University of South Florida

DIGITAL COMMONS
@ UNIVERSITY OF SOUTH FLORIDA

Digital Commons @ University of South Florida

USF Tampa Graduate Theses and Dissertations

USF Graduate Theses and Dissertations

November 2021

Novel Approach to Integrate CAN Based Vehicle Sensors with GPS Using Adaptive Filters to Improve Localization Precision in Connected Vehicles from a Systems Engineering Perspective

Abhijit Vasili
University of South Florida

Follow this and additional works at: https://digitalcommons.usf.edu/etd

Part of the Computer Sciences Commons, Electrical and Computer Engineering Commons, and the Urban Studies and Planning Commons

Scholar Commons Citation

Vasili, Abhijit, "Novel Approach to Integrate CAN Based Vehicle Sensors with GPS Using Adaptive Filters to Improve Localization Precision in Connected Vehicles from a Systems Engineering Perspective" (2021). *USF Tampa Graduate Theses and Dissertations.*
https://digitalcommons.usf.edu/etd/9250

This Dissertation is brought to you for free and open access by the USF Graduate Theses and Dissertations at Digital Commons @ University of South Florida. It has been accepted for inclusion in USF Tampa Graduate Theses and Dissertations by an authorized administrator of Digital Commons @ University of South Florida. For more information, please contact digitalcommons@usf.edu.

Novel Approach to Integrate CAN Based Vehicle Sensors with GPS Using Adaptive Filters to Improve Localization Precision in Connected Vehicles from a Systems Engineering Perspective

by

Abhijit Vasili

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy in Electrical Engineering
Department of Electrical Engineering
College of Engineering
University of South Florida

Major Professor: Wilfrido A. Moreno, Ph.D.
Chung Seop Jeong, Ph.D.
Zhenyu Wang, Ph.D.
Mahshid Rahnamay Naeini, Ph.D.
Fernando Falquez, Ph.D.

Date of Approval:
November 10, 2021

Keywords: V2V, V2X, Kalman Filter, SLAM, SE

Copyright © 2021, Abhijit Vasili

Dedication

To Nana, Amma, Annai. To my Guru and Mavayya. The journey would not be possible without their endless support and belief and my faith in them.

Acknowledgments

I would like to thank my advisor, Dr. Wilfrido A. Moreno, for the constant guidance and support throughout my research.

I would like to thank Dr. Zhenyu Wang for always being there to guide me in the path of research. It was every day learning to work under his guidance.

I want to take this opportunity to thank Dr. Pei Sang Lin, Dr. Zhenyu Wang, Dr. Sisinnio Concas, and Dr. Achilleas Kourtellis. They have paved research ideologies and constantly guided me through the abundance of research experiences in the field of transportation while/during? working in their group Intelligent Transportation Systems Traffic Operations and Safety as a Research Assistant.

I want to thank the team at Brandmotion for the opportunity to work in the team testing applications for the Tampa Hillsborough Expressway Authority Connected Vehicle pilot project in Tampa. Later, being a part of the hardware installation teams and being a core intern working with the entirety of the operations in the Connected Vehicles' community of work.

I would like to thank Dr. Chung Seop Jeong, Dr. Mahshid Rahnamay Naeini, and Dr. Fernando Falquez for taking the time to be a part of my Ph.D. guidance committee and for providing valuable suggestions and guidance.

Table of Contents

List of Tables

List of Figures

Abstract

Research and development in Connected Vehicles (CV) Technologies has increased exponentially, with the allocation of 75 MHz radio spectrum in the 5.9 GHz band by the Federal Communication Commission (FCC) dedicated to Intelligent Transportation Systems (ITS) in 1999 and 30 MHz in the 5.9 GHz by the European Telecommunication Standards Institution (ETSI). Many applications have been tested and deployed in pilot programs across many cities all over the world.

CV pilot programs have played a vital role in evaluating the effectiveness and impact of the technology and understanding the effects of the applications over the safety of road users. The evaluation of applications from the pilot program has resulted in the core interest in discussing the challenges CV technology faces. In CVs, the vehicle's position is monitored by Global Positioning System (GPS) via Global Navigation Satellite System (GNSS) antennas placed on the vehicle. While the precision of GPS presented by GPS.gov states that the accuracy of high-quality GPS has a horizontal accuracy of less than or equal to 1.8 meters 95% of the time. In comparison, GPS-enabled smartphones with built-in Inertial Measurement Units (IMU) have shown an accuracy of 4.9 meters [1] under open skies. Also, it is stated that the GPS accuracy is impacted by the interference of physical objects such as “buildings, bridges, trees, etc.”[2].

The current CV applications are predominantly dependent on the GPS for the vehicle’s location measurements, making the accuracy of the GPS a significant factor impacting the performance of the CV technology. For example, the average width of a car is about 1.65 meters, and the width of the lane ranges from 2.5 meters to 3.25 meters. Considering an estimated position error from the GPS being 1.8 meters, the car location could be estimated to be a lane away from its actual lane position.

In this research, a novel approach of coupling vehicular sensors to the GPS devised using adaptive algorithms is proposed. The simulations results support the proposed novel approach that can be adapted in CVs to increase vehicle localization accuracy. The results presented show the Root Mean Square Deviation (RMSD) of about 1.5 meters compared to the GPS values.

Using the Systems Engineering (SE) approach, the aim of this research topic is to contribute to increase the accuracy in CV localization. A conceptual design to integrate in vehicle sensors into CV technology is presented. A Python program framework to implement the sensor fusion techniques is contributed as a byproduct of the research. Using the framework analysis of the simulation results are presented and documented. Finally, RMSD is used to evaluate the goodness of the novel approach and a cost benefit analysis is provided to understand the benefit of the novel technique.

Chapter 1: Introduction

1.1 Background

The number one casualties of road accidents have been statistically proven to be caused by cars and light trucks, which comprise sports utility vehicles (SUV) and minivans. The main reason for these fatalities have been shown to be driver error in the form of inattentiveness, carelessness, and other reasons [3]. According to the data published by the United States Department of Transportation National Highway Traffic Safety Association (NHTSA), the total number of fatal crashes constituted about 52,000 in the year 2018. Of these crashes, the involvement of passenger vehicles and light trucks contributes to about 77 %.

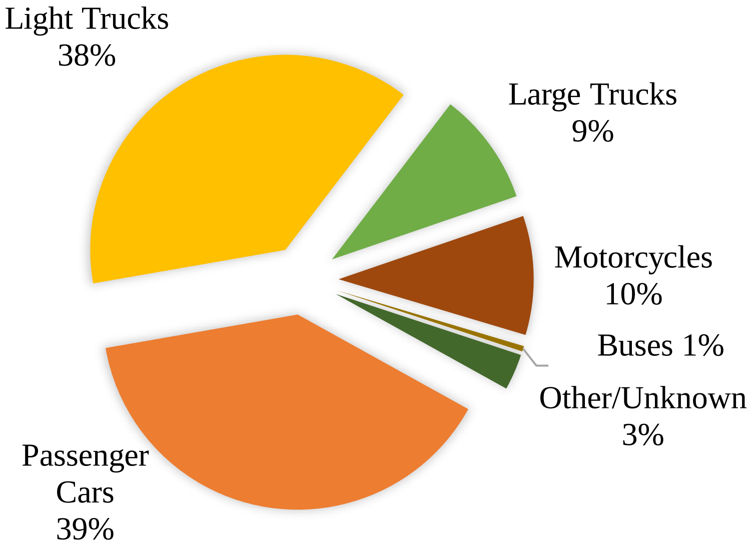


Figure 1.1 Fatalities divided into vehicle types reported by NHTSA.

While many engineering-based research projects deal with providing mitigating solutions to vehicle-related facilities, the answers do not only lie in the educational province of engineering but, it must require a holistic research approach where a cumulative solution(s) is found by multidisciplinary research teams such as economists, policymakers, engineers, public health personnel, all working together.

The major solution hovering over the problem of human carelessness driving is to reduce the involvement of humans in driving with the help of innovative solutions such as the ones being developed for autonomous vehicles. Although there has been advancement in the autonomous vehicle technologies in the research front and has started to enter the market by companies such Tesla, GM, etc., it is far from achieving complete autonomy, not until entirely or most of the vehicles that are placed on public roads are completely autonomous or have the capability to communicate within themselves, accurately determine localization, collision avoidance, and many other safety and mobility capabilities.

The primary goal for researchers approaching this issue is to increase the transportation system's safety and increase the transportation system efficiency. In this approach towards transportation systems, the safety of the transportation system deals with crash reduction and first responders time management; these processes must be mitigated under the umbrella of cost management. In [4], it is explained that there are different dimensions to safety in transportation systems, including incidents, crashes, incident emergency recourses, and human behavior. Further questions can be researched by considering the lighting levels, geometric parameters, traffic speeds, and weather conditions. Additionally, the effects of increasing safety to reduce the count on accidents and/or to reduce the severity of the accident and finally the impact of reduction of crashes on the economy. The second increase in efficiency deals with the research on understanding the causes of bottlenecks, queues, and slowdowns. This issue is addressed more in terms of travel time and traffic speed. These issues have been researched from the geometric layouts and using communication to devise vehicles and infrastructures with information from sensors.

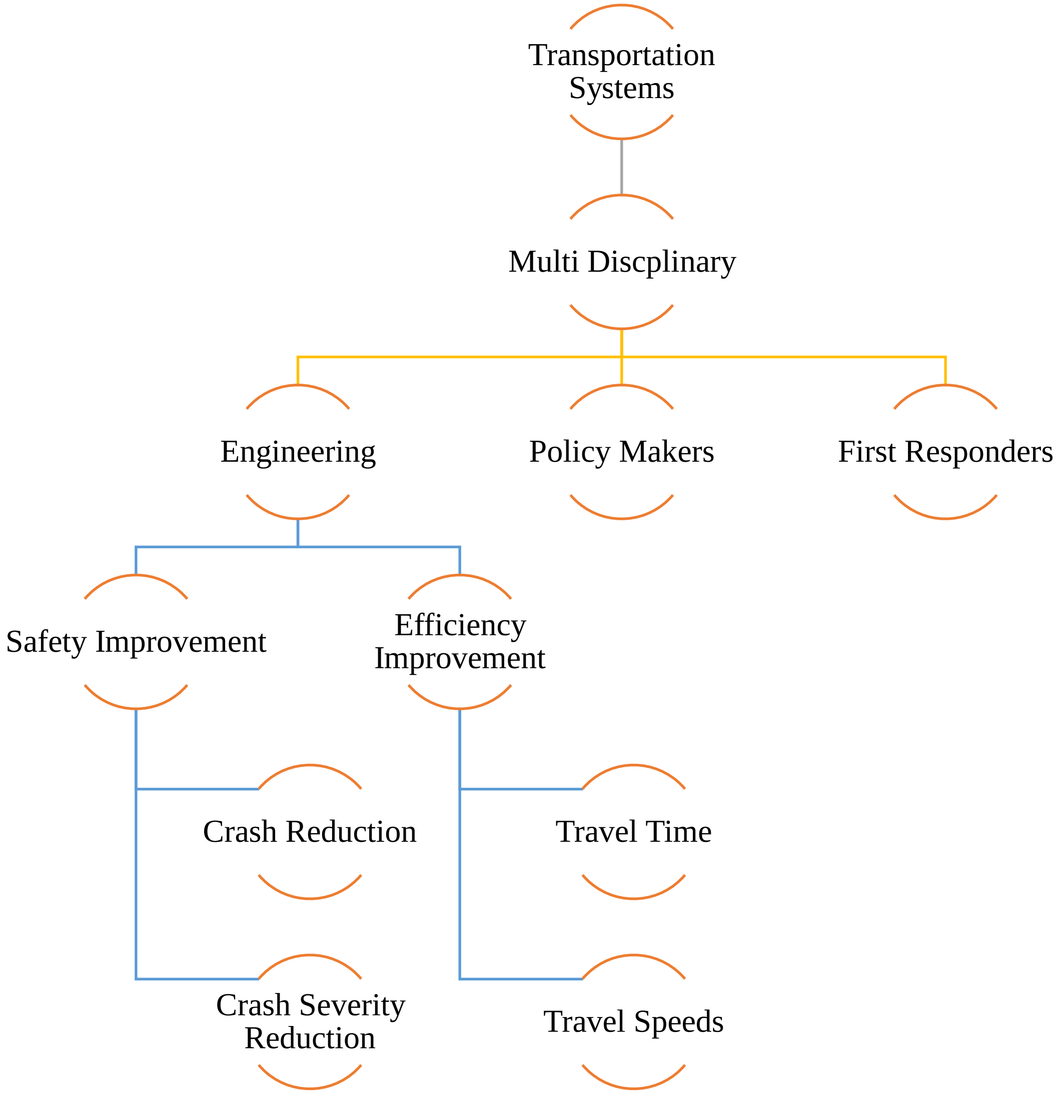


Figure 1.2 Research hierarchy in transportation systems.

Meanwhile, many major cities worldwide started adopting the concepts of smart cities, where sensors are used to detect and transfer information to the appropriate applications, first responders, governance, and other implementing organizations. The CV is a part of this urban movement which thrives on starting the channel of communication between the vehicles using this technology, communicate the information from these vehicles to the traffic control organizations and first responders, and communicate with the traffic signals and traffic control devices for bettering the mobility and safety of the traffic.

1.2 Contributing Statement

Though the idea is novel and the application for this technology is massive, the applications currently built to work with this technology are majorly dependent on the GPS. While for the testing of the applications, the quality of the data that the system delivers might work, in the real-world application, the error radius of the GPS is too high for the vehicle's use. Though there has been much research to increase the GPS's accuracy by adding IMUs and integrating the data to INS sensors, the systems have too many drift factors associated with them. In this research project, an adaptive Extended Kalman Filter (EKF) is used to optimize localization accuracy, which fuses the data from the vehicles' sensors and the GPS data to improve the accuracy of existing localization methodologies. The research also provides an approach to adapt to those vehicles with a limited number of sensors. However, the proposed solution is expected to be able to produce better accuracy than GPS.

A Systems Engineering (SE) approach was followed in the research process. The research closely incorporated the design engineering process from SE in transportation proposed by the International Council on Systems Engineering (INCOSE) Transportation chapter [5].

The main contribution in this research is the novel approach that integrates the in-vehicle sensors into the CV technology to enhance localization accuracy in autonomous vehicles following a Systems Engineering approach. The systems engineering approach was followed to identify the need for the novel solution. In addition, a by-product of this research is the design, development and validation of a research platform using Python for sensor fusion based research.

1.3 Overview

The dissertation focuses with an underlying accuracy limitation of the existing GPS. Further progressing into the document, emphasis is given on the present state-of-the-art technologies that have been used to improve the accuracy of the GPS

In Chapter 2 discussion about the different CV technologies and the benefits of Smart Cities will be made. Information about state-of-the-art technologies used in communications for the CVs will be discussed, their specific advantages and disadvantages will also be presented. Moreover, the applications used in the current CV pilot programs will be explained, and the challenges facing by these applications will be presented.

In Chapter 3, current technological research in Vehicle Localization will be discussed. A brief understanding of localization in self-driving cars will be presented, and the current state of localization in connected vehicles will be discussed. Further, the progression of localization from dead reckoning to fusion of GPS with Inertial Navigation System (INS) will be presented. The focus of this chapter will be a detailed description of the evolution of navigation systems and its limitations. Primary emphasis will be on the GPS and the errors the system accumulates. The accuracies of the current state-of-the-art techniques, their advantages, and limitations will be tabulated.

In Chapter 4, a brief introduction to Controller Area Network (CAN) with a general overview of the CAN fundamentals and examples of sensors within the CAN are described. In addition, a CV fusion block diagram, and the methodology used to integrate the CAN into the CV technologies are illustrated.

In Chapter 5, the vehicle kinematic bicycle model and Constant Turn Rate Constant Velocity (CTCV) models are presented, which will be used as a base to fuse the vehicle sensor data with the GPS. Further in the chapter, the Audi A2D2 dataset used in this research is described. Also, the data provided by the CV technologies and in specific, the data from the Tampa CV pilot project is also explained.

In Chapter 6, using the kinematic models, adaptive EKF are used to simulate the data from the A2D2 dataset. Two algorithm approaches are provided to explain the adaption of multiple sensors to perform optimization in localization. The results of the simulations are provided by breaking down the results into analysis and evaluation. Analysis is used to explain in detail the limitation of the algorithm with fewer sensors over an algorithm with more sensors. And evaluation was used to present the performance of the algorithms compared to each other and compared to GPS locations.

Chapter 2: Connected Vehicles[1]

2.1 Smart City

According to IBM, Smart City is defined as information and communication technologies (ICT) to sense, integrate and analyze the information in main systems in running cities [6]. The smart city comprises of three layers perception layer, application layer, and network layer, this can be the future measurable, connected, operable and intelligent [7]. Figure 2.1 shows the architecture of the three layers that form smart cities.

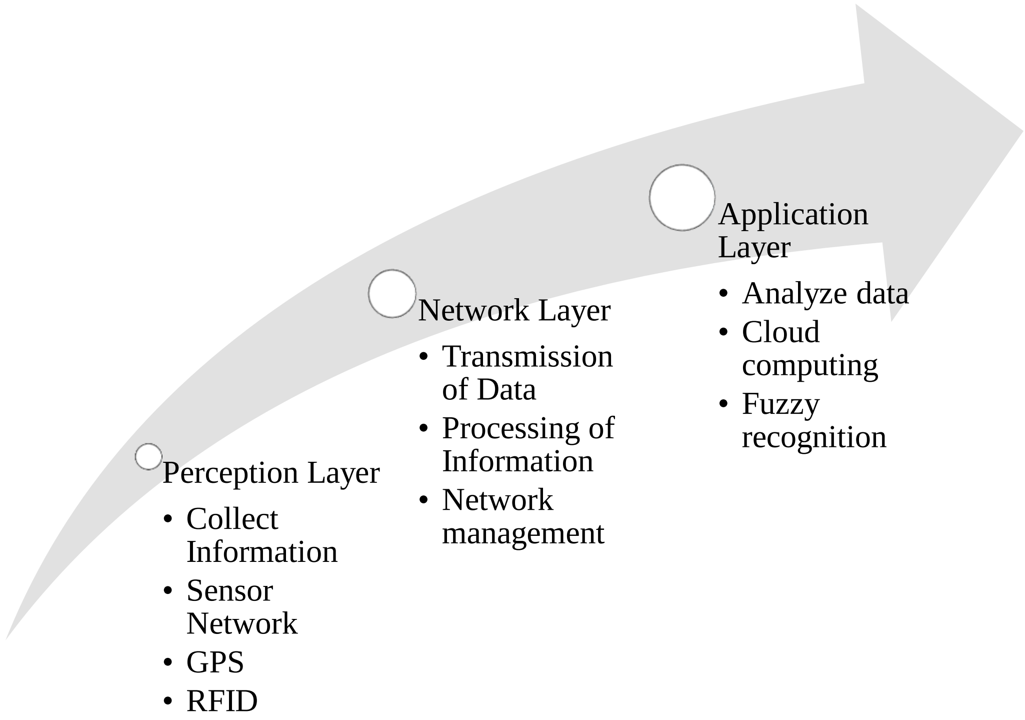


Figure 2.1 Smart city architecture.

[1] Parts of this chapter were published by IEEE in [8]. Permission is included in Appendex B

With the advent of Internet of Things (IoT), many cities are under extreme evolution to integrate technology into the routine of the residents. A brief framework to access the routines of a city to digitalize the cities then giving them capabilities to apply intelligent functions is a crucial step. For this purpose, smart cities adopt (ICT) [9]–[11]to collect data from different types of sensors to improve the quality of safety, mobility, emergency, and environmental issues. One of the parts of the smart cities initiative is the CV.

2.2 Connected Vehicle Technology Introduction

There are two driving issues for bringing communication between vehicles, the need for the safety of transportation and the increasing population in cities, which will lead to congestion, which will tamper with the economic and environmental problems. In a 2012 report [12], it is mentioned that the US accumulates 121 billion dollars in conjunction with the extra fuel costs and travel time because of congestion in 498 urban areas. Connected vehicles show a promising solution to traffic congestion and also using On-Board Units (OBU) to process driver assistance signals [13], [14].

CV refers to vehicles that are connected wireless, which communicates with other vehicles and also with other external environments. To encompass all the different communications the vehicle progresses to have, they can be bracketed into three major groups of communication. Vehicle to Vehicle (V2V), Vehicle to Infrastructure (V2I), and Vehicle to Everything (V2X). CVs are considered as the fundamental system that will form the Internet of Vehicles (IoV), a mobile system that collects, shares, analyzes, computes, and sends the information which will evolve the next generation ITS [15].

2.3 Vehicular Communication

It is believed that the advances in inter-vehicle communication will reinvent the future of transportation systems. With the introduction of inter-vehicle communication or V2V, information that is gathered in a vehicle is not isolated to itself anymore. The data collected from the onboard sensors, control systems, and onboard computers can be efficiently communicated to the vehicles present in the proximity. Without any addition to the road infrastructure, safety applications such as collision detection, blind-spot warning, information sharing, and interactive applications can be enabled [16], [17].

CV is a form of Ad-hoc network, a network structure that is decentralized wireless communication that creates temporary communication channels between the clients. In the CV, the Ad-hoc network is called Vehicular Ad-hoc Network (VANET)[18].

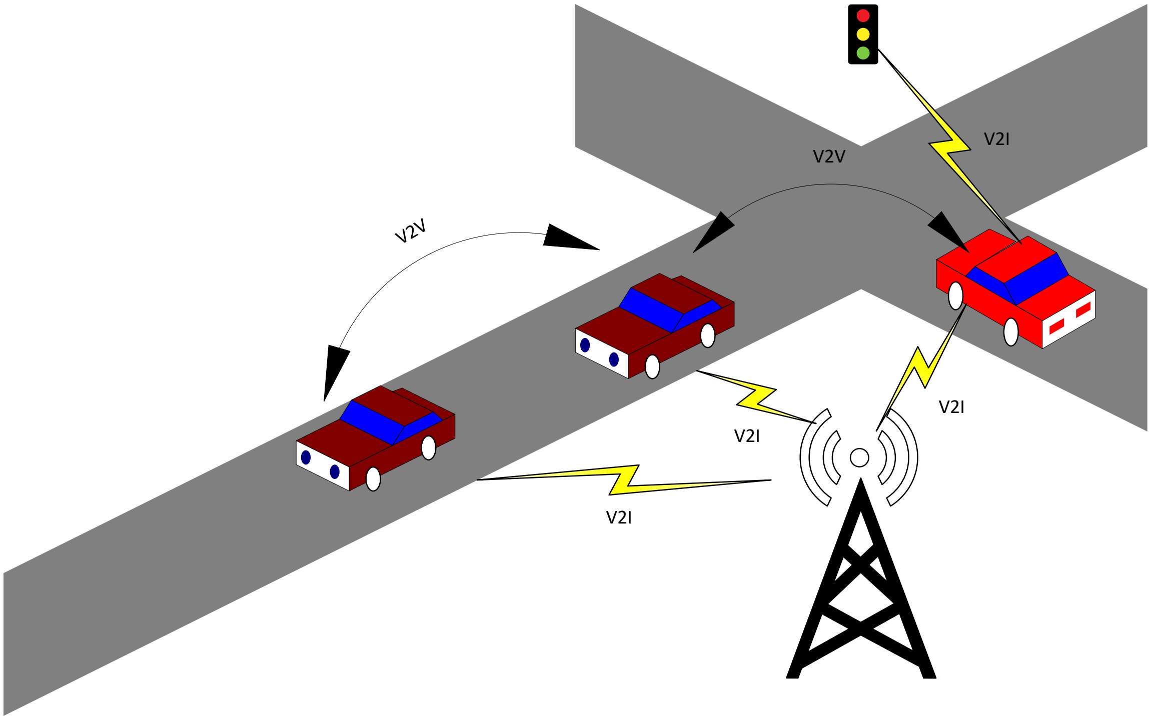


Figure 2.2 VANET illustration.

Figure 2.2 shows a simple example of VANET. Clients in VANET are called On-Board Units (OBU). These are embedded computers that are devices in every car to gather the information from the vehicle and the location data from the GPS. The nodes in VANET are called Road Side Units (RSU). These devices are used to gather the data from vehicles to further process in need for application use. When two cars communicate i.e., when two OBUs communicate, it is a V2V communication, and when the OBU is communicating with the RSU, it is V2I communication [19].

The most researched technologies for VANET communication are Dedicated Short Range Communication (DSRC), Long Term Evolution (LTE) [20], and 5G [15].

2.3.1 Dedicated Short Range Communication (DSRC)

DSRC is one of the oldest communication mediums for VANET. The amendment of IEEE 802 in 2011 to form the IEEE 802.11 standard which describes the DSRC protocol. IEEE 802.11p comprises of the physical layer and a part of data link layer called Medium Access Control (MAC) sublayer. IEEE 802.11p is a modification of IEEE 802.11, which covers the Wi-Fi standard [21]. This addition of the MAC sublayer to the physical layer creates the distinction of Vehicular Wi-Fi called Wireless Access for Vehicular Environments (WAVE). Different layers of the protocol use the WAVE addressed by the IEEE 1609 standard. There have been extensive academic and industry research that has been addressing the communication attributes of DSRC [22]–[26]. In addition, other research that has focused on enhancing the performance in MAC layer and physical layer have been performed. [27]–[30].

IEEE 1609.4 is used in MAC sublayer extension for channel switching. IEEE 1609.3 is used as the network layer, which includes the short message protocol for WAVE (WSMP). Further, 1609.2 is the security layer, but DSRC also uses Internet Protocol 6 (IPv6), Transmission Commission Protocol for the network and security layers.

