DRIVING ANALYTICS FOR IMPROVED ROAD SAFETY

A Thesis

by

ADITHYA AJITH

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Andrew L Johnson
Co-Chair of Committee,	Ivan Damnjanovic
Committee Member,	Sergiy Butenko
Head of Department,	Mark Lawley

May 2018

Major Subject: Industrial Engineering

ABSTRACT

Road safety has always been a globally growing concern and speeding is one of the major factors that causes road crashes. Road geometry is an important factor that influence speeding and thus the road safety. The purpose of this research is to access the sensitivity of horizontal and vertical road geometries on driving safety. A simulation framework was developed to imitate a safe human way of driving along 2-lane rural roads during free flow or no traffic conditions. This framework is in a computer simulation environment to generate a safe driving speed profile by using road geometry information. The framework also includes the real-time importing of the required road geometry data from the online map databases like Open Street Maps and Google Maps. Basically, if the model is inputted with latitude and longitude coordinates of starting and end point for a route, the model will output for every 1 meter along the route the simulated driving speed under a stipulated safe driving condition.

Based on the starting and the end coordinates of a driving route, the model queries the coordinates and the elevation of the equidistant waypoints along the route from online map databases. This queried information about the road geometry is used to evaluate the safe driving speed conditions along the route. The model tries to imitate the human way of driving by predicting at each point along the route if a driver will accelerate, maintain a constant speed, decelerate without braking or apply brake to stay within the limiting speeds due to road geometry. The simulation framework was validated against the real driving speed profile recorded on four routes. As an improvement to road safety, this framework could be deployed to warn drivers when they are having unsafe driving speeds.

CONTRIBUTORS AND FUNDING SOURCES

This work was supervised by a dissertation committee consisting of Professor Ivan Damnjanovic of the Zachry Department of Civil Engineering and Professors Andrew L Johnson and Sergiy Butenko of the Department of Industrial and System Engineering. All work for the thesis was completed independently by the student.

There are no outside funding contributions to acknowledge related to the research and compilation of this document.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

1. INTRODUCTION

Road traffic injuries and deaths are tragedies that tend to affect everyone around the world either directly or indirectly at some point in the lifetime. It causes not only a lot of mental and physical suffering but also considerable economic losses to the individuals, their friends, and families. The menace caused can be curbed by understanding the critical root causes and taking appropriate measures to improve the road safety. Human errors like speeding, driving under influence, non-use of the safety devices (like motorcycle helmets, seat-belts, and child restraints), distracted driving and drowsy driving are commonly known causes of road accidents. Other common aggravating factors are unsafe road infrastructure, unsafe vehicles, inadequate post-crash care, inadequate traffic laws and inadequate law enforcement. Despite all the ongoing efforts throughout the world, "without sustained action, road traffic crashes are predicted to become the seventh leading cause of death by 2030."[1]

The likelihood of a crash is more at higher driving speed [2]. At a higher speed, the driver has a shorter time to stop the vehicle if needed. The distribution of road fatalities (from the year 2014 to 2016) in the United States as per the National Highway Traffic Safety Administration is shown in Figure 1 [3]. Every year approximately speed contributes to about 30% of deaths on the US roads. Reports are more alarming for some low-income and middle-income countries, where speed is estimated to contribute to about half of all road crashes [4].

Drivers can have excess speed by exceeding the speed limit or inappropriate speed by having a driving speed unsafe for the prevailing road and traffic conditions. Haste, thrill or adapting to other traffic makes the drivers to intentionally drive at unsafe speed. Unsafe driving speed can be attained unintentionally too. A habitual unsafe driving, underestimating the current

driving speed and/or lack of awareness of speed limits can be the reasons [5]. The driver's perception of the speed can be skewed if driving fast for a long duration, driving cars with quiet and comfortable cabins, or particular road features like the absence of reference objects (buildings, trees, traffic etc.) alongside the road/driving lane [6]. Roadway agencies use various speed management strategies to control unsafe speeding. Such improvements include traffic control devices and road designs [7]. A preventive risk assessment tools, such as road safety audits and inspections can also be applied at successive stages of road planning, design, opening, and operations. Roadway analytics is also used to understand the traffic flow and impact of congestion.

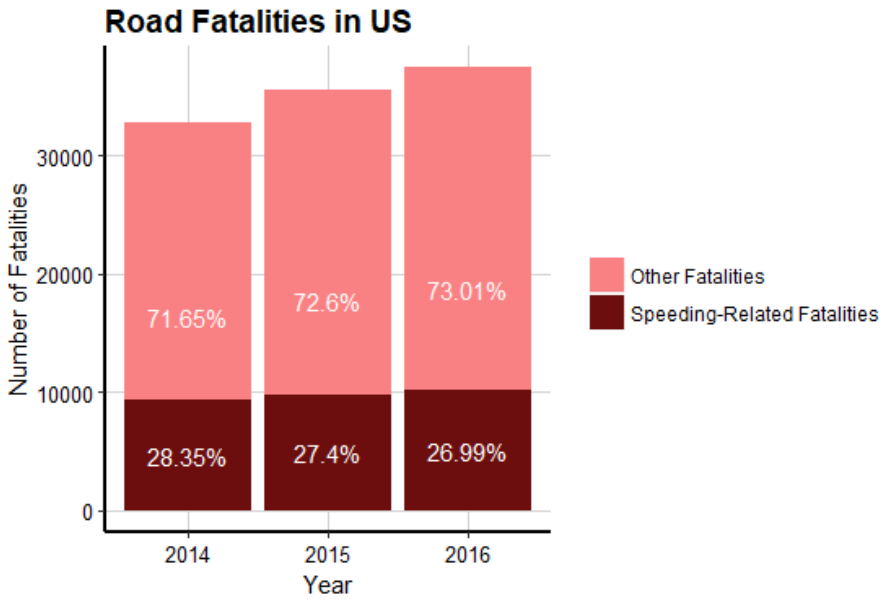

Figure 1. *Distribution of Road Fatalities in the United States during the period 2014-2016.*

Road environment is an important factor for safe driving conditions as "the driving task involves an interaction among the driver, the vehicle, and the road" [8]. A driver continuously adjusts the driving speed, from the information obtained visually. These visual cues include the road signs, weather, road surface conditions, traffic conditions and the geometry of the road. But the reaction to the visual cues depends on the driver's perception, experience, physiology, and

psychological characteristics. The dynamic relationship between these external and cognitive factors are illustrated in Figure 2.

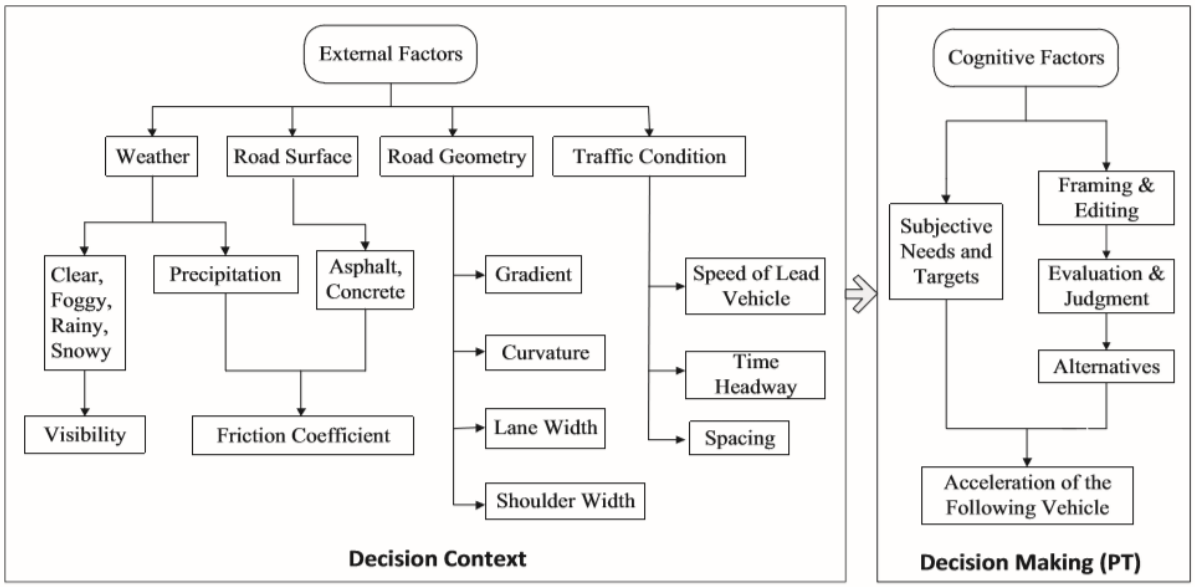

Figure 2. *The dynamic relationship between various factors that affect driving behavior, reprinted from [9]*

The work discussed in this report presents a framework that can be used to access the sensitivity of road geometry (gradient and curvature) on the road safety by reproducing safe human way of driving. This framework is in a computer simulation environment to generate a safe driving speed profile by just using road geometries. The framework also includes real-time importing of the required road geometry data from the online map databases like Open Street Maps and Google Maps. The simulation framework is developed for 2-lane rural roads, under free-flow traffic condition during daytime with the clear weather. This simulated speed profile could be used to warn a driver during an event of unintentional speeding in the real-world driving.

The framework developed in R language has three parts, discussed in detail in the later sections of the report:

1. Retrieving the driving route information from online map databases: Given the starting and the end coordinates of a driving route, the model queries and outputs the coordinates and the elevation of the equidistant waypoints along the route from online map databases.
2. Determining the limiting speeds due to the route geometry: Based on the route information from the online map database, for every equidistant waypoint, the model evaluates the safe driving speed conditions.
3. Simulating the safe driving speed profile: Based on the limiting speed conditions along the route, the model simulates a speed profile supposed to taken by a driver following safe driving conditions.

2. LITERATURE REVIEW

The dire necessity and the avidness to improve the road safety can be found in the literature base. The rapid development of embedded technologies like integrated infotainment and telematics continues to transform the driving experience into being more enriched, convenient, pleasurable and robust than ever before. Telematics has also become the core technology in an Intelligent Transportation System (ITS) [10]. ITS are smart systems that can not only improve road safety but also helps in avoiding traffic congestion, provision of effective traffic management and infotainment applications [11], [12]. ITS leverages the connected vehicles technology, which includes vehicle-to-vehicle and/or vehicle-to-infrastructure systems.

Road safety can also be improved by enhancing or even automating the vehicle system. This to an extent addresses the road safety concerns due to the human errors. Advanced driver assistance systems (ADAS) are such systems that incorporate features to prevent collisions and accidents by alerting the driver about the potential problems or to prevent collisions by taking over control of the vehicle. Apart from being warned about the potential collision, the alerts may include traffic warnings, correct lane departure, blind spots, excess speeding etc. [13]. An ADAS may use a combination of various sensors like proximity, image and/or optical sensors [14].

In a speed management perspective, an ADAS uses Intelligent Speed Adaptation (ISA). ISA is a system where, “the driver is warned and/or vehicle speed is automatically limited when the driver is, intentionally or inadvertently, traveling over the posted/safe speed limit” [5]. Studies have shown that ISA can have an optimistic effect on average driving speed and is also considered to help in speed harmonization [15][16]. It has been argued that although ISA warns the driver about speeding, the driver may not have enough time to avoid a collision. Intelligent speeding

prediction system (ISPS) is mathematical model suggested to address shortcomings of an ISA. ISPS tries to predict both intentional and unintentional speeding by measuring dynamic vehicle variables(speed, acceleration, throttle, and brake pedal inputs) and dynamic environmental variables (posted speed limit, traffic flow, and signal light) [17].

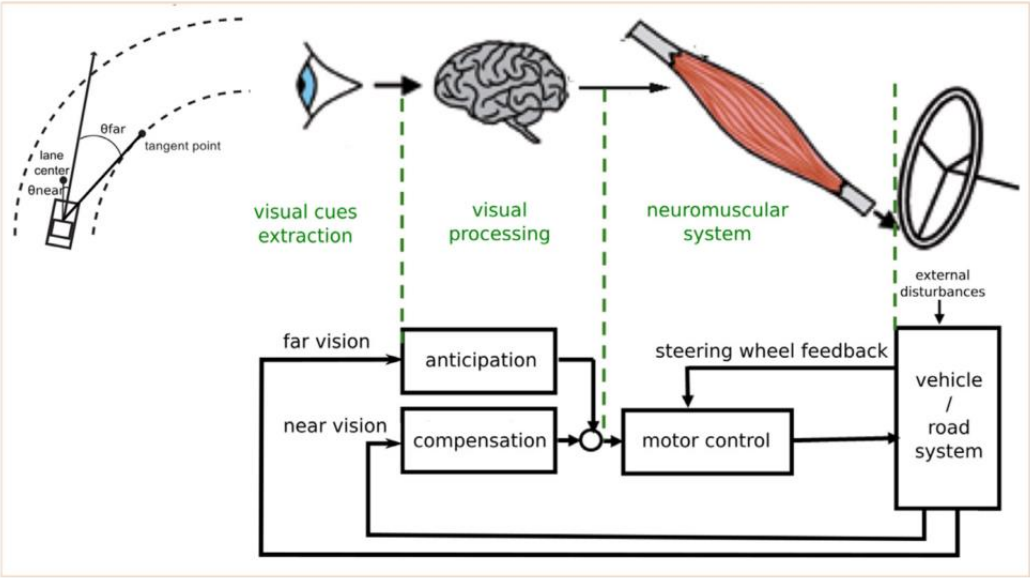

Figure 3. *The general architecture of the driver model for steering control, reprinted from [18]*

Good behaviors and attitudes in drivers can improve the road safety. Owing to the fundamental complexities, predicting the driver behavior is a challenging task for any ADAS. The developers of ADAS also has to consider the fact that introduction of ADAS changes to the driving environment and it may lead to changes in driver behavior [19]. The driver's state, behavior or intentions are predicted by measuring driving information and physiological signals of drivers. Driving information used are acceleration, speed, vehicle lane position, steering angles and steering wheel behaviors. Physiological signals, such as electroencephalogram (EEG), electromyography (EMG), heart rate variability (HRV), the percentage closure of the eyes averaged across a time (PERCLOS) can also be deployed [20],[21],[18].

The human way of driving, specifically applied to steering control is depicted Figure 3. This model can be extended to modeling the acceleration and braking behaviors as well. The three main processes involved include human perceptual, motor and cognitive processes. For a human driver, the visual and auditory perception acts as inputs, which is processed to determine where and how the driver wants to drive. This cognitive process leads to forces being applied to the steering wheels, brake and gas pedals through the neuromuscular systems [18]. It is essential to understand that the driver behavior is not static but evolves dynamically with situational as it is subjected not only to permanent but also temporary contributors[19].

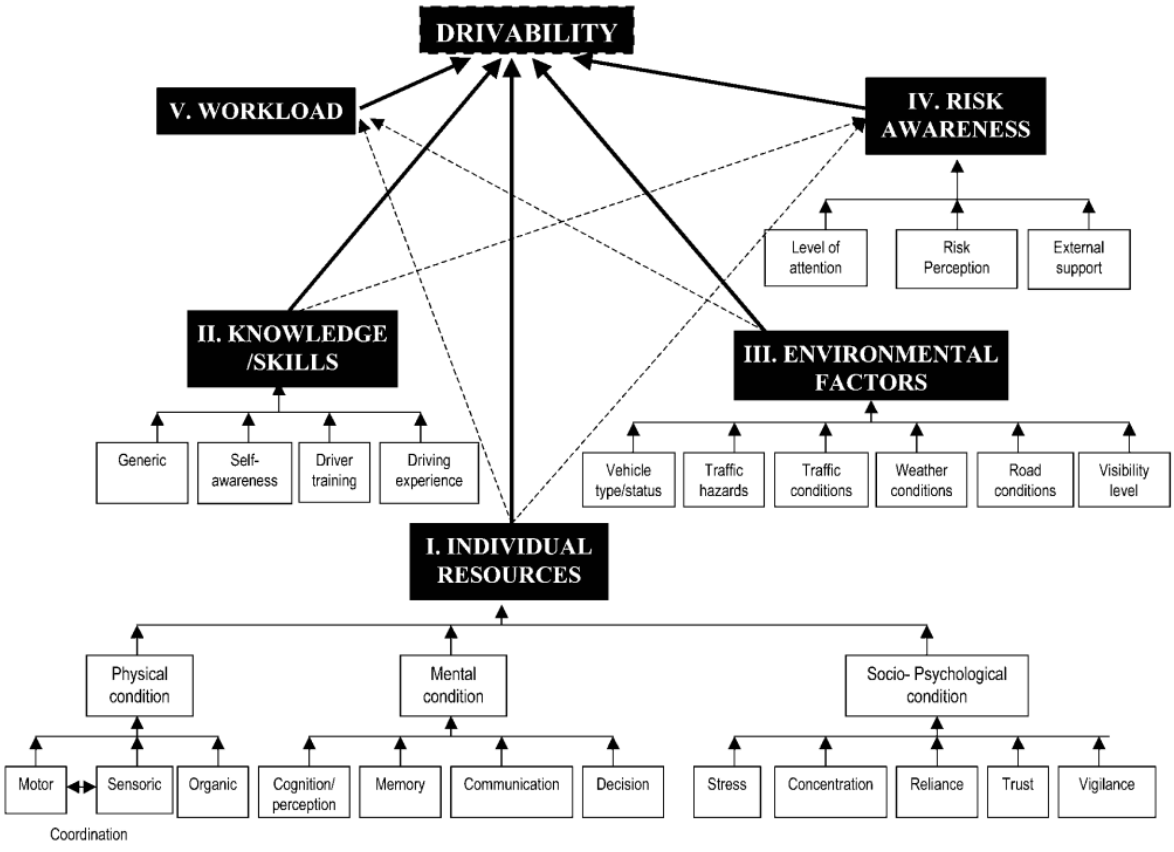

Figure 4. *DRIVABILITY: a concept for modeling dynamic driving performance, reprinted from [22]*

"DRIVABILITY", is a holistic concept for dynamic driver performance modeling and its contributors are represented in Figure 4. The surrounding environment affects a driver's behavior and at the same time, it affects each driver differently. These external factors are further elucidated in Figure 2. The driver performance is affected differently based on visibility depending on if the weather condition is clear, foggy, rainy or snowy [23], [24]. The visibility factor also influences the car following behavior of the drivers. For example, in a foggy weather condition, some driver in the fear of losing reference will follow a lead vehicle much closer than normal and some drivers may instead drop back to some larger distance headways accompanied with higher speed variability [25]. Precipitation decreases the friction coefficient of the road surface which in turn influences the vehicle's maneuverability. Studies have proven that precipitation tends to slow down the vehicles as the drivers become more cautious and aware of the decrease in traction [26], [27].

The driving performance is directly linked to the roadway layout and its geometry. The main influencing factors are lane and shoulder width, median existence, horizontal and vertical alignment [9]. Karlaftis et al. developed a mathematical model to predict the road accident rates based on the road geometric characteristics of rural roads [28]. Many independent studies can be found that tries to understand the relationship between the roadway design and the road accident rates [8], [29], [30], [31], [32] , [33], [34]. Miaou et at, proposed a Poisson regression model to establish the empirical relationships between truck accidents and road geometry, and traffic parameters. The model suggested that the accident rates are highly correlated with annual average daily traffic per lane, horizontal curvature and vertical curvature in rural interstate highways [35].

A sharp turn with a very small horizontal radius of curvature is found to have a high correlation to crash rates [33]. A study conducted on rural two-lane rural roads showed that

roadway crashes can be reduced by 33%, by widening lanes or shoulders on curves [34]. A report by Federal Highway Administration demonstrates that for unchanged roadway characteristics, the use of narrower lanes would lead to more crashes [36]. Likelihood of off-the-road crashes is more on freeways with narrower shoulder width [30]. Road medians that act a physical barrier separating opposing traffic streams, also influence the traffic flow and safety [37], [38].

Road curvature (both horizontal and vertical) and gradient certainly affects the driving behavior. Extra effort is required to keep the vehicle in the lane while driving through curves. Also, curves increase uncertainty in driving by reducing the visibility distances along the route, limiting anticipation the further route geometry and traffic situations. Drivers tend to enter the curve at high speeds, underestimating the turning angle involved. This results in abrupt braking behavior inside the curve. This indicates that the drivers do not reduce to a required safe speed. In such scenarios compared to the general speed signs, an advisory speeds signs are found more effective to make the drivers reduce speed. Advisory speed signs make the driver realize the safety reason for the warning or the restriction that may not be apparent or expected [39].

Even though such advisory warnings and marking are cost-effective methods, they cannot address the inherent inefficiencies of a poorly designed curve [34]. Bonneson et al. investigated the driver behavior on horizontal curves on rural two-lane highways to develop criteria for determining the curve advisory speeds, when they are needed for safe driving operations [40].

Similar to horizontal curvature, with rising gradients, the visibility distance is reduced. The increased uncertainty in road condition due to the shorter line of sight causes the driver to slow down the vehicle. Apart from the line of sight distance, along the rising gradients speed decreases due to gravity also. While going downhill the gravity increases the speed. If the driver does not compensate for the gravitational force, there will be the difference in the driving speeds between

vehicles in the same direction. This deteriorates the traffic flow and increases the probability of collisions [39].

Bidulka et al. have validated that horizontal curvature can be perceived differently when overlapped with vertical alignments. When a crest vertical curve overlaps with the horizontal curve, the latter appears sharper. Horizontal curve appears flatter when it overlaps with a sag vertical curve. This misperception is more predominant with sag vertical curves overlapping with horizontal curves and thus are more prone to roadway crashes [8].

Studies have been done to understand and estimate the vehicular speed based on the road curvatures and tangents to the curves. The speed adopted by drivers along a curve or a tangent not only depends on the curve geometry and tangent length respectively but also on the prior geometric conditions [41]. This knowledge can be used to design safer road geometric elements. The driving behavior is also affected by the number of bends or curves along the route, the percentage of the bend length to the total route and extremes (maximum and minimum) radius of curvatures along the route [33].

Summala states that drivers always adopts speed control to avoid dangerous and risky situations [42]. This can be observed on routes with narrower lane and shoulder widths. Compared to driving in a wider lane, the driver needs more mental effort to keep the vehicle in a narrower lane [43]. A study has shown that there is a reduction of 1.1mph in speed when the lane width decreases by 1 foot [44]. Medians give a perception of a protective barrier against the opposing traffic. Thus drivers tend to drive faster on roads with median [45], [46].

