objective Function Pr.2
fun = @(x)  x(1) - (53.7*L +
U*x(2)^2/g)*K_temp;
%C.1 (Bounds)
% lb = [-3,25];
% ub = [3,60];
lb = [-3,25]; % -3 < x1 < 3;
ub = [3,35];  % 55 < x2 < 80; mph
% There are no linear constraints,
so set those arguments to |[]|.
A = [];  b = []; % Linear In-
equality Constraints
Aeq = []; beq = [];  % Linear
Equality Constraints
%Initial Conditions
x0 = [1/4,1/4];
%Constraints as an annoynomous
function
nonlcon = @EqConstraint;
options =
optimoptions('fmincon','Display','o
ff');
Op(i,:) =
fmincon(fun,x0,A,b,Aeq,beq,lb,ub,no
nlcon,options);
end
fprintf('Pr. 2 Has finalized \n');
vOpt = Op(:,2);
figure;
plot(snew,vOpt,'linewidth',3)
ylim([max(vOpt)-10 max(vOpt)+10])
title('Segment Length vs Velocity
Optimized'); grid on
xlabel('S-Segment (m)'); ylabel
('Velocity (m/s)');
figure; plot(M1(x,snew),vOpt)
ylim([max(vOpt)-10 max(vOpt)+10])
title('Curvature vs Velocity
Optimized'); grid on;
xlabel('S-Segment (m)'); ylabel
('Curvature(m^{-1})');

% Nonlinear Constaints (Not bounds)
function [c,ceq] = EqConstraint(x)
global K_temp e g mu
%Pr.2
% Nonlinear Inequality Constraints
c = x(2)^2*K_temp/g - (mu +
0.01*e)/(1-0.01*mu*e);
% Nonlinear Equality Constraints
ceq = [];
end
```