The US Federal Communication Commission (FCC) has dedicated the 5.85 GHz to 5.925 GHz band, a 75 MHz bandwidth allocation for DSRC [31]. Also the European Telecommunication Standards Institute (ETSI) has dedicated the band from 5.795 to 5.805 GHz of DSRC [32].

2.3.2 Long Term Evolution (LTE)

The LTE is an evolution from Global System for Mobile Communication (GSM), which uses the Universal Mobile Telecommunication (UMTS) standards[33]. In [34], it is mentioned that the LTE has an uplink speed of about 50 Mbps and a downlink speed of about 150 Mbps. In the practical approach, the speeds depend on the availability of bandwidth from the carrier tower.

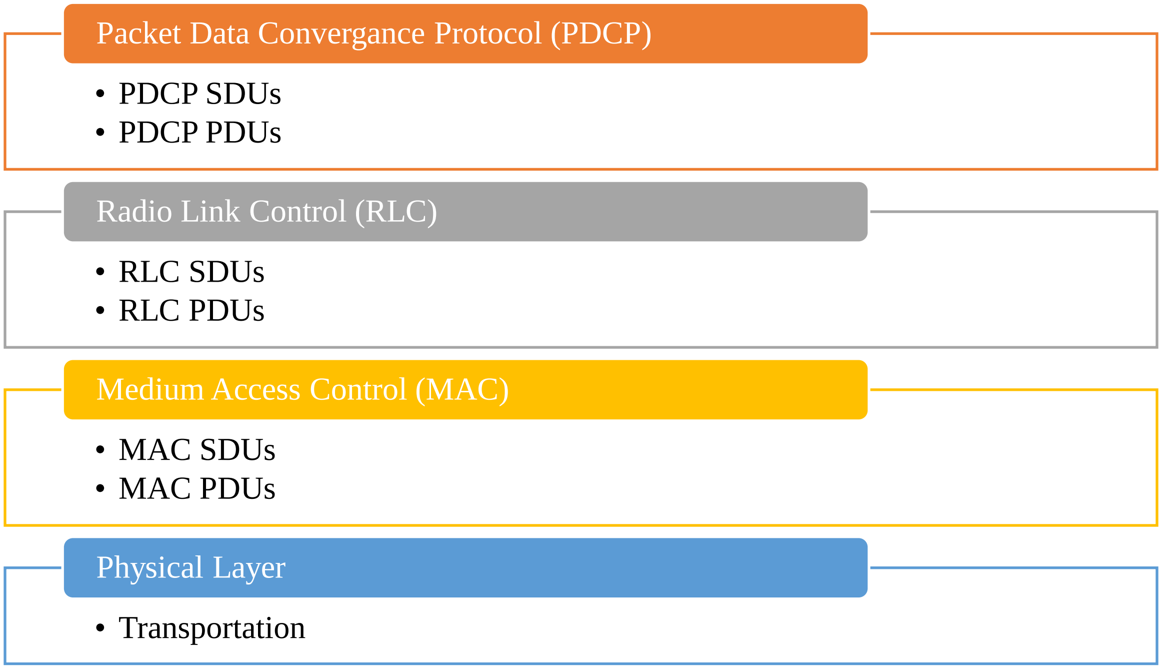


Figure 2.3 LTE communication layers.

Figure 2.3 shows the four different layers in LTE[35], [36]. The first layer is the physical layer, where all the information from the second layer Medium Access Control (MAC) layer, is carried over the air. It controls protocols such as link adaption, synchronization with cell phone and power control [37]. In the second layer, MAC data from the transport layer is added with a specific MAC header, and at the end of the data, padding is added. In the RLC or the third layer, operations such as segmentation and reassembly are performed. Here there are three modes for operation, i.e., acknowledgment, un-acknowledgment, and transparent modes. In the final layer PDCP layer, operations such as decryption, redundancy removal, sequence numbering, and decompression are performed.

Table 2.1 shows the outline differences between the DSRC and LTE pertaining to the CV technologies.

Table 2.1 DSRC vs LTE.

DSRC	**LTE**
No additional infrastructure is required (Decentralized)	A base station is required to be present
Onboard Communication using Antenna	Cell network
3 – 27 Mbps	50Mbps uplink 150 Mbps downlink
Good beacon transfer until medium volume	Good beacon transfer even in high volume

2.3.3 5G

With the rapid increase in vehicular communications, a new application for IoT was referred to as Internet of Connected Vehicles (IoCV) [38], [39]. The IoCV is defined as a mobile communication system in which vehicles communicate within themselves and the RSUs with the help of public networks[40]. It is specified that 5G network is crucial to handle the heavy loads of connectivity requirements in IoCV [41], [42]. 5G has high bandwidth and low latency compared to 4G, with an increase in quality for users [43]. With the help of 5G, the huge data generated by vehicles can be transferred to cloud for processing to tackle the applications of IoCV [44]. The major limitation for present implementation is the distance of the macro base station and the cloud, which will limit feedback operations to complete tasks in IoCV [45]. To provide real-time application feedback and services, the implementation of edge computing is crucial. By implanting edge computing and local computing capabilities at RSU and user level, data uploading tasks can be reduced and more applications can be implemented [46], [47].

2.4 Applications

The major advantage for CV is that opens a gateway to implement various number of applications. There have been about 40 applications that have been showcased in the United States Department of Transportation (US DOT) CV pilot Deployment Program [48]. The applications are categorized majorly into V2I safety, V2V safety, Environment applications, Mobility applications. Most of the applications listed in the spec of the CV scope are conceptual research approaches to apply the problem but were not actually developed to be used in real tests. A list of applications that are being tested in the current Pilot Programs are explained below.

2.4.1 V2V Safety Applications

Below is a list of safety application that use V2V communication.

- Emergency Brake Lights (EEBL) [49]: In case a vehicle brake suddenly, this application will provide not just the car behind but also cars on the same line a little further behind with an alert for the drivers to be aware of the situation.
- Forward Collision Warning (FCW) [49]: This application alerts the following vehicle with a warning if the driver is driving too close to the leading vehicle in order to alert a potential collision between the cars.

- Intersection Movement Assist (IMA) [49]: This application alerts the drivers when it is unsafe to enter the intersection in case of a large building or any other object physically blocking the view of the driver entering the intersection.
- Blind Spot Warning (BSW) [49]: This application triggers an alert if a vehicle is trying to change lanes and turns on the indicator and a car is present in the blind spot area.
- Turning In Front of Bus [50]: This application provides an alert to the bus driver when a car is turning in front when the bus is stationed at the bus stop.
- Left Turn Assist [49]: It is similar to the IMA, but in this case, the vehicle is trying to make a left turn, and the alert is posted if there is traffic coming in the opposite direction.

2.4.2 V2I Safety Applications

Here, a list of safety applications that use V2I communication.

- Red Light Violation (RLVW) [51]: The application uses signal timing to alert the vehicles if the car is positioned crossing the stop line.
- Curve / Banking Speed Warning (CSW) [51]: The application uses speed detection radars to alert vehicles if the speed of the vehicles is more than the posted speed limit for the curve.
- Stop Sign Assist (SSGA) [51]: This application uses cameras and lidars to alert the vehicles approaching a stop line.
- Pedestrians on Crosswalk [52]: A LiDAR is usually used in this application to keep monitoring the crosswalks and send alerts to cars approaching the crosswalk in the presence of bicycles and pedestrians.

- Spot Weather Warning (SWIW) [53]: Hazardous weather warnings are broadcasted to the vehicles entering areas with extreme weather conditions.
- Do Not Pass: This application alerts the drivers if a vehicle is trying to pass and there is a vehicle coming in the opposite direction.
- Work Zone Alert (RSWZ) [53]: The application sends work zone alerts to the vehicles that are approaching work zones.

2.4.3 Mobility

Henceforth are listed mobility related applications.

- Traffic Signal System (I-SIG) [54]: This application changes the signal timing, depending on the priority of the signal which can be set to increase the movement of cars in one direction.
- Transit/ Fright Signal Priority (TSP) [54]: The application changes the signal when transit vehicles are approaching the intersection.
- Dynamic Speed Harmonization (SPD-HARM) [55]: This application recommends vehicles with appropriate speed in case of congestion or incidents to maximize mobility and reduce crashes.
- Platooning/ Cooperative Cruise Control (CACC) [55]: This application is used to dynamically monitor and control the cruise control in platooning vehicles to improve the traffic flow.
- Queue Warning (Q-WARN) [55]: The application warns the driver about the queues that are already present on the roads.
- RESCUEME Application [56]: This application is an mixture of applications that are designed to be used by first responders.

  - Routing decisions to emergency vehicles.
  - An incident scene alert is broadcasted to the drivers that are approaching the incident with the appropriate traffic behavior operations.
  - In case of evacuation, vehicles are provided with safe routes to the nearest shelter.

- Transit and Ride Sharing (T-DISP) and (D-RIDE) [57]: The application coordinates between passengers and different public transport providers to increase transit operations.

2.4.4 Environmental

The applications that are designed to enhance the environmental impact are mentioned below.

- Eco Traffic Signal Priority & Timing [58]: Using the same technologies from traffic signal priority, increasing signal timing for lanes with more congestion which invariantly will reduce the fuel consumption.
- Eco Cruise Control [59]: Derived from the cooperative cruise control application, the platooning vehicles variables are used to optimize the cruise control parameters to reduce emissions.
- Eco Ramp Metering [59]: This application is also a derivative of traffic signal priority where the signal priority for the queued vehicles entering the ramps and exiting the ramp are given priority to reduce emissions.
- Eco Lane Management [59]: This application is designed to establish pre-defined eco lanes, which dynamically change the lane parameters regarding weather conditions.
- Dynamic Eco Routing [59]: A navigation application that provides vehicles with the most eco-friendly route.

In Table 2.2, a list of the applications that have been used in the pilot projects in the US [8] is listed. Detailed information on the performance of the applications has been provided in the respective pilot project reports [60]–[64].

Table 2.2 Applications tested in CV pilot projects in the US.

Application Name	**NYCDOT CV**	**THEA CV**	**WYDOT CV**
Forward collision warning	✓	✓	✓
Emergency break light	✓	✓	
Blind spot warning	✓		
Lane change assist	✓		
Intersection movement assist	✓	✓	
Vehicle turning in front of transit	✓	✓	
Intelligent traffic signal system	✓	✓	
Speed compliance	✓		
Work zone warning	✓		✓
Banking speed	✓	✓	
Red light violation	✓		
Oversize vehicle compliance	✓		
Emergency evacuation information	✓		
Pedestrian at crosswalk	✓	✓	
Mobile access pedestrian signals	✓	✓	
Transit signal priority		✓	
Situation awareness			✓
Spot weather warning			✓
Distress notification			✓
Wrong way entry		✓	
End of ramp speed warning		✓	

2.5 GPS Impacting CV Applications

Figure 2.4 illustrates a scenario where a vehicle is trying to make a right turn, and another vehicle is in the lane. The BSW application will alert the driver making the right turn once the indicator is turned on about the car that is present in the blind spot. But in this scenario, because of the error of the GPS, the location of the vehicle is assumed to be behind the car that is making the right turn. Since the application has been misinformed about the position of the vehicle being behind the car making a right turn and it assumes that the right lane has no car and no alarm will be triggered, which will defeat the purpose of the application. In Figure 2.5, a reverse scenario if the vehicle was actually behind the vehicle turning and the GPS gives the wrong position of the vehicle showing the vehicle in the blind spot area, a false BSW alert will be sent to the driver, thereby causing confusion.

In Figure 2.6, the illustration shows another scenario in which the cars are following each other in the same lane. The leading car makes a sudden brake because of an obstacle, the car that is following should immediately receive an EEBL warning. In this scenario, the error from the GPS has given the application the location of the following car to be not behind the leading car. In this case, the application will not send an EEBL to the following car, which can result in a crash defeating the purpose of the application.

In Figure 2.7, a similar scenario as previously is presented where there is a leading car and a car following the leading car. A FCW will be alerted to the following car when the leading car slows down, or the two cars come too close to each other. In our scenario, the GPS error sends the location of the vehicle further behind the vehicle than the actual position. If the vehicle is not in the range for FCW, no signal will be given to the following car.

Figure 2.8 illustrates another scenario, where a queue of cars are backed up making a right turn or trying to exit an interstate, and the exit ramp is backed up. GPS keeps jumping locations of the vehicles within the error range. In this scenario, the locations of a group of vehicles are shown in a lane where there is no queue, but an approaching vehicle 200 meters away will be sent a queue warning of the wrong lane.

All the illustrations are pointers as to the importance of the precision of the GPS. With the current accuracies, the applications will perform with a lot of false alarms. There is an urgent need to address the problem of GPS accuracy.

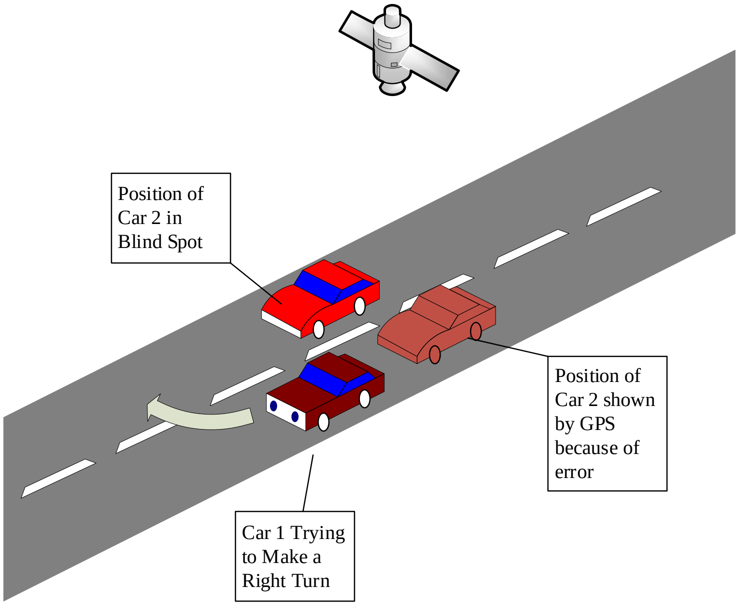

Position of Car 2 in Blind Spot
Position of Car 2 shown by GPS because of error
Car 1 Trying to Make a Right Turn

Figure 2.4 Illustration of GPS error not giving blind spot warning.

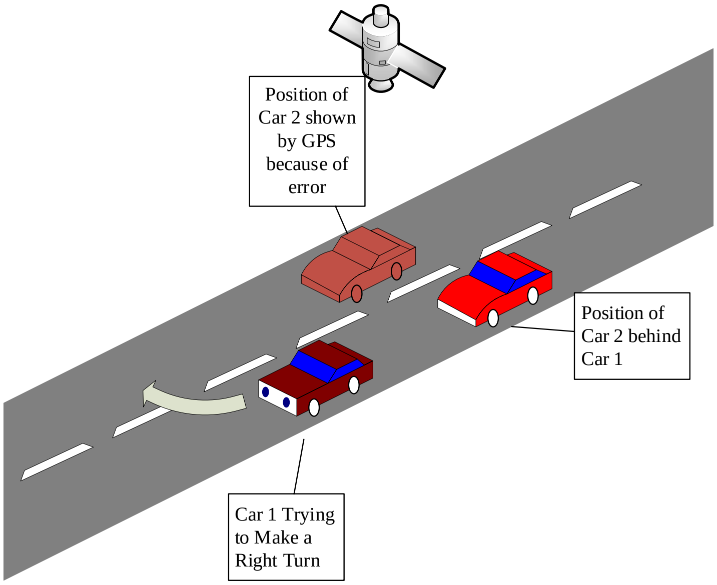

Position of Car 2 shown by GPS because of error
Position of Car 2 behind Car 1
Car 1 Trying to Make a Right Turn

Figure 2.5 Illustration of GPS error giving false blind spot warning.

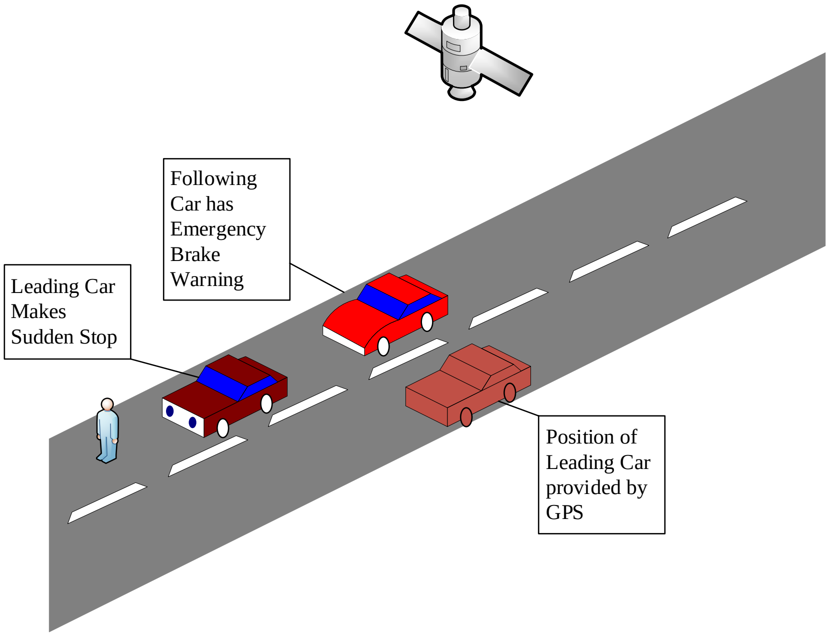

Following Car has Emergency Brake Warning
Leading Car Makes Sudden Stop
Position of Leading Car provided by GPS

Figure 2.6 Illustration of GPS error hindering performance of EBW.

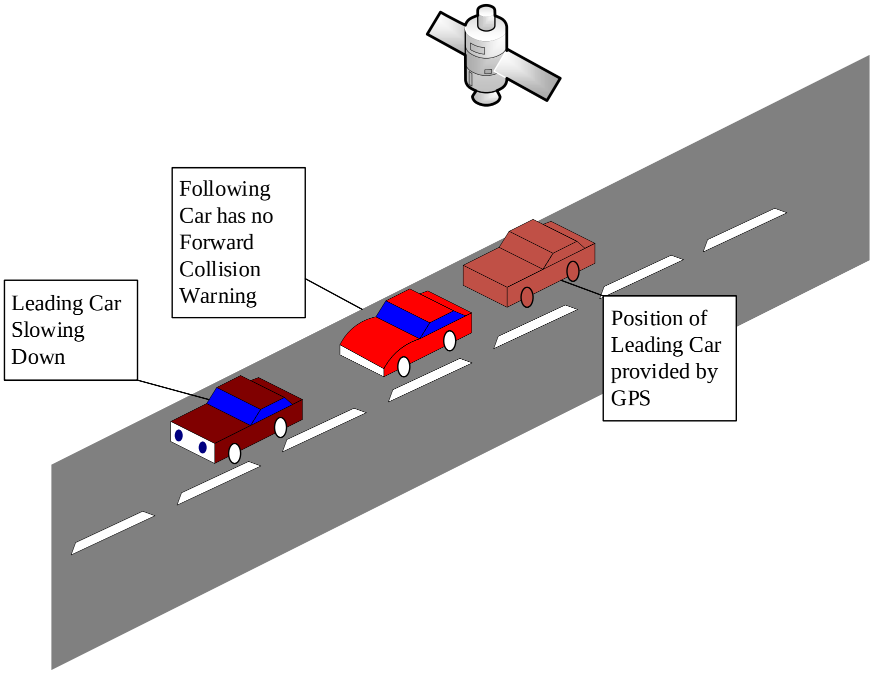


Figure 2.7 Illustration of GPS error hindering performance of FCW.

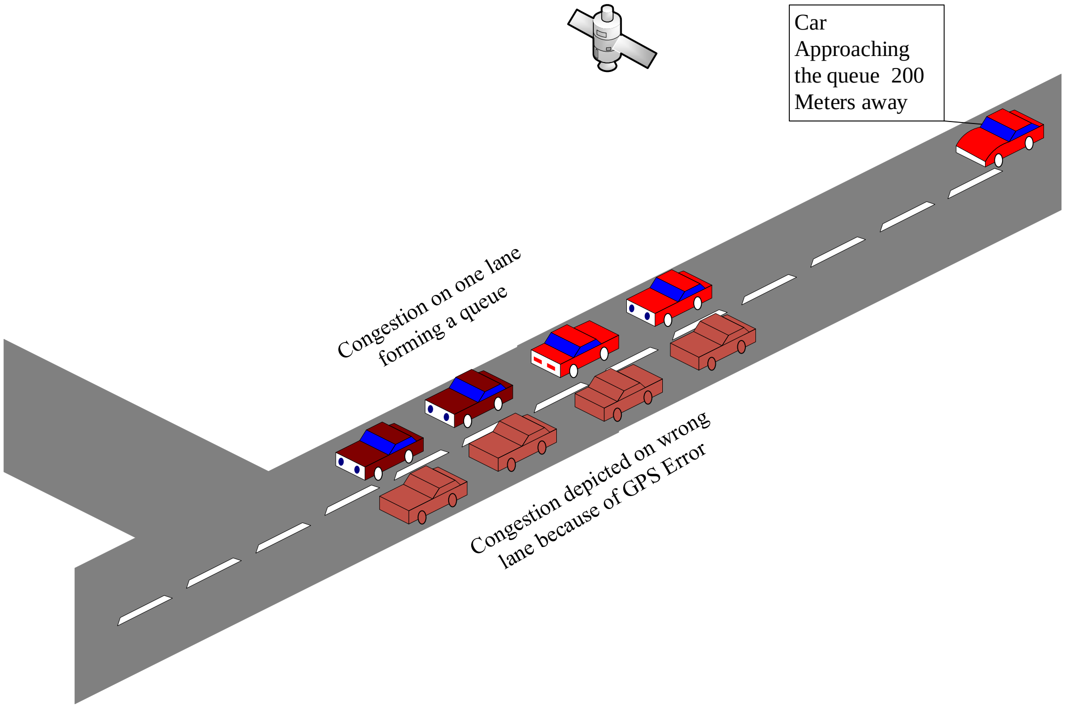


Figure 2.8 Illustration of GPS error hindering performance of queue warning.

Chapter 3: State-of-the-Art Localization

Vehicle navigation is one of the most important aspects of the automobile industry[65]–[73]. A lot of innovative technologies have evolved over time. As mentioned in [74], with information like where and when to make a lane change in coordination with the planned route, the accuracy of the navigation systems and maps has to be improved [75]–[78]. Also, it is mentioned that with the development of ITS applications such as Advance Driver Assistance Systems (ADAS), not only depend on the accuracy of navigation systems but also the reliability of the systems[79]. The major commercially available navigation systems use the information received from the GPS and match it to the digital maps called map matching [77], [78], [80], [81]. In map matching, the most likely position of the vehicle on the road is estimated with the input from the GPS and is matched with the position of roads in the digital maps. Urban environments pose a bigger problem with the presence of obstacles like buildings which will force the GPS to work with a poor geometric constellation of satellites. This will increase the errors in estimations [82]–[85]

3.1 Dead Reckoning

It is one of the oldest navigation techniques. In this technique heading sensors or compass sensors and distance measurement sensors are used to calculate the displacement vector. This method is used in a recursive method to determine the position of the vehicle. However, the accuracy of the dead reckoning method constantly degrades because of the cumulative errors in the sensors [86].

3.2 Global Positioning System (GPS)

GPS is a part of GNSS. GPS is a US-owned utility that provides navigation, timing, and position[2]. The GPS uses six orbital planes around the earth with 4 satellites in each orbital plane [87]. The GPS needs three satellites to produce location on earth. The fourth satellite is used to validate the other three satellites.

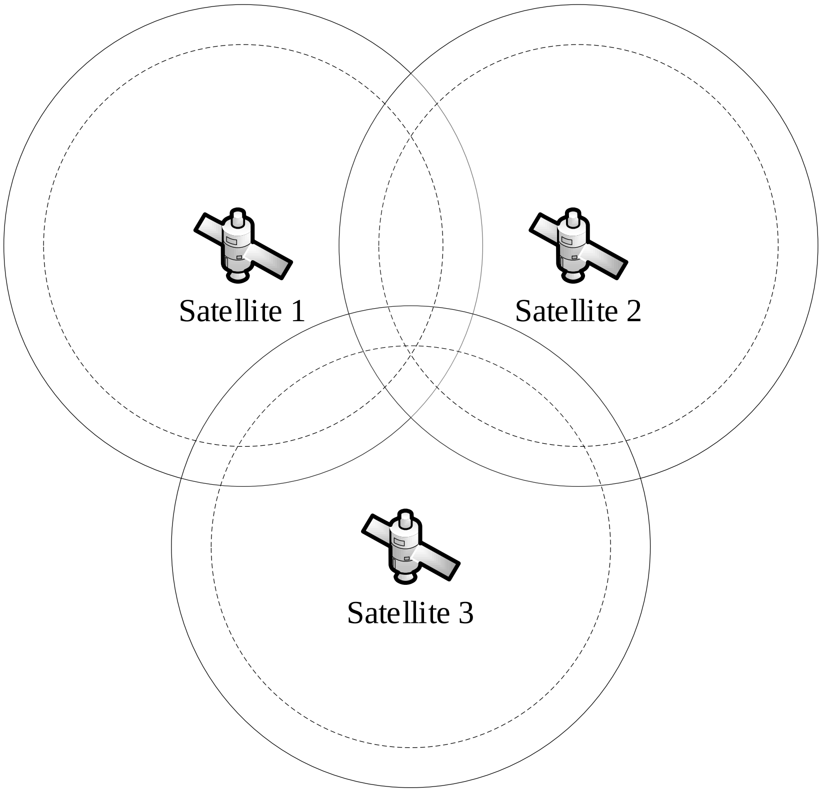


Figure 3.1 GPS positioning methodology.