Influenced by the cost-effectiveness and ease of data collection, driver simulators along with traffic simulators are extensively used to develop understanding of the driver behavior and traffic flow characteristics [47]. The knowledge acquired is being used to design better ADAS and

self-driving cars. But these technologies still tend to use the reactive approach (mostly depending on computer vision) to address road safety concerns. Instead, studies could also focus on building predictive models where potential hazards will be recognized and prevented much before the condition fully develops. There is a lack of a framework for driving performance simulation that focuses on the road geometry data easily available from online map databases.

3. RETRIEVING THE DRIVING ROUTE INFORMATION FROM ONLINE MAP DATABASES

A driver who knows his current location and destination, but not the route can use various resources available online to fetch the optimized route. The popular online resources that are used are Google Maps, Waze, Mapbox, Mapquest, Maps.me, project-OSRM etc. These resources were analyzed and the feasibility of using them in a computer simulation model was determined. Mapbox's Direction API was chosen to get the route. Mapbox Direction API relies on the Open Street Maps (OSM), which is free and has no restriction on usage like the most popular Google Maps. The route obtained from the Mapbox Direction API is passed to Google Maps Elevation API, which will give the information of the elevation along the route. The steps in the model used to get the route information is shown in Figure 5.

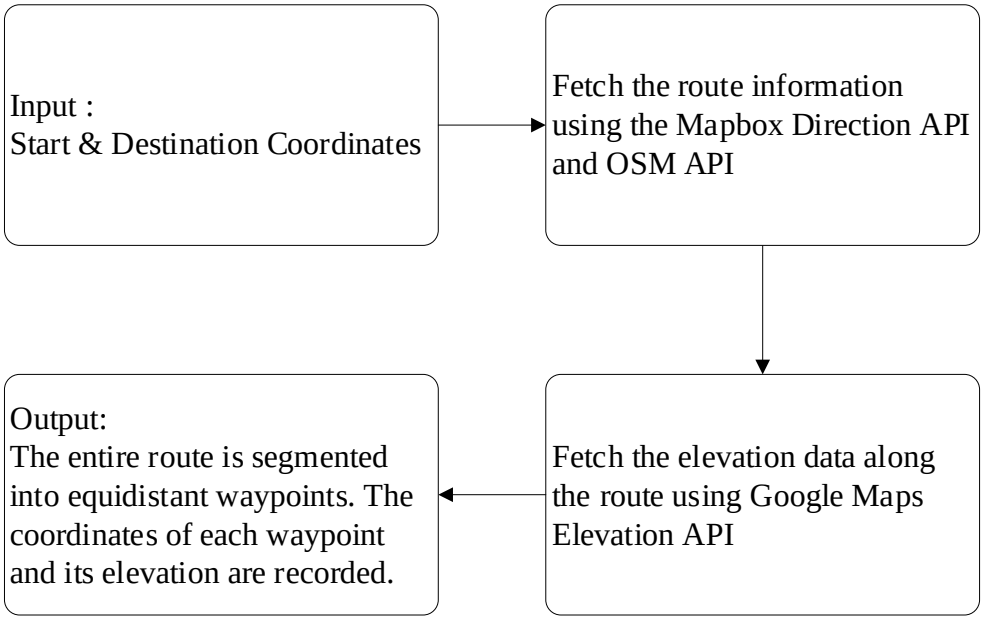

Figure 5. *The model for retrieving the route information*

The following subsections present more details about the OSM, Mapbox API and Google Maps Elevation API**.**

3.1 Open Street Map Data

Open Street Map [48] is a free to use and editable map of the whole world. OSM primarily uses crowdsourcing to maintain an up to date and rich variety of information about traffic lights, road types, points of interest etc. Currently, it has more than 4.7 million users across the globe [49]. Apart from these users, OSM relies on several commercial and government bodies for the map data, especially spatial landscape information. For the United States, the main data contributors are – TIGER Map Services, USGS, MASSGIS, and so on [50]. Apart from its distributed servers across the world, OSM has its primary servers located at the University College London (UCL) and Imperial College London (ILC). OSM is used as a base map for many companies like Facebook, Wikipedia, Mapillary, Foursquare, Strava, Runkeeper, Snapchat, Weather.com, etc.

The OSM map data can be accessed through several tools and application program interfaces (APIs). The data structure of OSM consists of the following core elements:

i. Nodes: Points on the map. Each node has a unique node ID and it contains a geographic coordinate (pairs of a latitude and a longitude). Thus, nodes are the smallest structural units in OSM.

ii. Ways: They represents roads, rivers, railway lines, roundabouts, walls, buildings, park etc. A way is nothing but an ordered list of nodes. If the last node in a way is also the first nodes on that way, the way is called a closed way, otherwise an open way. A way can have 2 to 2000 nodes in OSM.

iii. Relations: They represents how nodes, ways and/or even other relations work together. This data structure holds the relationship between two or more data elements. It consists of an ordered list of the data elements and tag that defines the

relation's meaning. Some examples of a relation are bus routes, cycle routes, numbered highways, etc.

The geographic attribute of each data element is described in OSM using tags. Tags are the metadata for the data elements representing the physical features. A tag consists of a string pair (a key and a value) stored in the format "key = value". A key represents the type of feature. For example, a node or a way belonging to a road, street or a path will have a "highway" as its key. The value stores information about the data element with respect to its key. For example, a road along which people live will have a value "residential".

There are multitudes of map features in OSM described using tags. Some of the commonly found features include – aerial ways, aero ways, amenities, barriers, buildings, highways, historic, military, railways, shops, waterways, tourism, etc. OSM is still growing in terms of the information it holds, thus there is a "lack of comprehensive data availability, completeness, and correctness" [51]. The OSM community has informal standards for key and value combinations that are commonly used. The lack of standardization complicates the feature extraction from OSM. Nevertheless, the raw OSM dataset is a valuable data source for roadway analytics.

There is no argument that OSM provides a lot of good content, but it is hard to have an effective quality control as in any crowdsourced database. There is no company or organization dedicated to alter and complete the data. But new features are being added constantly by a group of developers. OSM is accessed by many third-party applications for customization and specific uses [52]. Leveraging the customizable map features, many thematic maps have been created like OpenCyleMap, OpenSkiMap, Wheelmap, Philly Fresh Food map, etc. [53].

Value	Data Element	Description
motorway	way	A restricted access major divided highway, normally with 2 or more running lanes plus an emergency hard shoulder.
trunk	way	The most important roads in a country's system that aren't motorways. (Need not necessarily be a divided highway.)
primary	way	The next most important roads in a country's system.
secondary	way	The next most important roads in a country's system
tertiary	way	The next most important roads in a country's system. (Often link smaller towns and villages)
unclassified	way	The least most important through roads in a country's system – i.e. minor roads of a lower classification than tertiary, but which serve a purpose other than access to properties.
residential	way	Roads which serve as an access to housing, without the function of connecting settlements. Often lined with housing.
service	way, relation	For access roads to, or within an industrial estate, campsite, business park, car park etc.
footway	way	For designated footpaths; i.e., mainly/exclusively for pedestrians. This includes walking tracks and gravel paths.
give_way	node	A "give way", or "Yield" sign
stop	node	A stop sign
traffic_signals	node	Lights that control the traffic signals

Table 1. *List of some common values found with the "highway" key in OSM*

For the roadway analytics, the data elements with tag key as “highway” should be queried. Among the various commonly used values against “highway” key, some of the critical once are listed in Table 1.

OSM offers a few REST (Representational state transfer) web services to interact with its data. REST services are based on HTTP verbs to perform certain actions, such as reading, creating, updating, and deleting records. GET requests are used to *read* data from the server; PUT requests are used usually to *update* data; POST requests are used to *create* a new record; and DELETE requests are used to *delete* a record. Some of the useful OSM’s HTTP GET requests are discussed below. All the response will be in outputted in XML format.

- Retrieving map data by bounding box: “A bounding box (usually shortened to bbox) is an area defined by two longitudes and two latitudes” [54]. This API call can be used to get all the nodes inside a bbox; all the ways that reference at least one node in the bbox; and all the relations that references the nodes and the ways in the bbox.

  Syntax: GET *http://api.openstreetmap.org/api/0.6/node/map?bbox={min Longitude},{min Latitude},{max Longitude},{max Latitude}*

- Retrieving data specific to a node: This can be used to extract the geographic coordinates and the associated tags with a given node. This can be used to get the traffic signals, stop signs, etc.

  Syntax: GET *http://api.openstreetmap.org/api/0.6/node/{node ID}*

- Retrieving all the ways referenced to a given node: This can be used to get information like name of the road, number of lanes, speed limits, type of road, etc. pertaining to each way associated with the node.

  Syntax: GET *http://api.openstreetmap.org/api/0.6/node/{node ID}/ways*

3.2 Mapbox Directions API

Mapbox is a commercial mapping service provider for mobile and web application. The biggest competitor to Mapbox is Google Maps. Mapbox is chosen by its users as the Google Maps does not offer any form of customization. It is popular for its variety of customization options, including the color theme, features, displayed information, and even let upload and display user-specific data. The Mapbox services are being used by companies like Lonely Planet, Snapchat, Weather.com, Lyft, Bloomberg and so on. Mapbox gets its roads, buildings, and places around the globe from open street maps database [54]. Therefore, all the data updates in OSM will be directly reflected in Mapbox. Mapbox also offers services where it displays the real-time traffic conditions on the map. This is possible using the telemetry from all the Mapbox SDKs running on mobile devices, where its users' map data and device locations are anonymously captured.

For the purpose of the framework discussed in this report to simulate the driving speed, the Mapbox direction API is used. It's basically a routing service that is used in various mobile and web applications that help people to get from one place to another with turn-by-turn directions on the map. Apart from Mapbox, the other few routing services are offered based on open street map data by companies like MapQuest, Skobbler, Graph Hopper Maps etc. Mapbox was chosen as its reliable and offer a lot more other development kits in Android and iOS platforms, which could be later easily deployed in the application discussed later this is report. Apart from the direction API, Mapbox offers other direction related services like Mapbox Matrix API, Mapbox Optimization API and Mapbox Map Matching API. The Matrix API outputs the travel times between points on the map (up to 25 coordinates points). The Optimization API provides a solution to the traveling salesperson problem. Map Matching API outputs the closest road path on the OSM map if a set of coordinates are inputted. All the direction related services' responses are obtained in GeoJSON

format and are powered by the Open Source Routing Machine (OSRM). It is a completely free and open source routing algorithm in C++ programming language developed for open street maps data[55].

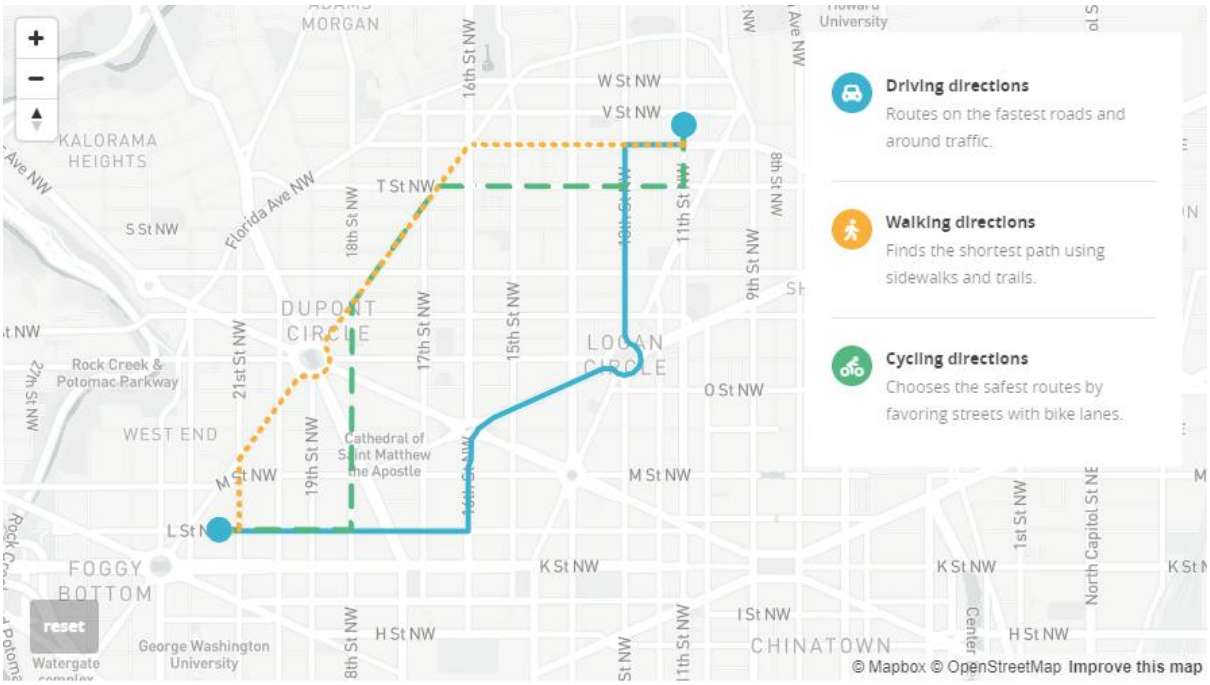

Figure 6. *Various routing profiles available through Mapbox Direction API, reprinted from [56]*

The Mapbox Direction API could be used for walking, cycling or driving profiles as shown in Figure 6. This report discusses only the driving profile. This API is a commercial service and hence has a limited 50,000 free requests per month after which it charges $0.5 per 1000 additional direction request. The usage is tracked through the user account's access token, which has to be inputted in each Mapbox API request. The HTTP GET API syntax used the request routing data from Mapbox in the speed profile simulation framework is as follows

https://api.mapbox.com/directions/v5/mapbox/driving/{coordinates},?alternatives=false&geometries=polyline&steps=true&overview=full&annotations=distance&access_token={access-token}

Where *{coordinates}* represents a semicolon-separated list of *{longitude}, {latitude}* coordinate pairs to visit in order. A maximum of 25 coordinates is supported currently. The other parameters are chosen so as to get all the OSM nodes and its IDs along with the outputted route and also to identify the nodes which are intersections. The API response will also give the distance between each OSM node. The node IDs obtained here can be used to get used to retrieve more information about the route using the OSM APIs as discussed in the previous sections.

Figure 7. *Visualization of route information on the map*

The responses from the Mapbox direction API and the OSM API along the highway TX 21 through the city of Bryan, Texas in visualized in Figure 7. At each node, the information captured and displayed include speed limits, number of lanes and highway type. The blue and red markers correspond to OSM nodes, where the red ones denote intersection.

3.3 Google Maps Elevation API

The Elevation API provides elevation data for locations on the surface of the earth, including depth locations on the ocean floor (which return negative values). In those cases where Google does not possess exact elevation measurements at the precise location requested, the service will interpolate and return an averaged value using the four nearest locations. The API can be used to query discrete locations on the earth for elevation data. Additionally, it is possible to request sampled elevation data along paths, allowing you to calculate the equidistant elevation changes along routes [57]. This feature is utilized in the simulation framework described in this report.

This API is also a commercial service and hence has a limited 2,500 free requests per day after which it charges $0.5 per 1000 additional direction request. The usage is tracked through the user account's API key, which has to be inputted in each API request. The route path obtained from the Mapbox direction API is inputted to the Google Maps Elevation API using the following syntax.

https://maps.googleapis.com/maps/api/elevation/json?&key={API_Key}&path=enc:{mapbox_route }&samples={number_of_samples}

Number of samples value inputted in the API request determines the equidistance between the waypoints. The maximum number of samples currently supported by the API is 512. The response to the API request is in JSON format and it is parsed to get coordinates of the equidistant waypoints and its elevation along the driving route.

4. DETERMINING THE LIMITING SPEEDS DUE TO ROUTE GEOMETRY

Previous research studies have come up with empirical formulas that give the driving speed based on the horizontal radius of curvature and the line of sight distances due to changes in the vertical elevation [57]. These equations were developed through experimental studies that captured the driving speed profiles of drivers following the safe driving conditions. These equations are incorporated in the speed profile simulation framework to determine the limiting speeds for safe driving conditions at each waypoint along the route.

The list of variables used in the calculations of the limiting speeds is listed in Table 2.

Variables	Description
α	Horizontal turning angle (degree)
R_h	Horizontal radius of curvature (m)
S_r	Limiting speed due to horizontal geometry (m/s)
Smax	Limiting speed at the waypoint (m/s)
θ	Vertical turning angle (degree)
Rv	Vertical radius of curvature (m)
θ_g	Limiting vertical turning angle
h	Eye sight height = 1.2 m
P_z	Maximum line of sight distance to a crest (m)
S_v	Limiting speed due to horizontal geometry (m/s)

Table 2. *List variables used in the limiting speed calculations*

For each waypoint obtained as discussed in the previous section, the horizontal turning angle and the radius of curvature are calculated. This radius of curvature is used to find the limiting speed conditions due to the horizontal geometry. From the elevation data, the waypoints

corresponding to the crests due to the vertical curvature are found. The limiting speed conditions due to the crest vertical curvature are found. The limiting speed conditions for the route is finally determined by considering both the vertical and horizontal limiting speeds and taking the smaller values, as described in the following equation.

$$Smax = \min(\ S_r\ , S_v\ )$$

The model used to determine the limiting speed along the route is shown in Figure 8.

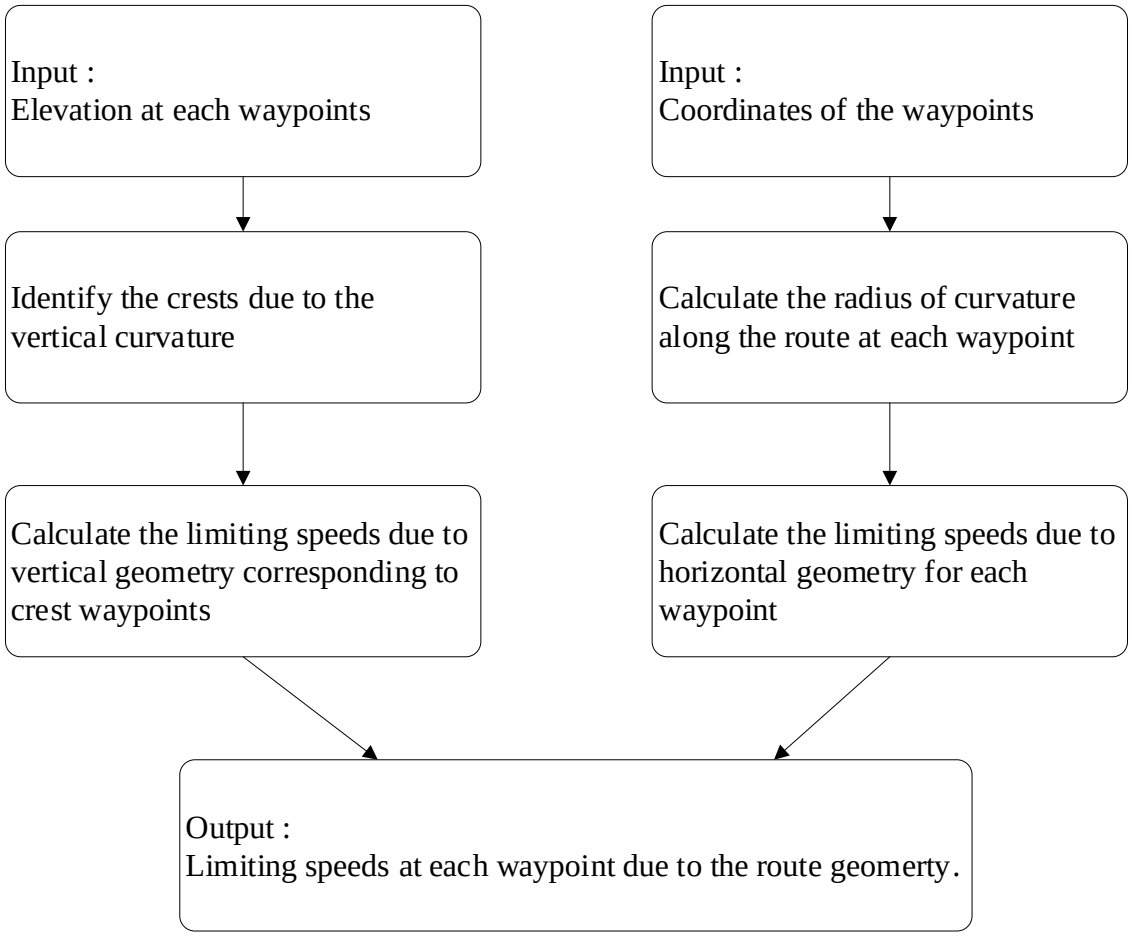

Figure 8. *The Model for determining the limiting speed due to route geometry*

4.1 Limiting Speeds Due To Horizontal Geometry

Horizontal turning angle α, is calculated between three consecutive waypoints as shown in Figure 8, using the following equations. The coordinates of waypoint 1, waypoint 2 and waypoint 3 are denoted by $(latitude_1, longitude_1)$, $(latitude_2, longitude_2)$ and $(latitude_3, longitude_3)$ respectively in horizontal plane.

$$\overrightarrow{H_1} = (longitude_2 - longitude_1)i + (latitude_2 - latitude_1)j$$

$$\overrightarrow{H_2} = (longitude_3 - longitude_2)i + (latitude_3 - latitude_2)j$$

$$\alpha = \cos^{-1} \frac{\overrightarrow{H_1}.\overrightarrow{H_2}}{|\overrightarrow{H_1}||\overrightarrow{H_2}|}$$