As shown in Figure 3.1, as a basic concept, GPS determines the position of the GPS-enabled device using a process of triangulation. This is calculated by measuring the distance of an unknown location to three known locations. Then the distance being the radius of the intersection of the three circles will be the position of the GPS device. Table 3.1 shows the errors that are accumulated in a GPS [87].

Table 3.1 GPS errors description.

Error Name	**Description**
Ionosphere delays	Atmosphere slows the satellite speed.
Signal multipath	GPS signal reflecting off buildings
Receiver clock error	Improper timing error at receiver’s device
Orbital error	The satellite’s reported wrong position

3.2.1 Radio Technical Commissions of Maritime (RTCM)

This method uses a correction stream called RTCM on top of the base GPS signals to improve accuracy. It is a government protocol where base stations are placed at various places that send out a correction stream that has information about the errors which could have been added in the transmission of the data from the satellite. These correction streams of data have information about the turbulence in the ionosphere and troposphere. GPS devices can parse the data by themselves and correct the errors presented by the RTCM.

The drawbacks of this method are, the RTCM is not easily accessible to the public, the GPS has to be equipped with either radio communication or connected to the internet, and also there should be a base station located close enough to the area where the GPS is located. With the modern-day advancement in manufacturing sensors, an RTCM base station can be built at one's own convenience with the help of a GPS module capable of generating the RTCM correction data, a controller to gather the RTCM data, and push it to the internet or the radio signal. The drawbacks are that a very sensitive antenna and a clear sky will be needed. Even with the modern-day advancement, the base system can be built at a fairly low price, but the major drawback is the correction message can only be generated at a speed of 1 Hz. [88]

3.2.2 Precise Point Service (PPP)

The official term for PPP is Standard Positioning Service (SPS). This technique is capable of meter-level accuracy. Another official service is based on L1, and L2 P-code measurements and the broadcast data called Precise Positioning Service (PPS)[88].

PPS is capable of more precise (and accurate) positioning. In principle, dual-frequency data will remove all the ionospheric errors, and the P-code provides higher precision code measurements. However, since the P-code is encrypted, PPS is only available to authorized military and government users.

3.2.3 Differential GPS (DGPS)

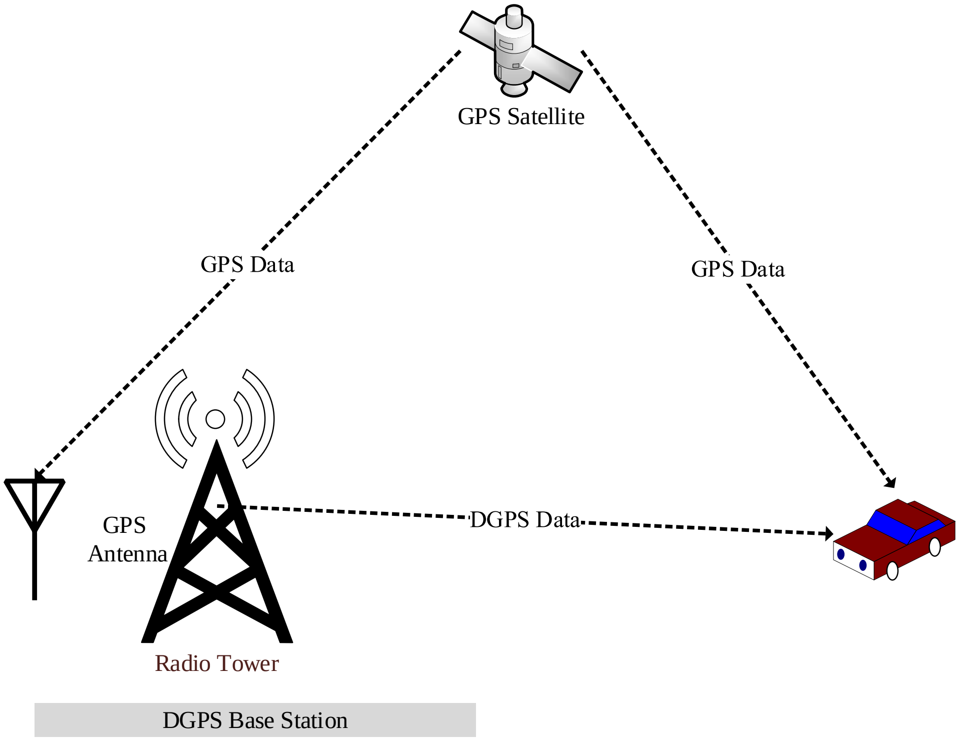


Figure 3.2 DGPS communication diagram.

In DGPS, a GPS receiver position is fixed and called the Base station, and the position of this GPS base station is determined at high accuracy[88]. The base station compares the position received from the satellite to its actual position, as shown in Figure 3.2. Differences between the positions can be calculated as satellite errors. These errors are sent to other receiver's stations from the base station, these errors are then incorporated, and the corrections into their position can be calculated.

Differential positioning requires correspondence between the base station and vehicles. Assuming errors should be applied progressively, something like four GPS satellites ought to be in view at both the base station and the vehicles. The outright accuracy of the vehicle's processed position will rely upon the outright accuracy of the base station's position.

3.2.4 Real-Time Kinematics (RTK)

At the basic level, the range is calculated by observing the carrier cycles between the vehicle and the satellite and then multiplying the wavelength to the number of cycles. The calculated errors will not reduce errors from satellite clock, ionospheric errors, and tropospheric errors[88]. To take advantage and eliminate the errors, the RTK needs measurements from the base station to the rover station. Like the DGPS, the accuracy depends on the distance of the vehicle from the base station.

3.3 Inertial Navigation System (INS)

An INS involves two particular frameworks, the IMU (inertial measurement unit) and the microcontroller. The IMU involves accelerometers, gyros, and magnetometers that give acceleration and angular velocity estimations. The controller that is the subsequent part utilizes these estimations to process the situation of the framework.

The most notable arrangement obliged for this accuracy issue is to combine an INS to the GPS. Numerous examinations [83], [89]–[93] have taken care of this issue with altogether different filters techniques. In any case the INS is an expensive sensor technology, and INS application in the vehicle application is unreasonable.

In the ego sensors such as vision sensors, radar, and laser scanners can only be utilized to localization when the data that is acquires from the ego sensors are fused in some way with the GPS and then use the Map matching techniques to find the exact road features. This data fusion was achieved using Bayesian filter. The filter based sensor fusion approaches have been extensively used with the positioning computation [74], [94], [103]–[105], [95]–[102].

The Extended Kalman Filter (EKF) is the incredible and obviously used data combination and filtering calculation. It uses the Taylor series to linearize the non-direct conditions related with the limitation issues [94]–[96], [105]. But the EKF gives viable and strong execution to viable applications, the linearization cycle can incite contrasts in an especially nonlinear system. To prevent this issue, progressed non-direct filtering systems, for instance, the molecule filter was introduced [102], [106]. Nonetheless, this strategy requires pointless computational capacity to be executed in installed structures.

3.4 Light Detection and Ranging (LiDAR)

The LiDAR system uses a laser to generate laser pulses to measure the distances to an object and generating a 3D map of the environment[107]. The sensor uses the Technique of Time of Flight (TOF), which is the time taken for the signal from the transmission of the light pulses to the reception of the light back to the sensor. Though LiDAR is primarily used in environment mapping, obstacle avoidance, and detection. The sensor can be integrated with GPS and INS to increase the accuracy of localization applications[108]–[110]. Though the accuracies of the LiDAR sensors are good the major drawback for the solution is that it is a highly expensive sensor and also requires high computational power which can affect real-time applications[108]. Another drawback of the LiDAR solution is that it can fail to scan objects such as glass [110]–[112].

3.5 Vision Based Localization

Research on vision based localization has been conducted because the systems are more robust and reliable and do not encounter interferences which could be a drawback in laser-based proximity sensors[113], [114]. The vision-based systems have many applications including edges, lanes, and intersections on roads. In fact, it has been mentioned [115] that the major goal of computer vision is vision-based navigation. It is also mentioned that visual odometry provides online estimations of the position of the vehicle by analyzing the image sequences captured by the cameras[116].

The major limitations that are posed by visual localization are the analysis is computationally expensive since it requires multiple steps of analysis such as gathering the image, analyzing for the features such as lines, corners etc. then matching the frames of the images and then calculating the position from the pixel displacement between different sequential frames. Also, vision-based localization is highly sensitive to the environment, illuminations, textures, quality of the images, and shadows to name a few. This will make the localization unreliable in extreme environmental conditions [108], [109].

3.6 State-of-the-Art Comparison

The RTK framework is highly accurate and simple to initialize however is perplexing to set up and costly. The RTK framework needs around two recipient frameworks, a base station, vehicle, a GPS antenna for every collector, with a data link between them. An exceptionally exact area of the base station turns into a pivotal component for the accuracy of vehicle localization.

A PPP framework has a more straightforward setup; a solitary recipient, an antenna for getting GPS and L-Band frequencies. The PPP additionally requires a membership to a correction service provider. Be that as it may, PPP has a more convergence time and somewhat lower precision.

Another differentiator is the standard length. The distance between the base station and vehicle (standard length) on a RTK framework straightforwardly impacts framework exactness. Short standard lengths, a couple of kilometers, RTK is extremely precise. Like referenced before, as the length expands, the accuracy of the arrangement diminishes. At long lengths, RTK will be obsolete. Since PPP doesn't utilize a base station, the separation from the baseline length doesn't influence and can give total exactness anyplace.

The design DGPS and RTK frameworks are comparative. Both the strategies require a base station arrangement at an exactly known area. This vehicle recipient gets revisions from the base station and an data transfer interface between the two receivers. The thing that matters is that RTK utilizes the carrier phase technique, fundamentally more exact than the DGPS, which utilizes a code-based strategy. The benefit of DGPS is that it is profitable over a more drawn out distance between the base station and vehicle recipients, and the framework is more affordable.

Table 3.2 State-of-the-art accuracies of navigation devices.

Method	**Precision**	**Advantages**	**Limitations**
GPS	< 7.8 meters	Low Cost	Satellite signal blockage due to buildings, bridges, trees, etc.
DGPS	< 1 meter	Range 100 to 200 km	The accuracy of the vehicles' position will depend on the precise location of the Base Station
RTK	1 to 2 cm	Range 10 to 20 km	Very expensive and limited range

Table 3.2 (Continued)

PPP	< 1 meter	GPS devices can correct the data by themselves.	Data is not easily available for public use, and 1 Hz speed
INS		Can work in GPS denied areas	Even the best accelerometers, with a standard error of 10 micro-g, would accumulate a 50-meter error within 17 minutes
GPS + INS	1 to 2 cm	Can work in GPS denied areas	Expensive equipment
LiDAR	1 to 2 cm	Provides very good accuracy	Very Expensive and high computational costs
Cameras	1 to 2 cm	Provides very good accuracy	Poor performance in different lighting conditions and high computational costs

Chapter 4: CAN Bus Design Model

4.1 Background

In 1980 Robert Bosch developed the Controller Area Network (CAN) bus to accommodate the needs of many auto manufacturers[117]. Though CAN was developed for automobile networking needs, and its initial development objective was to achieve standardization in communication between the nodes in an embedded system, i.e., nodes are a sensor, actuators, motors, etc., [118]. The CAN protocol developed by Bosh became an ISO standard called ISO 11898. An initial version was released in 1993 called CAN 2.0A, and an extended version was released in 1995 called CAN 2.0B[119]. CAN is a serial multicast, multi-master protocol. Multi-master means any node can send a message when the bus is free, and multicast means all nodes receive the message transmitted on the bus.

Figure 4.1 and Figure 4.2 can be observed as a brief example of vehicular systems without CAN and vehicular systems with CAN. In a vehicle, the most common component is an Electronic Control Unit (ECU) which is essentially an embedded system that comprises of numerous microcontrollers that communicate over the CAN network to control the powertrain, stability control, information panel, etc. [120]. Since the microprocessors cannot communicate directly, a protocol needed to be developed to send and receive data packets. Therefore, an Open System Interconnection (OSI) model, which consists of seven layers, including network layer, data link layer, physical layer, was developed within the CAN protocol to fuse the data link layer and the physical layer.

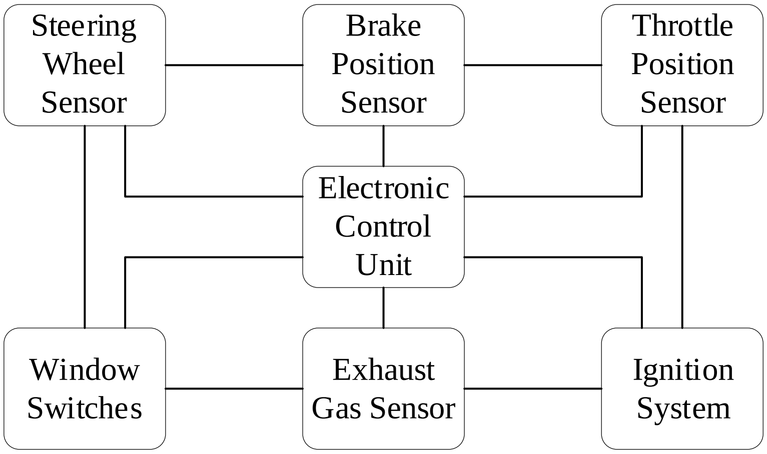


Figure 4.1 Vehicular network without CAN.

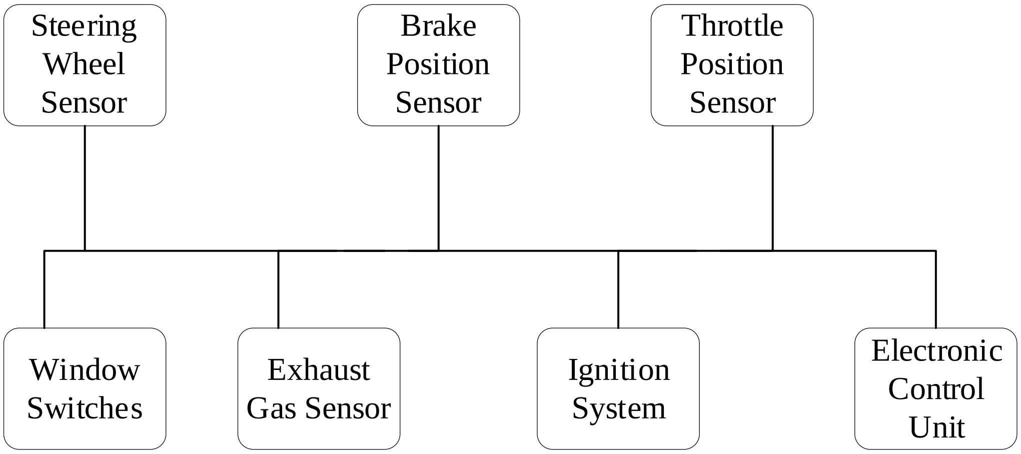


Figure 4.2 Vehicular network with CAN.

4.2 CAN Fundamentals

A brief description of how CAN work is presented, while a much detailed explanation is provided in [121]. A twisted pair of copper cables with a common ground is used as the physical layer. All the nodes connected to the network have the same data rate. The CAN 2.0A has 11-bit identifiers, and CAN 2.0B supports both 11 bit and 29-bit identifiers.

In a CAN data frame, seven fields complete a message. Please refer to Table 4.1 and Table 4.2 for further reference of the bits. All the bits in a CAN network are set to either (0) meaning dominant and (1) meaning recessive. The Start of Frame (SOF) bits in a CAN message are the dominant bits. All the nodes that are waiting to transmit will synchronize with the SOF, and the arbitration will determine which message to transmit first. Data Frame starts with a Start of Frame bit and follows a sequence of identification bits which comprise of eleven bits followed by the remote transmission request bit. After the remote transmission request bit comes the control bits, which are six bits that specify the number of bytes the data field consists of, the data field can transmit anywhere from zero to eight bytes of information. After the data field, there is a cyclic redundancy check field, a two-bit field, which is used to find corrupted data in the data field. After the corrupted data check field, the acknowledgment bits are 2 bits to validate the frame received. In the end, there is a total of seven-bit End of Frame bits.

Table 4.1 Message frame format for CAN 2.0A.

Name	**Bits**
Start of Frame	1
Identifier	11
Remote Transmission Request	1
Reserved	2
Data Length Code	4
Data Field	0 to 8 bytes
Cyclic Redundancy Check	15
Cyclic Redundancy Check Delimiter	1
Acknowledgement	1
Acknowledgement Delimiter	1
End of Frame	7

Table 4.2 Message frame format for CAN 2.0B.

Name	**Bits**
Start of Frame	1
Identifier Standard	11
Identifier Extended Format	18
Remote Transmission Request	1
Substitute Remote Request Extended Format	1
IDE	1
Reserved r0	1
Reserved r1 Extended Format	1
Data Length Code	4
Data Field	0 to 8 bytes
Cyclic Redundancy Check	15
Cyclic Redundancy Check Delimiter	1
Acknowledgement	1
Acknowledgement Delimiter	1
End of Frame	7

The CAN network is formed by combining the last two layers of the Open System Interconnection (OSI), i.e., the Physical layer and the Data Link layer. The CAN protocol is a standardized protocol ISO 1189-1. The Can protocol is reliable, and it is used widely in embedded systems as the communication protocol because of its low-cost implementation.

The CAN is not a master-slave protocol; it is a peer-to-peer protocol. This means that the nodes in the network have the capabilities to read and write data on the network. When a node is ready to transmit data, it checks if the bus is busy and transmits its data frame on the network. The data that is transmitted does not contain information about the nodes from which it is being sent, nor does it contain data about the receiving node. All it contains is an arbitrary ID that is specific to each node within the entire network. Since it is a peer-to-peer network, all the nodes receive all the data that is being transmitted in the network., the nodes identify the information from other nodes by checking the arbitration identification and decide if the information from an specific node is useful for its own purpose.

4.3 System Block Diagram

Figure 4.3 shows an overview block diagram of the GPS and INS fused system. The INS uses the gyroscopes and accelerometers present in the IMU, the GPS positioning data, and uses data fusion algorithms such as Kalman filters to improve localization accuracy. As the limitations of the INS were discussed earlier in Chapter 2, the drift factor account for cumulative errors, in addition, the cost of the INS systems is higher as better accuracy is required. This research proposes a novel approach to enhance localization accuracy by replacing the IMU with vehicle sensors that can provide information on the movement of the vehicle. Figure 4.4 shows a brief block diagram approach that can be used as an approach for the proposed system is to replace the IMU and INS blocks with the sensors that are present on the vehicle to accomplish the increase in accuracy of the navigation. Further, this approach has to be applied to CV to have access to the applications in CV.

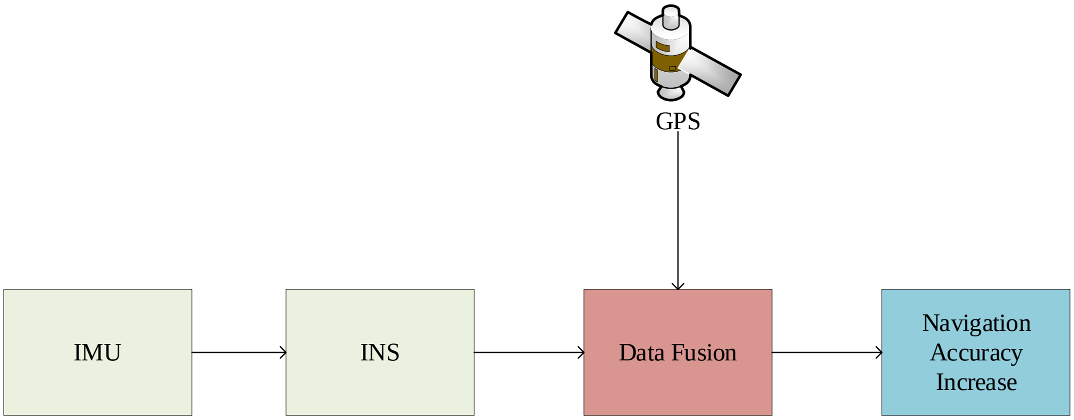


Figure 4.3 GPS + INS block diagram.

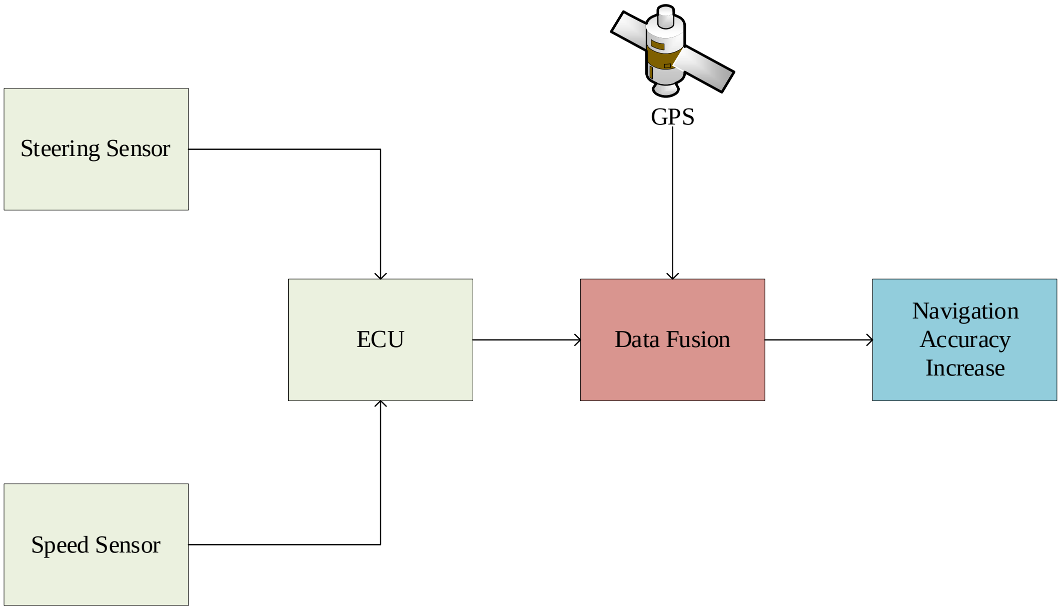


Figure 4.4 Modified system to accommodate vehicle sensors.

4.4 Connected Vehicle Design

Figure 4.5 shows the integral parts that transform a vehicle into a connected vehicle. An On-Board Unit (OBU) is added to the car, and this is the computer onboard that receives the information from the GPS via a GPS antenna placed on the car. Two Dedicated Short-Range Communication (DSRC) antennas are placed on the car. These antennas are used to communicate with the CV-enabled vehicles nearby within 200 meters radius and communicate with Road Side Units (RSU) for communication with the infrastructure. OBU gathers data from RSUs, and the locations of the CV-enabled vehicles and uses the information to apply the appropriate application that is downloaded on the system.

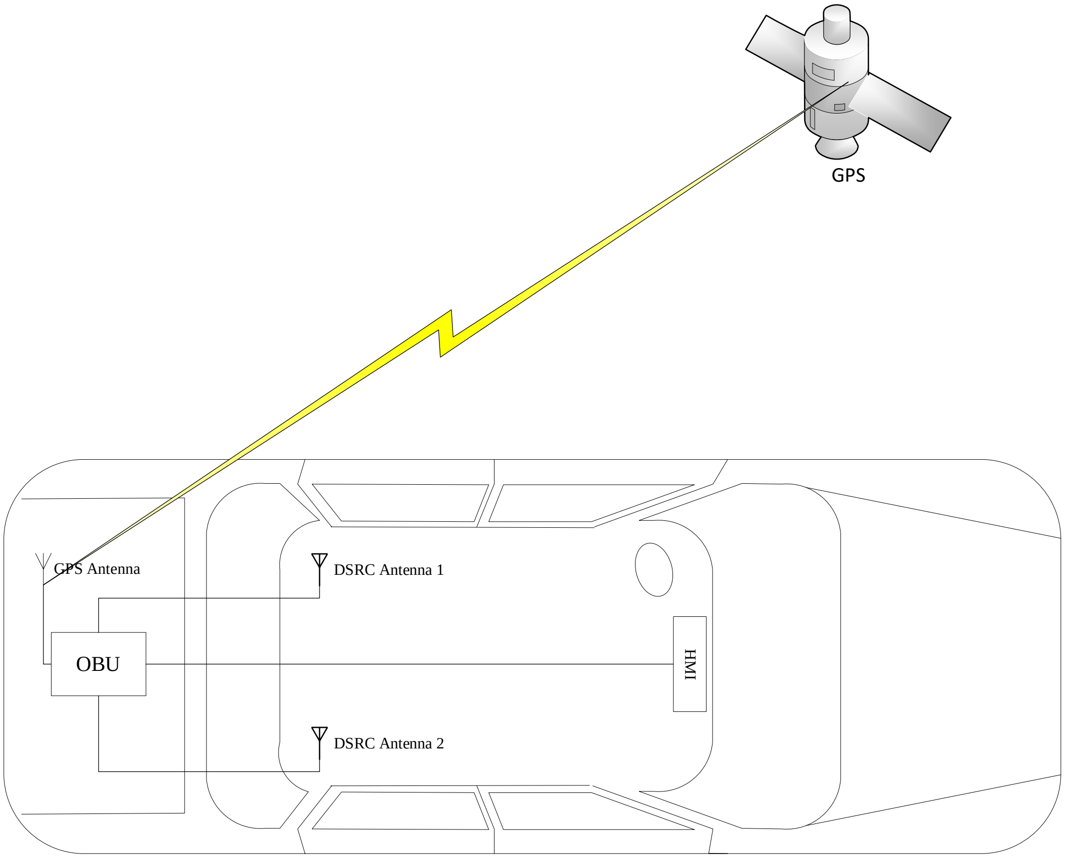


Figure 4.5 Illustration of a connected vehicle.