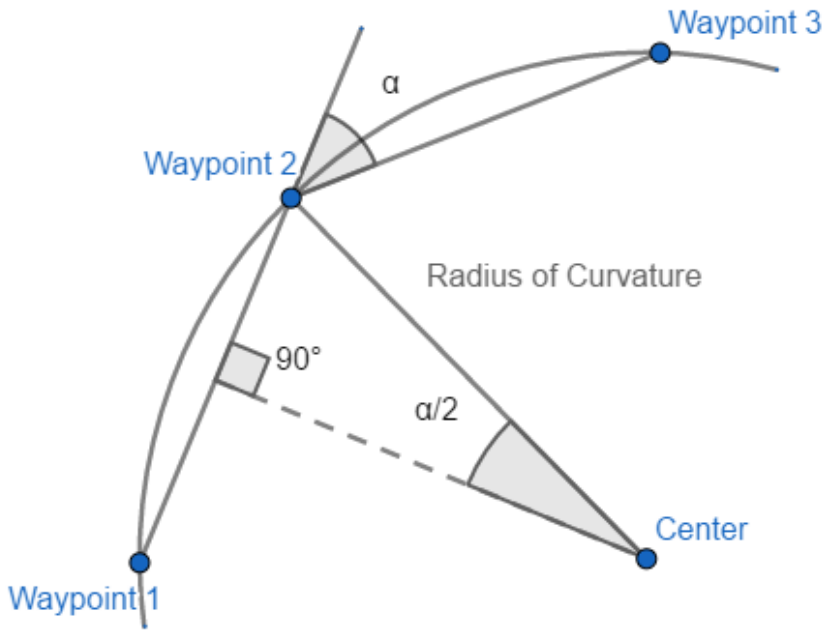

Figure 9. *Turning angle and radius of curvature at the waypoint 2*

The horizontal radius of curvature at waypoint 2 in Figure 9 is calculated using the following equation. These calculations are made by assuming that 3 consecutive waypoints are on a perfect circular curvature.

$$R_h = \frac{Distance\ between\ the\ waypoints\ /\ 2}{\sin {}^{\alpha}/_{2}}$$

The limiting speed based on the horizontal radius of curvature is found using the following empirical equation [57].

$$S_r = 9.15(\log_{10} R_h)^2 + 17.68 \log_{10} R_h - 11.93$$

4.2 Limiting Speeds Due To Vertical Geometry

Vertical turning angle θ, is calculated between three consecutive waypoints using the following equations. The coordinates of waypoint 1, waypoint 2 and waypoint 3 are denoted by $(distance_1, elevation_1)$, $(distance_2, elevation_2)$ and $(distance_3, elevation_3)$ respectively in vertical plane.

$$\overrightarrow{V_1} = (distance_2 - distance_1)i + (elevation_2 - elevation_1)j$$

$$\overrightarrow{V_2} = (distance_3 - distance_2)i + (elevation_3 - elevation_2)j$$

$$\theta = \cos^{-1} \frac{\overrightarrow{V_1}.\overrightarrow{V_2}}{|\overrightarrow{V_1}||\overrightarrow{V_2}|}$$

From the elevation data, the waypoints corresponding to the crests due to the vertical curvature are found. The vertical radius of curvature at a waypoint corresponding to a crest is evaluated based on the following equation.

$$R_v = \frac{Distance\ between\ the\ waypoints\ /\ 2}{\sin {}^{\theta}/_{2}}$$

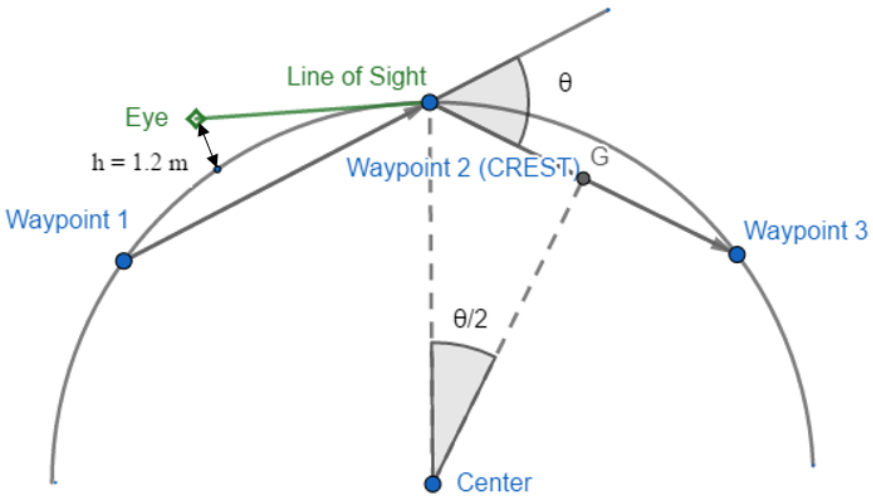

Figure 10. *Crest Vertical Curvature*

Using the vertical radius of curvature and turning angle, the maximum line of sight distance is evaluated as follows and depicted in Figure 10. This line of sight distance corresponds to the distance from the crest point where the driver realize the presence of the crest vertical curvature [57].

$$P_z = \begin{cases} \sqrt{(R_v + h)^2 - R_v^2}, & \theta \geq \theta_g \\ \frac{\theta^2 R_v + 2.4}{2\theta}, & \theta < \theta_g \end{cases} \qquad \text{Where, } \theta_g = \frac{1.55}{\sqrt{R_v}}$$

The limiting speed based on the vertical line of sight distance is found at the maximum line of sight distance before the crest waypoint point using the following empirical equation. For all the non-crest waypoints, there is no limiting speed due to vertical curvature [57].

$$S_v = 1.25(36.51 \ln P_z - 78.09)$$

5. SIMULATING THE DRIVING SPEED PROFILE

The limiting speeds due to route geometry at each waypoint along with the actual statutory speed limits and the stop sign locations are inputted to the simulation model. It is an iterative model, where for every 1 meter along the route the current speed and the acceleration for the next 1meter along the route is determined. The list of variables used in the model is explained in Table 3.

Variable Name	Description
D	The total distance of the route. (m)
$Smax_k$	Limiting speed at the k^{th} equidistant waypoint. (m/s)
S_i	Simulated speed in i^{th} iteration or i^{th} meter from the starting point. (m/s)
a_i	Simulated acceleration of the next 1 meter distance in the i^{th} iteration. (m/s^2)
R_i	Response Distance during the i^{th} iteration. (m)
n_i	Number of waypoints within the response distance from the current position or i^{th} meter.
pn_i	Number of waypoints for which the speed has been simulated already. This is equivalent to number of waypoints already traveled along the route in the real driving scenario.
$(nBD_k)_i$	The distance that has to be traveled by the vehicle when no brake and accelerator applied knowing the S_i and $Smax_k$. (m)
$(ba_k)_i$	Breaking acceleration of the k^{th} waypoint in the i^{th} iteration. (m/s^2)
$(d_k)_i$	Distance from the k^{th} waypoint in the i^{th} iteration. (m)

Table 3. *List of variables used in speed profile algorithm*

From previous studies, it is known that it takes 7 seconds for a driver to respond to any visual stimuli while driving [58]. In other words, a driver will have a dynamic response distance based on the varying driving speed as described in the following equation.

$$R_i = 7S_i$$

At the beginning of each iteration, the current speed is evaluated using the following equation.

$$S_i^2 = S_{i-1}^2 + 2a_{i-1}$$

For every iteration of the simulation, the current speed is compared to the limiting speeds corresponding to the waypoints within the response distance from the current vehicle position. If the current speed(S) is higher than the limiting speeds (Smax) found within the response distance, the model determines if the lower speed (Smax) should be attained without applying braking by evaluating the distance that has to be traveled by the vehicle when no brake and accelerator applied using the flowing equation.

$$(nBD_k)_i = S_i^2 - Smax_k^2$$

If braking is required, the model evaluates the braking acceleration $(ba_k)_i$ for the kth waypoint within the response distance using the following equation. In this scenario, the rate acceleration for the ith iteration will be $\min_{\forall k}(ba_k)_i$

$$(ba_k)_i = \frac{Smax_k^2 - S_i^2}{2(d_k)_i}$$

To avoid intricacies, the following assumptions are incorporated into the simulation model.

- The vehicle is always starting from a rest condition (speed of 0 m/s) with an acceleration of 1 m/s^2.
- There is no traffic along the route, i.e. free flow speed condition.
- The rate of acceleration is 1 m/s^2 (from a lower speed to a higher speed when required)
- Under no braking and no accelerator condition, deceleration is 0.5 m/s^2.
- When braking is necessary, deceleration will be a constant value greater than 0.5 m/s^2 depending upon the current speed and the required speed after braking (which can be the limiting speed based on road geometry or the speed limits).

Based on the inputs and the equations described above, the model basically determines if the vehicle is going to be in any one of the following states of motion for every 1 meter along the entire route, as shown in Figure 11.

1. Accelerating to a higher speed at a rate of 1 m/s^2.
2. Maintain the current speed.
3. Decelerating to a lower speed (no braking nor acceleration condition) at a rate of 0.5 m/s^2.
4. Braking to reduce the speed at a deceleration rate higher than 0.5 m/s^2.

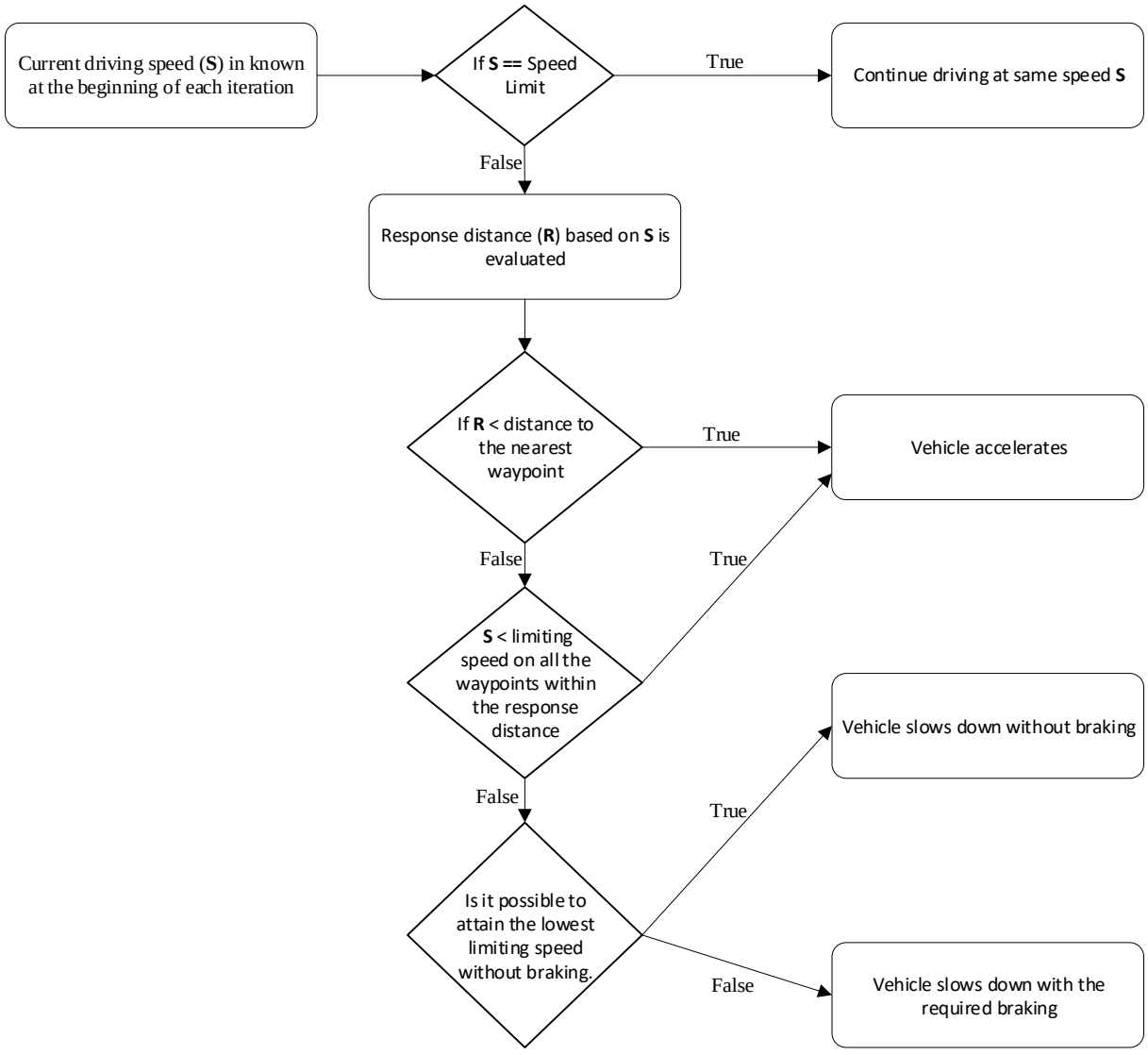

Figure 11. *Flowchart for the four possible states of motion in the simulation model*

The algorithm developed in R programming language can be found in the Appendix of this report and the pseudocode for the same is described below. The algorithm is also depicted using a flowchart in Figure 12.

$a_1 = 0, S_1 = 0, pn_1 = 1;$
for i = 2 to D **do**
evaluate S_i;
if $S_i < 0$ **then**
$S_i = 0$;
end if
evaluate R_i
n_i
pn_i;
if $n_i == 0$ **then**
if $S_i < Sl_i$ **then**
$a_i = 1$;
else
$S_i = Sl_i$;
$a_i = 0$;
end if
else
for k = (pn_i+1) to n_i **do**
if $S_i > Smax_k$ **then**
evaluate $(nBD_k)_i$, $(d_k)_i$;
if($(nBD_k)_i > (d_k)_i$)
evaluate $(ba_k)_i$;
else
$(ba_k)_i = 0$;
end if
end if
end for
if $(ba_k)_i == 0 , \forall$ k **then**
if $S_i < Sl_i$ **then**
$a_i = 1$;
else
$S_i = Sl_i$;
$a_i = 0$;
end if
else
$a_i = \min(ba_k)_i \forall$ k;
end if
end if
end for

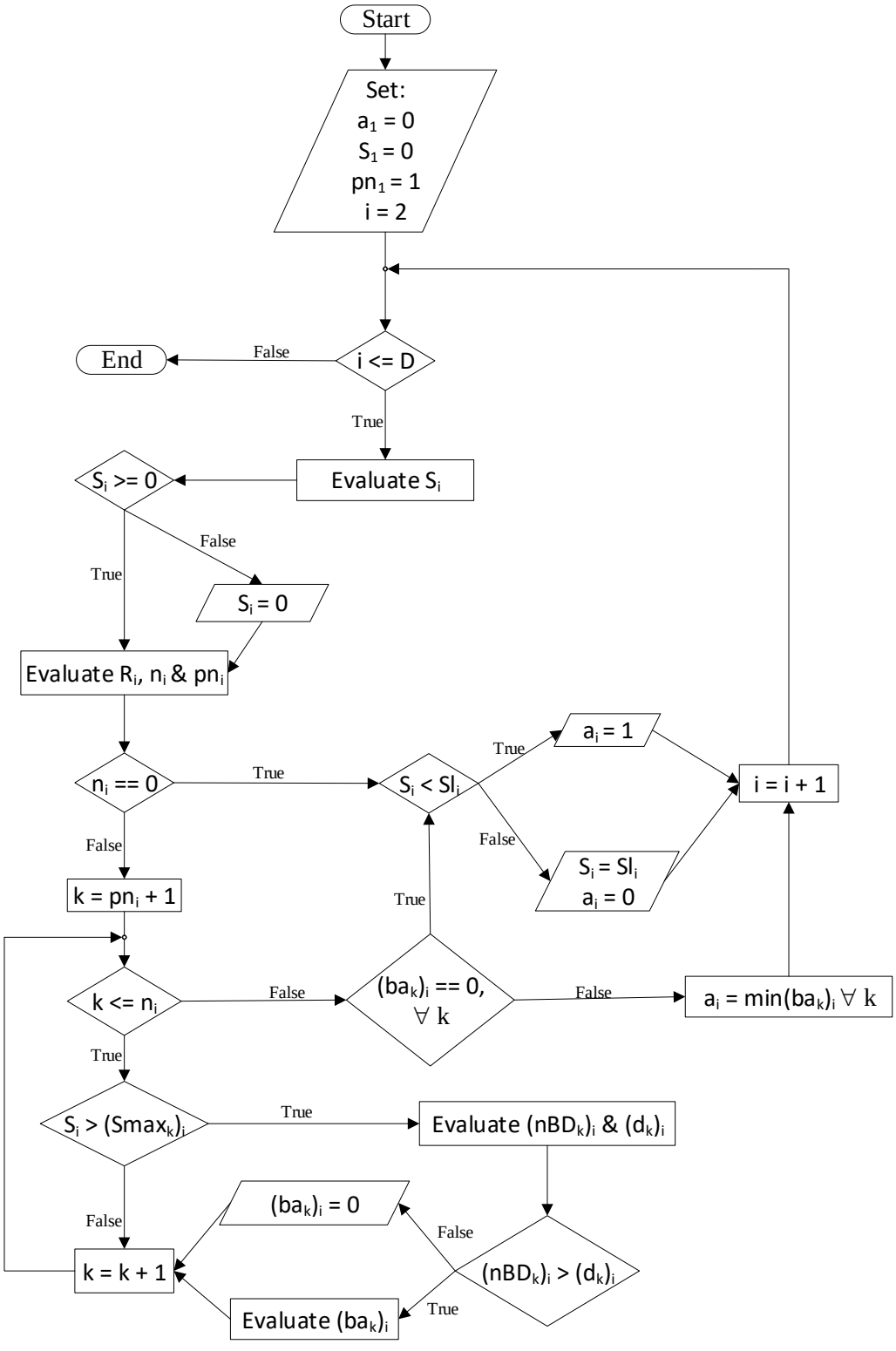

Figure 12. *Flowchart for the speed profile algorithm*

6. SIMULATION RESULTS AND VALIDATION

The developed simulation framework discussed in the previous sections is summarized in Figure 13. Basically, if the model is inputted with latitude and longitude coordinates of starting and end point for a route, the model will output every 1 meter along the route the simulated driving speed under a stipulated safe driving condition.

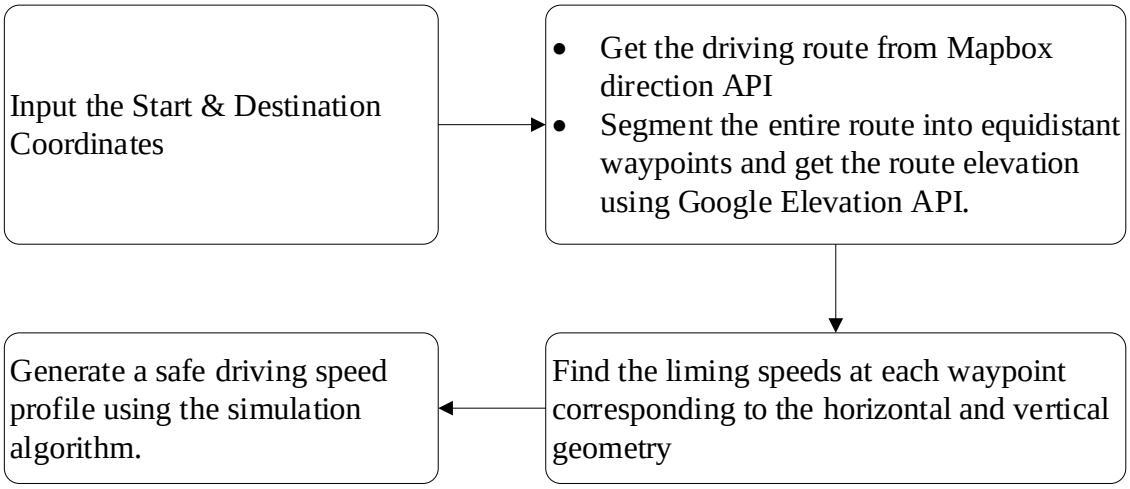

Figure 13. *The simulation framework for generating a safe driving speed profile*

	Route 1	Route 2	Route 3	Route 4
Name of Road	Sandy Point Road, Bryan, TX	Sandy Point Road, Bryan, TX	FM 159, Brazos County, TX	FM 159, Brazos County, TX
Total Distance (Km)	10.675	10.972	15.542	16.409
Driving Direction	West	East	South	North
Starting Coordinate	(30.678184, -96.391981)	(30.699651, -96.489106)	(30.506977, -96.197433)	(30.393913, -96.214739)
Ending Coordinate	(30.699787, -96.488752)	(30.677187, -96.389613)	(30.395658, -96.231938)	(30.498911, -96.198245)

Table 4. *Details of the routes used to demonstrate the simulation model*

Two rural roads with 2-lanes and no median were identified for validating the simulation model as detailed in Table 4. The simulated speed profile is validated on both the roads by safely

driving on both the lanes (i.e. both directions). For standardization, all the routes were segmented into equidistant points that are 72m apart. The speed limits and the stop sign locations were manually inputted to the model as the required data was not updated in the OSM database.

The OSM nodes retrieved using Mapbox direction API for Route 1 and Route 2 along the Sandy Point Road, Bryan, Texas is shown on the map in Figure 14. The equidistant waypoints obtained using the Google Elevation API is shown in Figure 15.