The alerts produced from the system are then sent to the driver of the vehicle via a Human Machine Interface (HMI) in the form of visual aid using an LCD screen integrated into the rearview mirror and a sound alert with a speaker. The whole CV system is standalone in the vehicle, only drawing power from the vehicle. It can also be observed that there is no input from the car.

4.5 On-Board Diagnostics (OBD II)

OBD II port is an SAE standard present in all vehicles that have been produced in the US after 1996 [122]. It is a port that is usually on the driver's side of the vehicle. The OBD II standard defines messages, electrical interface, and Diagnostic Codes from SAE Standard J1972. The port was created to standardize any manufacturer to diagnose the vehicle.

The OBD II is a 16 pin port, of which the standard only mandated nine pins. The pin configuration for the OBD II is mentioned in Table 4.3.

Table 4.3 OBD II pin description.

OBD II Pin	**Description**
1	Manufacturer Option
2	J1850 Bus +
3	Manufacturer Option
4	Chassis Ground
5	Signal Ground
6	CAN High (J 2234)
7	ISO 9141-2 K Line
8	Manufacturer Option
9	Manufacturer Option
10	J850 Bus -
11	Manufacturer Option
12	Manufacturer Option
13	Manufacturer Option
14	CAN Low (J 2234)
15	ISO 9141-2 Low
16	Power

As observed in the Pin layout of the OBD II port, The CAN bus high and CAN bus low are available in pins 6 and 14, respectively. Many embedded system approaches can access the CAN data by connecting the OBD II port to a microcontroller. This data is readily available to access to perform diagnostics.

4.6 Proposed System Design

Figure 4.6 shows the simplistic hardware capability for the CV technology to be added to the vehicles without the CV technology. In practice, the transformation of a non-CV vehicle to a CV as performed in the THEA CV pilot project was less than 2 hours for most of the cars. With this simplistic design, the solution to be added should also be very easy to implement. Figure 4.6 shows a system block diagram where the two networks are differentiated by a gray block representing the In-Vehicle sensor network connected by the CAN. And the maroon block represents the CV technology.

The system diagram shows the possibility of the in-vehicle sensors to be available to the CV technologies to increase the precision of location information to better the application performance.

Figure 4.7 shows a simple integration of adding a communication channel from the OBD II port to the OBU will enable the CV technology to have access to the in-vehicle sensors. Since there are already embedded system approaches to access the CAN, it can be only a software solution over the communication channel that will be needed on the OBU to use the in-vehicle sensors.

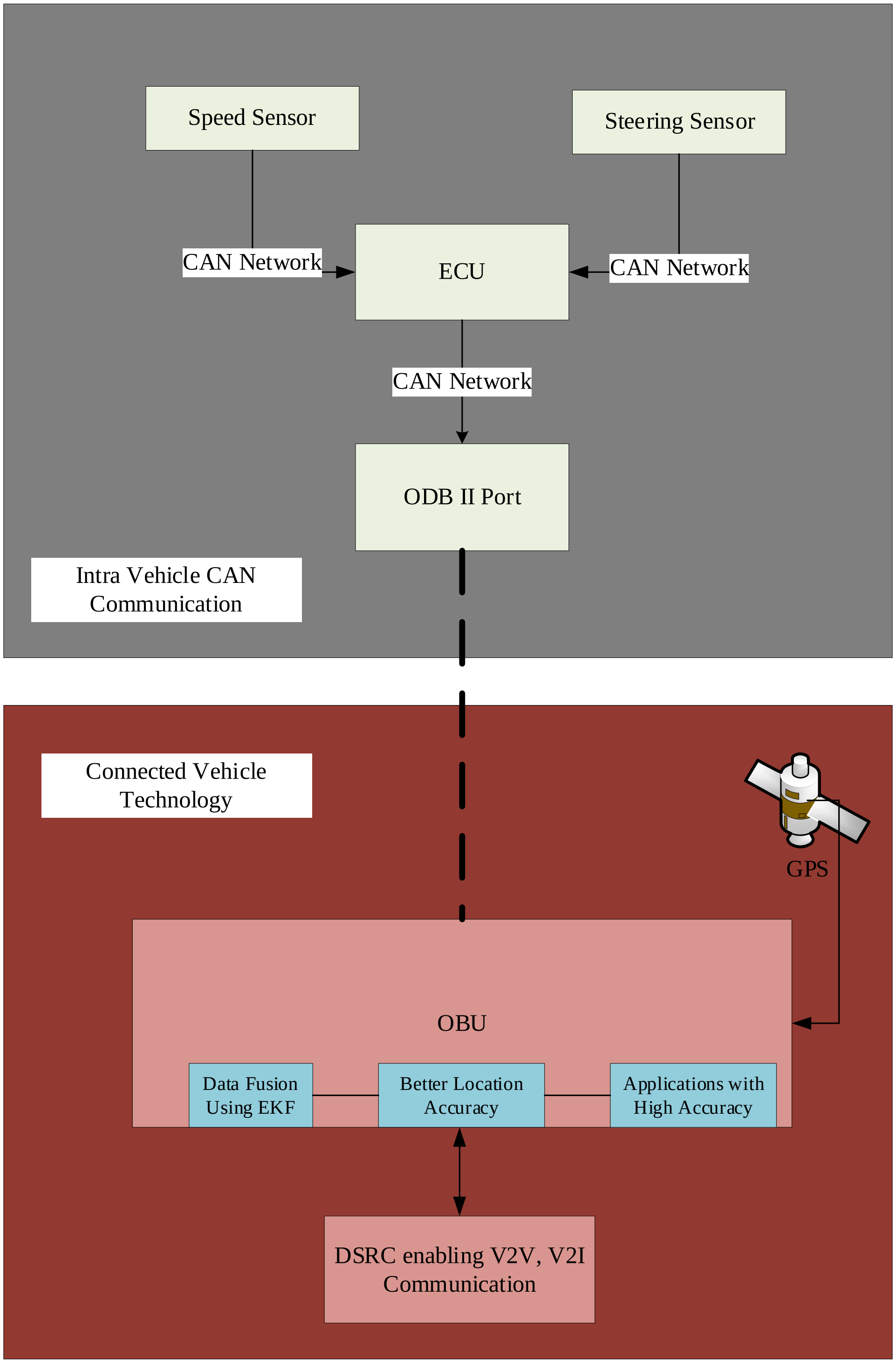

Speed Sensor
Steering Sensor
CAN Network
ECU
CAN Network
CAN Network
ODB II Port
Intra Vehicle CAN Communication
Connected Vehicle Technology
GPS
OBU
Data Fusion Using EKF
Better Location Accuracy
Applications with High Accuracy
DSRC enabling V2V, V2I Communication

Figure 4.6 System block diagram.

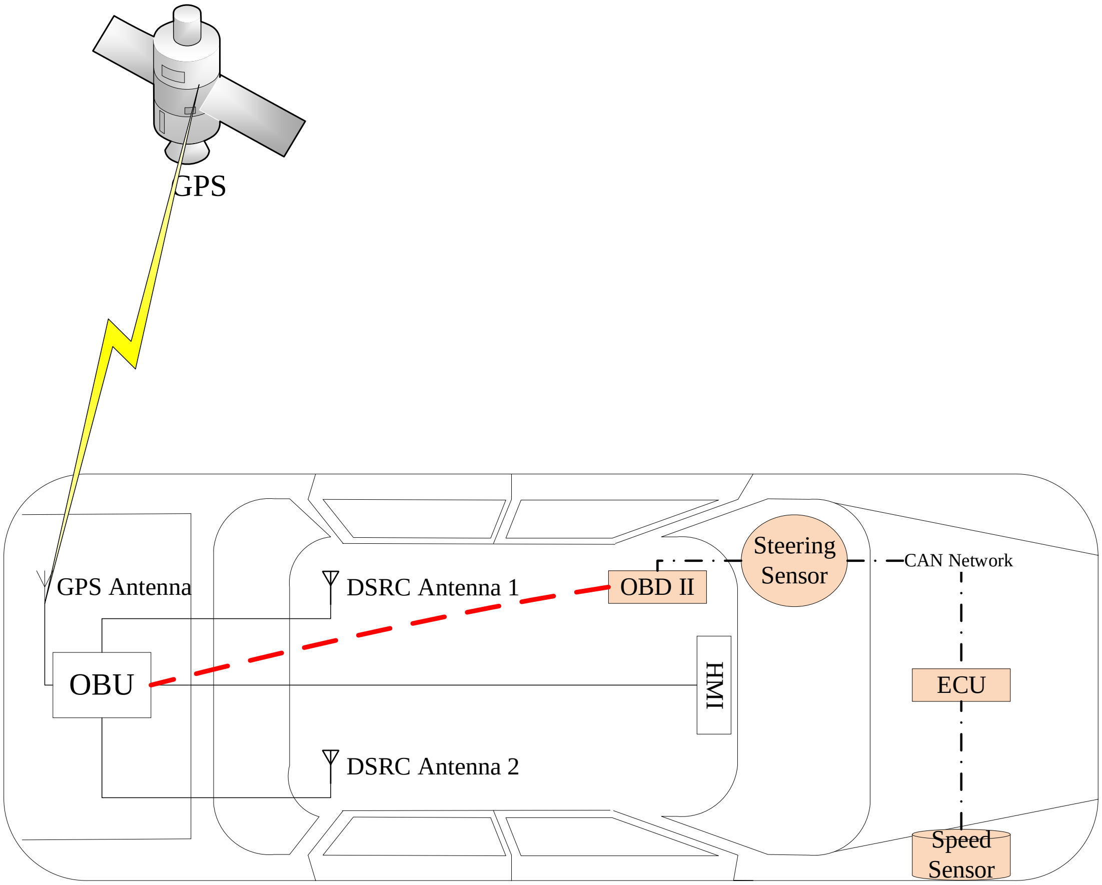


Figure 4.7 CV system integration with CAN.

4.7 Systems Engineering for Transportation Design Engineering

According to INCOSE, systems engineering is a process involving multidisciplinary teams to create a successful system [123]. Figure 4.8 shows the systems engineering V diagram for transportation design. The diagram resonates with a seven phases methodology employed by the systems engineering process to develop a successful design. The first phase is to understand the problem. Then the second step is to investigate the state-of-the-art or alternative solutions, the third step is to define and agree on system architecture, the fourth step is to manage requirements, the fifth step is to manage interfaces the sixth step is to prepare a test system. The seventh step is to trace the progress [123].

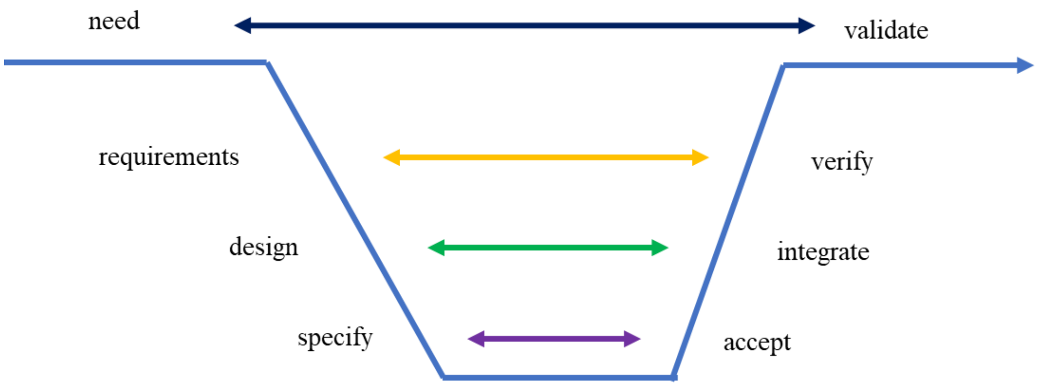
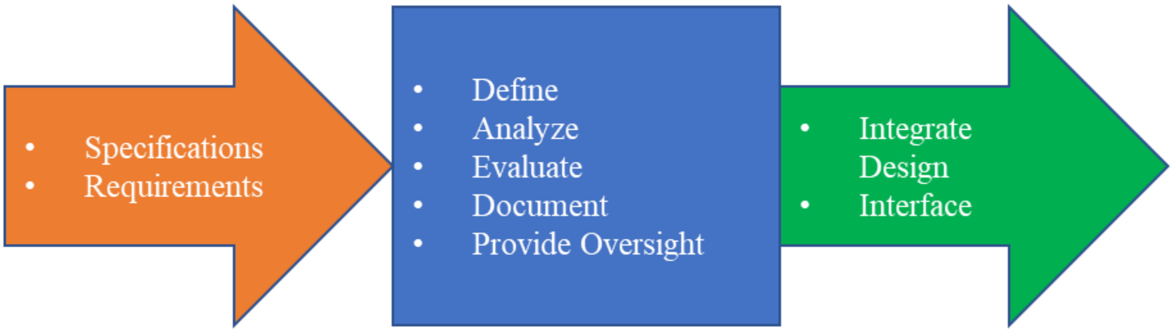


Figure 4.8 Systems engineering 'V' diagram

The V-diagram leads to the much more robust high-level concept for design engineering as a process, as illustrated in Figure 4.9. The defined part of the system consists of the requirements and specifications for the system to be designed. Then an analysis is performed to determine the achievements of the requirements or specifications. Evaluate the design alternatives by establishing a criterion and selecting the best option. Next, define a strategy to integrate the system elements and resolve integration issues. Then document the changes and provide oversight for quality and design management.


- Specifications
- Requirements

- Define
- Analyze
- Evaluate
- Document
- Provide Oversight

- Integrate Design
- Interface

Figure 4.9 Design engineering block diagram

Chapter 5: Vehicle Kinematic Modelling for Localization

5.1 Vehicle Kinematic Model

The model of the vehicle is based on the Ackerman steering geometry. While purely a kinematic model, it is a good representation of slow speeds. Figure 5.1 shows the vehicle scheme and the main variables. The pose is defined by its position (x, y) and the bearing φ. The actuations are the car speed υ and the steering angle ψ. The only parameter is the wheelbase L. The kinematic bicycle model is called the front wheel steering model. The front-wheel orientation is observed relative to the heading of the vehicle.

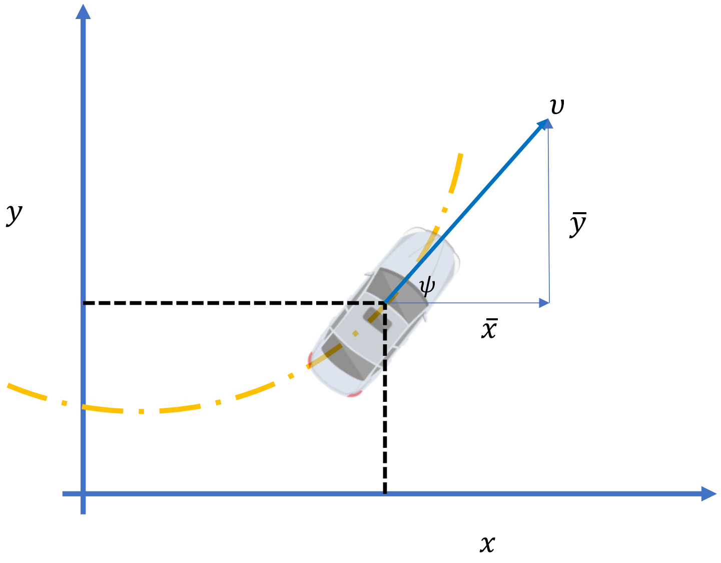


Figure 5.1 Kinematic model of the path traveled. Data taken from [124]

The target is to compute state vector, $[\bar{x}, \bar{y}, \bar{\theta}]$. The inputs to the system are (x, y), υ is the velocity, ψ is the heading angle or the baring.

To calculate the kinematics of the bicycle model, a starting reference point (x, y) must be selected. Consider the rear axle as the reference of the vehicle (x, y).

The first approximation is the automobile steers by turning the front tires while moving forward. The front of the car changes the direction that the wheels are pointing while turning around the rear wheels.

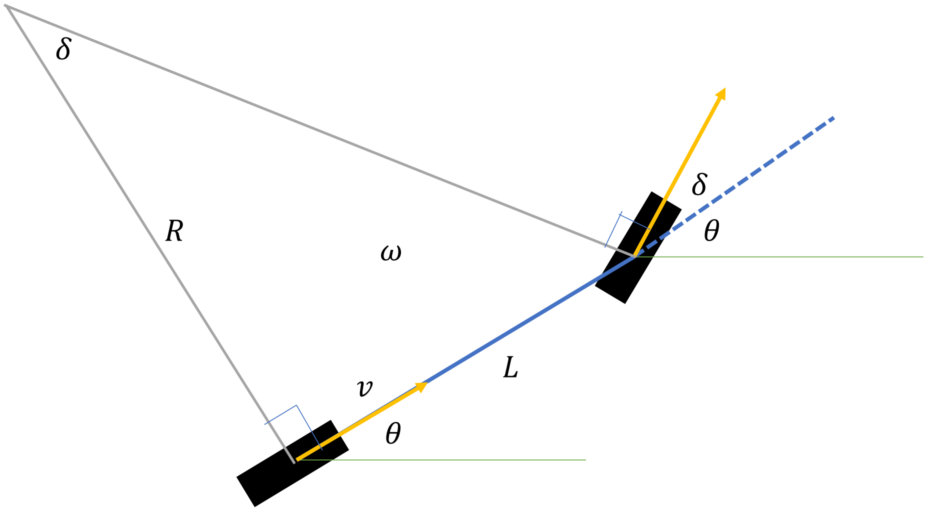


Figure 5.2 Illustration of ICR. Data taken from [124]

From Figure 5.2, it can be observed that the front tire is pointing at an angle δ compared to the wheelbase. Over a short time period, the car moves forward, and the rear wheel ends up further ahead and slightly turned inward. For this short time, approximation that this is a turn around a radius R can be made. By applying Instantaneous Center of Rotation (ICR), state change rate can be computed as,

$$\bar{x} = v * \cos{(\theta)}$$

$$\bar{y} = v * \sin{(\theta)}$$

is equal to rotation rate ω,

$$\omega \ = \ v \ / \ R$$

where R is the radius, considering the length of the vehicle as L, δ is steering angle so that R can be computed as,

$$R = L / tan(\delta)$$

now $\boldsymbol{\bar{\theta}}$ can be computed as,

$$\bar{\theta} = \frac{v}{\left(\frac{L}{\tan(\delta)}\right)}$$

$$= v * tan(\delta) / L$$

5.2 Constant Turn Rate Constant Velocity

The constant turn rate constant velocity model (CTCV) is a motion prediction model in vehicles. The model assumes a constant turn rate and constant velocity to predict the next step. Consider the function,

$$x_{k|k} = f(x_k, v_k)$$

the state for the model is given by,

$$x = [p_x \quad p_y \quad v \quad \psi \quad \dot{\psi}]^T$$

To find the process model, the change of the state vector, x, should be derived. From Figure 5.3 to find the change in the rate of x denoted as, $\dot{x}$. And $\dot{x}$ is the differential equation.

$$\dot{x} = \begin{bmatrix} \dot{p_x} \\ \dot{p_y} \\ \dot{v} \\ \dot{\psi} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} v \cdot \cos\ \psi \\ v \cdot sin\ \psi \\ 0 \\ \dot{\psi} \\ 0 \end{bmatrix}$$

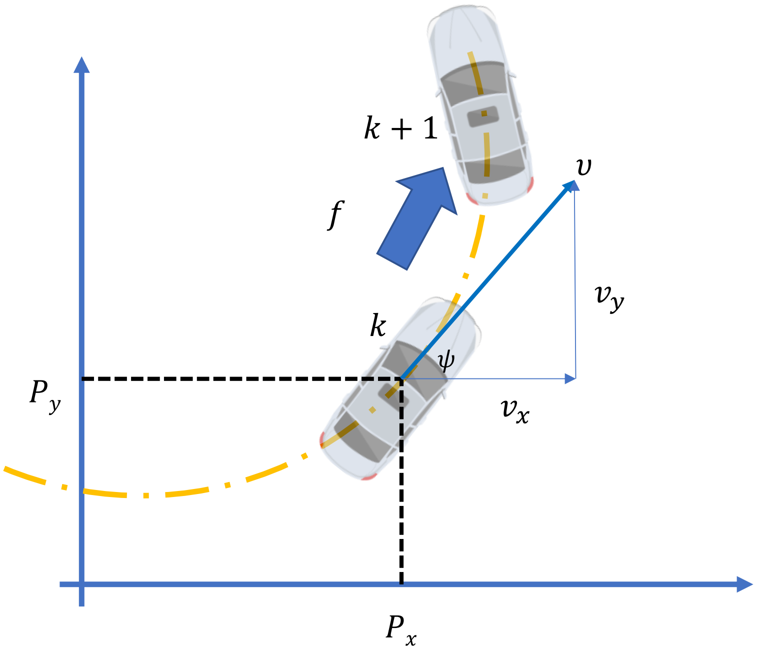


Figure 5.3 CTCV model diagram. Data taken from [125]

Since the change in velocity and turn rate will become zero, by assuming the velocity and the turn rate to be constant, and the derivative of a constant is zero. Then we use the difference in time elapsed

$$\Delta t = t_{k+1} - t_k$$

and integrate $\dot{x}$.over this time period.

$$x_{k+1} = x_k + \int_{t_k}^{t_{k+1}} \begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{v} \\ \dot{\psi} \\ \ddot{\psi} \end{bmatrix} \mathrm{d}t$$

$$\int_{t_k}^{t_{k+1}} \begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{v} \\ \dot{\psi} \\ \ddot{\psi} \end{bmatrix} \mathrm{d}t = \begin{bmatrix} \int_{t_k}^{t_{k+1}} \dot{p}_x \mathrm{d}t \\ \int_{t_k}^{t_{k+1}} \dot{p}_y \mathrm{d}t \\ \int_{t_k}^{t_{k+1}} \dot{v}\, \mathrm{d}t \\ \int_{t_k}^{t_{k+1}} \dot{\psi}\, \mathrm{d}t \\ \int_{t_k}^{t_{k+1}} \ddot{\psi}\, \mathrm{d}t \end{bmatrix}$$

$$\int_{t_k}^{t_{k+1}} \begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{v} \\ \dot{\psi} \\ \ddot{\psi} \end{bmatrix} \mathrm{d}t \quad = \begin{bmatrix} \int_{t_k}^{t_{k+1}} v \cdot \cos\, \psi \mathrm{d}t \\ \int_{t_k}^{t_{k+1}} v \cdot \sin\, \psi \mathrm{d}t \\ \int_{t_k}^{t_{k+1}} \dot{v}\, \mathrm{d}t \\ \int_{t_k}^{t_{k+1}} \dot{\psi}\, \mathrm{d}t \\ \int_{t_k}^{t_{k+1}} \ddot{\psi}\, \mathrm{d}t \end{bmatrix}$$

$$\int_{t_k}^{t_{k+1}} \begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{v} \\ \dot{\psi} \\ \ddot{\psi} \end{bmatrix} \mathrm{d}t = \begin{bmatrix} \int_{t_k}^{t_{k+1}} v \cdot \cos\, \psi \mathrm{d}t \\ \int_{t_k}^{t_{k+1}} v \cdot \sin\, \psi \mathrm{d}t \\ 0 \\ \int_{t_k}^{t_{k+1}} \dot{\psi}\, \mathrm{d}t \\ 0 \end{bmatrix}$$

5.3 Audi Autonomous Driving Dataset (A2D2)

Audi published the Audi Autonomous Driving Dataset (A2D2) to support researchers working on autonomous driving to find innovative solutions[126]. In the research, an Audi Q7 was equipped with a sensor suite. Since recording a dataset and labeling, it is time-intensive and laborintensive. The Audi A2D2 dataset removes the high entry barrier and frees researchers and developers have to incur. Instead, invest the time and money to focus on developing new technologies. The dataset features, amongst many other features, vehicle bus sensors, GPS, 3D bounding boxes, etc.

5.3.1 Sensor Setup

The vehicle is equipped with six cameras, an embedded system to record bus data, and five LiDAR sensors. The bus data collected by the embedded system gives information about the vehicle's current state and driver control input. A GNSS clock is synchronized to the time master.

The bus signals have the following data, timestamped in sequence. Acceleration sensor data in x, y, and z directions, the position of accelerator, the gradient sign of the accelerator, angular velocity in x, y, and z. The brake pressure sensor data, the distance pulse from the front, right, left, and back. GPS Latitude in degrees, GPS longitude in degrees, the pitch angle, the roll angle, the steering angle, the steering sign, and the vehicle speed. Each sample is associated with a timestamp, and each signal has its unit, which is specified in the JSON file.

5.3.2 Dataset Information

The dataset contains sequences recorded from three locations in Germany, Munich (Dataset 1), Gaimersheim (Dataset 2), and Ingolstadt (Dataset 3). For the rest of the document, the recorded sequences will be referred to as Dataset 1 for Munich data sequence, Dataset for Gaimersheim data sequence, and Dataset 3 for Ingolstadt data sequence.

Table 5.1, Table 5.2, and Table 5.3 detail the information about the sample size of each dataset. From the sample sizes, the frequency of the sensors can be determined. The frequency of the GPS signals is 5 Hz, the frequency of the speed sensor in the bus is 50 Hz, and the frequency of the yaw rate sensor is 200 Hz.

Table 5.1 Dataset 1 (Munich data sequence) information.

Data Type	**Sample Size**
Time	15:19.53
Longitude	4599
Latitude	4599
Speed	45984
Yaw Rate	183937

Table 5.2 Dataset 2 (Gaimersheim data sequence) information.

Data Type	**Sample Size**
Time	08:45.60
Longitude	2629
Latitude	2629
Speed	26288
Yaw Rate	105153

Table 5.3 Dataset 3 (Ingolstadt data sequence) information.

Data Type	**Sample Size**
Time	12:24.97
Longitude	3726
Latitude	3726
Speed	37260
Yaw Rate	149041

5.4 Tampa CV Pilot Data

The main reason to deploy the CV Pilot program is to provide open-source data from the three Pilots programs to researchers and developers. THEA CV became the second pilot project to complete its data pipeline and sent continuous data to ITS DataHub [127]. It is an online platform that provides datasets publicly available USDOT ITS research data[48].

The message types THEA CV Pilot has provided follow the SAE J2735 data standards, which include the Basic Safety Messages (BSM), the Signal Phasing and Timing Messages (SPaT), and Traveler Information Messages (TIM)[48]

5.4.1 Basic Safety Messages (BSM)

BSM data set includes data generated by OBUs transmitted to RSUs located throughout the pilot area. This data is used to exchange data regarding vehicles' current attributes, such as speed and path predicted. BSM data from THEA CV Pilot can be found in the ITS Sandbox.[48]

5.4.2 Traveler Information Messages (TIM)

TIM is used to transfer important traffic information and provide situational awareness alerts to the driver; the THEA CV TIM data set includes messages sent from RSUs located throughout the Tampa CV pilot study area to OBUs of equipped vehicles.[48]

5.4.3 Signal Phasing and Timing (SPaT) Messages

SPaT are used to convey the current status signalized intersections. The THEA CV SPaT data set includes messages transmitted by RSUs located throughout the Tampa pilot area. Note that Tampa is the first of the CV Pilots to upload SPaT data.[48]

Chapter 6: Adaptive Filtering to Couple GPS with Vehicle Sensors

Analyzing the datasets from A2D2, density distribution graphs for longitude and latitude values of dataset 1 are shown in Figure 6.1 and Figure 6.2, respectively. In the same way, density distribution graphs for longitude and latitude values of dataset 2 are shown in Figure 6.3, and Figure 6.4, and density distribution graphs for longitude and latitude values of dataset 3 are shown in Figure 6.5 and Figure 6.6. The common observation that can be made from all the individual density distribution graphs is that none of the distributions are linear distributions. There is a blue line in front of the histogram, representing the distribution's probability density function. If the line has more than one significant peak, then the system is not a normal distribution. All the six density distribution graphs show that the blue line has more than one peak. This means that none of the distributions are normal distributions.

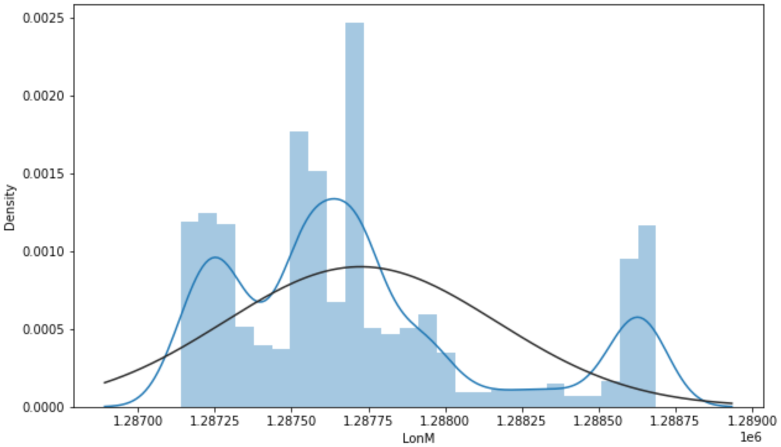


Figure 6.1 Density distribution of longitude from dataset 1.

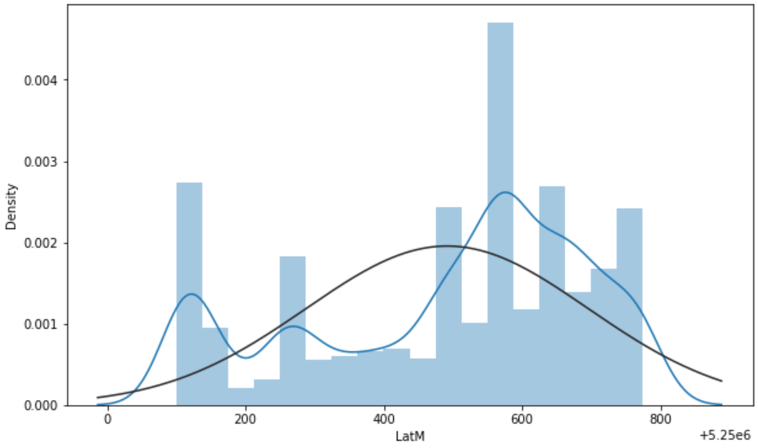


Figure 6.2 Density distribution of latitude from dataset 1.

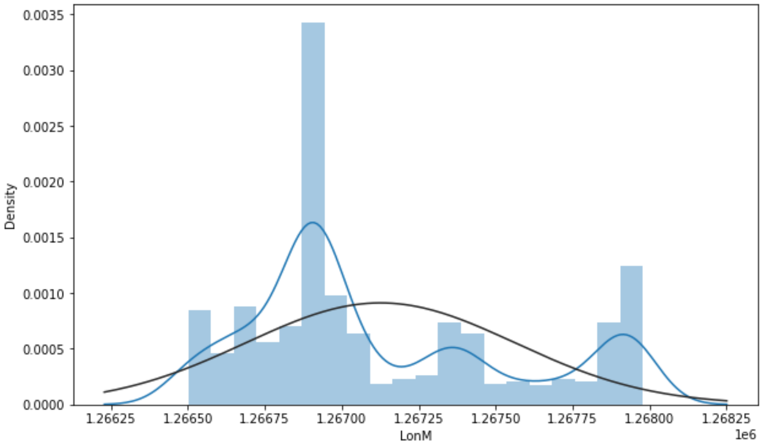


Figure 6.3 Density distribution of longitude from dataset 2.

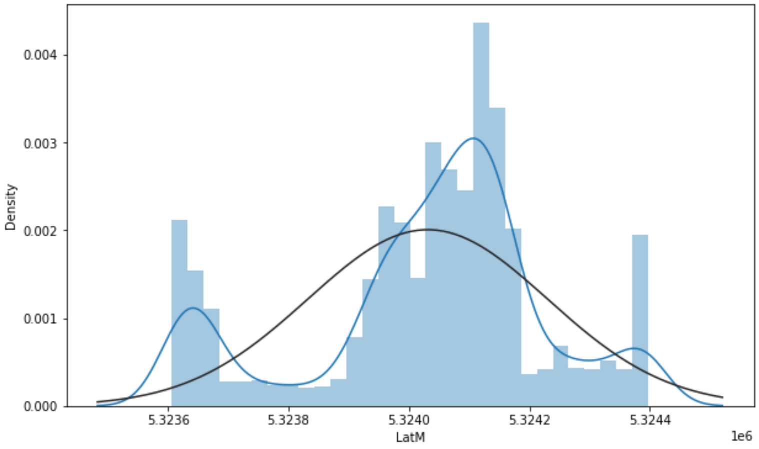


Figure 6.4 Density distribution of latitude from dataset 2.

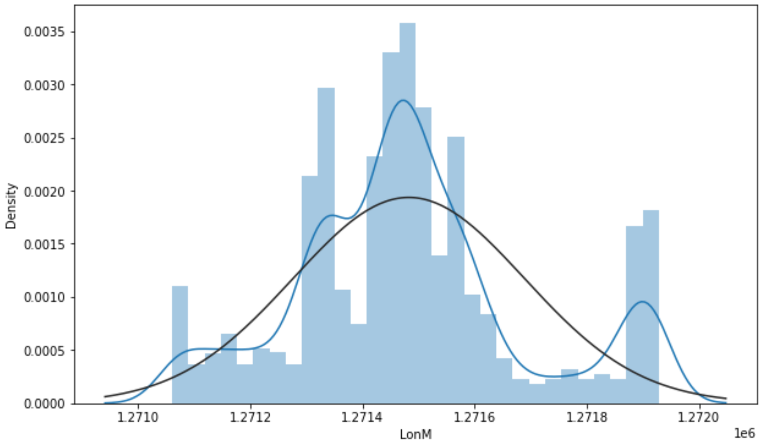


Figure 6.5 Density distribution of longitude from dataset 3.

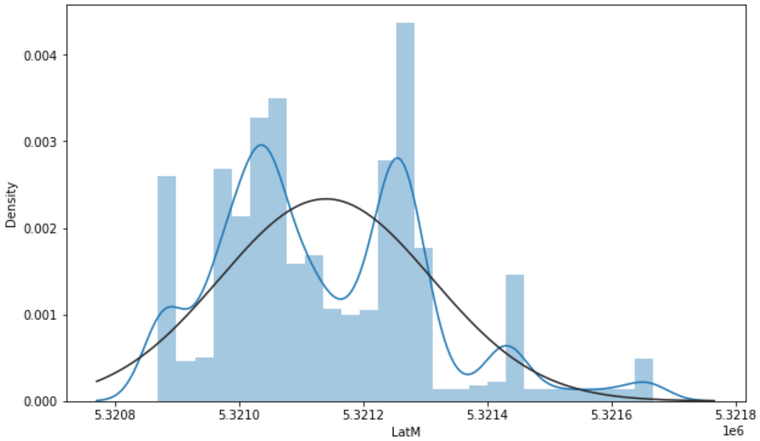


Figure 6.6 Density distribution of latitude from dataset 3.

Kalman filter has a more mathematical name called the Linear Quadratic Estimator. As the name suggests, the prediction and update steps can only be made when the state and measurement estimates are normal distribution functions. Since the position inputs for the Kalman Filter are not a normal distribution, an Extended Kalman filter must be used to normalize the inputs.

The equations for the EKF will be modified to accommodate the linearization of the state transition matrix and the observation matrix.

- Prediction Equations
    - Predicting the next state

$$x_{k+1} = g(x_k, u)$$

    - Predicting the covariance error

$$P_{k+1} = J_A P_k J_A^T + Q$$

- Update Equations
    - Calculating Kalman gain

$$K_k = P_k J_H^T (J_H P_k J_H^T + R)^{-1}$$

    - Updating the estimation with the inputs from the sensors

$$x_k = x_k + K_k\big(z_k - h(x_k)\big)$$

    - Updating the covariance error

$$P_k = (I - K_k J_H) P_k$$

Earlier in Chapter 5 it was calculated that the frequency of the GPS values provided by the datasets is 1 Hz. The first EKF approach increases the prediction speed from 1 Hz (GPS speed) to 5 Hz. Since the position inputs from the GPS are 1 Hz, a kinematic bicycle model is used to predict the next states when the system has no position inputs.

Later, another EKF is used to predict trajectories of the datasets while having the GPS as the position input. Also inputs from the speed sensor and the yaw rate sensor, which have been collected from the CAN network, are used to predict the trajectories at 5 Hz.

6.1 Filter Model 1 (Algorithm 1) Bicycle Model

The algorithm firstly defines the states (x, y), where (x, y) represent the positional vectors, the latitude and longitude from the GPS will be converted to East and North respectively. For the purposes of mathematical simplification, East will be called x, and North will be called y, where x and y are converted from decimal degrees to meters.

ψ is the heading angle of the vehicle concerning the North. And v is the speed of the vehicle. The matrix representation is presented to explain the implementation of the EKF easily.

$$States\ (x_k) = \begin{bmatrix} x \\ y \\ v \\ \psi \end{bmatrix}$$

The state transition function, this matrix is the predicted state from the previous state.

$$x_{k+1} = \begin{bmatrix} \mathrm{Tvcos}(\psi) + x \\ \mathrm{Tvsin}(\psi) + y \\ v \\ \psi \end{bmatrix}$$

For the prediction step of the Kalman filter, the Jacobian of the equation is required.

$$J_A = \begin{bmatrix} 1 & 0 & \mathrm{Tcos}(\psi) & -\mathrm{Tvsin}(\psi) \\ 0 & 1 & \mathrm{Tsin}(\psi) & \mathrm{Tvcos}(\psi) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The adaptive nature of the Kalman filter is implemented to observe the movement in the distance to calculate the next step. i.e., the GPS values of the previous step (x_{k-1}, y_{k-1}) and the GPS values from the present step (x_k, y_k) are used, and distance is calculated only when D > 0 the next step values are used.

$$Dist\ D = \sqrt{(x_k - x_{k-1})^2 + (y_k - y_{k-1})^2}$$

The time measurement T in this adaptive filter is assumed from the difference in time T_{k-n} between the first previous (x_{k-n}, y_{k-n}) values and the time T_k of current (x_k, y_k) value when the distance between them is not zero.

$$T = T_k - T_{k-n}$$

The measurement matrix for the system is very simple as the position vectors are used as inputs.

$$h(x_k) = \begin{bmatrix} x \\ y \end{bmatrix}$$

The Jacobian of this matrix will be,

$$J_H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

The error covariance of the system is something that can be tuned to change the attributes of the filter. It is the maximum variance which the filter allows the predictions to be from the previous state.

The covariance for the position states is calculated at 4.4 meters displacement. Likewise, the speed was set to be 9 meters per second. The heading of the vehicle was set to be 0.1 radian.

The process noise covariance of the systems is set to 1000, showing the uncertainty of the system on the initial accuracy of the GPS. If the process noise covariance were to be set to zero, the filter would have complete confidence in the initial input measurements.

6.2 Filter Model 2 (Algorithm 2) CTCV

This is a data fusion algorithm where the inputs of the yaw rate and vehicle speed with the (x, y) positional vectors are used. The state transition equations were derived from adjusting the equations presented in [128]

$$x_{k+1} = \begin{bmatrix} x + \dfrac{v\left(-\sin(\psi) + \sin\left(T\dot{\psi} + \psi\right)\right)}{\dot{\psi}} \\ y + \dfrac{v\left(\cos(\psi) - \cos\left(T\dot{\psi} + \psi\right)\right)}{\dot{\psi}} \\ T\dot{\psi} + \psi \\ \dot{\psi} \\ v \end{bmatrix}$$

The states of the system are represented by which include the

$$x_k = \begin{bmatrix} x \\ y \\ \psi \\ \dot{\psi} \\ v \end{bmatrix}$$

The Jacobian the of the state transition matrix

$$J_A = \begin{bmatrix} 1 & 0 & \dfrac{v\left(-\cos(\psi) + \cos\left(T\dot{\psi} + \psi\right)\right)}{\dot{\psi}} & \dfrac{\mathrm{Tvcos}\left(T\dot{\psi} + \psi\right)}{\dot{\psi}} - \dfrac{v\left(-\sin(\psi) + \sin\left(T\dot{\psi} + \psi\right)\right)}{\dot{\psi}^2} & \dfrac{-\sin(\psi) + \sin\left(T\dot{\psi} + \psi\right)}{\dot{\psi}} \\ 0 & 1 & \dfrac{v\left(-\sin(\psi) + \sin\left(T\dot{\psi} + \psi\right)\right)}{\dot{\psi}} & \dfrac{\mathrm{Tvsin}\left(T\dot{\psi} + \psi\right)}{\dot{\psi}} - \dfrac{v\left(\cos(\psi) - \cos\left(T\dot{\psi} + \psi\right)\right)}{\dot{\psi}^2} & \dfrac{\cos(\psi) - \cos\left(T\dot{\psi} + \psi\right)}{\dot{\psi}} \\ 0 & 0 & 1 & T & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The input observation matrix of this algorithm takes input from GPS as the position vectors (x, y), the yaw rate sensor from the bus input $\dot{\psi}$ and the vehicle speed v.

$$h(x_k) = \begin{bmatrix} x \\ y \\ v \\ \dot{\psi} \end{bmatrix}$$

The adaptive nature of the filter is implemented to only consider the GPS inputs from the sensors when distance D is greater than zero.

The covariance matrix for the position states is calculated at 4.4 meters displacement. The heading of the vehicle was set to be 0.1 radian and the yaw rate at 1 radian per second. Likewise, the speed was set to be 9 meters per second.

The process noise covariance of the systems is set to 1000, showing the uncertainty of the system on the accuracy of the GPS. If the process noise covariance was to be set to zero, then the filter would have complete confidence on the initial input measurements.

6.3 Results

6.3.1 Analysis

The trajectory predictions for dataset 1, dataset 2, and dataset 3 have been shown in Figure 6.7, Figure 6.8, and Figure 6.9, respectively. Since the number of samples involved in all three datasets is in the thousands, analysis for each dataset has been presented in the form of short segments.

Dataset 1 has been divided into five straight segments, eight curves or turns, and one circle. Likewise, dataset 2 has been divided into three straight segments, twelve curves, and three circles around roundabouts. And dataset 3 has been divided into nine curves and six straight segments. In Figure 6.7, Figure 6.8, and Figure 6.9, the GPS locations are marked by a blue '+' symbol, the predictions made by the EKF without using sensors are marked by yellow '.' symbols and the predictions made by the EKF using the CAN based vehicle sensors are marked by green '-' symbol. A green dot is used to represent the starting point of the predictions, and a red dot is used to mark the end of the predictions. The same notations will be used in every trajectory comparison pictures in the rest of the document. For convenience purposes, the EKF without using in-vehicle sensors will be referred to as EKFNS (EKF no sensors), and the EKF using the CAN based in-vehicle sensors will be referred to as EKFCAN.

Observing Figure 6.7, Figure 6.8, and Figure 6.9, both the predictions by EKFNS and EKFCAN resemble each other, and the GPS values are embedded in the trajectories. This proves that both the predictions from EKFNS and EKFCAN have not failed in predicting some other path not resembling the GPS. The second observation that can be made is the starting predictions in all the three datasets by the EKFNS and EKFCAN vary a little. A detailed explanation of the reason for the variation will be provided later in the chapter.

6.3.1.1 Heading Angle

Predictions made on the vehicle's heading angle are represented in Figure 6.10, Figure 6.11, and Figure 6.12. The blue line represents the predictions made by EKFNS, and the broken orange line represents predictions made by EKFCAN. The first few steps or predictions made by both EKFNC and EKFCAN fluctuate a lot. This is because the algorithms are initiated to start predictions assuming the vehicle is facing north. This creates the heading error at the start of the predictions from the algorithm. Therefore, the algorithm needs some time to converge to the true heading angle. Predictions made by EKFNS can be observed to fluctuate more than the predictions made by EKFCAN. This is because the yaw rate sensor from the car is used to correct the heading angle in EKFCAN.

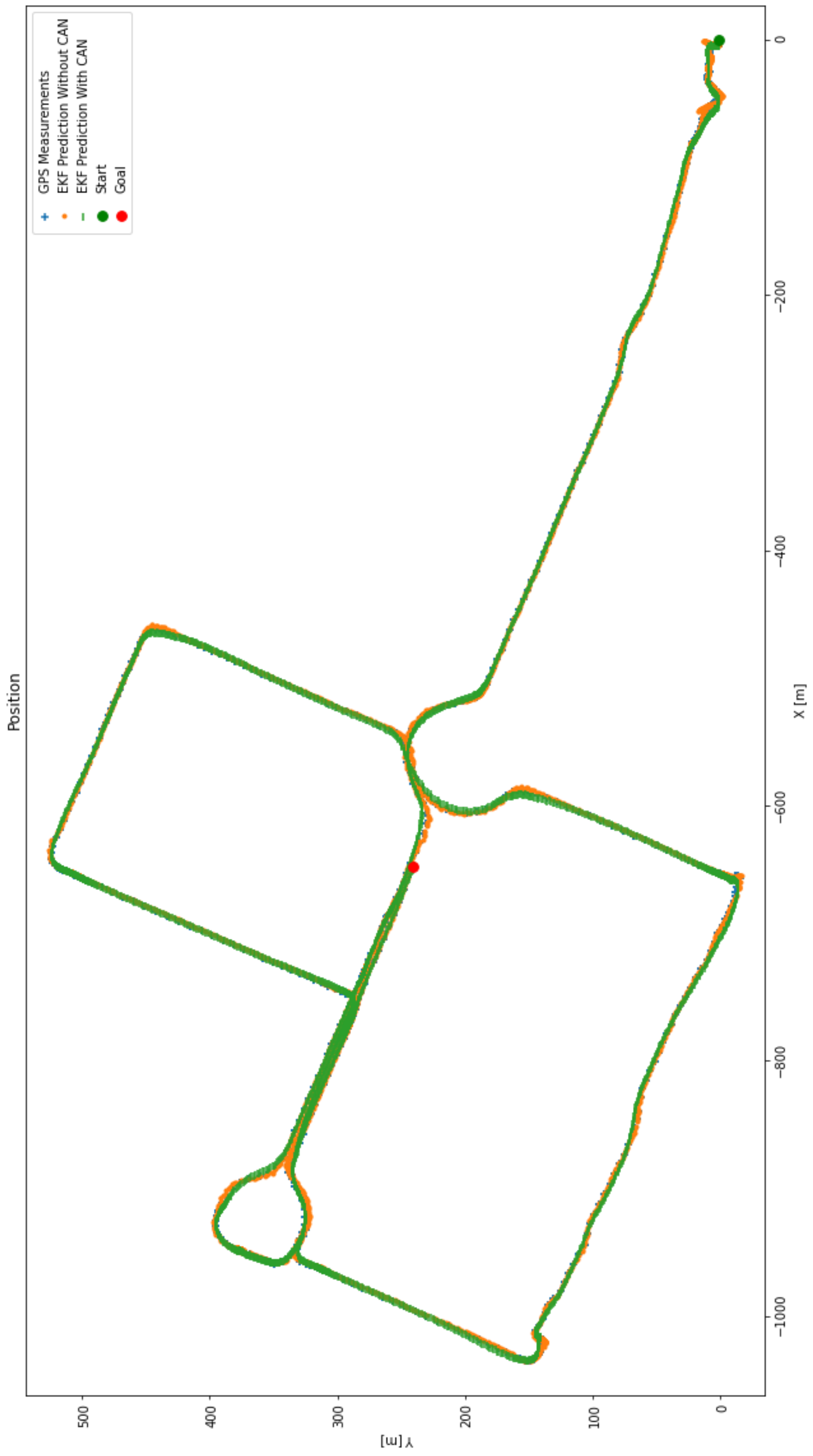


Figure 6.7 Trajectory predictions comparison for dataset 1.

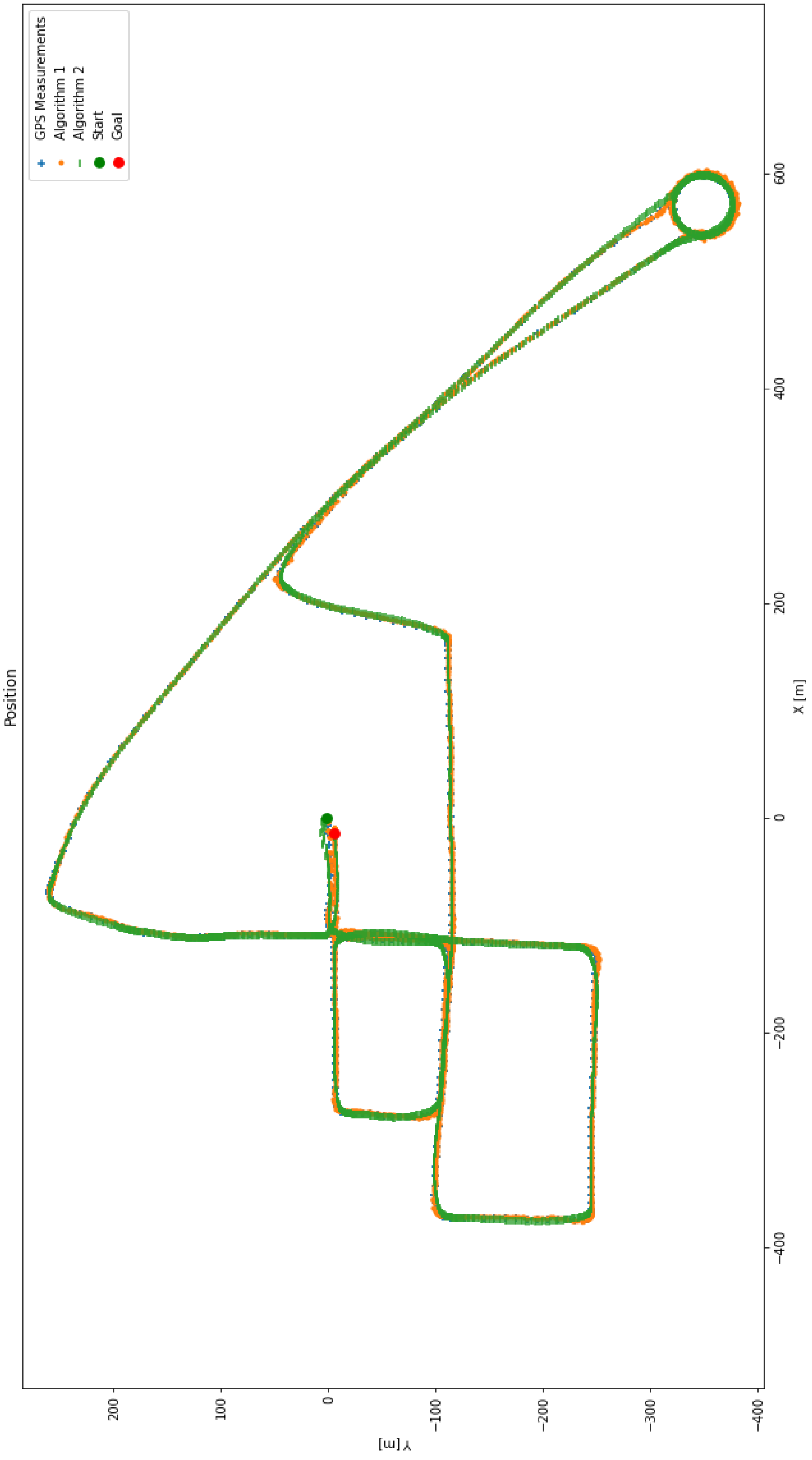


Figure 6.8 Trajectory predictions comparison for dataset 2.