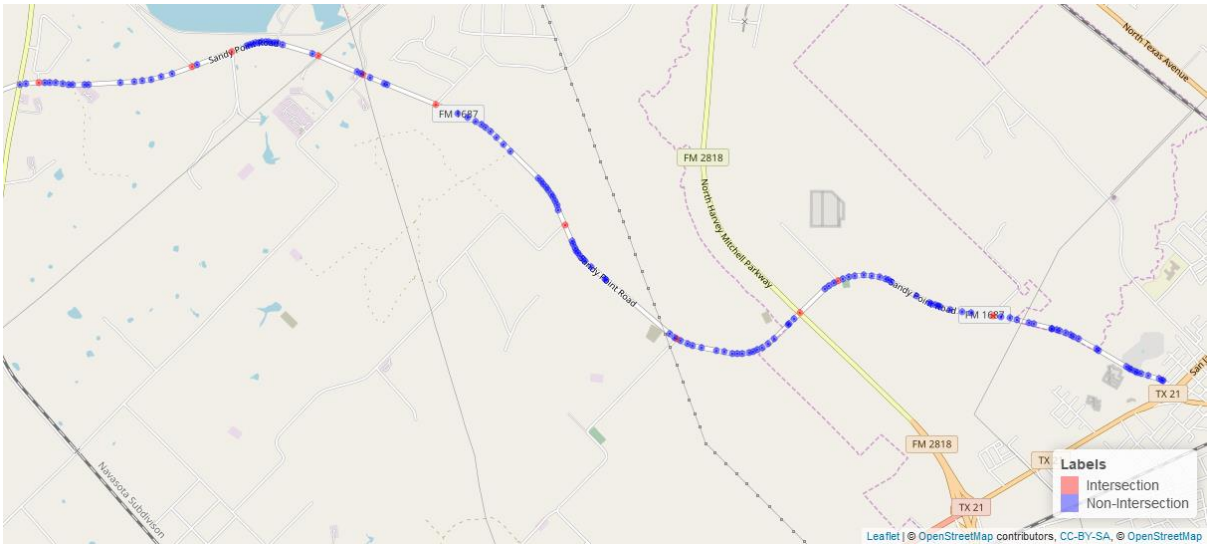

Figure 14. *OSM nodes for Route 1 and Route 2 along Sandy Point Road, Bryan, TX*

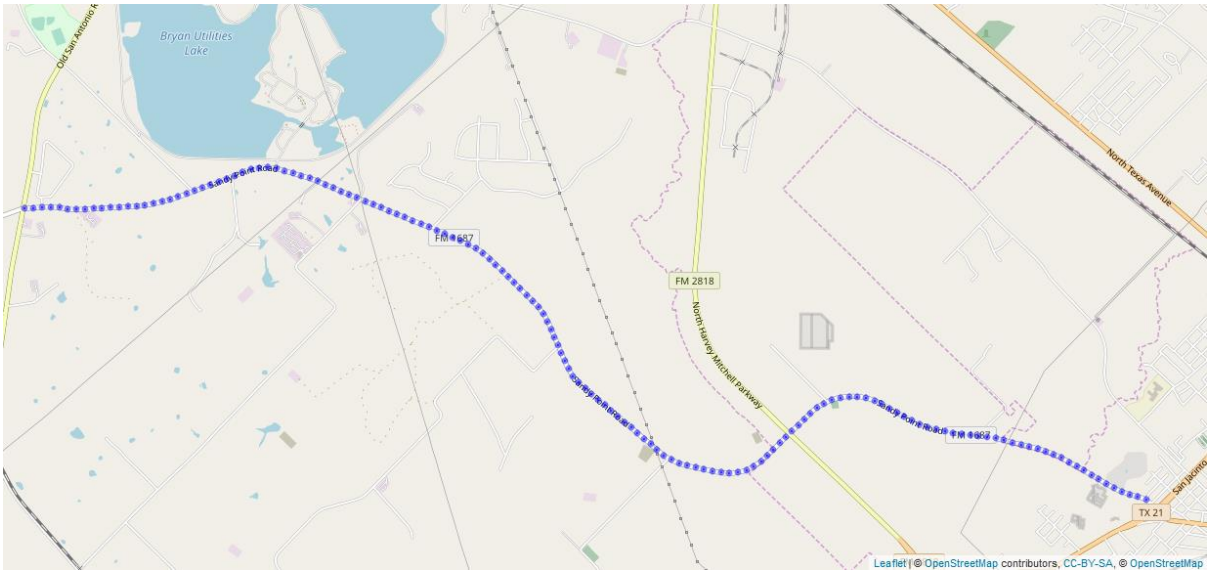

Figure 15. *Equidistant waypoints for Route 1 & Route 2 along Sandy Point Road, Bryan, TX*

The horizontal turning angle and radius of curvature for Route 1 is shown in Figure 16 and Figure 17 respectively. In the radius of curvature plot, all the radiuses above 2000m have been trimmed of the plot and marked as 2000m, so as to fit all the points in a graph. The horizontal line marked at the radius of 701m corresponds to the limiting curvature below which the maximum speed limit of 112.64 Km/hr (70 mph) for the route is not a safe driving speed. For the waypoints below this line will have a limiting speed less than the 112.64 Km/hr.

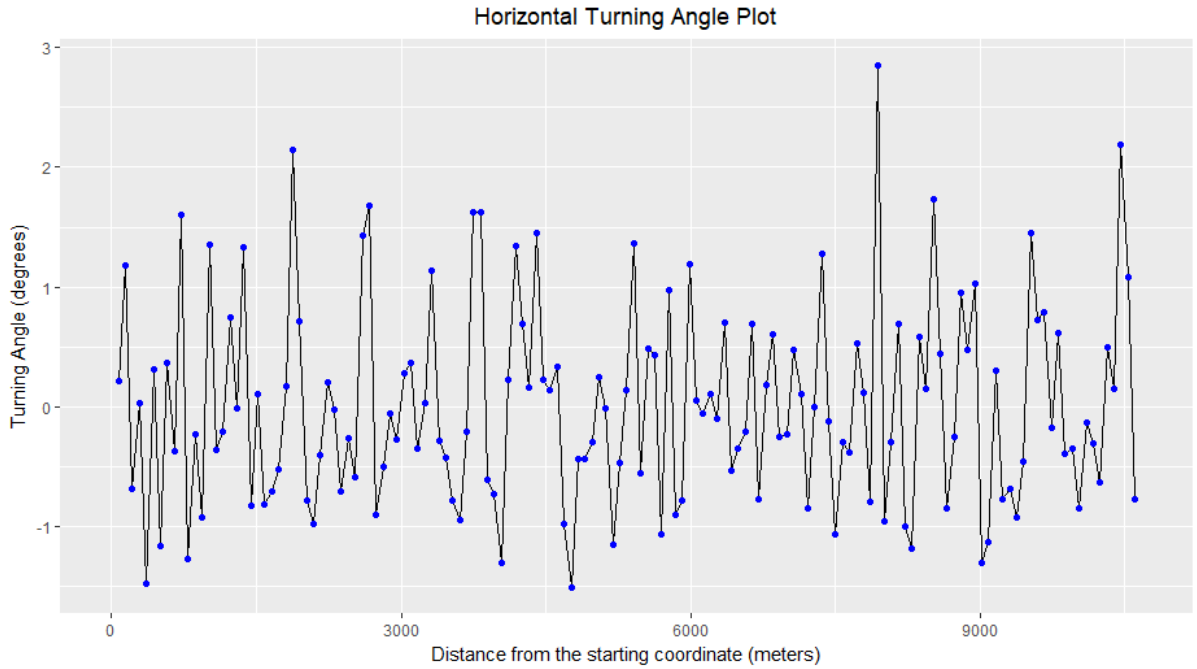

Figure 16. *Horizontal turning angle vs Distance plotted for Route 1*

Figure 18 shows the elevation plot for Route 1. The crest due to vertical curves is marked with a red marker. The blue marker represents the maximum line of sight distance from the adjacent crest point. Limiting speeds for Route 1, based on the road geometry is plotted in Figure 19. A cut-off of 120 Km/hr (75 mph) as the maximum speed limiting speed is used for the plotting purpose.

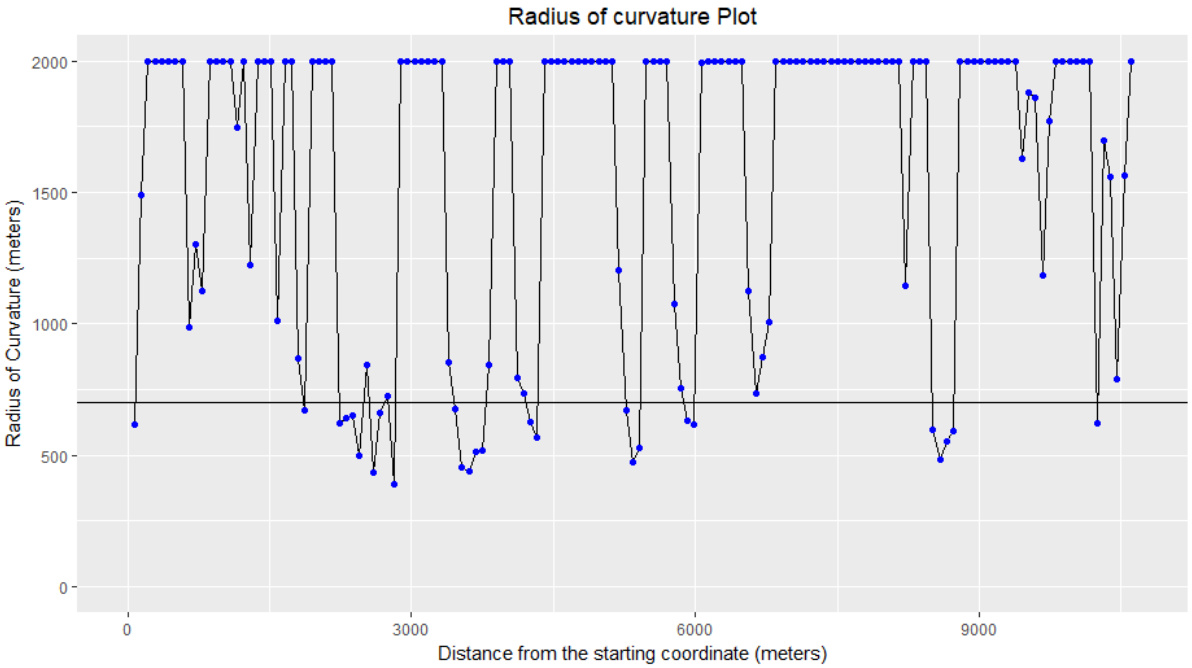

Figure 17. *Horizontal turning angle vs Distance plotted for Route 1*

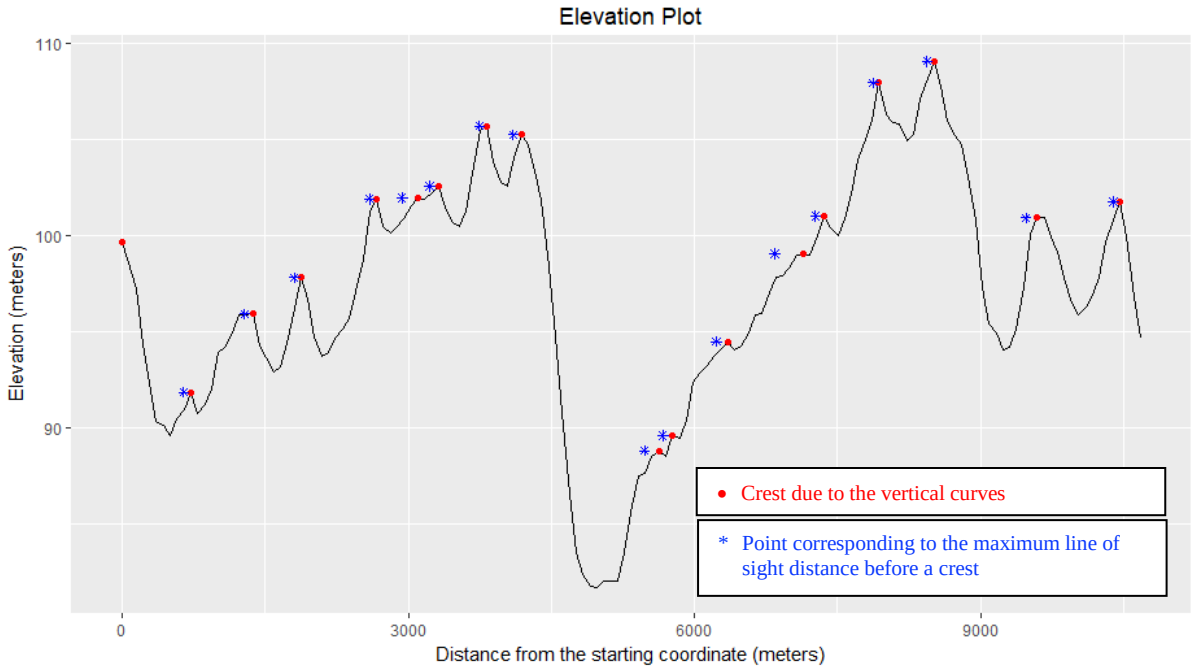

Figure 18. *Elevation vs Distance plotted for Route 1*

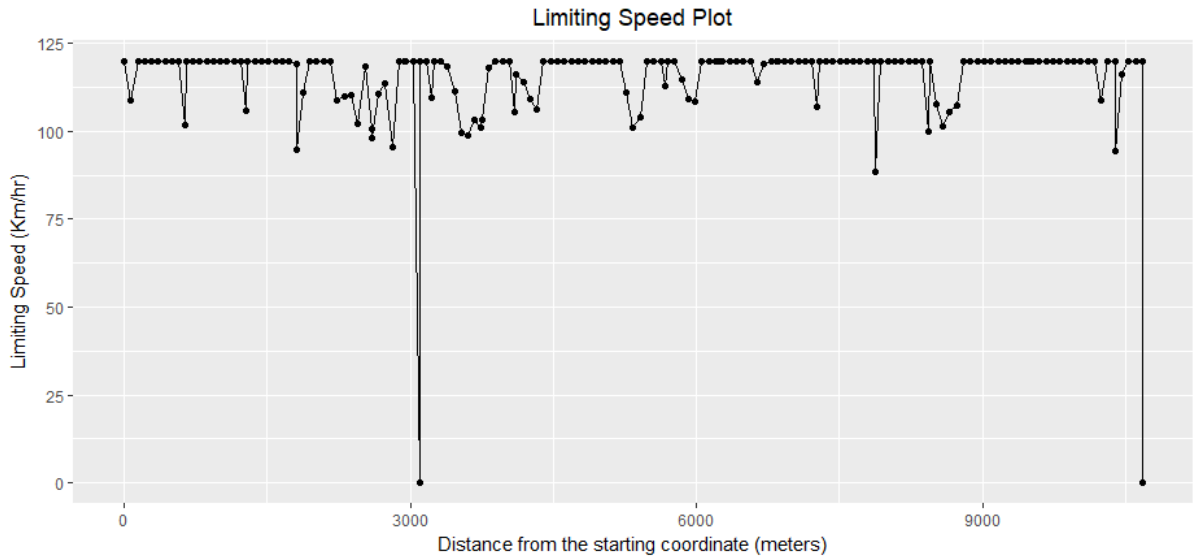

Figure 19. *Limiting Speed due to road geometry vs Distance plotted for Route 1*

Based on the simulation algorithm, for Route 1 the generated acceleration is plotted in Figure 20. The speed profile, generated by the simulation algorithm is plotted in Figure 21. The speed profile observed during a trial drive along the same route is shown in Figure 22.

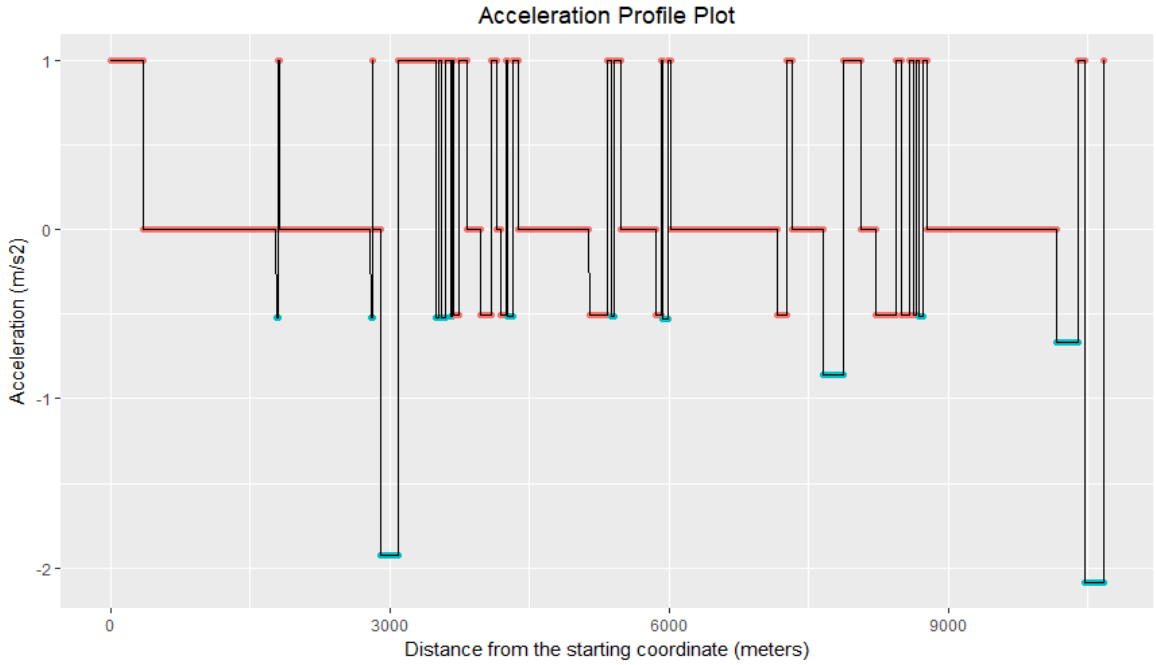

Figure 20. *Simulated Acceleration vs Distance plotted for Route 1*

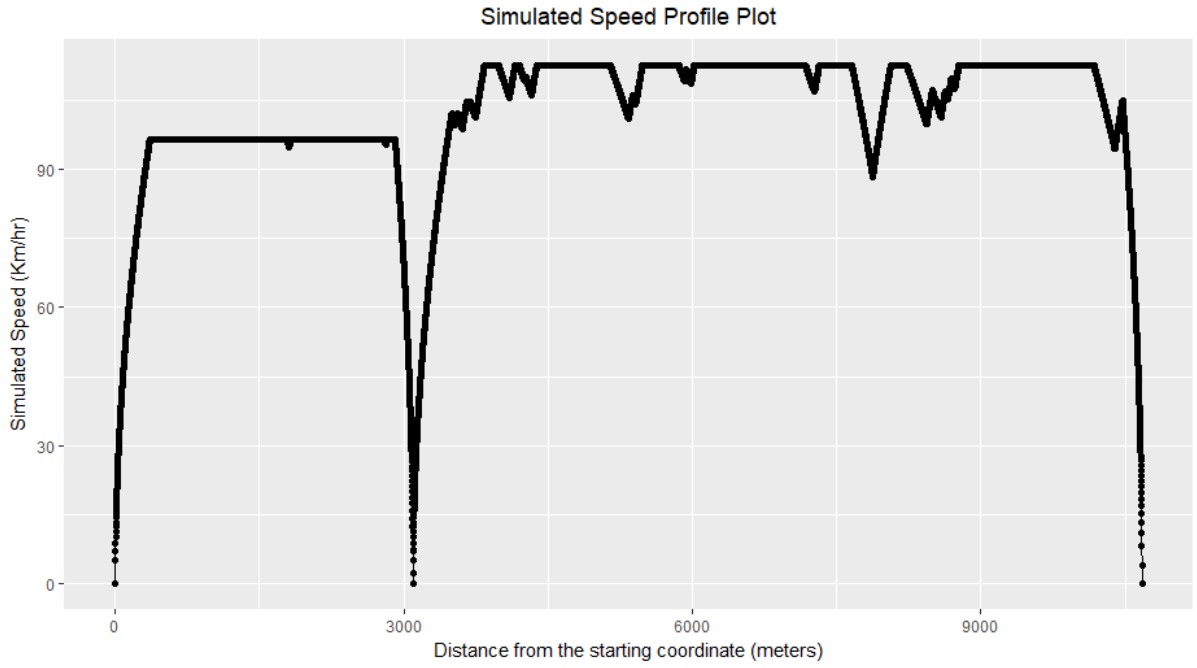

Figure 21. *Simulated Speed vs Distance plotted for Route 1*

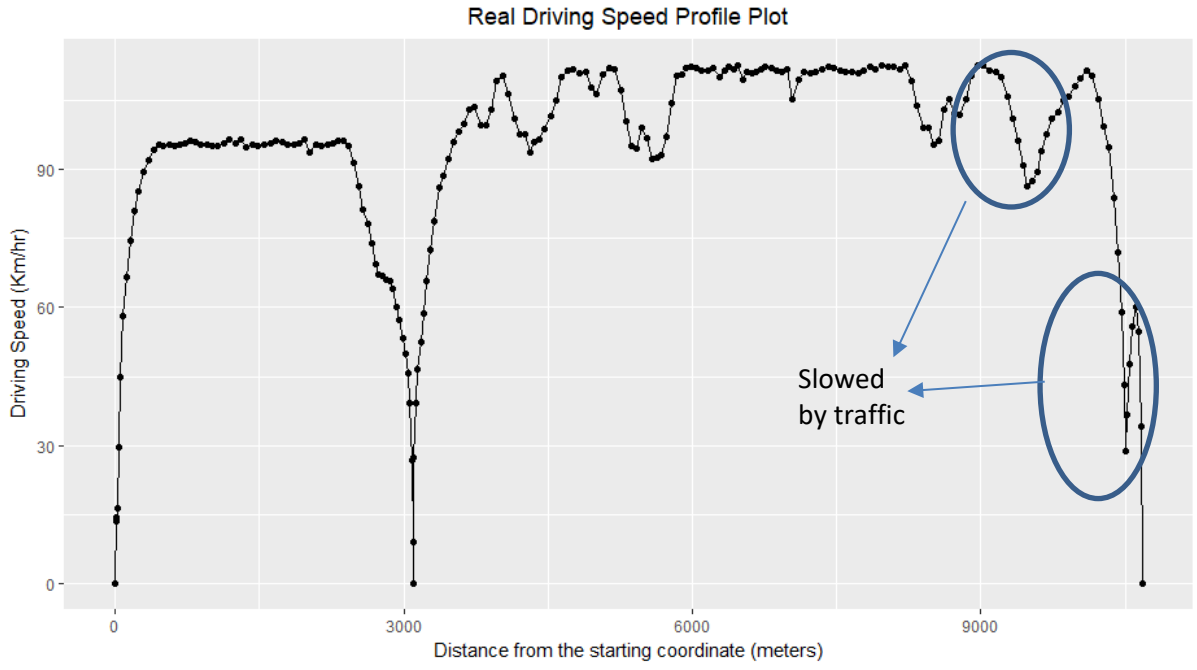

Figure 22. *Speed profile obtained during a trial drive along Route 1*

The RMSE of the simulated driving profile compared to the real driving speed profile is found to be 5.33 Km/hr. The difference between the simulated and the real driving speed profiles for Route 1 is plotted in Figure 23. This concludes the validation of the simulation framework with respect to Route 1.