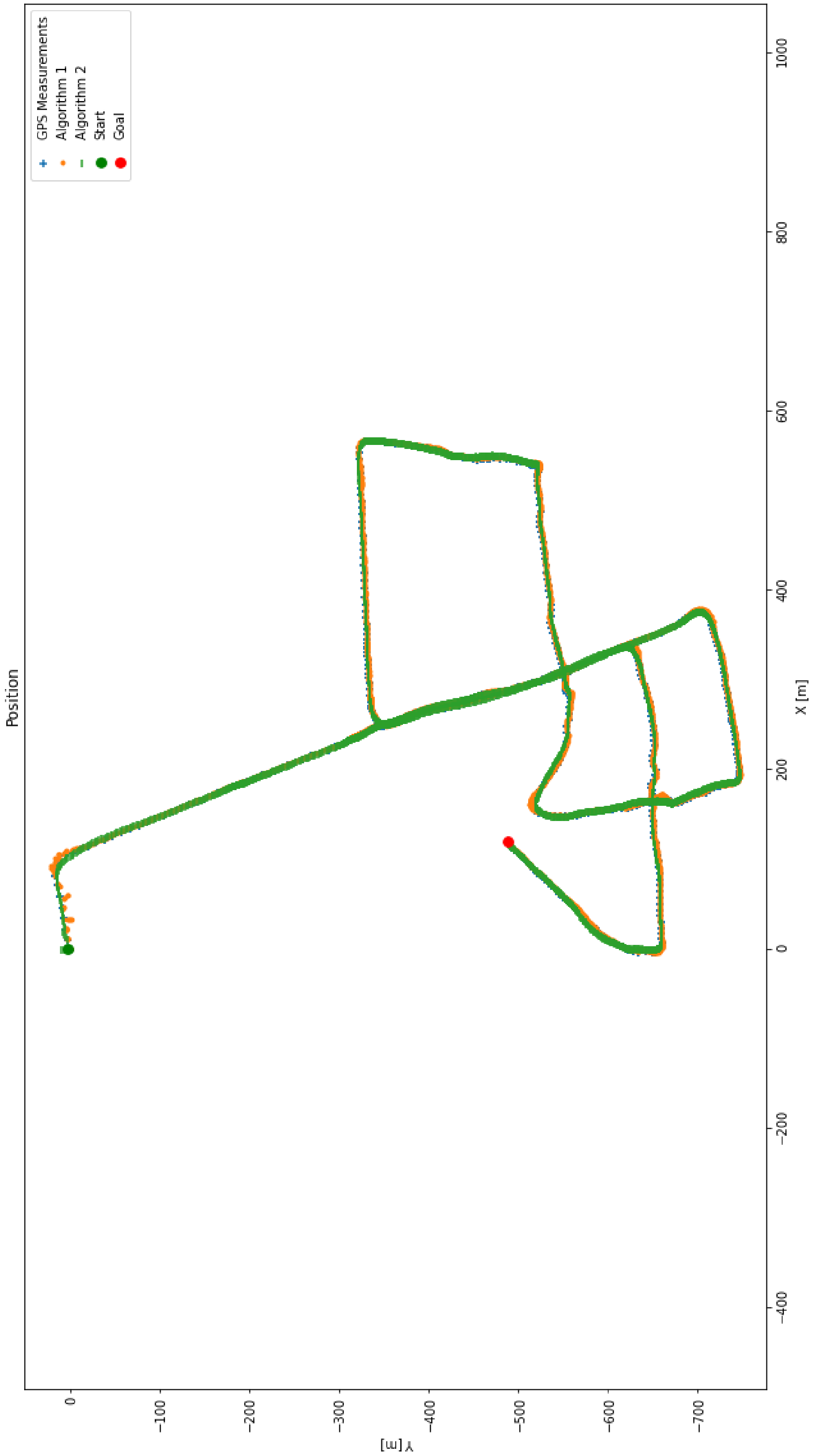


Figure 6.9 Trajectory predictions comparison for dataset 3.

It can also be observed that the minimum angle of the predictions does not fall below -3.14 radians, and the maximum angle of the predictions does not exceed 3.14 radians. The numbers represent the 360-degree range that a vehicle can change its heading in one step. Where North is 0 radians, any angle to the right is positive radians, and any angle to the left is negative. The vehicle can only change course by 180 degrees to left or right. But the predictions in EKFNS have values that exceed the limits of 3.14, which is a clear indication that the algorithm is throwing some errors.

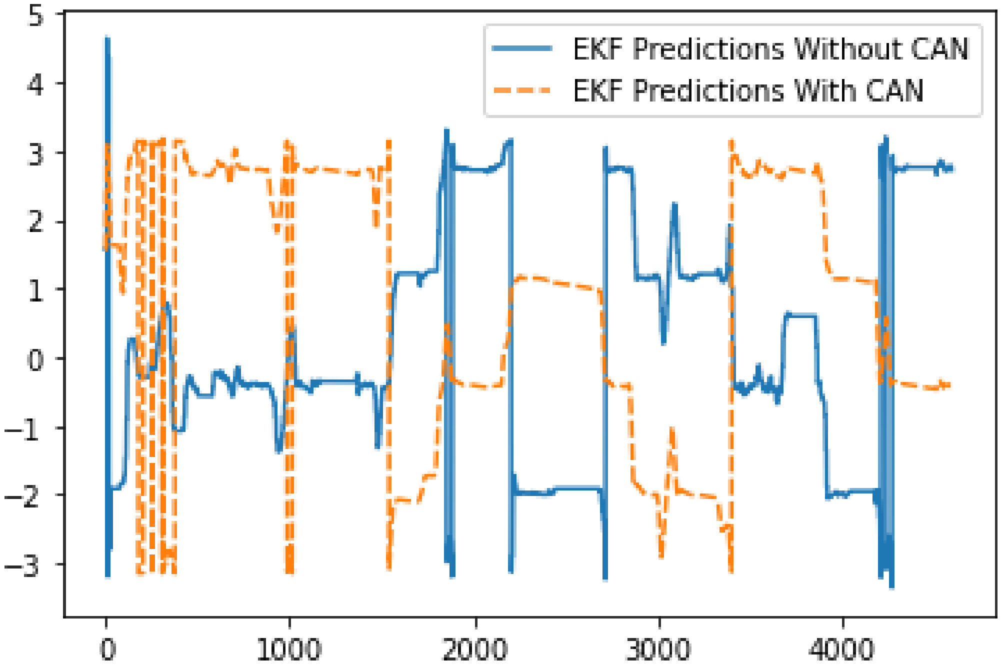


Figure 6.10 Predictions for heading in dataset 1.

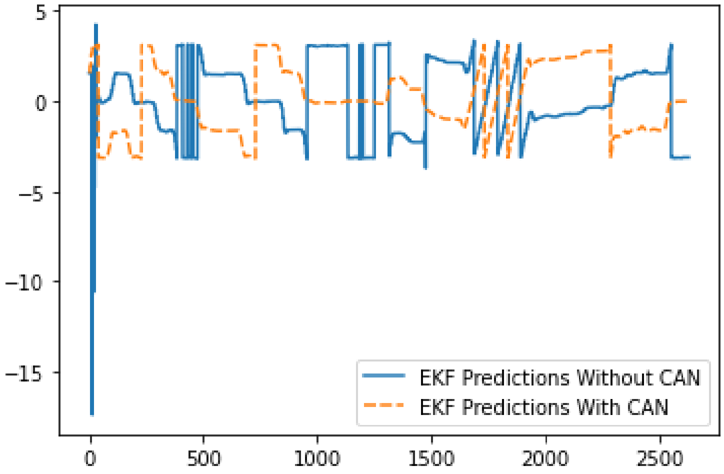


Figure 6.11 Predictions for heading in dataset 2.

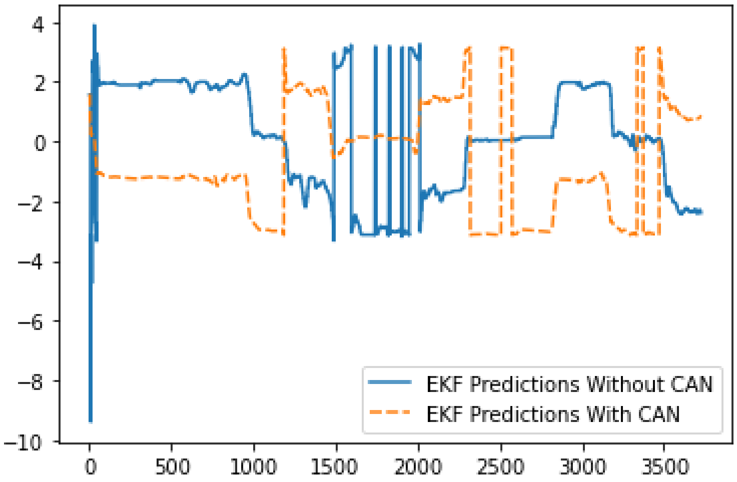


Figure 6.12 Predictions for heading in dataset 3.

Figure 6.13, Figure 6.14, and Figure 6.15 represent the trajectories of the predictions for the initial iterations. Figure 6.13 shows the first 600 predictions for dataset 1, Figure 6.14 represents the first 200 predictions for dataset 2, and Figure 6.15 shows the first 200 predictions for dataset 3. These figures enumerate all the above-mentioned observations. The first observation is that the predictions start by assuming north as the initial heading. The second observation is that the predictions made by EKFNS seem to fluctuate a lot and represent an unrealistic movement made by a vehicle's path. The predictions made by EKFCAN converge into the correct heading within a few steps and smooths during turns, representing the movement of a vehicle. Figure 6.13 shows the GPS values being stagnant makes the predictions in EKFNS more inconsistent. In contrast, the predictions from EKFCAN seem to have a smoother path, even when compared to the GPS representation.

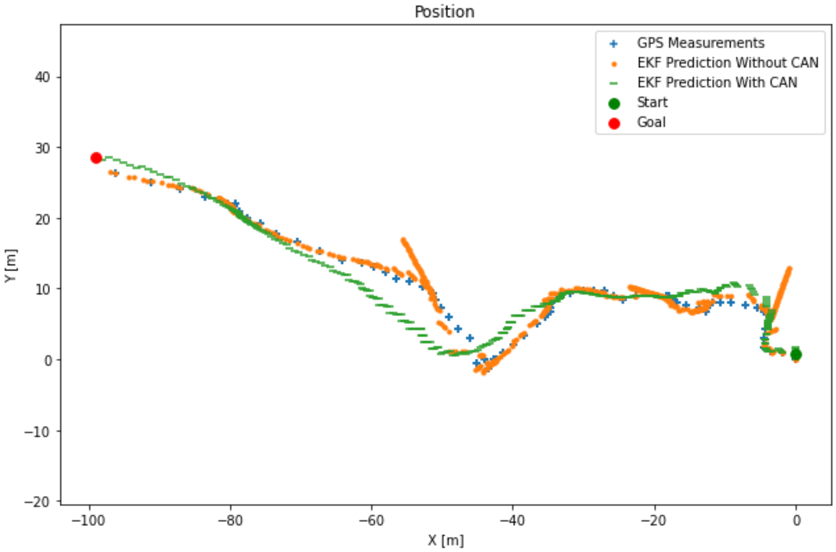


Figure 6.13 Starting uncertainty for dataset 1.

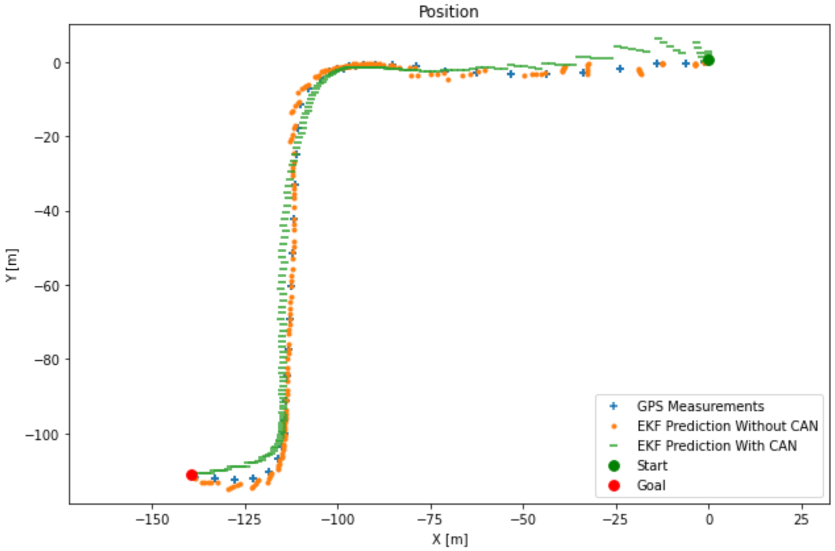


Figure 6.14 Starting uncertainty for dataset 2.

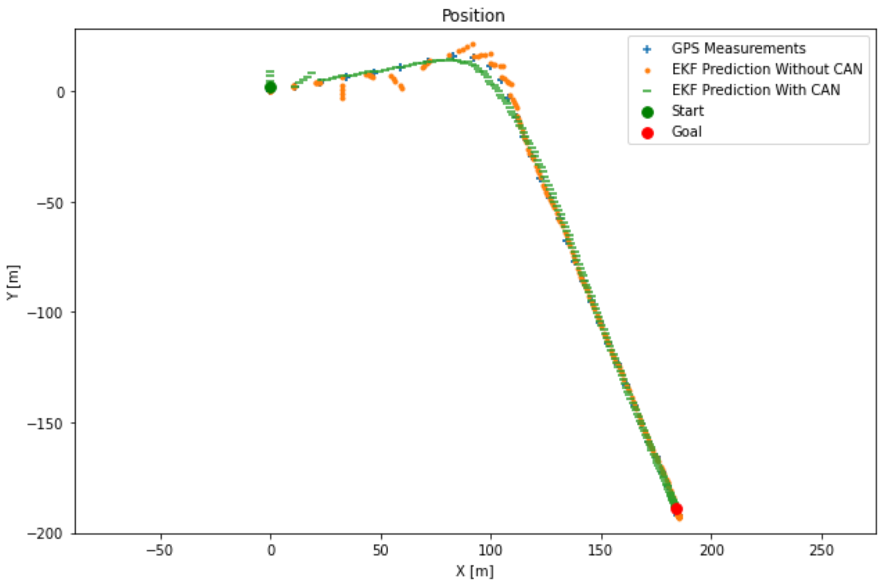


Figure 6.15 Starting uncertainty for dataset 3.

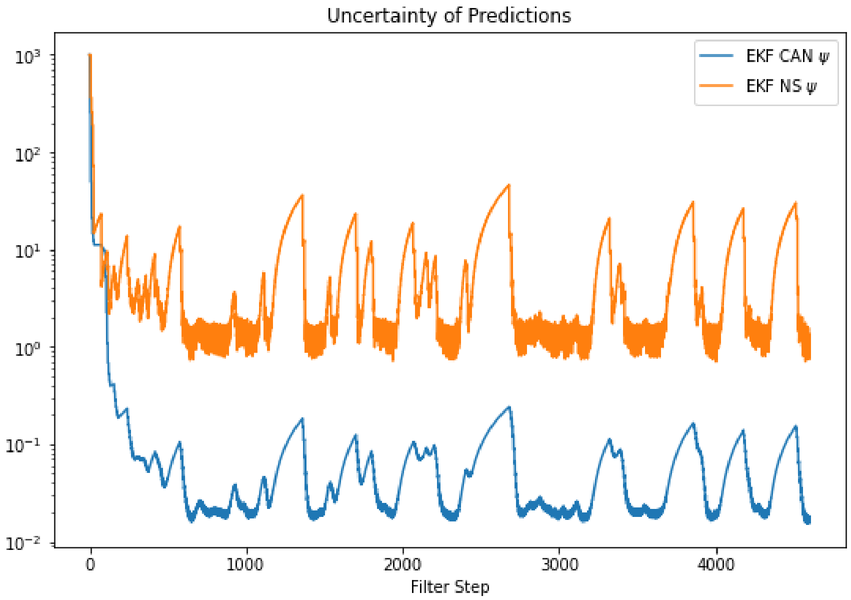


Figure 6.16 Heading prediction uncertainty dataset 1

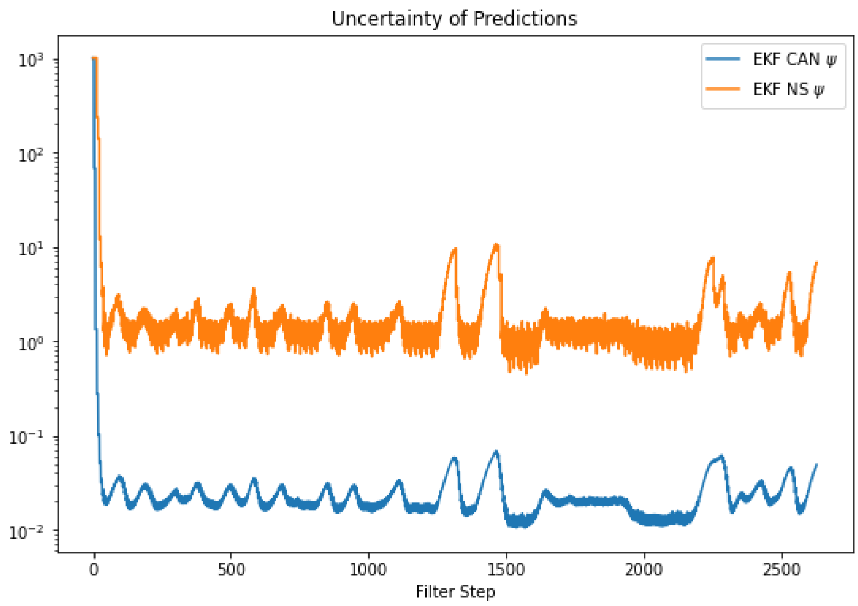


Figure 6.17 Heading prediction uncertainty dataset 2

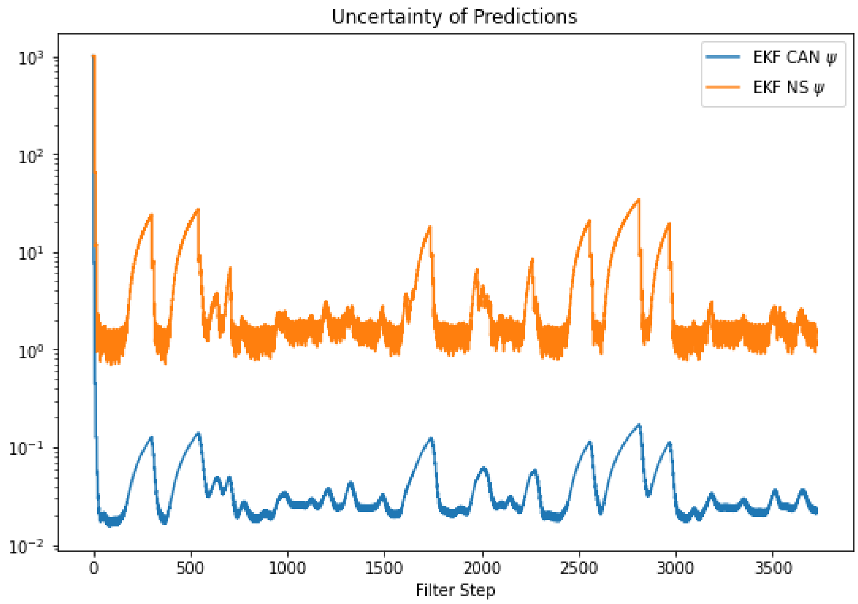


Figure 6.18 Heading prediction uncertainty dataset 3

Figure 6.16, Figure 6.17, and Figure 6.18 show the prediction uncertainty for dataset 1, dataset 2, and dataset 3 respectively. It can be observed that the uncertainty starts at 1000 this is because the initial uncertainty of the filter is set at 1000 which indicates the position of the vehicle is uncertain. After a few steps both the predictions of EKFNS and EKFCAN converge to a set variance. The EKFCAN uncertainty varies in the 0.001 to 0.01 range where as the uncertainty of the EKFNS varies in the range of 1 to 10. From the uncertainty graphs, it can be seen that the predictions made by EKFCAN have very low uncertainty values compared to the uncertainties by ELFNS. Another important observation that has to be made is the spikes in the uncertainties are indicators of a shift in the trajectories from a straight path to a turn. It can be observed from the Figure 6.16 that the initial uncertainty takes more time to converge when compared to the uncertainties of Dataset 2 and Dataset 3 this is because of the initial iterations of the Dataset 1 are very noisy as shown in Figure 6.13.

6.3.1.2 Vehicle Speed

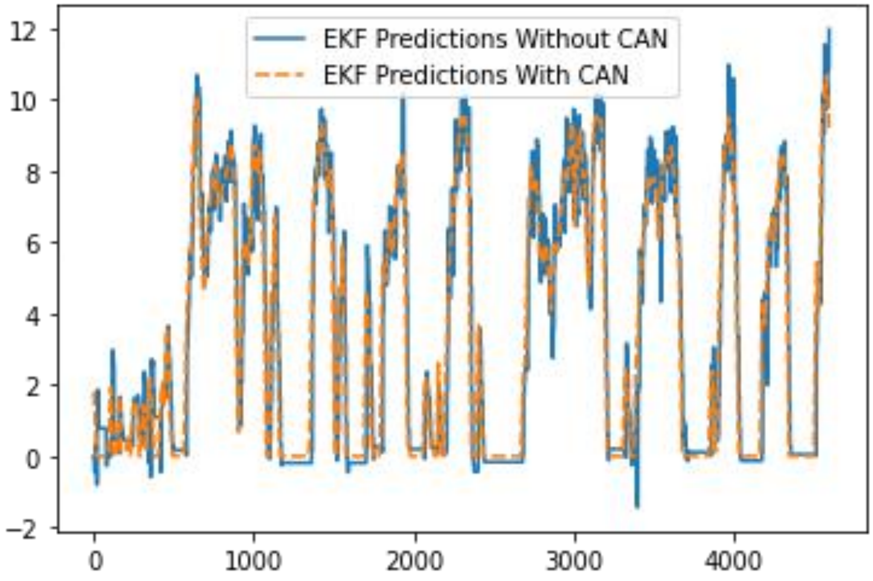


Figure 6.19 Speed prediction comparison for dataset 1.

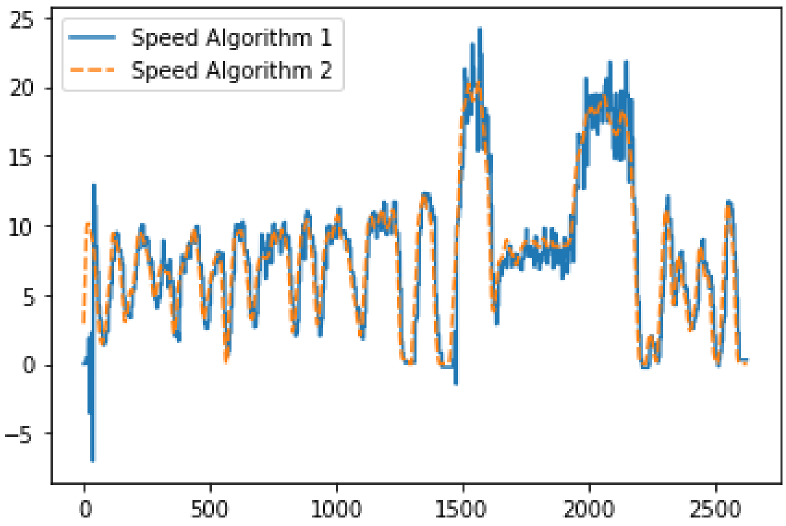


Figure 6.20 Speed prediction comparison for dataset 2.

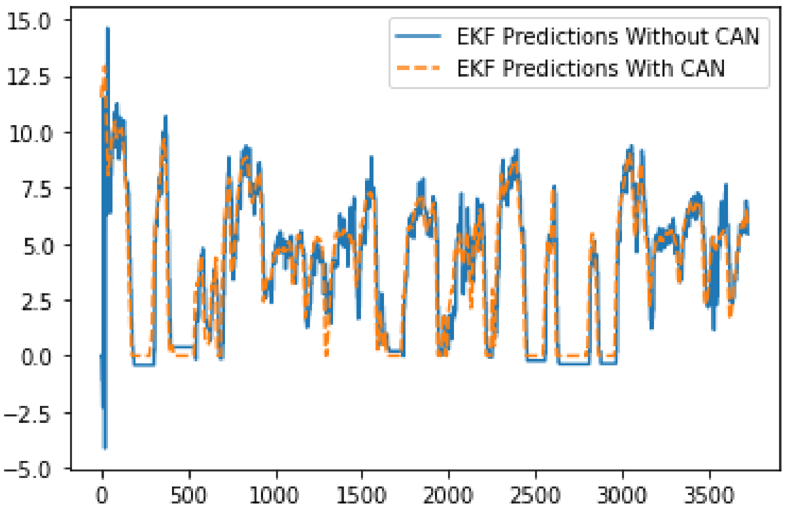


Figure 6.21 Speed prediction comparison for dataset 3.