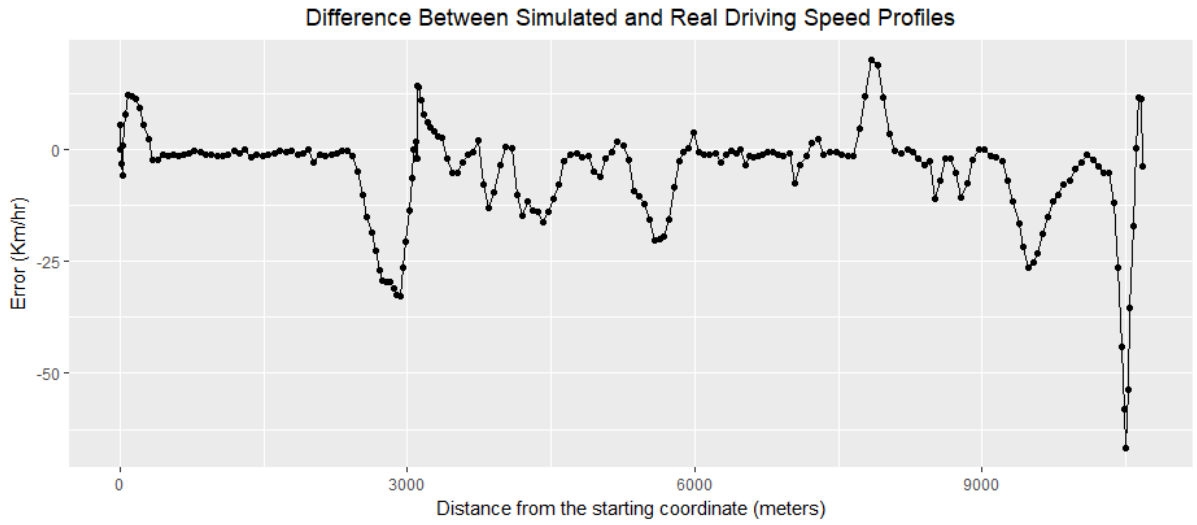

Figure 23. *Difference between simulated and real driving speed profiles for Route 1*

The horizontal turning angle and radius of curvature for Route 2 is shown in Figure 24 and Figure 25 respectively.

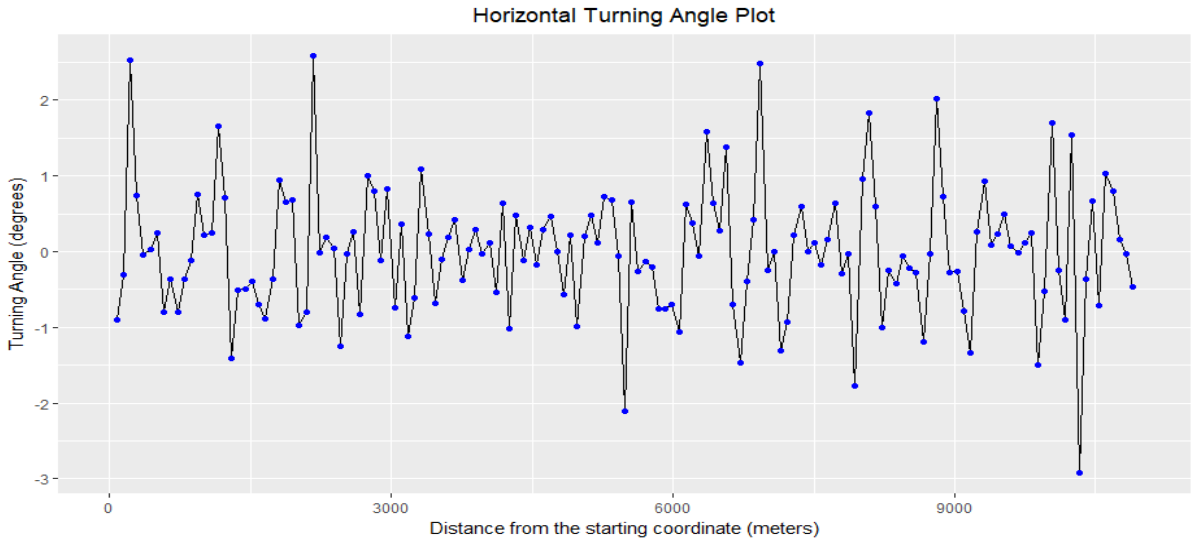

Figure 24. *Horizontal turning angle vs Distance plotted for Route 2*

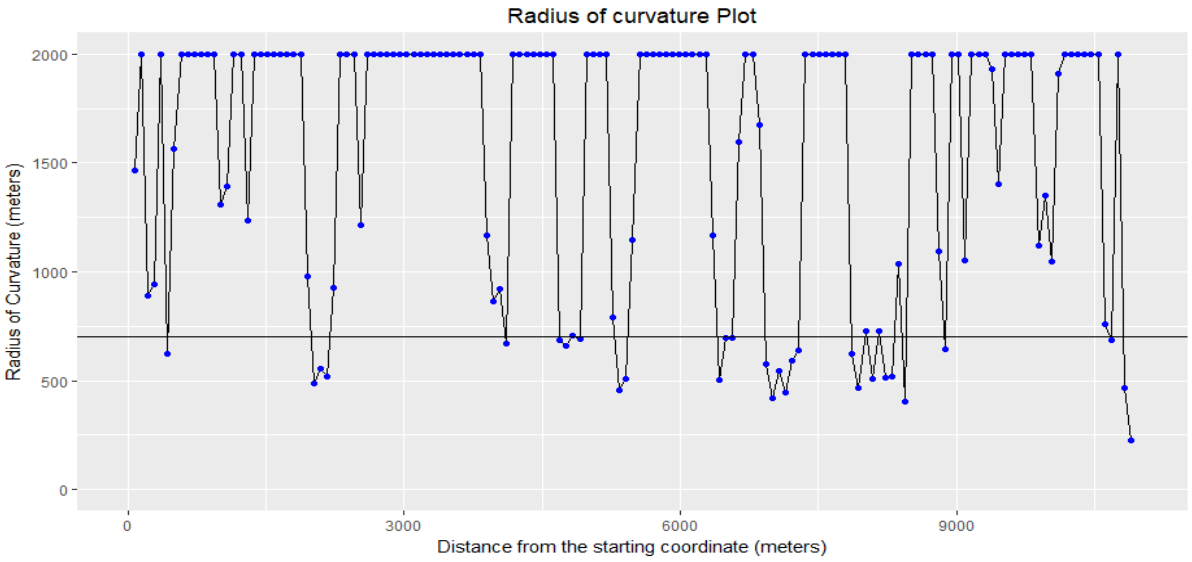

Figure 24. *Horizontal turning angle vs Distance plotted for Route 2*

Figure 25 shows the elevation plot for Route 2. The crest due to vertical curves is marked with a red marker. The blue marker represents the maximum line of sight distance from the adjacent crest point. Limiting speeds for Route 2, based on the road geometry is plotted in Figure 26. A cut-off of 120 Km/hr (75 mph) as the maximum speed limiting speed is used for the plotting purpose.

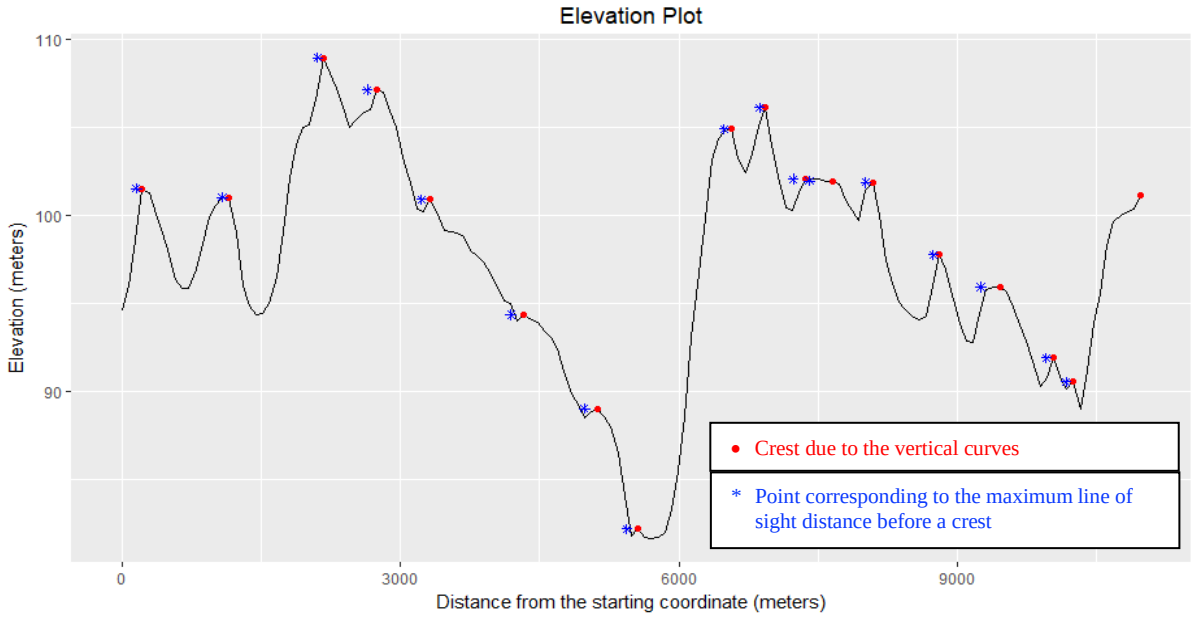

Figure 25. *Elevation vs Distance plotted for Route 2*

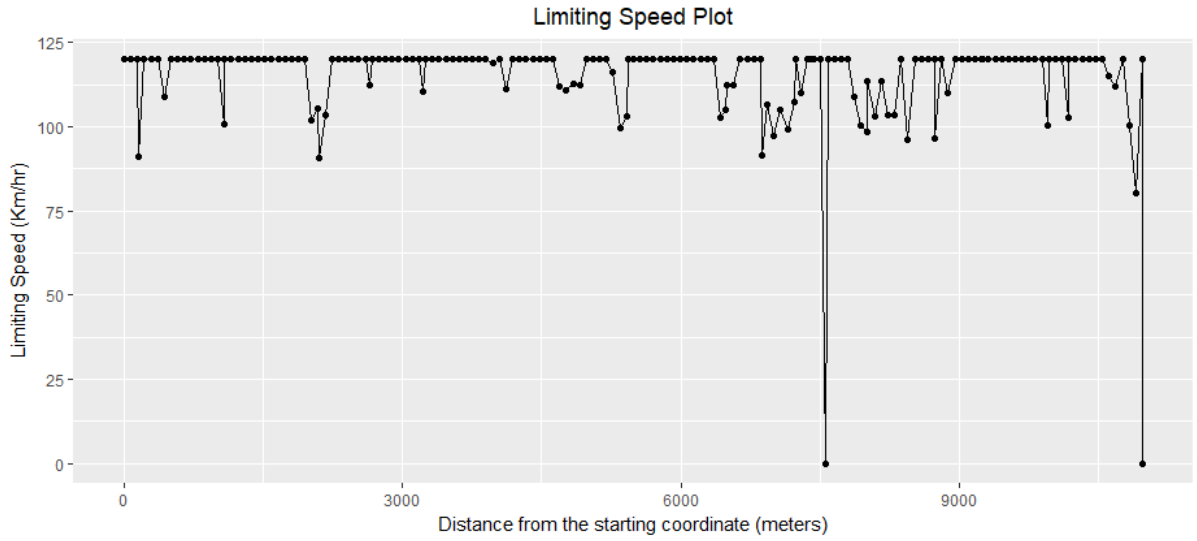

Figure 26. *Limiting Speed due to road geometry vs Distance plotted for Route 2*

Based on the simulation algorithm, for Route 2 the generated acceleration is plotted in Figure 27. The speed profile, generated by the simulation algorithm is plotted in Figure 28. The speed profile observed during a trial drive along the same route is shown in Figure 29.

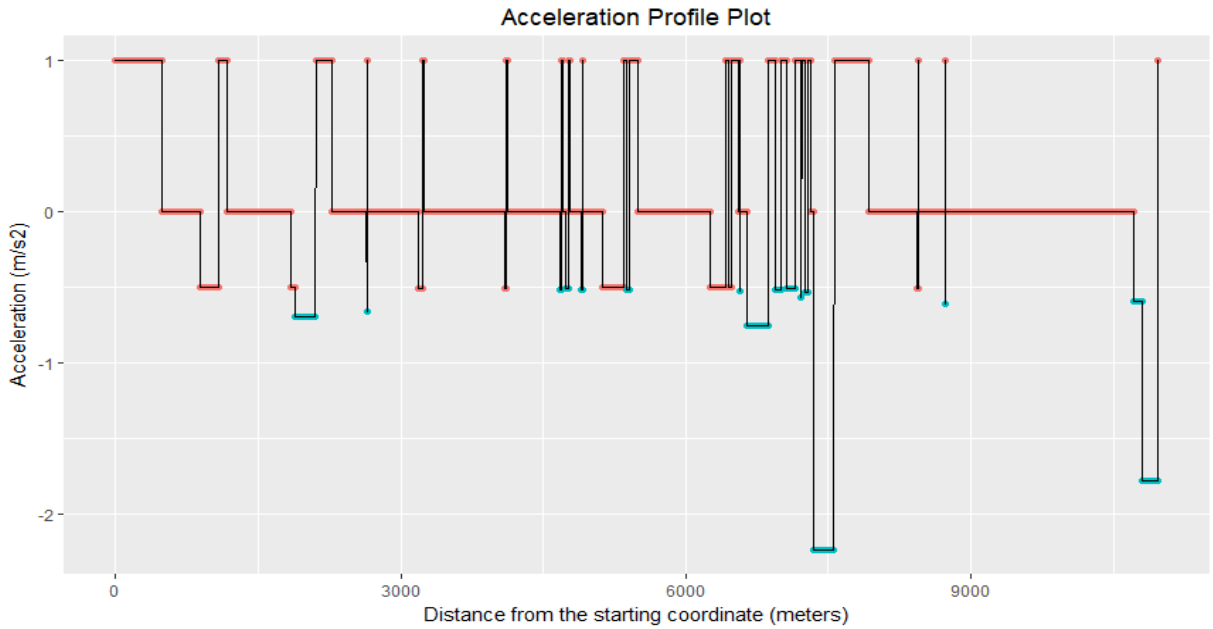

Figure 27. *Simulated Acceleration vs Distance plotted for Route 2*

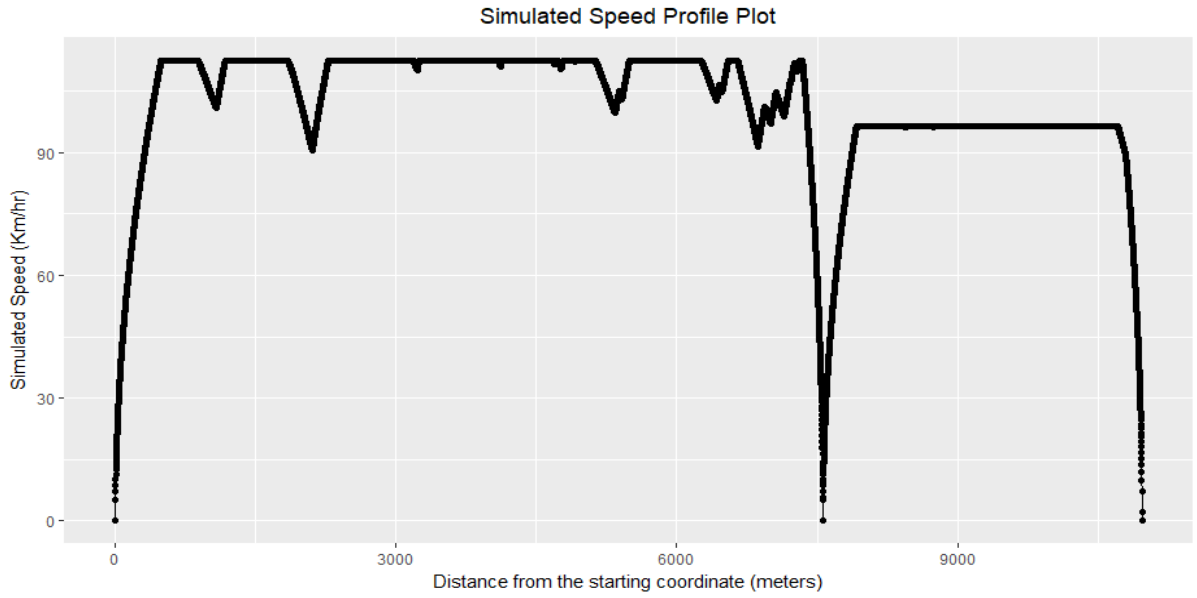

Figure 28. *Simulated Speed vs Distance plotted for Route 2*

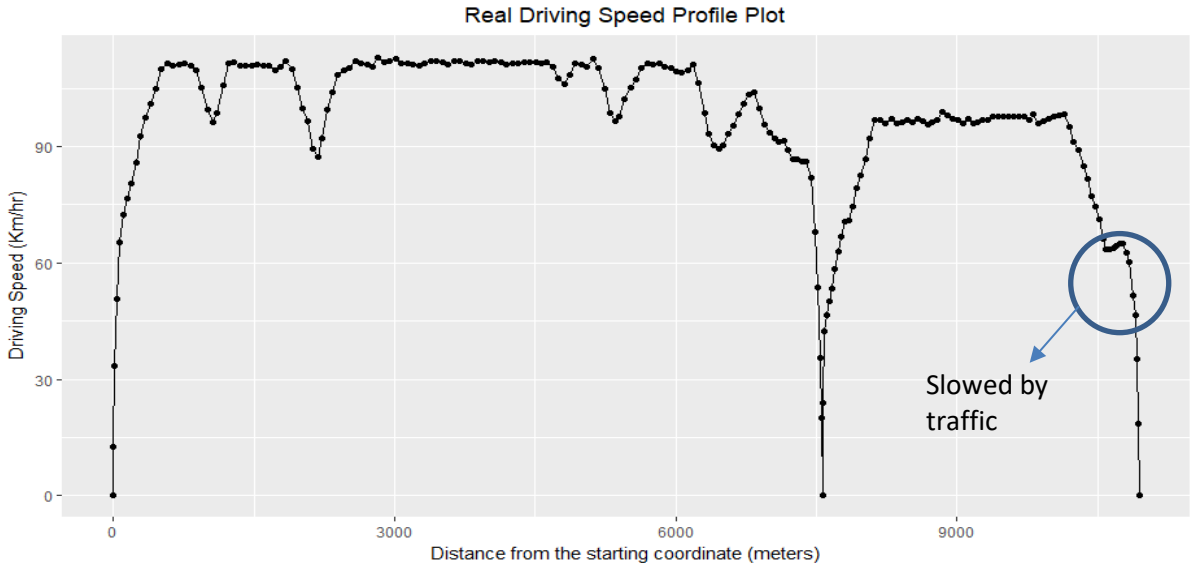

Figure 29. *Speed profile obtained during a trial drive along Route 2*

The RMSE of the simulated driving profile compared to the real driving speed profile is found to be 4.13 Km/hr. The difference between the simulated and the real driving speed profiles for Route 2 is plotted in Figure 30. This concludes the validation of the simulation framework with respect to Route 2.

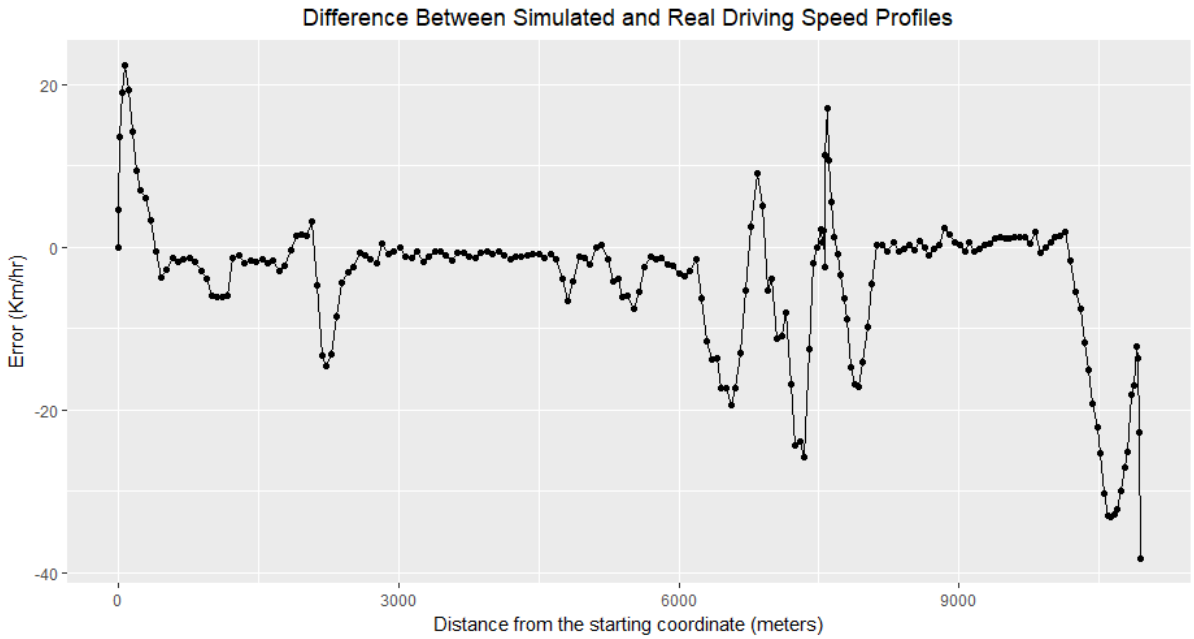

Figure 30. *Difference between simulated and real driving speed profiles for Route 2*

The OSM nodes retrieved using Mapbox direction API and the equidistant waypoints obtained using the Google Elevation API for Route 3 and Route 4 along the FM 159 road, Texas is shown on the map in Figure 31.