The speed predictions of the EKFNS algorithm is compared to the speed from the EKFCAN algorithm in Figure 6.19 for dataset 1, Figure 6.20 for dataset 2, and Figure 6.21 for dataset 3. From the plots, it can be observed that the values predicted by EKFNS are also below zero, and EKFCAN has no values that are less than zero. This is because EKFCAN has inputs from the vehicle's speed sensor sending out speed data, and a vehicle's speed cannot dip below zero. The unique observation about EKFNS speed predictions is that the values dip to negative values when the speed in EKFCAN dips to zero and stays at zero for a continuous period. The speed of a vehicle should never reach below zero. These predictions are wrong and show how the EKFNS has errors in predictions.

6.3.1.3 General Trajectory Analysis

Analyzing the trajectory predictions for the roundabouts from dataset 2, let us call the trajectory represented in Figure 6.26 as circle 1, trajectory represented in Figure 6.27 as circle 2, and trajectory represented in Figure 6.28 as circle 3. Observing the trajectory of predictions EKFCAN, it is very smooth in circle 1, but for the initial 30 prediction points, the EKFCAN seems to be away from the trajectory of the GPS. This can be more evidently seen in Figure 6.29. The predictions made by EKFCAN seem to flow into the roundabout rather than seem abrupt turn shown by the GPS trajectory. But from all three circles, it can be seen EKFNS to be wrongly predicting the heading. This is because EKFNS only depends on the values of the next available GPS value to refresh the heading, and this heading will remain the same for the next five predictions made by the algorithm. In the case of EKFCAN, the heading almost always changes for every prediction in the circle since the vehicle is updating the change in heading. The trajectory in circle 2 EKFCAN completely resembles the trajectory of GPS. This is because the GPS corrected the position in circle 1. This statement should be observed to be true because in circle 3, at the end of the trajectory, the EKFCAN predictions start to move away from the trajectory of GPS and EKFNS at least by 20 predictions, The change in the trajectory can only be attributed to the change in the heading angle provided by the vehicle sensor.

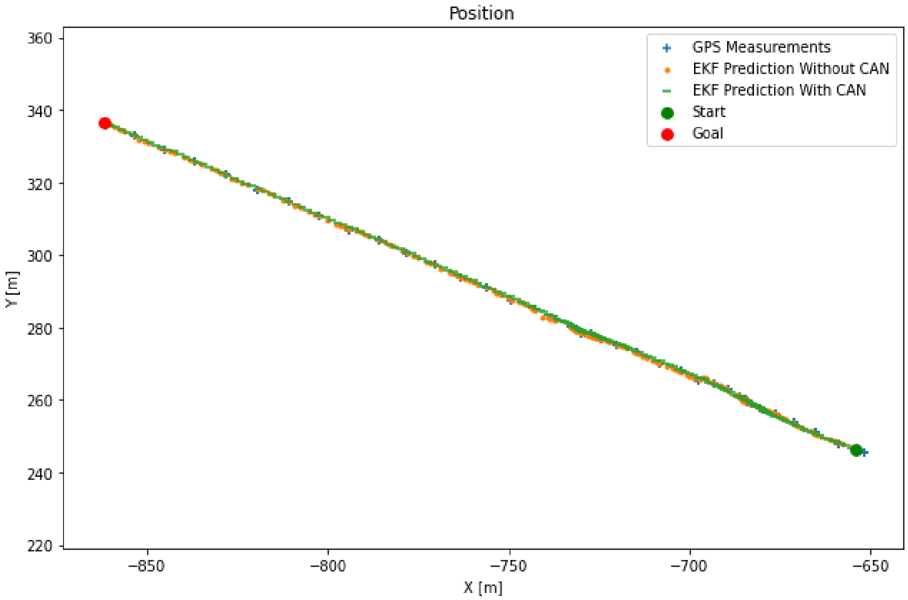


Figure 6.22 Trajectory of predictions from iterations 1100 to 1500 in dataset 1.

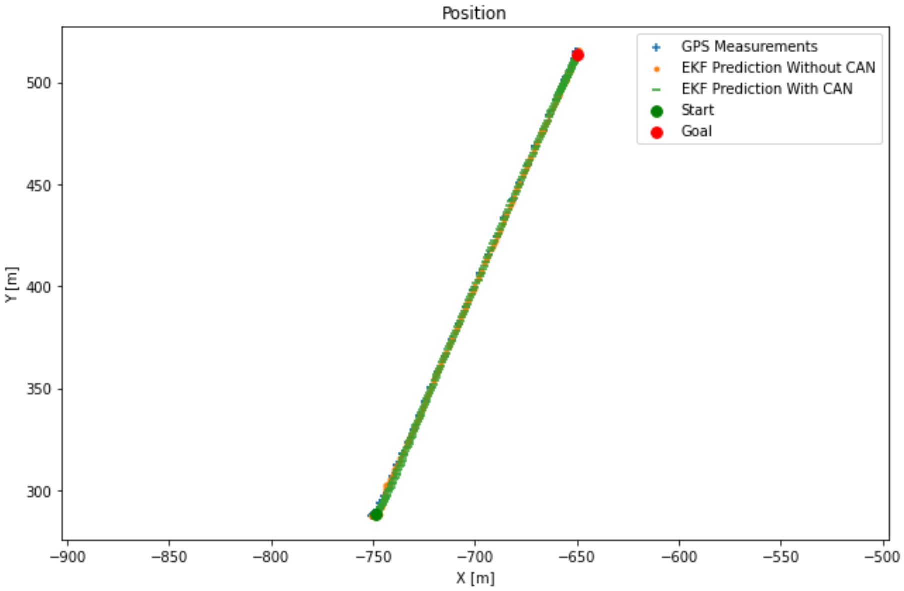


Figure 6.23 Trajectory of predictions from iterations 2200 to 2500 in dataset 1.

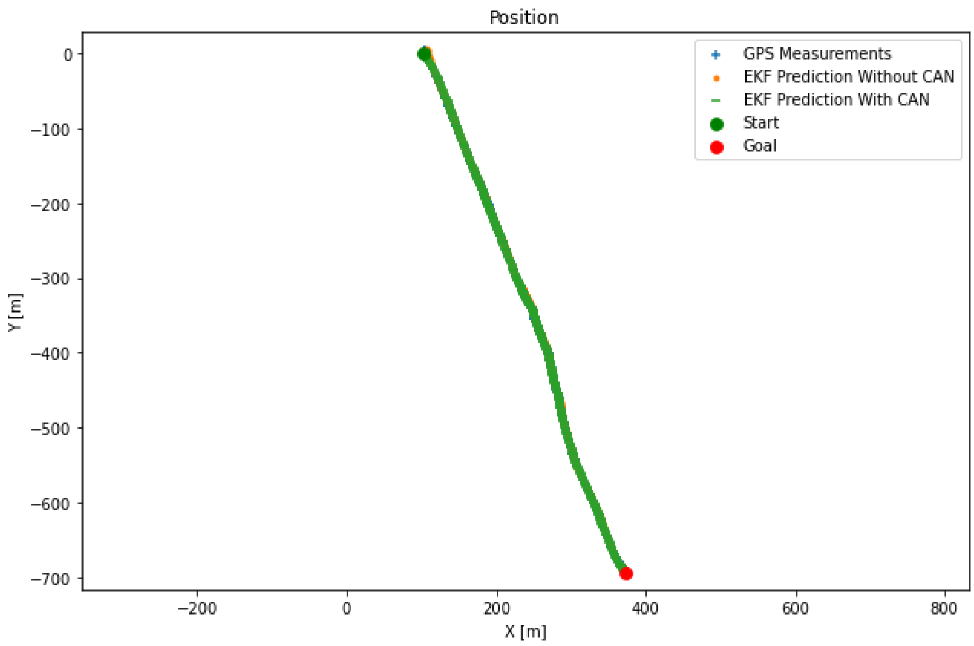


Figure 6.24 Trajectory of predictions from iterations 50 to 950 in dataset 3.

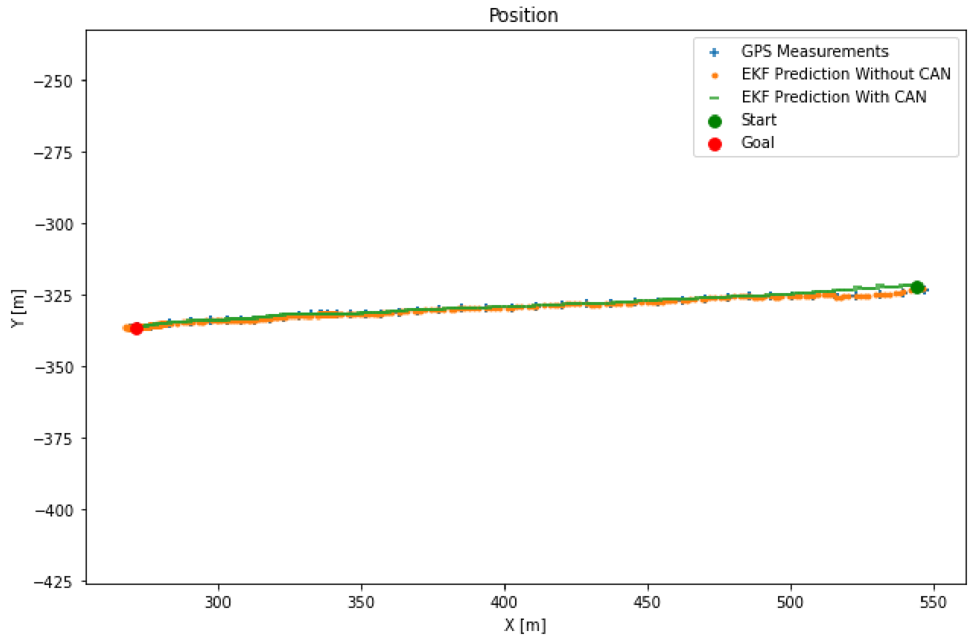


Figure 6.25 Trajectory of predictions from itirations 2300 to 2800 in dataset 3.

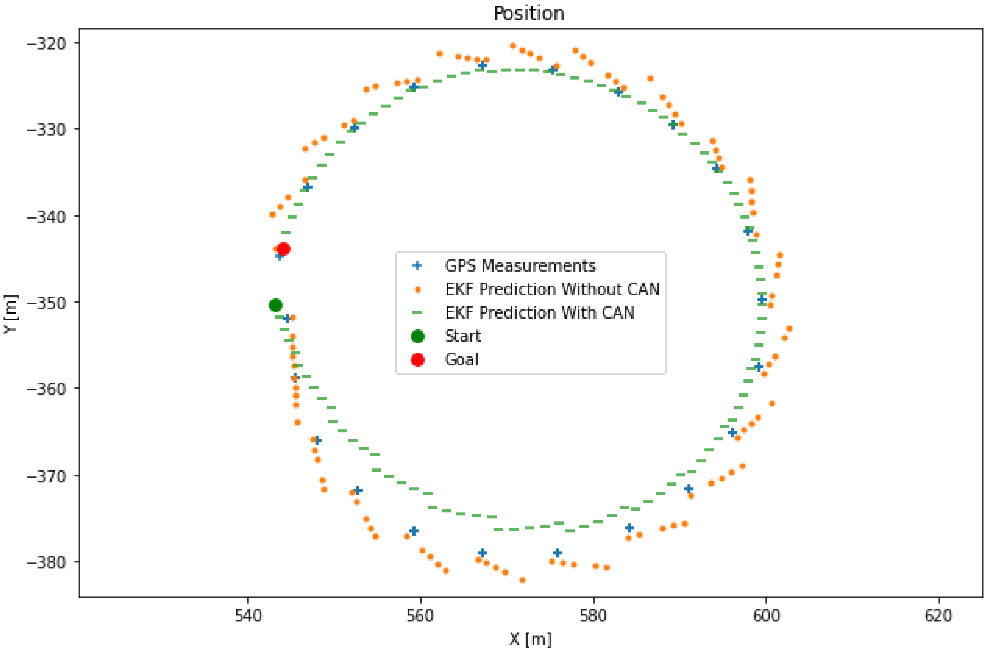


Figure 6.26 Trajectory of predictions from 1650 to 1755 in dataset 2.

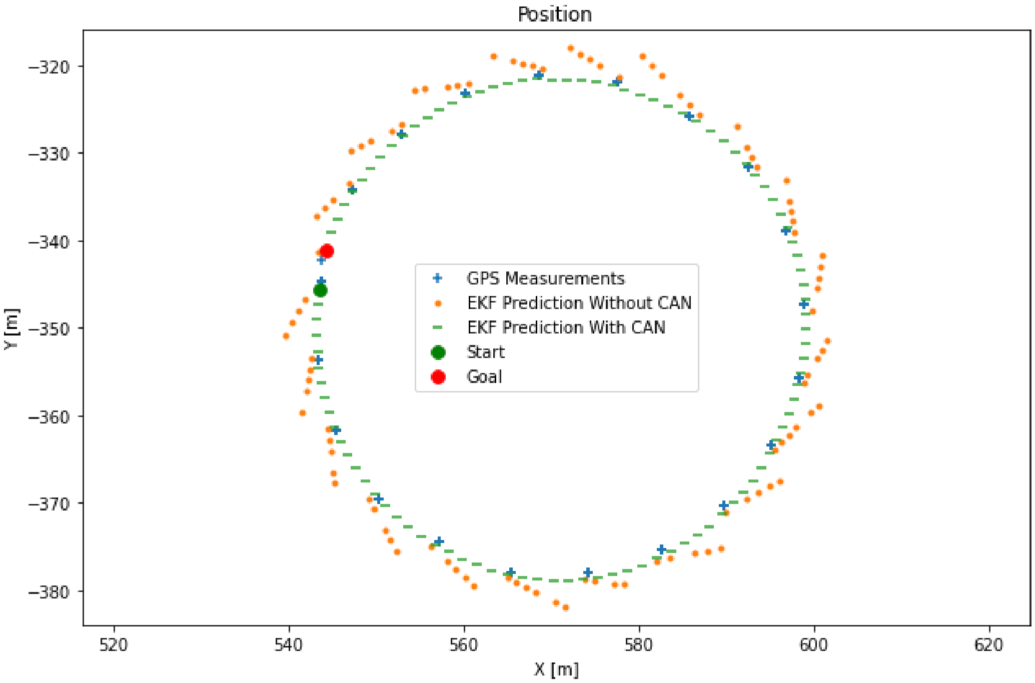


Figure 6.27 Trajectory of predictions from 1755 to 1855 in dataset 2.

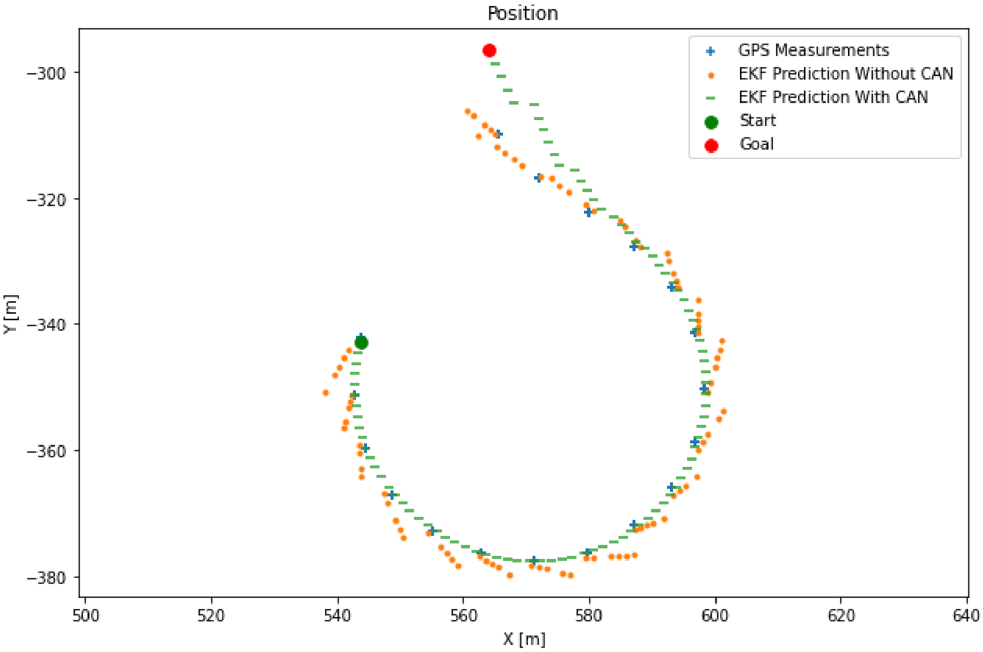


Figure 6.28 Trajectory of predictions from 1855 to 1945 in dataset 2.

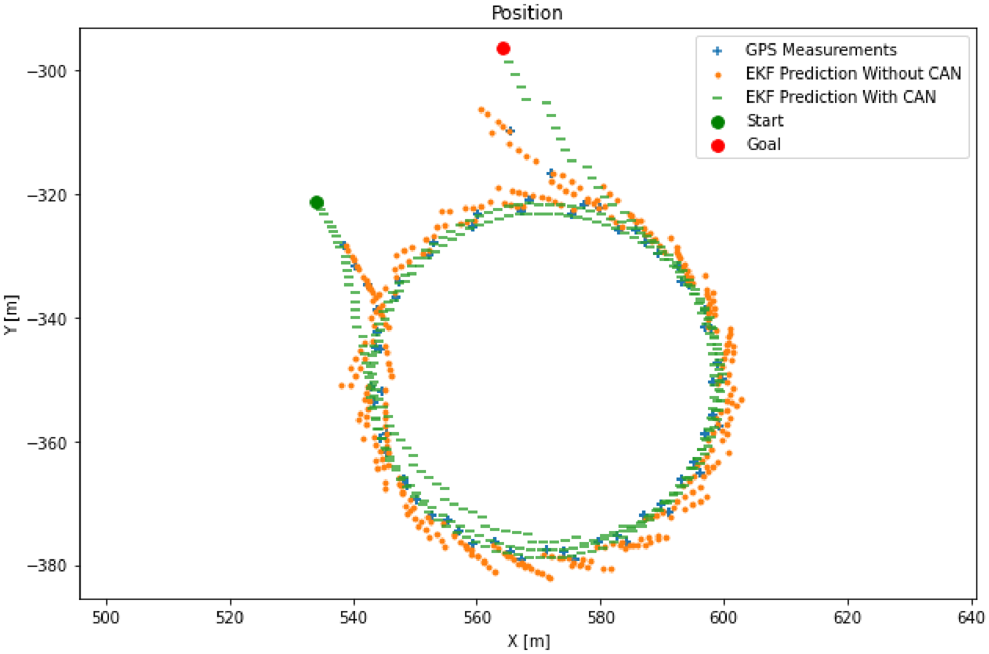


Figure 6.29 Trajectory comparison for all the circles combined in dataset 2.

From Figure 6.22, Figure 6.23, Figure 6.24, and Figure 6.25, it can be said that both EKFNS and EKFCAN tend to predict almost the same path. This is an important observation because the predictions don't have any change in the heading angle and only rely on the speed of the vehicle. Observe the trajectories presented in Figure 6.30 turn 5 for dataset 1. The trajectory for the EKFCAN starts to turn before the trajectory of the EKFNS or GPS. This can be observed in all the turns, which are preceded by long straight trajectories. Since speed is input to EKFCAN, the algorithm predicts the slowing of speed and evidently shows in the predictions made, whereas the predictions of EKFNS and GPS trajectories take more time to refresh and measure the speed of the vehicle. Figure 6.31 shows another important problem that occurs in GPS predictions. When the vehicle comes to a halt or when the vehicle slows down considerably, the GPS predictions tend to jump back and forth, which results in the EKFNS predicting the negative values for the speed prediction. But it can be observed that EKFCAN predictions do not jump back and forth but move in a much smoother trajectory.

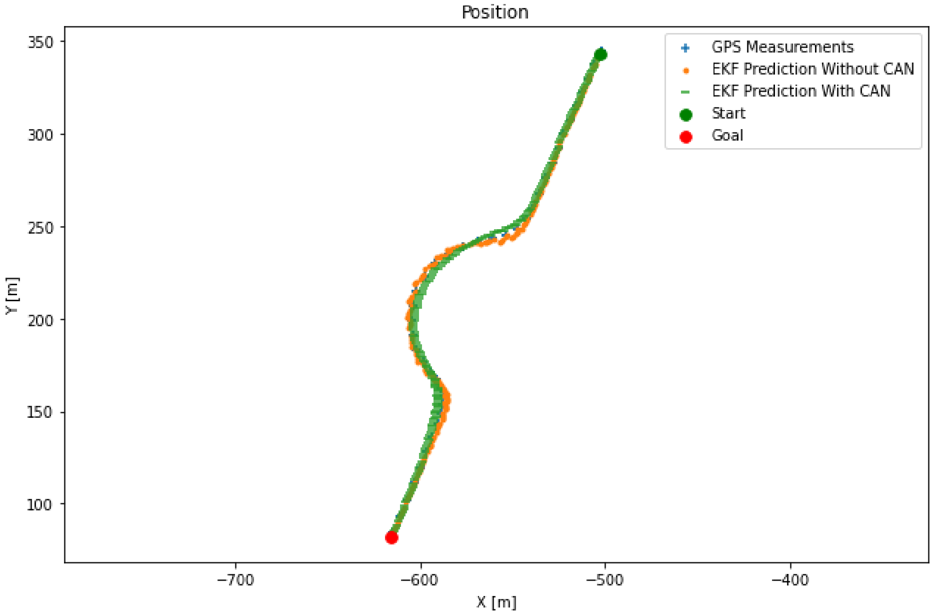


Figure 6.30 Trajectory comparisons for iterations 2950 to 3150 in dataset 1.

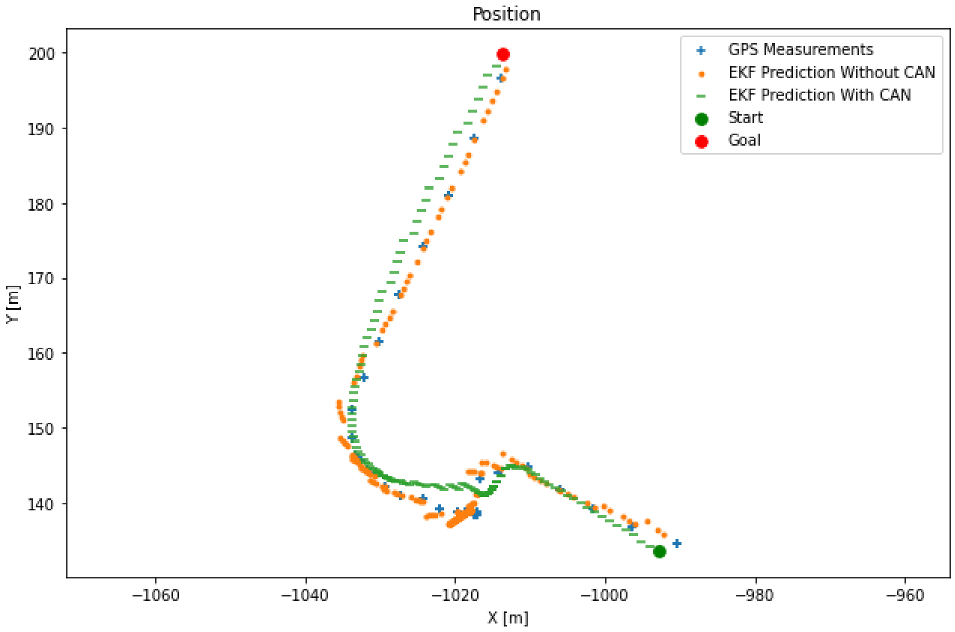


Figure 6.31 Trajectory comparisons for iterations 3650 to 3950 in dataset 1.

From all the analysis, it can be deduced that the predictions made by EKFCAN are more valid and tend to filter the errors of the GPS. Further, EKFCAN predicts trajectories in which a car can be driven without having any improbable difference.

6.3.2 Evaluation

The Root Mean Square Deviation (RMSD) is a statistical means to refer to the difference between two trajectories. It is calculated by finding the standard deviation of the residuals.

$$RMSD = \sqrt{\frac{\sum_{i=0}^{N}(O_i - P_i)^2}{N}}$$

where O_i is the ith observation and P_i is the ith prediction. For our study, the observations are the GPS values and the predictions taken from the values predicted by EKFNS and values predicted by EKFCAN. Later on, in the discussion also the RMSD between EKFNS and EKFCAN is also presented. Though the formula states the values as Observation and Prediction, the RMSD is a measure of how different the two series are. The RMSD of EKFNS with GPS is represented by blue bars, and the RMSD of EKFCAN is represented by orange bars.

6.3.2.1 Dataset 1

In Figure 6.32 and Figure 6.33, RMSD values for the segments where the trajectories were turning are compared. Except for Turn 4 in X direction, RMSD of EKFCAN is higher when compared to RMSD of EKFNS. Observe Figure 6.34 at the turn the variation for the values of EKFNS were greater with respect to the X direction, compared to the variation of values of EKFCAN to the GPS. This resulted in more difference in RMSD of EKFNS to RMSD of EKFCAN in Turn 4. In Figure 6.33, except for Turn 3, RMSD of EKFCAN is higher than EKFNS. Inspecting Turn 3 Figure 6.35, the same reasoning can be applied but with respect to the Y direction. By observing all the RMSD values for turn segments, it can be said that EKFCAN has more than one meter difference than the GPS.