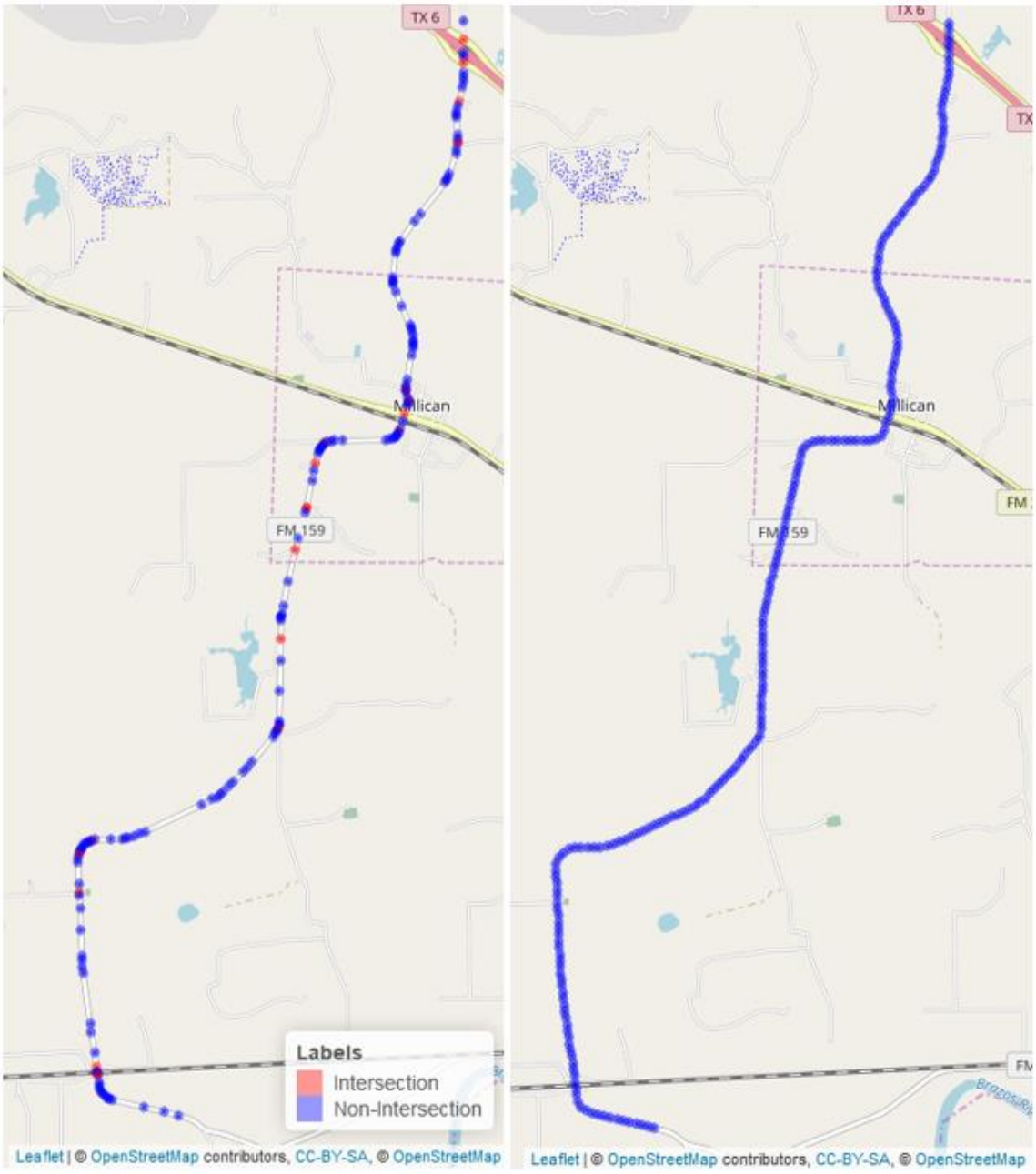

Figure 31.*Route 3 & 4 along FM 159 road;(Left) OSM nodes; (Right) Equidistant waypoints*

The horizontal turning angle and radius of curvature for Route 3 is shown in Figure 32 and Figure 33 respectively. In the radius of curvature plot, all the radiuses above 2000m have been trimmed of the plot and marked as 2000m, so as to fit all the points in a graph.

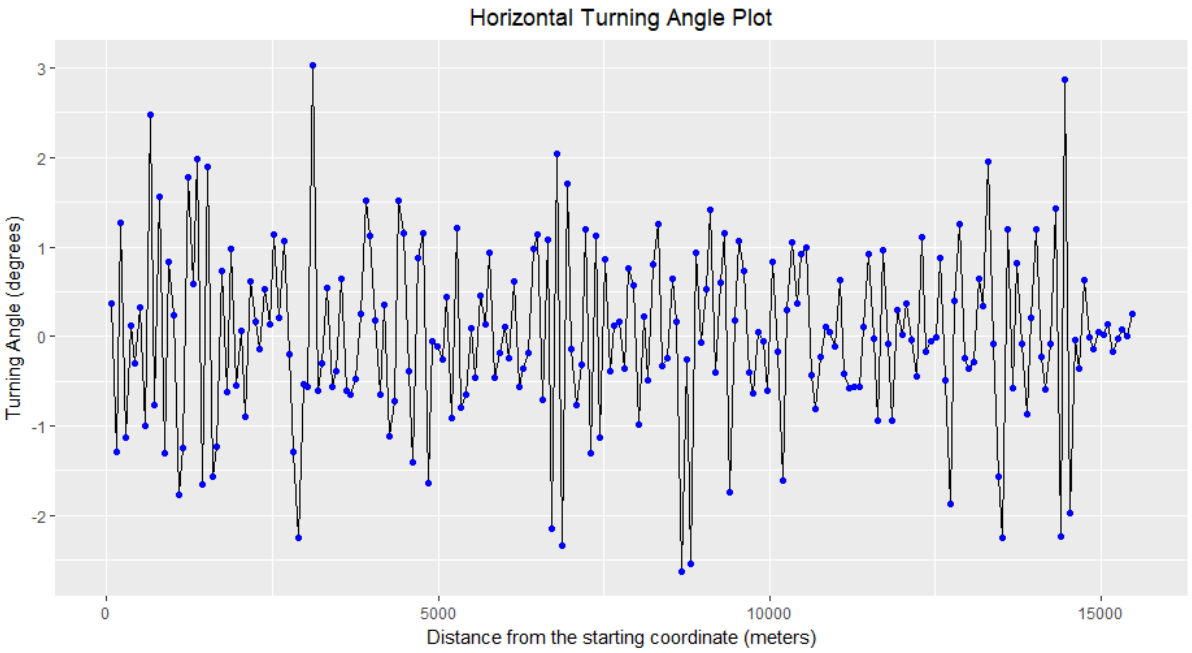

Figure 32. *Horizontal turning angle vs Distance plotted for Route 3*

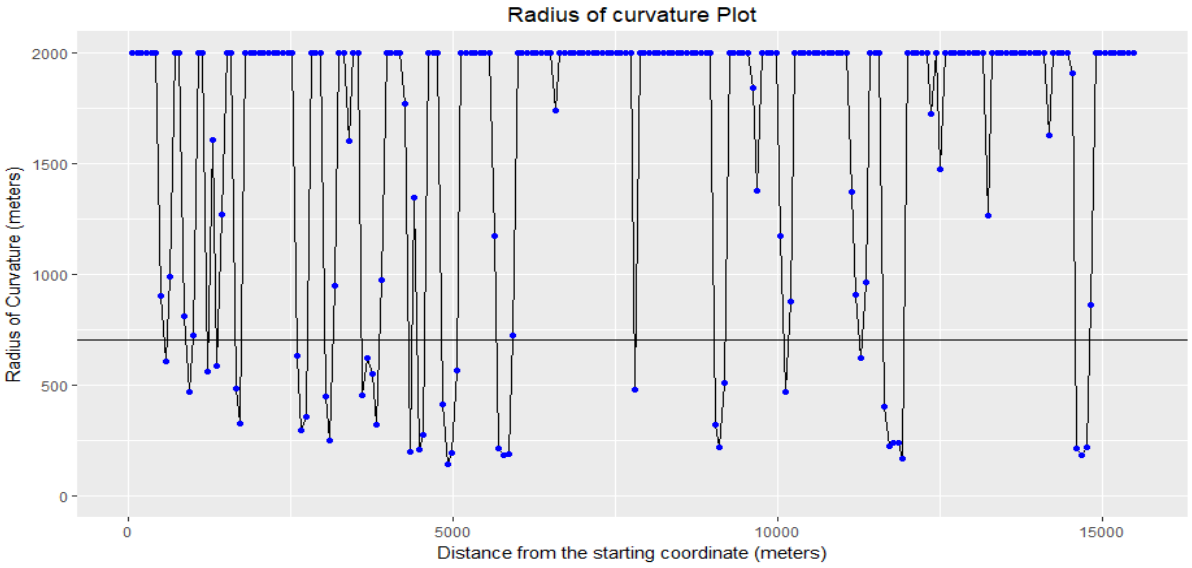

Figure 33. *Horizontal turning angle vs Distance plotted for Route 3*

Figure 34 shows the elevation plot for Route 3. The crest due to vertical curves is marked with a red marker. The blue marker represents the maximum line of sight distance from the adjacent crest point. Limiting speeds for Route 3, based on the road geometry is plotted in Figure

35. A cut-off of 120 Km/hr (75 mph) as the maximum speed limiting speed is used for the plotting purpose.

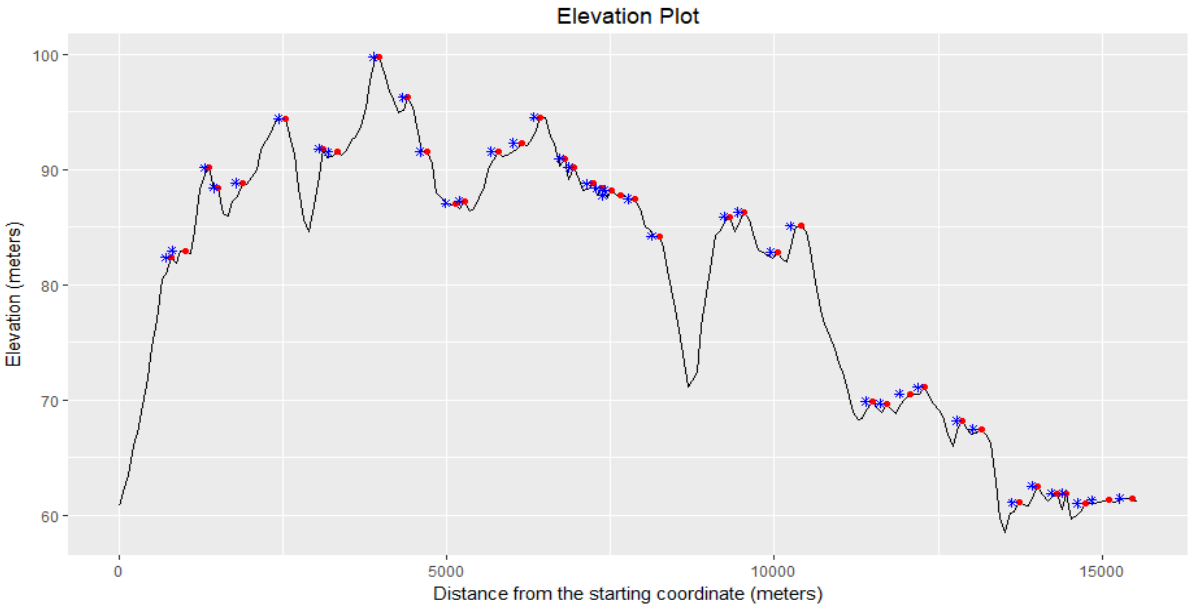

Figure 34. *Elevation vs Distance plotted for Route 3*

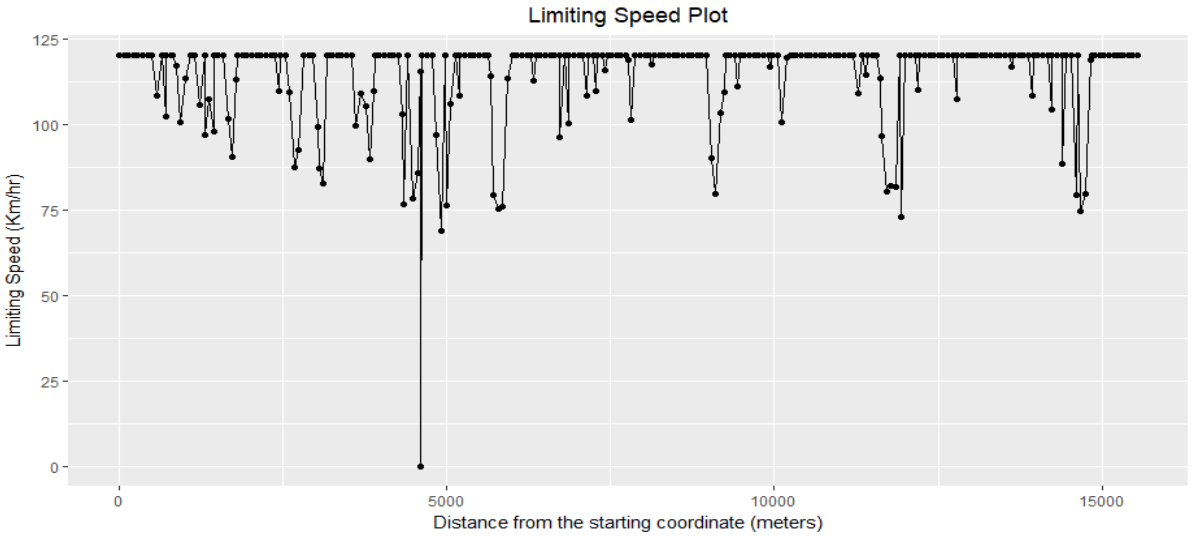

Figure 35. *Limiting Speed due to road geometry vs Distance plotted for Route 3*

Based on the simulation algorithm, for Route 3 the generated acceleration is plotted in Figure 36. The speed profile, generated by the simulation algorithm is plotted in Figure 37. The speed profile observed during a trial drive along the same route is shown in Figure 38.

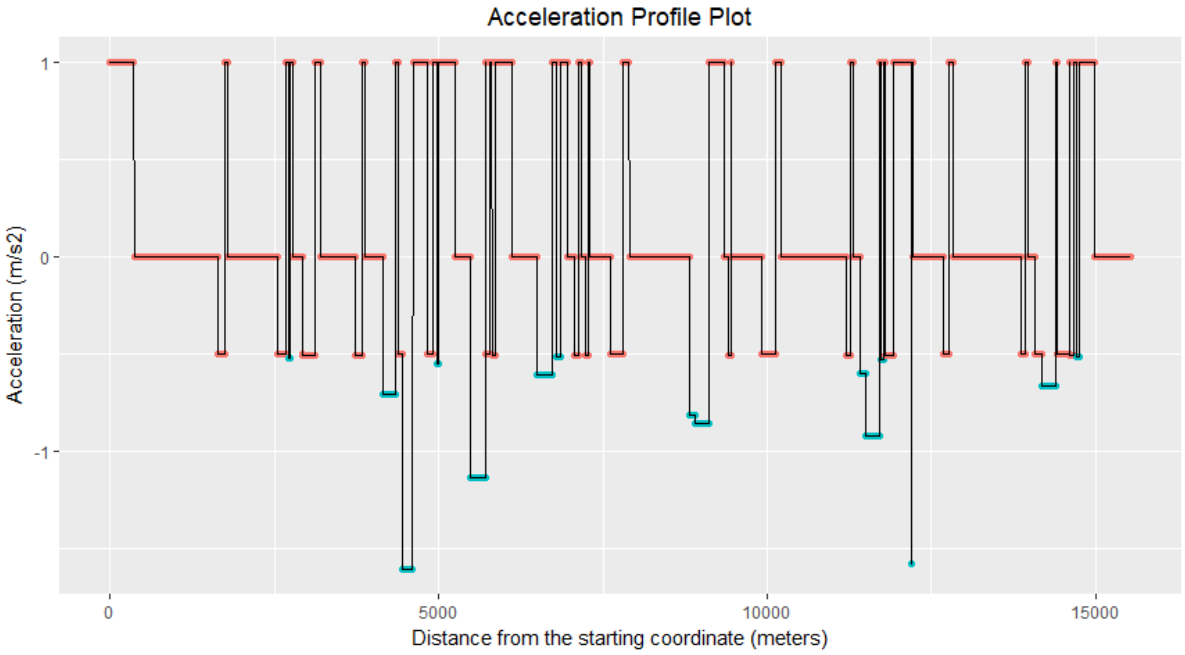

Figure 36. *Limiting Speed due to road geometry vs Distance plotted for Route 3*

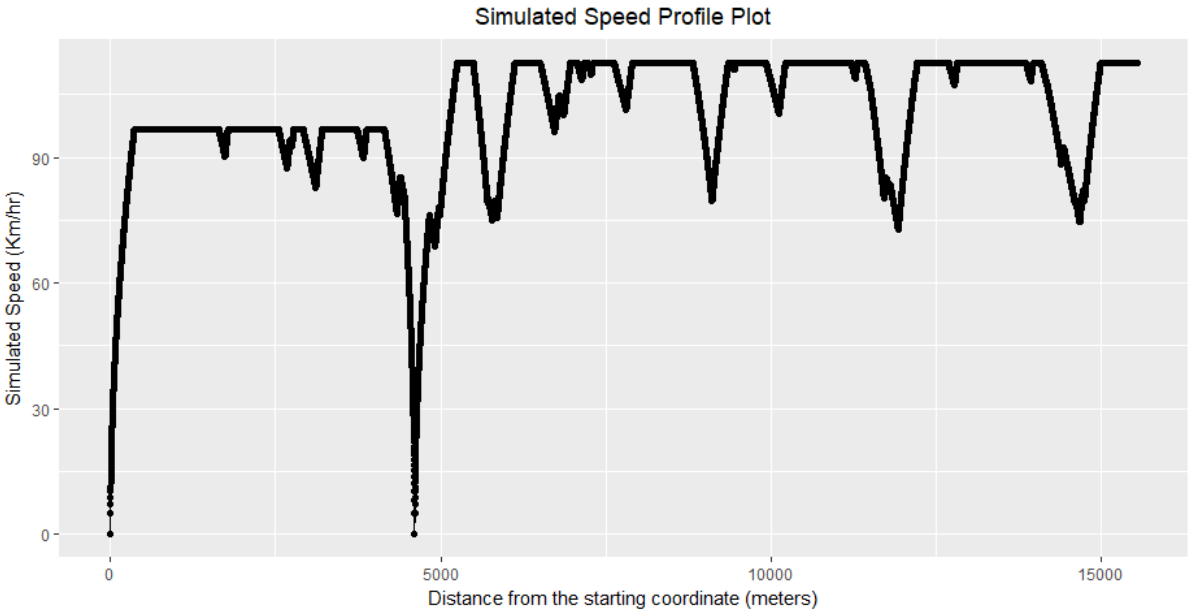

Figure 37. *Simulated Speed vs Distance plotted for Route 3*

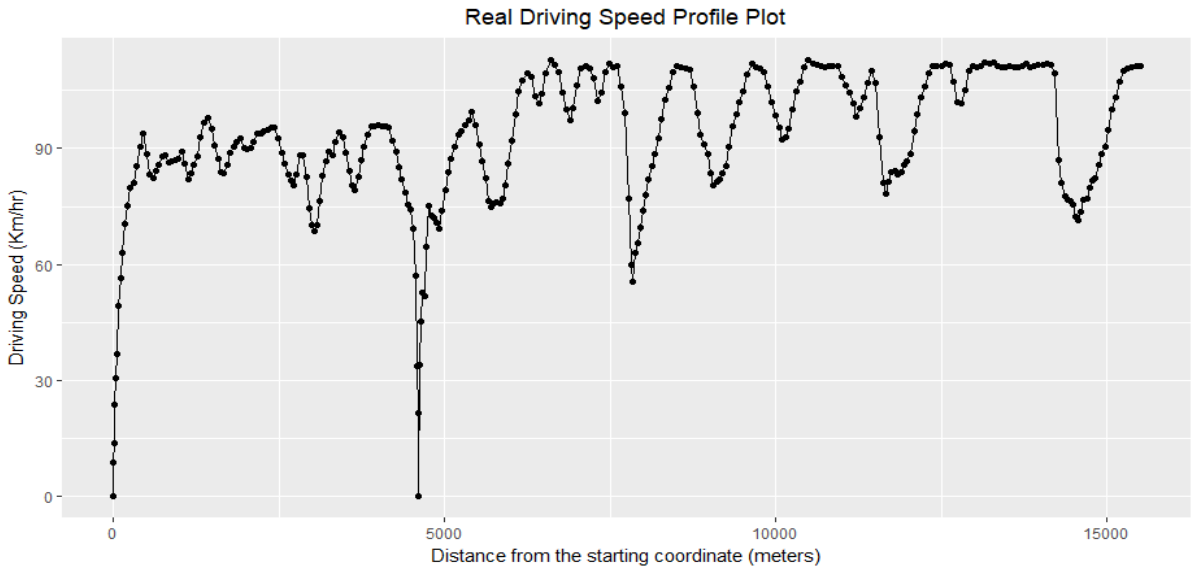

Figure 38. *Speed profile obtained during a trial drive along Route 3*

The RMSE of the simulated driving profile compared to the real driving speed profile is found to be 6.108 Km/hr. The difference between the simulated and the real driving speed profiles for Route 3 is plotted in Figure 39. This concludes the validation of the simulation framework with respect to Route 3.

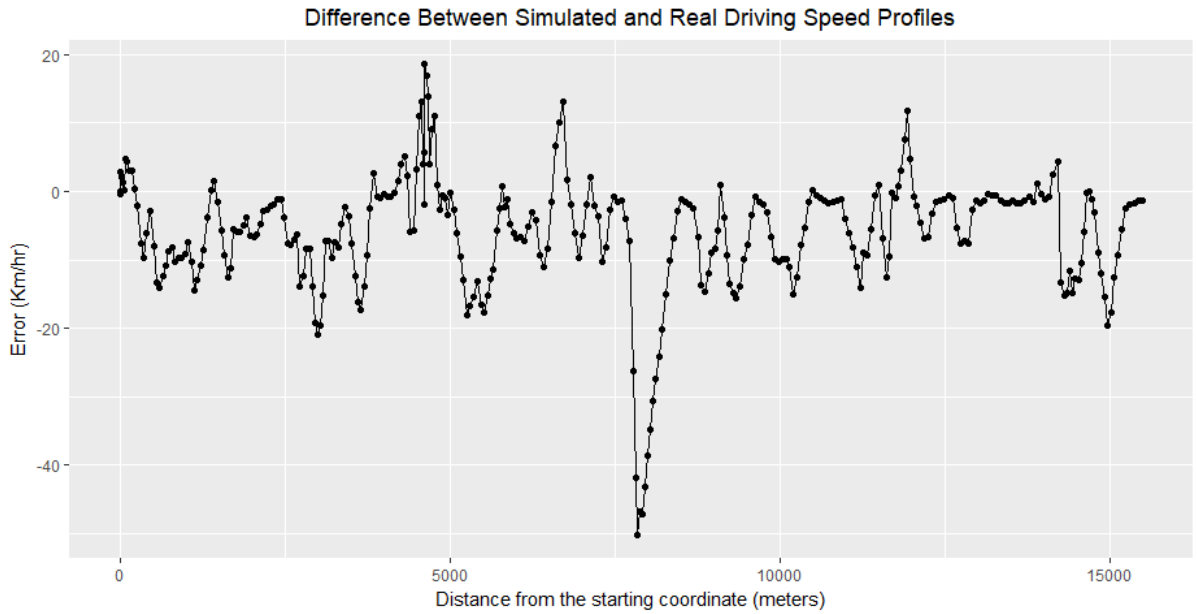

Figure 39. *Difference between simulated and real driving speed profiles for Route 3*

The horizontal turning angle and radius of curvature for Route 4 is shown in Figure 40 and Figure 41 respectively.

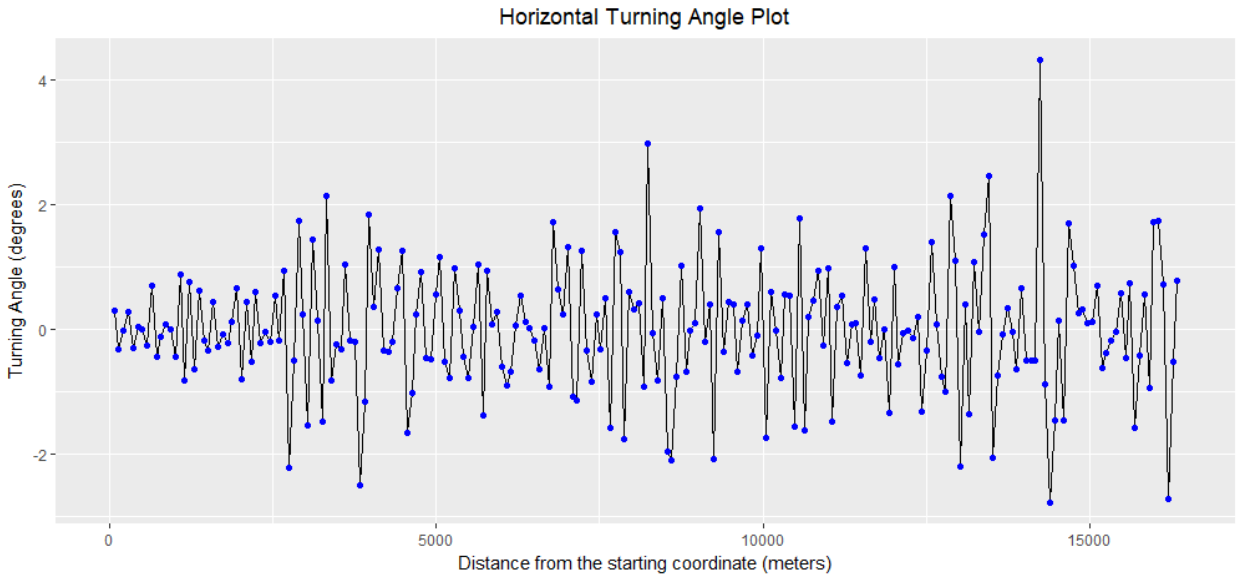

Figure 40. *Horizontal turning angle vs Distance plotted for Route 4*

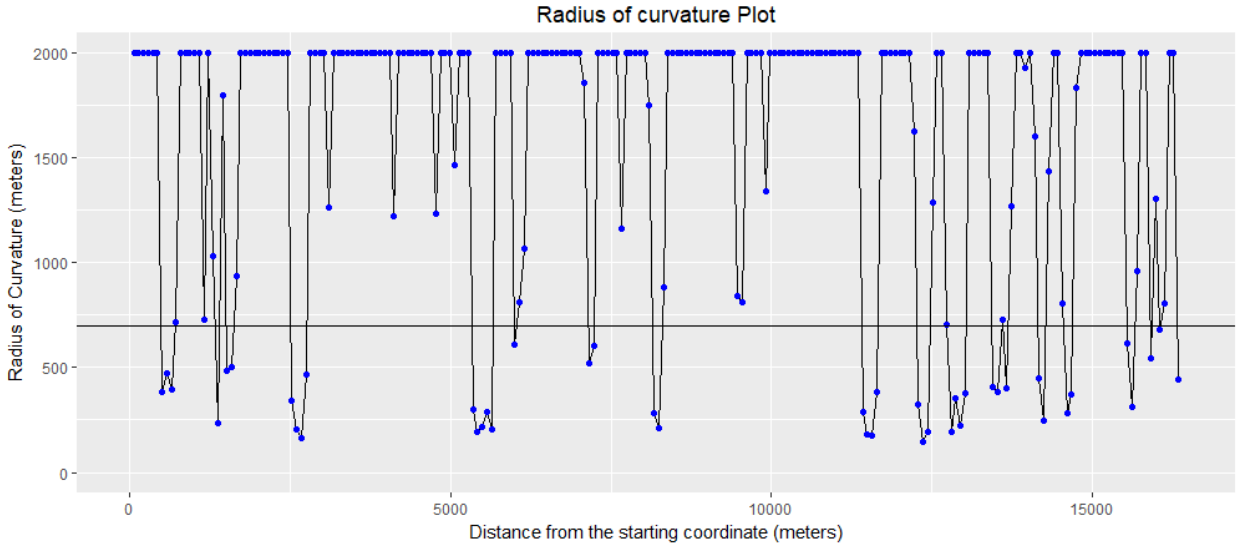

Figure 41. *Horizontal turning angle vs Distance plotted for Route 4*

Figure 42 shows the elevation plot for Route 4. The crest due to vertical curves is marked with a red marker. The blue marker represents the maximum line of sight distance from the adjacent crest point. Limiting speeds for Route 4, based on the road geometry is plotted in Figure 43. A cut-off of 120 Km/hr (75 mph) as the maximum speed limiting speed is used for the plotting purpose.

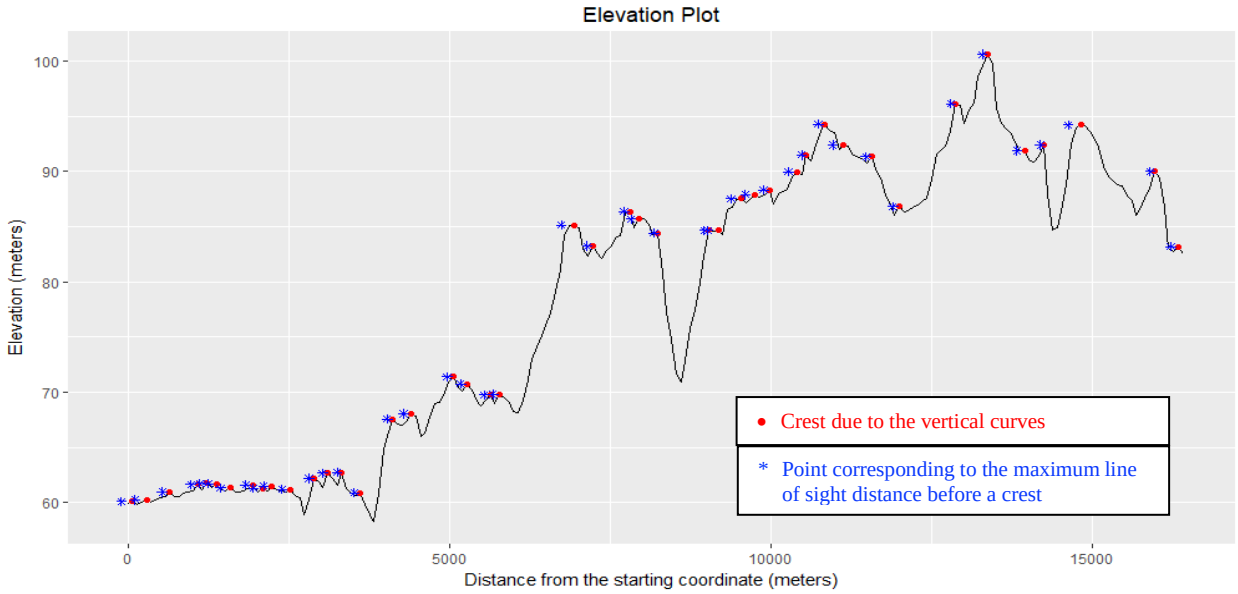

Figure 42. *Elevation vs Distance plotted for Route 4*

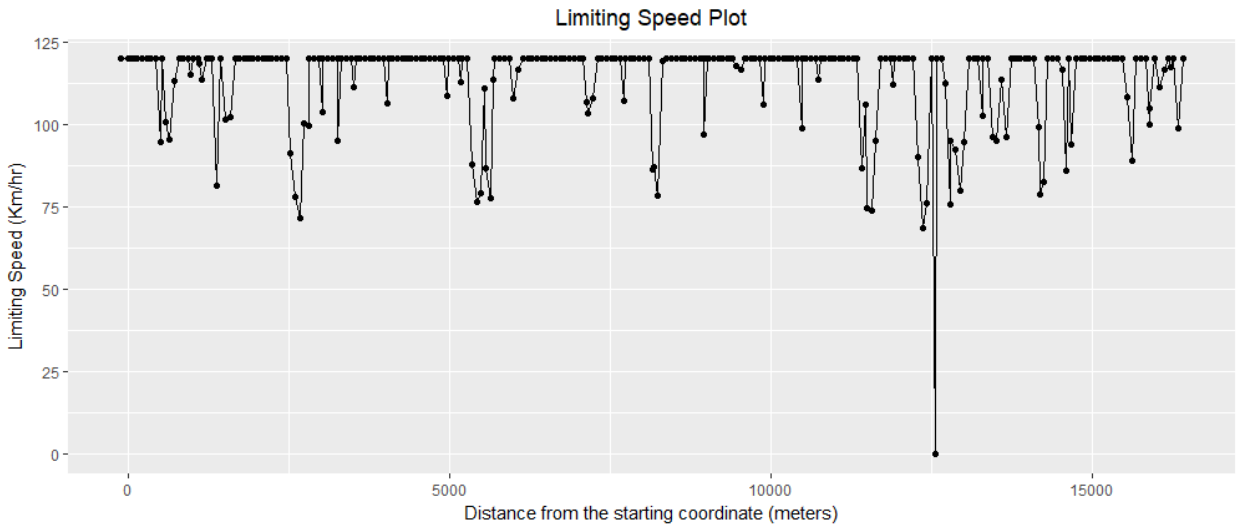

Figure 43. *Limiting Speed due to road geometry vs Distance plotted for Route 4*

Based on the simulation algorithm, for Route 4 the generated acceleration is plotted in Figure 44. The speed profile, generated by the simulation algorithm is plotted in Figure 45. The speed profile observed during a trial drive along the same route is shown in Figure 46.

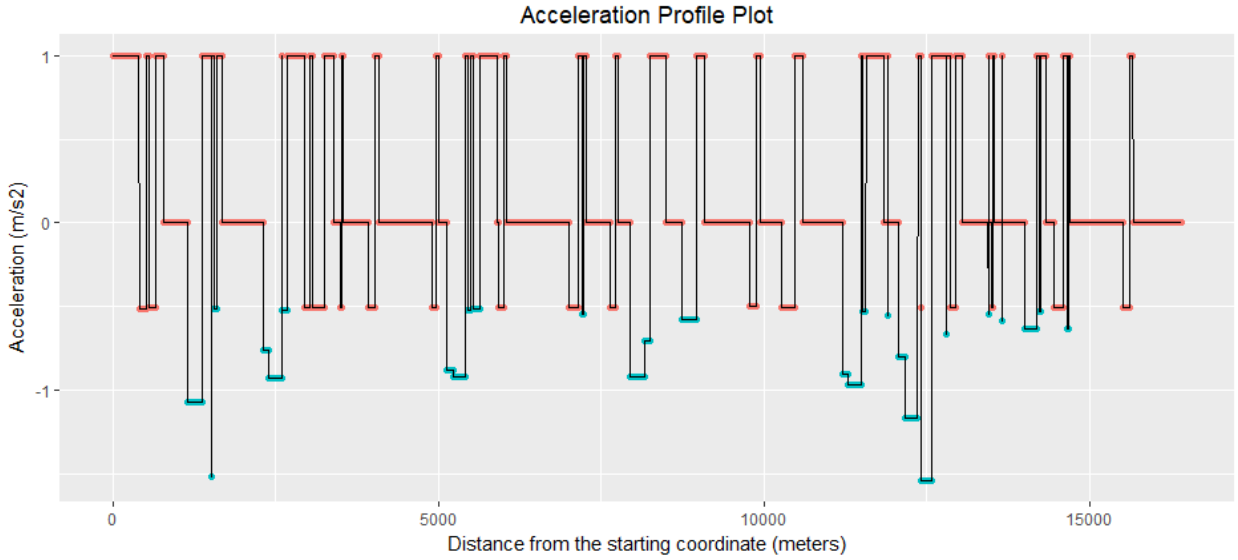

Figure 44. *Simulated Acceleration vs Distance plotted for Route 4*

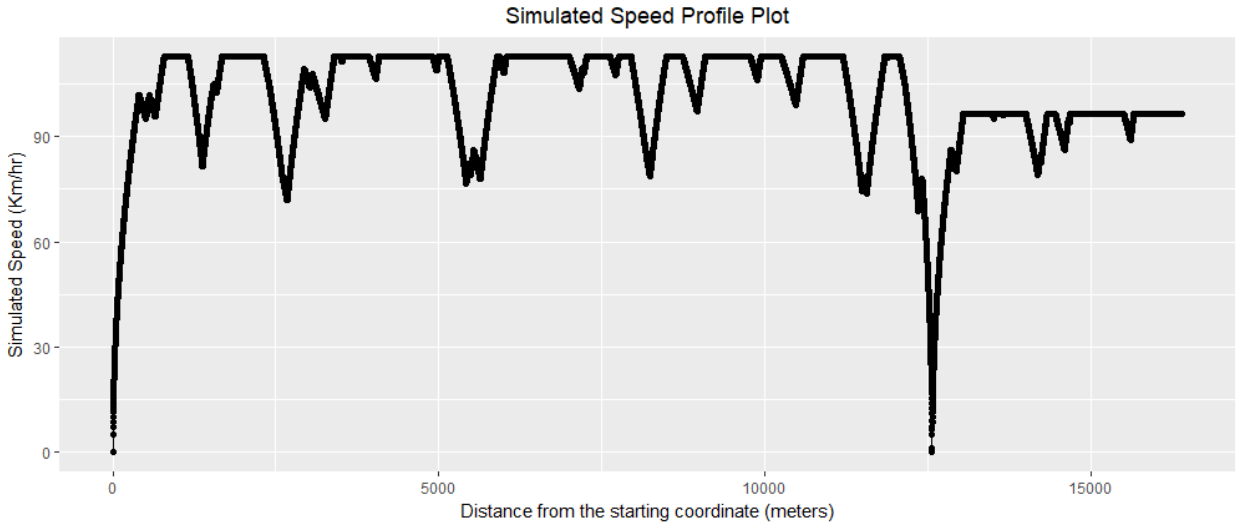

Figure 45. *Simulated Speed vs Distance plotted for Route 4*

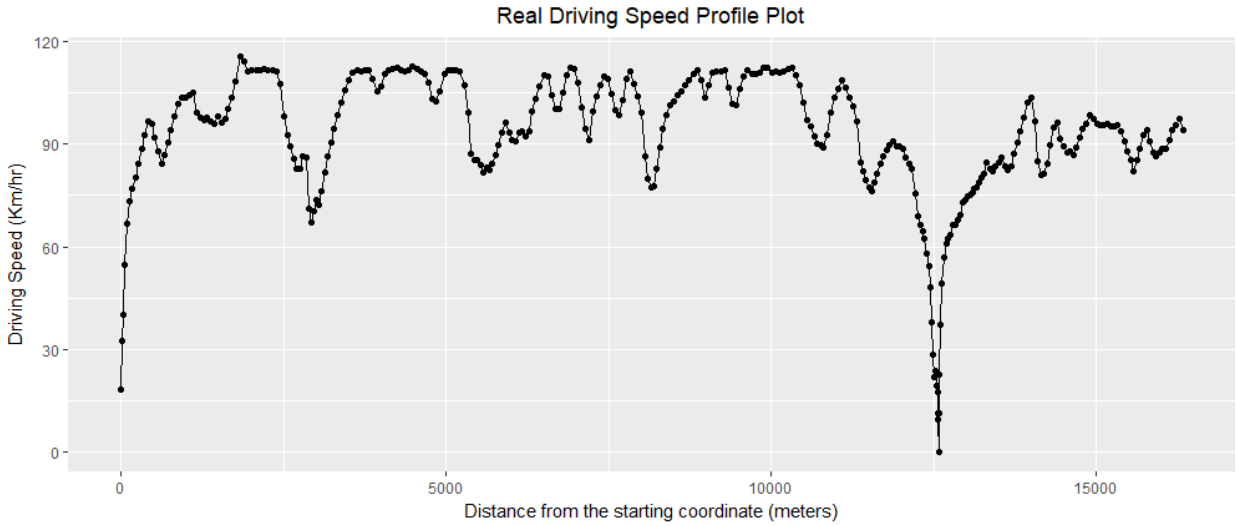

Figure 46. *Speed profile obtained during a trial drive along Route 4*

The RMSE of the simulated driving profile compared to the real driving speed profile is found to be 6.16 Km/hr. The difference between the simulated and the real driving speed profiles for Route 4 is plotted in Figure 47. This concludes the validation of the simulation framework with respect to Route 4.

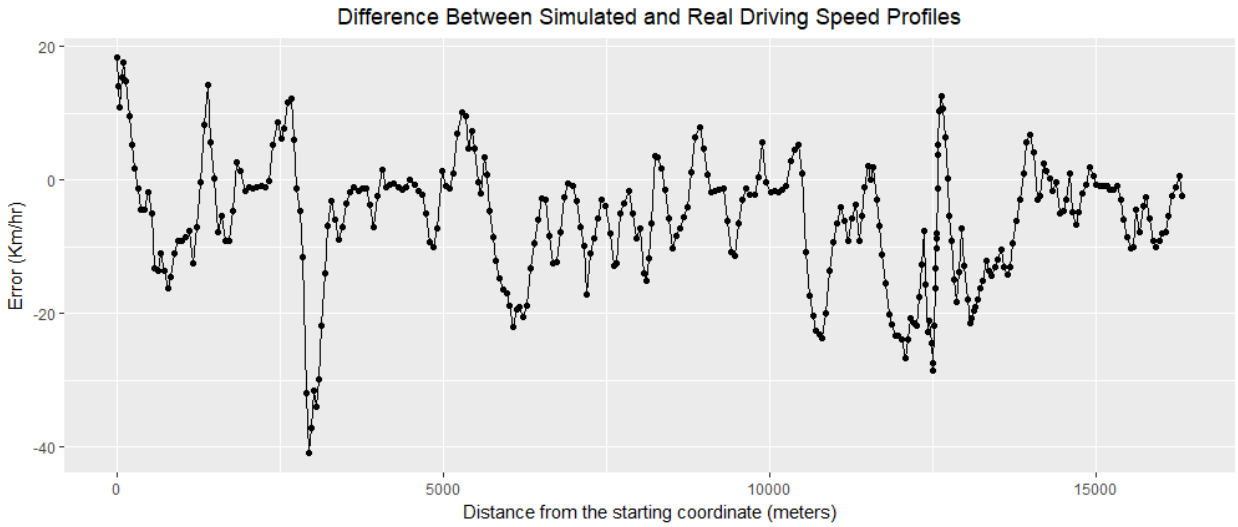

Figure 47. *Difference between simulated and real driving speed profiles for Route 4*

For all the routes remarkable improvement over the design speed profile is observed, as shown in Table 5. Design Speed is the maximum safe speed that can be maintained over a specified section of highway. For this analysis, the speed limits are taken as design speed

	Route 1	**Route 2**	**Route 3**	**Route 4**
Name of Road	Sandy Point Road	Sandy Point Road	FM 159	FM 159
Total Distance (Km)	10	10	14.2	14.2
Direction	West	East	South	North
Design Speed RMSE (Km/hr)	16.35	13.73	16.68	19.07
Simulated Speed RMSE (Km/hr)	5.33	4.13	6.11	6.16

Table 5. *RMSE of simulated and design speed profiles compared to real driving speeds*

7. CONCLUSION

This study presents a framework that simulates the human way of driving in order to generate a safe driving profile for 2-lane rural roads under free-flow condition. The knowledge that driving speeds are constrained by horizontal and vertical road geometries is leveraged in this model successfully.

The speed profile generated using this simulation framework was validated against the real driving speed profile recorded on four routes. In all the four cases the simulated speed profiles were observed to be similar to the real speed profiles and the error rates were quite satisfactory. As an improvement to road safety, this framework could be deployed to warn drivers when they are having unsafe driving speeds.

REFERENCES

[1] World Health Organisation, “WHO | Road traffic injuries,” 2017. [Online]. Available: http://www.who.int/mediacentre/factsheets/fs358/en/. [Accessed: 11-Jan-2018].

[2] L. Aarts and I. Van Schagen, “Driving speed and the risk of road crashes: A review,” *Accid. Anal. Prev.*, vol. 38, pp. 215–224, 2006.

[3] NHSTA, “Quick Facts 2016,” 2017. [Online]. Available: https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812451. [Accessed: 13-Jan-2018].

[4] World Health Organisation, “Global status report on road safety 2015,” 2015.

[5] K. L. Young, M. A. Regan, T. J. Triggs, K. Jontof-Hutter, and S. Newstead, “Intelligent speed adaptation - Effects and acceptance by young inexperienced drivers,” *Accid. Anal. Prev.*, vol. 42, pp. 933–941, 2010.

[6] SWOV, “Fact sheet Speed choice: the influence of man, vehicle, and road,” Leidschendam, 2012.

[7] J. Bagdade, D. Nabors, H. Mcgee, R. Miller, and R. Retting, *Speed Management: A Manual for Local Rural Road*. 2012.

[8] S. Bidulka, T. Sayed, and Y. Hassan, “Influence of vertical alignment on horizontal curve perception: Phase I examining the hypothesis,” *J. Transp. Res. Board*, vol. 1796, pp. 12–23, 2002.

[9] S. H. Hamdar, L. Qin, and A. Talebpour, “Weather and road geometry impact on longitudinal driving behavior: Exploratory analysis using an empirically supported acceleration modeling framework,” *Transp. Res. Part C Emerg. Technol.*, vol. 67, pp. 193–213, 2016.

[10] R. C. Hsu and L. R. Chen, “An integrated embedded system architecture for in-vehicle telematics and infotainment system,” *IEEE Int. Symp. Ind. Electron.*, vol. IV, pp. 1409–1414, 2005.

[11] E. Ndashimye, S. K. Ray, N. I. Sarkar, and J. A. Gutiérrez, “Vehicle-to-infrastructure communication over multi-tier heterogeneous networks: A survey,” *Comput. Networks*, vol. 112, pp. 144–166, 2017.

[12] D. Hale *et al.*, “Introduction of Cooperative Vehicle-to-Infrastructure Systems to Improve Speed Harmonization,” 2016.

[13] F. Biondi, R. Rossi, M. Gastaldi, and C. Mulatti, “Beeping ADAS: Reflexive effect on drivers’ behavior,” *Transp. Res. Part F Traffic Psychol. Behav.*, vol. 25, no. PART A, pp. 27–33, 2014.

[14] M. A. Taie, E. M. Moawad, M. Diab, and M. ElHelw, “Remote Diagnosis, Maintenance and Prognosis for Advanced Driver Assistance Systems Using Machine Learning Algorithms,” *SAE Int. J. Passeng. Cars - Electron. Electr. Syst.*, vol. 9, no. 1, pp. 2016-01–0076, 2016.

[15] H. W. Warner and L. Åberg, “The long-term effects of an ISA speed-warning device on drivers’ speeding behaviour,” *Transp. Res. Part F Traffic Psychol. Behav.*, vol. 11, no. 2, pp. 96–107, 2008.

[16] F. Lai, O. Carsten, and F. Tate, “How much benefit does Intelligent Speed Adaptation deliver? - An analysis of its potential contribution to safety and environment,” *Accid. Anal. Prev.*, vol. 48, pp. 63–72, 2012.

[17] G. Zhao, C. Wu, and C. Qiao, “A mathematical model for the prediction of speeding with its validation,” *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 2, pp. 828–836, 2013.

[18] F. Mars and P. Chevrel, “Modelling human control of steering for the design of advanced driver assistance systems,” *Annu. Rev. Control*, vol. 44, pp. 292–302, 2017.

[19] P. C. Cacciabue, *Modelling driver behaviour in automotive environments: Critical issues in driver interactions with intelligent transport systems*. London, 2007.

[20] M. . Wang *et al.*, “Drowsy Behavior Detection Based On Driving Information,” *Int. J. Automot. Technol.*, vol. 17, no. 2, pp. 165–173, 2016.

[21] Y. Yang, M. McDonald, and P. Zheng, “Can drivers’ eye movements be used to monitor their performance? A case study,” *IET Intell. Transp. Syst.*, vol. 6, no. 4, pp. 444–452, 2012.

[22] E. Bekiaris, A. Amditis, and M. Panou, “DRIVABILITY: a new concept for modelling driving performance,” *Cogn. Technol. Work*, vol. 5, no. 2, pp. 152–161, 2003.

[23] A. T. Ibrahim and F. L. Hall, “Effect of Adverse Weather Conditions on Speed-Flow-Occupancy Relationships,” *Transp. Res. Rec.*, vol. 1457, pp. 184–191, 1994.

[24] “The effect of environmental factors on driver speed: a case study,” *Transp. Res. Rec.*, vol. 1635, pp. 155–161, 1998.

[25] K. L. M. Broughton, F. Switzer, and D. Scott, “Car following decisions under three visibility conditions and two speeds tested with a driving simulator,” *Accid. Anal. Prev.*, vol. 39, pp. 106–116, 2007.

[26] C.-G. Wallman, P. Wretling, and G. Öberg, “Effects of Winter Road Maintenance State-of-the-Art,” 1998.

[27] M. Tanaka, P. Ranjitkar, and T. Nakatsuji, “Comparison of Driving Behavior and Safety in Car-Following Platoons Under Icy and Dry Surface Conditions,” 2010, p. 22.

[28] M. G. Karlaftis and I. Golias, “Effects of road geometry and traffic volumes on rural roadway accident rates,” *Accid. Anal. Prev.*, vol. 34, pp. 357–365, 2002.

[29] A. Polus, M. A. Pollatschek, and H. Farah, “Traffic Injury Prevention Impact of Infrastructure Characteristics on Road Crashes on Two-Lane HighwaysL a n e H i g h w a y s,” *Traffic Inj. Prev.*, vol. 63, no. 6, 2005.

[30] J. F. Kraus, C. L. Anderson, S. Arzemanian, M. Salatka, P. Hemyari, and G. Sun, “Epidemiological Aspects of Fatal and Severe Injury Urban Freeway Crashes,” *Accid Anal. Prev*, vol. 25, no. 3, pp. 229–239, 1993.

[31] K. W. Ogden, “The Effects of Paved Shoulders on Accidents on Rural Highways'," *Accid. Anal. Prev*, vol. 29, no. 3, pp. 353–362, 1997.

[32] A. Polus, “The relationship of overall geometric characteristics to the safety level of rural highways,” *Traffic Q.*, vol. 34, no. 4, pp. 555–585, Oct. 1982.

[33] V. Shankar, F. Mannering ’, and W. Barfield, “Statistical Analysis of Accident Severity on Rural Freeways,” *Accid. Anal. Prev*, vol. 28, no. 3, pp. 391–401, 1996.

[34] C. Zegeer *et al.*, “Cost-effective geometric improvements for safety upgrading of horizontal curves,” no. FHWA/RD-90-021, Oct. 1991.

[35] S.-P. Miaou *et al.*, “Relationship Between Truck Accidents and Highway Geometric Design: A Poisson Regression Approach.”

[36] D. . Harwood, F. . Council, E. Hauer, W. . Hughes, and A. Vogt, “Prediction of the Expected Safety Performance of Rural Two-Lane Highways,” 2000.

[37] F. Council and J. Stewart, “Safety Effects of the Conversion of Rural Two-Lane to Four-Lane Roadways Based on Cross-Sectional Models,” *Transp. Res. Rec. J. Transp. Res. Board*, vol. 1665, pp. 35–43, Jan. 1999.

[38] J. G. Strathman, K. Dueker, J. Zhang, T. Williams, J. G. ; Strathman, and K. ; Dueker, “Analysis of Design Attributes and Crashes on the Oregon Highway System,” *Cent. Urban Stud. Publ. Reports*, vol. 81, 2001.

[39] M. Martens, N. Kaptein, and S. Comte, “The Effects of Road Design on Speed Behaviour: A Literature Review,” 1997.

[40] J. Bonneson, M. Pratt, J. Miles, and P. Carlson, “Development of Guidelines for Establishing Effective Curve Advisory Speeds,” College Station, 2007.

[41] P. J. Andueza, “Mathematical Models of Vehicular Speed on Mountain Roads,” *Transp. Res. Rec.*, vol. 1701, pp. 104–110, 2000.

[42] H. Summala, “Accident Risk and Driver Behaviour,” *Saf. Sci.*, vol. 22, pp. 103-117, 1996.

[43] D. De Waard, M. Jessurun, F. J. J. M. Steyvers, P. T. F. Reggatt, and K. A. Brookhuis, “Effect of road layout and road environment on driving performance, drivers’ physiology and road appreciation,” *Ergonomics*, vol. 38, no. 7, pp. 1395–1407, Jul. 1995.

[44] S. Yagar and M. Van Aerde, “Geometric and environmental effects on speeds of 2-lane highways,” *Transp. Res. Part A Gen.*, vol. 17, no. 4, pp. 315–325, Jul. 1983.

[45] K. Fitzpatrick, P. Carlson, M. Brewer, and M. Wooldridge, “Design Factors That Affect Driver Speed on Suburban Streets,” *Transp. Res. Rec. J. Transp. Res. Board*, vol. 1751, pp. 18–25, Jan. 2001.

[46] R. Tay and A. Churchill, “Effect of Different Median Barriers on Traffic Speed,” *Can. J. Transp.*, vol. 1, no. 1, Mar. 2007.

[47] “OpenStreetMap Wiki.” [Online]. Available: https://wiki.openstreetmap.org/wiki/Main_Page. [Accessed: 07-Feb-2018].

[48] “OpenStreetMap Statistics.” [Online]. Available: https://www.openstreetmap.org/stats/data_stats.html. [Accessed: 07-Feb-2018].

[49] “Contributors - OpenStreetMap Wiki.” [Online]. Available: https://wiki.openstreetmap.org/wiki/Contributors#United_States. [Accessed: 07-Feb-2018].

[50] V. Hagenmeyer *et al.*, “Information and Communication Technology in Energy Lab 2.0: Smart Energies System Simulation and Control Center with an Open-Street-Map-Based Power Flow Simulation Example,” *Energy Technol.*, vol. 4, no. 1, pp. 145–162, 2016.

[51] Darwin García, “OpenStreetMap: an alternative to Google Maps,” 2013. [Online]. Available: https://www.webilop.com/openstreetmap-an-alternative-to-google-maps/. [Accessed: 11-Feb-2018].

[52] Sterling Quinn, “OpenStreetMap and its use as open data | GEOG 585: Web Mapping.” [Online]. Available: https://www.e-education.psu.edu/geog585/node/738. [Accessed: 11-Feb-2018].

[53] “Bounding Box - OpenStreetMap Wiki.” [Online]. Available: https://wiki.openstreetmap.org/wiki/Bounding_Box. [Accessed: 11-Feb-2018].

[54] “Our map data | Mapbox.” [Online]. Available: https://www.mapbox.com/help/how-mapbox-data-works/. [Accessed: 16-Feb-2018].

[55] “Elevation Service | Google Maps JavaScript API | Google Developers.” [Online].

Available: https://developers.google.com/maps/documentation/javascript/elevation. [Accessed: 22-Feb-2018].

[56] OSRM, “Project OSRM.” [Online]. Available: http://project-osrm.org/. [Accessed: 18-Feb-2018].

[57] D. Damnjanovic, A. Milicevic, and D. Cvetkovic, *Uskladjivanje Konstruktivnih Elementa Puta Prema Ocekivanoj Brzini U Slobodnom Toku*. 2002.

[58] F. Bella, “Safety in Work Zones: Experiences with Driving Simulator.”

APPENDIX

The R code developed for the simulation framework discussed on this report.

```
################## API Keys ################################

mapbox_key = "enter_your_key"

google_elevation_key = "enter_your_key"

############## Input the coordinates ######################

#routes used for validation

waypoints = "-96.391981,30.678184;-96.488752,30.699787"    #Sandy pt rd, westward

waypoints = "-96.489106,30.699651;-96.389613,30.677187"    #Sandy pt rd, eastward

waypoints = "-96.197433,30.506977;-96.231938,30.395658"   #FM159 southward

waypoints = "-96.214739,30.393913;-96.198245,30.498911"   #FM159 northward

################## MapBox Direction API ############################

annotation ="distance,duration,speed,congestion,nodes"

req = paste("https://api.mapbox.com/directions/v5/mapbox/driving/",

waypoints,"?alternatives=false&geometries=polyline&steps=true&overview=full&annotations="

,annotation,"&access_token=", mapbox_key,sep = "")

resRaw = RCurl::getURL(utils::URLencode(req), useragent = "R-User")

vres = jsonlite::validate(resRaw)[1]

if (!vres)

  resRaw = gsub(pattern = "[\\]", replacement = "tilapia", x = resRaw)

route_json = jsonlite::fromJSON(resRaw)

if (!vres)
```

```
  route_json$routes$geometry = gsub(pattern = "tilapia", replacement = "\\\\", x =
route_json$routes$geometry)
distance = route_json$routes$distance
rm(vres,resRaw,req,waypoints, mapbox_key, annotation)
############## Google elevation service ############################
k72 = ifelse( floor(distance/72) + 1 > 512 , 512 , floor(distance/72) + 1)
req = paste0("https://maps.googleapis.com/maps/api/elevation/json?&key=",
google_elevation_key , "&path=enc:" , route_json$routes$geometry , "&samples=" , k72)
resRaw = RCurl::getURL(utils::URLencode(req), useragent = "R-User")
elevation_json = jsonlite::fromJSON(resRaw)

ele_dump = cbind(elevation_json$results$location ,
elevation_json$results[c("elevation","resolution")] )
colnames(ele_dump)[2] = "lon"
ele_dump$distance = seq(0, distance , length.out = k72)
rm(elevation_json,google_elevation_key,resRaw,req)
############ Functions used to find Turning  Angles ###########
anglefun <- function(xx,yy,bearing=TRUE,as.deg=FALSE){
   ## Options:
  ## bearing = FALSE returns +/- pi instead of 0:2*pi
  ## as.deg = TRUE returns degrees instead of radians
  c = 1
  if (as.deg){
```

```
  c = 180/pi
 }

 b<-sign(xx)
 b[b==0]<-1  #corrects for the fact that sign(0) == 0
 tempangle = b*(yy<0)*pi+atan(xx/yy)
 if(bearing){
  #return a compass bearing 0 to 2pi
  #if bearing==FALSE then a heading (+/- pi) is returned
  tempangle[tempangle<0]<-tempangle[tempangle<0]+2*pi
 }
 return(tempangle*c)
}
bearing.ta <- function(loc1,loc2,loc3,as.deg=T){
 ## calculates the bearing and length of the two lines formed by three points
 ## the turning angle from the first bearing to the second bearing is also calculated
 ## locations are assumed to be in (X,Y) format.
 ## Options:
 ## as.deg = TRUE returns degrees instead of radians
 if (length(loc1) != 2 | length(loc2) != 2 | length(loc3) !=2){
  print("Locations must consist of either three vectors, length == 2,
       or three two-column dataframes")
  return(NaN)
```

```
  }
  c = 1
  if (as.deg){
    c = 180/pi
  }
  locdiff1<-loc2-loc1
  locdiff2<-loc3-loc2
  bearing1<-anglefun(locdiff1[1],locdiff1[2],bearing=F)
  bearing2<-anglefun(locdiff2[1],locdiff2[2],bearing=F)
    if(is.data.frame(locdiff1)){
    dist1<-sqrt(rowSums(locdiff1^2))
    dist2<-sqrt(rowSums(locdiff2^2))
  }else{
    dist1<-sqrt(sum(locdiff1^2))
    dist2<-sqrt(sum(locdiff2^2))
  }
  ta=(bearing2-bearing1)
  ta[ta < -pi] = ta[ta < -pi] + 2*pi
  ta[ta > pi] = ta[ta > pi] - 2*pi
  return(list(bearing1=unlist(bearing1*c),bearing2=unlist(bearing2*c),
              ta=unlist(ta*c),dist1=unlist(dist1),dist2=unlist(dist2)))
}
```

```
####### Turning angle, ROC , Max speeds at each Google elevation point #########
#Turning Angle horizontal
theta=numeric()
for(i in 1:(nrow(ele_dump) - 2)){
  a=bearing.ta(ele_dump[i,c("lon","lat")], ele_dump[i+1, c("lon","lat")] , ele_dump[i+2 ,
c("lon","lat")])  #should pass (longitude,lattitude)
  theta[i]=a$ta
}
ele_dump$theta = c(NA,theta,NA)
#Turing angle for elevation
theta=numeric()
for(i in 1:(nrow(ele_dump) - 2)){
  a=bearing.ta(ele_dump[i,c("distance","elevation")], ele_dump[i+1, c("distance","elevation")] ,
ele_dump[i+2 , c("distance","elevation")])
  theta[i]=a$ta
}
ele_dump$ele_theta = c(NA,theta,NA)
rm(theta , i,a)

ele_dump$ele_roc = c(NA ,  abs(distance/((k72-
1)*2*sin(ele_dump$ele_theta[2:(nrow(ele_dump)-1)] *pi/360))),NA)
```

```
ele_dump$roc =c(NA ,  abs(distance/((k72-1)*2*sin(ele_dump$theta[2:(nrow(ele_dump)-1)]
*pi/360))),NA)
ele_dump$raw_speed = c(NA , 9.15*(log10(ele_dump$roc[2:(nrow(ele_dump)-
1)]))^2+17.68*log10(ele_dump$roc[2:(nrow(ele_dump)-1)])-11.93 , NA)
ele_dump$max_speed = c(120, ifelse(ele_dump$raw_speed[2:(nrow(ele_dump)-1)] >=120, 120 ,
ele_dump$raw_speed[2:(nrow(ele_dump)-1)]) , 120)
ele_dump$max_speed = ele_dump$max_speed*5/18
ele_dump$ele_theta_g_1 = c(NA , 1.55/sqrt(ele_dump$ele_roc[2:(nrow(ele_dump)-1)]) , NA)
ele_dump$ele_los =c(NA,
                ifelse(abs(ele_dump$ele_theta[2:(nrow(ele_dump)-1)]) <
ele_dump$ele_theta_g[2:(nrow(ele_dump)-1)] ,
                        (ele_dump$ele_theta[2:(nrow(ele_dump)-
1)]*ele_dump$ele_theta[2:(nrow(ele_dump)-1)]*ele_dump$ele_roc[2:(nrow(ele_dump)-1)]
+2.40)/2*ele_dump$ele_theta[2:(nrow(ele_dump)-1)] ,
                        sqrt((ele_dump$ele_roc[2:(nrow(ele_dump)-1)] +1.2 )*(
ele_dump$ele_roc[2:(nrow(ele_dump)-1)] +1.2) - ele_dump$ele_roc[2:(nrow(ele_dump)-
1)]*ele_dump$ele_roc[2:(nrow(ele_dump)-1)])),
                NA )


#Locating crest vertical curvature points
localMaxima <- function(x) {
  # Use -Inf instead if x is numeric (non-integer)
  y <- diff(c(-.Machine$integer.max, x)) > 0L
```

```
  rle(y)$lengths
  y <- cumsum(rle(y)$lengths)
  y <- y[seq.int(1L, length(y), 2L)]
  if (x[[1]] == x[[2]]) {
    y <- y[-1]
  }
  y
}
maxima = localMaxima(ele_dump$elevation)
ele_dump$distance_los = c(NA, ele_dump$distance[2:(nrow(ele_dump)-1)] -
ele_dump$ele_los[2:(nrow(ele_dump)-1)],NA)
library(dplyr)
ele_dump = ele_dump %>%
  dplyr::select(distance,max_speed,ele_los,distance_los)
for(i in 1:length(maxima))
  ele_dump = rbind(ele_dump, c(ele_dump$distance_los[maxima[i]],
1.25*(36.51*log(ele_dump$ele_los[maxima[i]]) - 78.09 )*5/18 ,NA , NA))
ele_dump =  subset(ele_dump, !is.na(distance))
ele_dump = ele_dump[order(ele_dump$distance),]
rm(anglefun,bearing.ta,localMaxima,maxima,i,k72,route_json)
```

```
############Speed profile simulation########
speed_profile <- data.frame(matrix(ncol = 5 , nrow = distance+1))
colnames(speed_profile) <- c("node_id" , "acceleration","speed", "time", "resp")
speed_profile[1, ] = c( 1 , 1 , 0 ,0 ,1)
#Speed Limits for the routes used in validation.
#speed_limit = c( rep(96.56, 3096), rep(112.65, (distance+2-3096)))  #Sandy pt rd, westward
#speed_limit = c( rep(112.65, 7562), rep(96.56, (distance+2-7562)))  #Sandy pt rd, eastward

#speed_limit = c( rep(96.56, 4550), rep(112.65, (distance+2-4550)))  #FM159 southward
speed_limit = c( rep(112.65, 12570), rep(96.56, (distance+2-12570)))  # FM159 northward
for(i in 2: (distance+1)){
  cat(paste(i," "))
  speed_profile$node_id[i] = ifelse( i >= ele_dump$distance[(speed_profile$node_id[i-1]+1)] ,
speed_profile$node_id[i-1]+1 , speed_profile$node_id[i-1])
  speed = speed_profile$speed[i-1]**2 + 2*speed_profile$acceleration[i-1]*1
  if(speed >=0)
    speed_profile$speed[i] = sqrt(speed)
  else
    speed_profile$speed[i] = 0
  respDist = speed_profile$speed[i]*7
  nodesInView = sum( ele_dump$distance < (respDist + i))
  if(nodesInView == speed_profile$node_id[i]){
    if(speed_profile$speed[i] < speed_limit[i]*5/18)
```

```
      speed_profile$acceleration[i] = 1
    else{
      speed_profile$speed[i] = speed_limit[i]*5/18
      speed_profile$acceleration[i] = 0
    }
  }
  else {
    brake_acc = numeric(nodesInView -speed_profile$node_id[i])
    for( k in (speed_profile$node_id[i]+1):nodesInView){
      if( speed_profile$speed[i] > ele_dump$max_speed[k]){
        noBrakeDistance = (speed_profile$speed[i]**2 - ele_dump$max_speed[k]**2)/(2*0.5)
        distFromNode =  ele_dump$distance[k] - i
        if(noBrakeDistance > distFromNode)
          brake_acc[k - speed_profile$node_id[i]] = (ele_dump$max_speed[k]**2 -
speed_profile$speed[i]**2)/(2*distFromNode)
      }
    }
    if(sum(brake_acc == 0) ==  (nodesInView -speed_profile$node_id[i])){
      if(speed_profile$speed[i] < speed_limit[i]*5/18)
        speed_profile$acceleration[i] = 1
      else{
        speed_profile$speed[i] = speed_limit[i]*5/18
        speed_profile$acceleration[i] = 0
```

```
    }
   }
   else
     speed_profile$acceleration[i] = min(brake_acc)
  }

  if(speed_profile$acceleration[i] ==1)
    speed_profile$resp[i] = 1
  else if (speed_profile$acceleration[i] == 0 )
    speed_profile$resp[i] = 1
  else if (speed_profile$acceleration[i] < 0 &&speed_profile$acceleration[i] >= -0.51)
    speed_profile$resp[i] = 1
  else if(speed_profile$acceleration[i] < -0.51)
    speed_profile$resp[i] = 4

  speed_profile$time[i] = ifelse(speed_profile$acceleration[i-1] ==0,1/speed_profile$speed[i-1] ,
(speed_profile$speed[i] - speed_profile$speed[i-1])/speed_profile$acceleration[i-1])
}
```