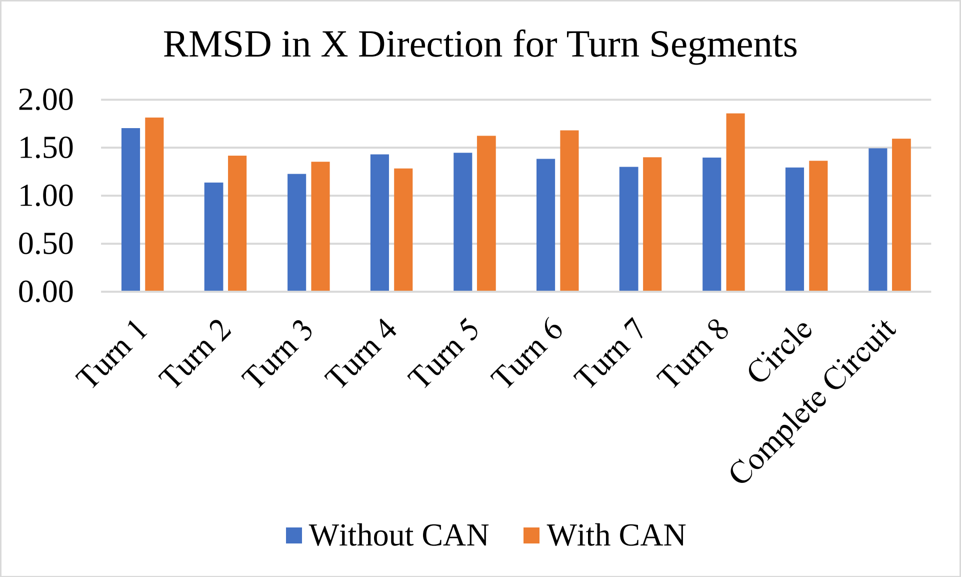


Figure 6.32 RMSD comparison of X for turns segments dataset 1.

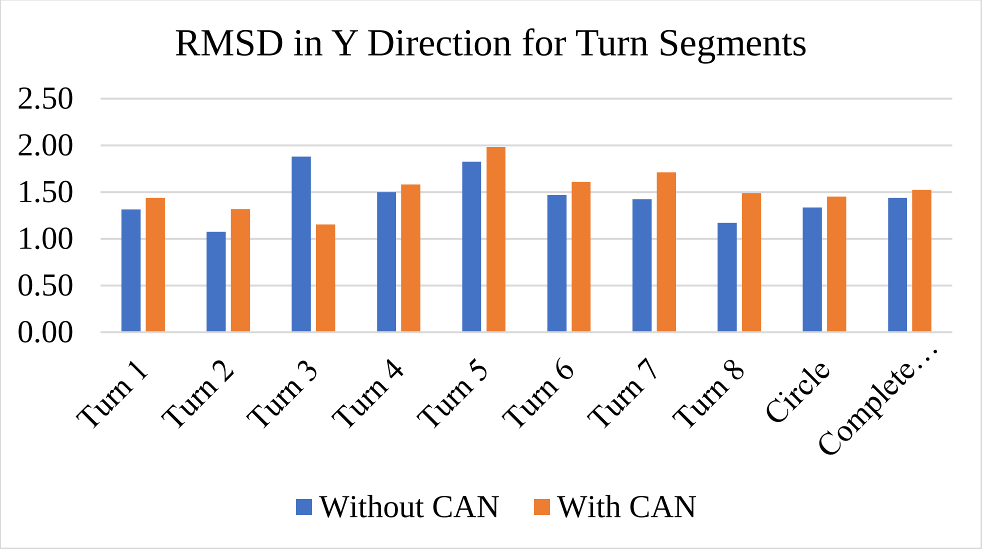


Figure 6.33 RMSD comparison of Y for turn segments dataset 1.

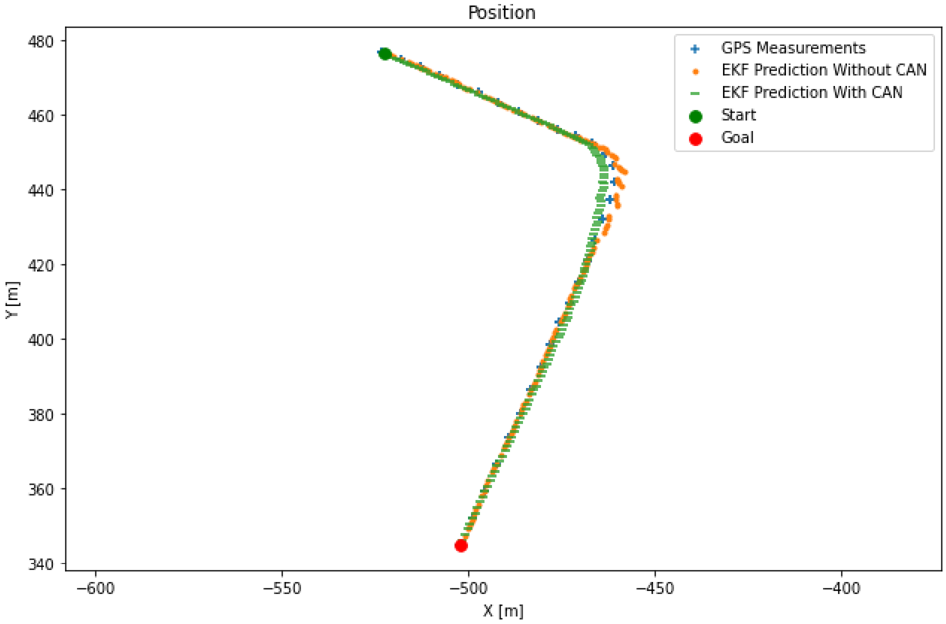


Figure 6.34 Trajectory comparison for dataset 1 at turn 4.

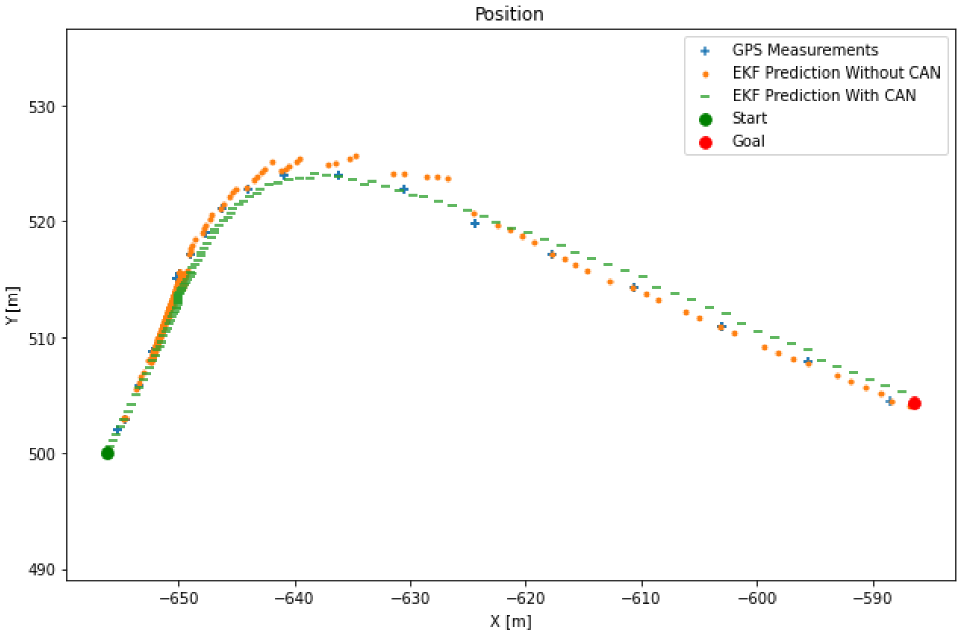


Figure 6.35 Trajectory comparison for dataset 1 at turn 3.

By observing the RMSD for the straight segments in the X and Y direction, straight segment 4 has lower EKFCAN RMSD than EKFNS RMSD. This is attributed to the segment not being completely straight but having light curves, and please refer to Figure 6.38.

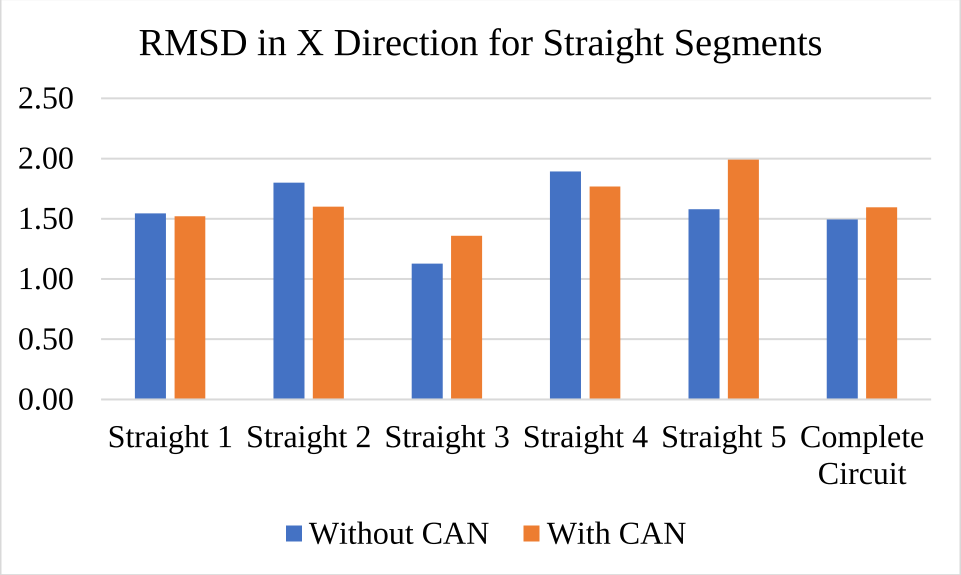


Figure 6.36 RMSD comparison of X for straight segments dataset 1.

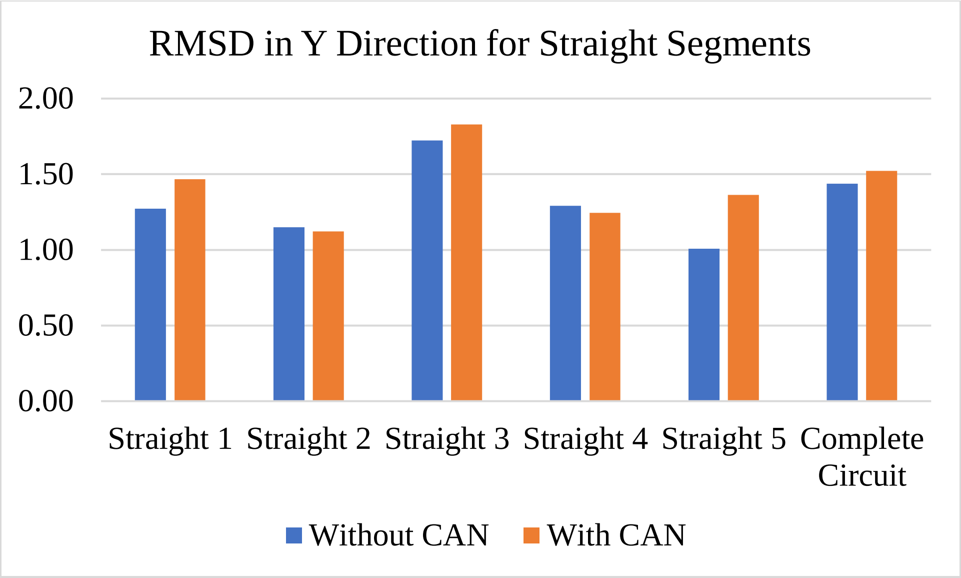


Figure 6.37 RMSD comparison of Y for straight segments dataset 1.

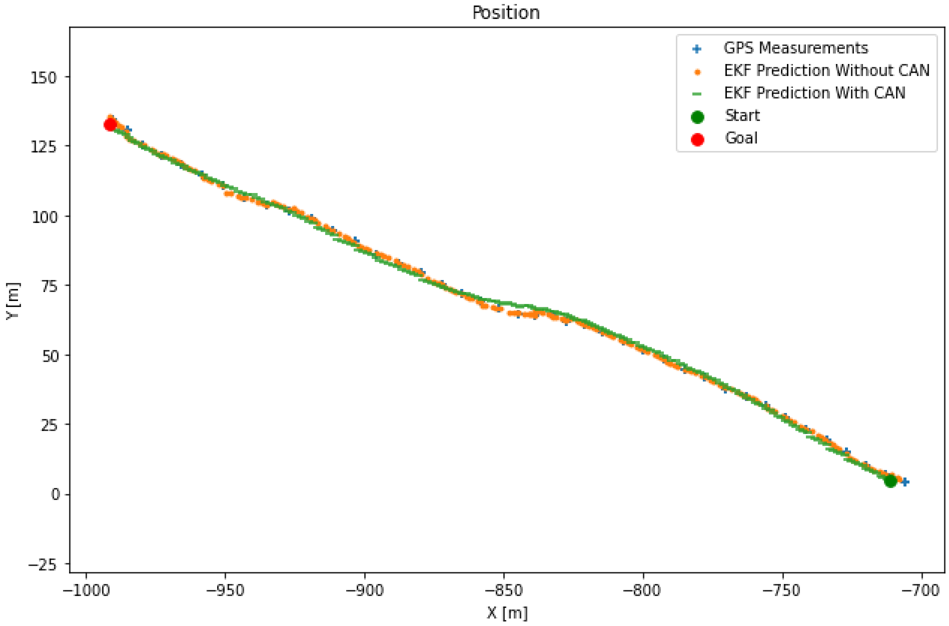


Figure 6.38 Trajectory comparison for dataset 1 at straight 4.

6.3.2.2 Dataset 2

From Figure 6.39 and Figure 6.40, the first observation that can be made is when comparing the EKFCAN RMSD for turn segments amongst all the three datasets, the EKFCAN RMSD in dataset 2 has the highest values. It is almost always close to two meters compared to the EKFCAN

RMSD values in dataset 1, most of the segments are a little more than 1.5 meters, and EKFCAN RMSD values in dataset 3 which are mostly a little less than 1.5 meters. And except for turn 4 and turn 8 in the Y direction, the EKFCAN RMSD is always more than the EKFNS RMSD. Both turn 4 and turn 8 (as shown in Figure 6.41) resemble a similar trajectory where the trajectory shifts from a vertical path to a horizontal path in almost a right-angle turn. Because of the heading error for the EKFNS, the change in trajectory takes time, and the variation from the GPS and EKFCAN is higher.

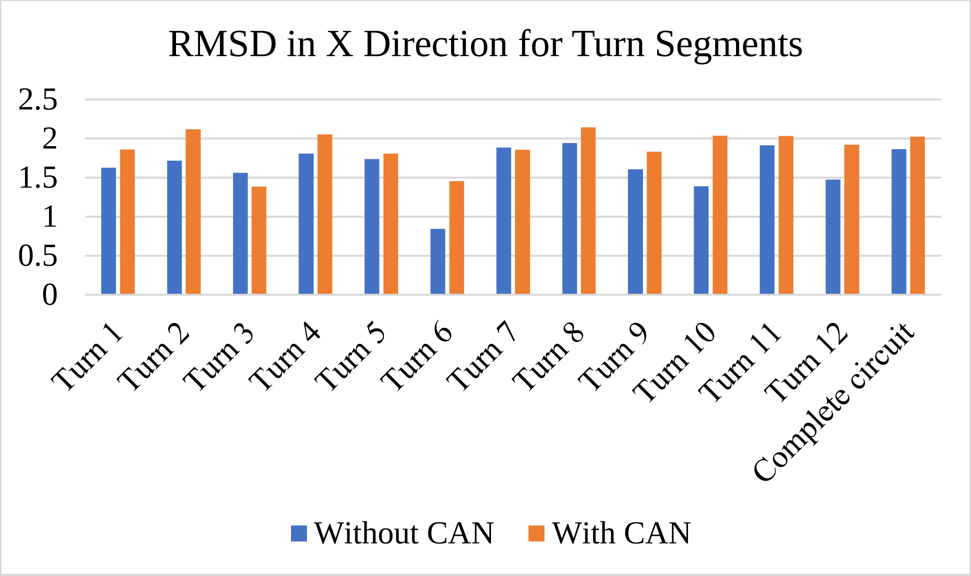


Figure 6.39 RMSD comparison of X for turn segments dataset 2.

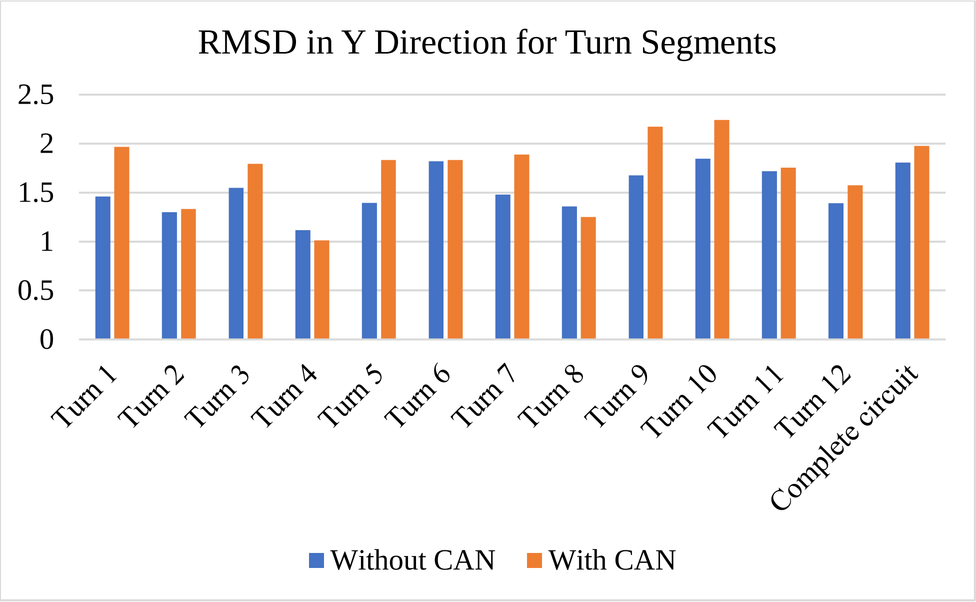


Figure 6.40 RMSD comparison of Y for turn segments dataset 2.

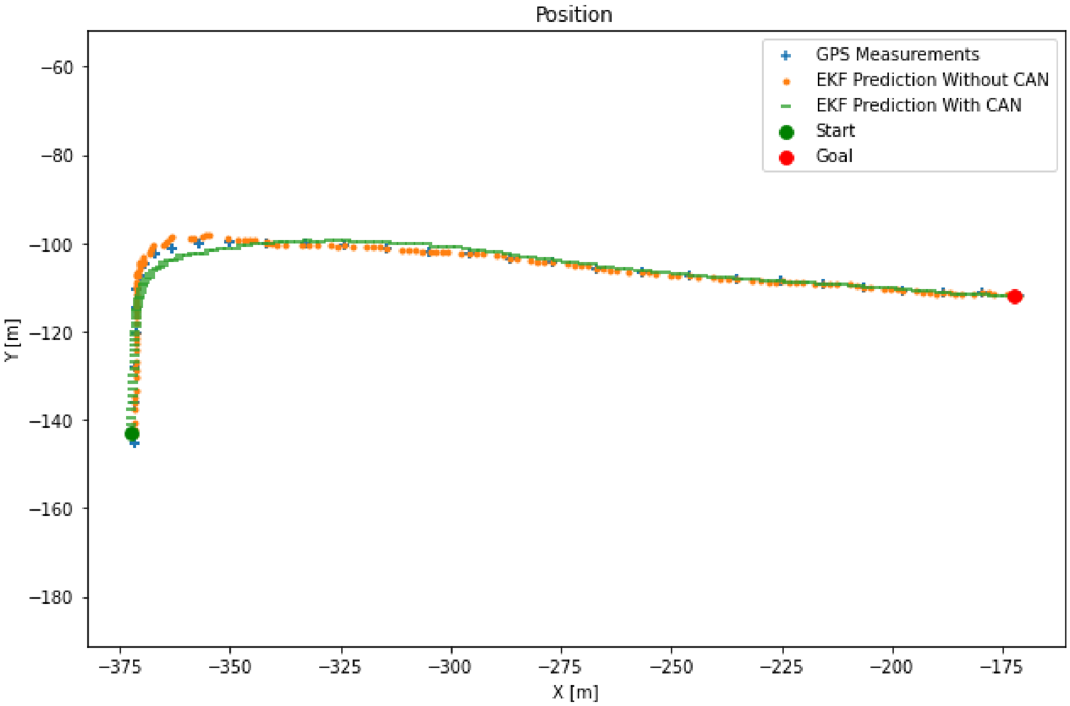


Figure 6.41 Trajectory comparison for dataset 2 at turn 8.

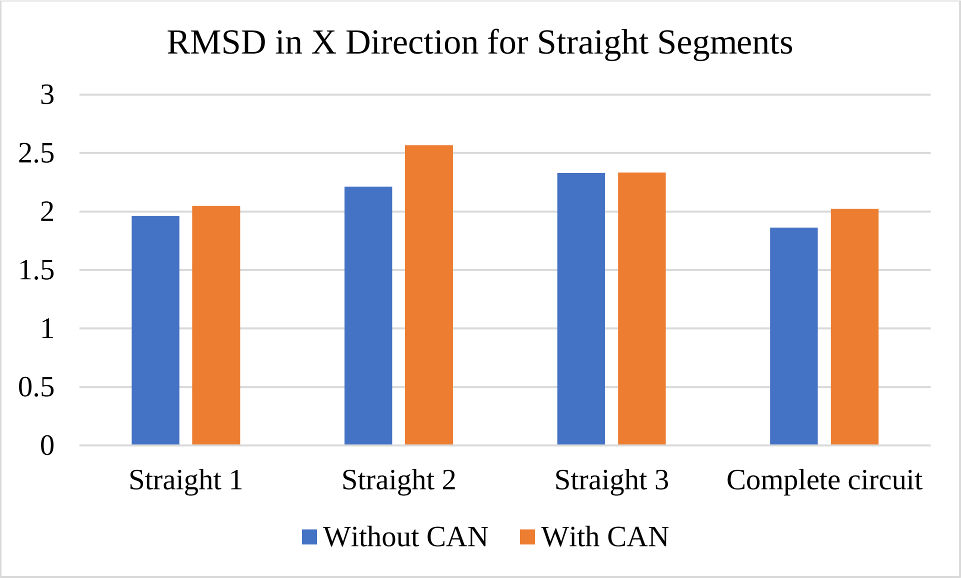


Figure 6.42 RMSD comparison of X for straight segments dataset 2.

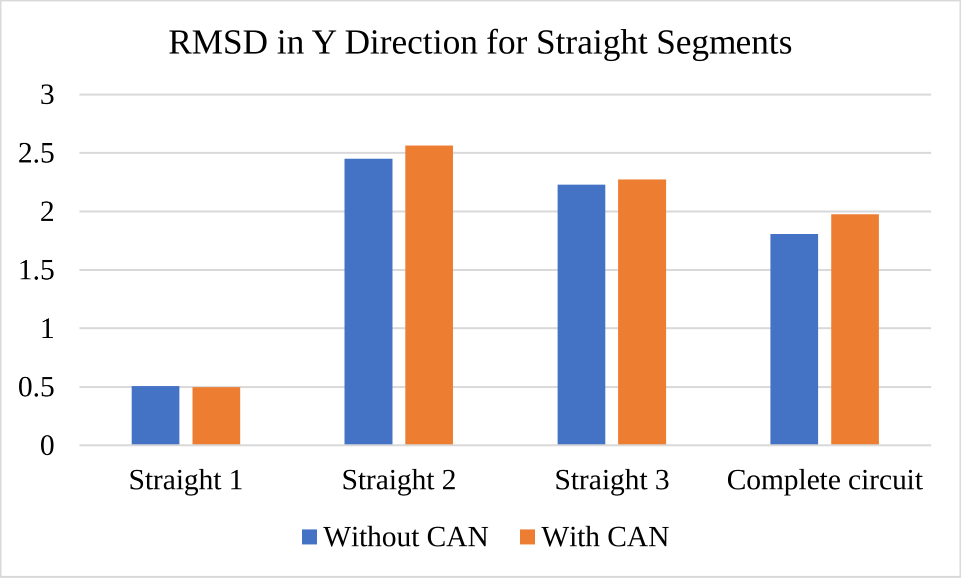


Figure 6.43 RMSD comparison of Y for straight segments dataset 2.

The observations that can be made from the RMSD comparisons for the straight segments for dataset 2 are the EKFCAN RMSD are always higher than the EKFNS RMSD. Another observation that can be made is that segment straight 1 and segment straight 3 have almost the same RMSD. This is because the two segments resemble very straight lines and the segment straight 2 is a little curved. Another observation that can be made is that the RMSD for segment straight 1 in the Y direction is about 0.5 meters. This is because the trajectory of this segment resembles the closest to a horizontal line. Therefore, there is less deviation in the Y direction, but there is a huge deviation in the X direction where both the RMSD are close to two meters.

Like the straight segments RMSD, the RMSD comparison for the circle segments also almost all the time have the EKFCAN RMSD higher than the EKFNS RMSD. Except for circle 3 X direction. Observing the trajectory from Figure 6.28, at the final steps of the prediction in circle 3 segment, the trajectory of EKFCAN remains in a vertical position, whereas the trajectory of EKFNS moves with the GPS and turns more horizontal.

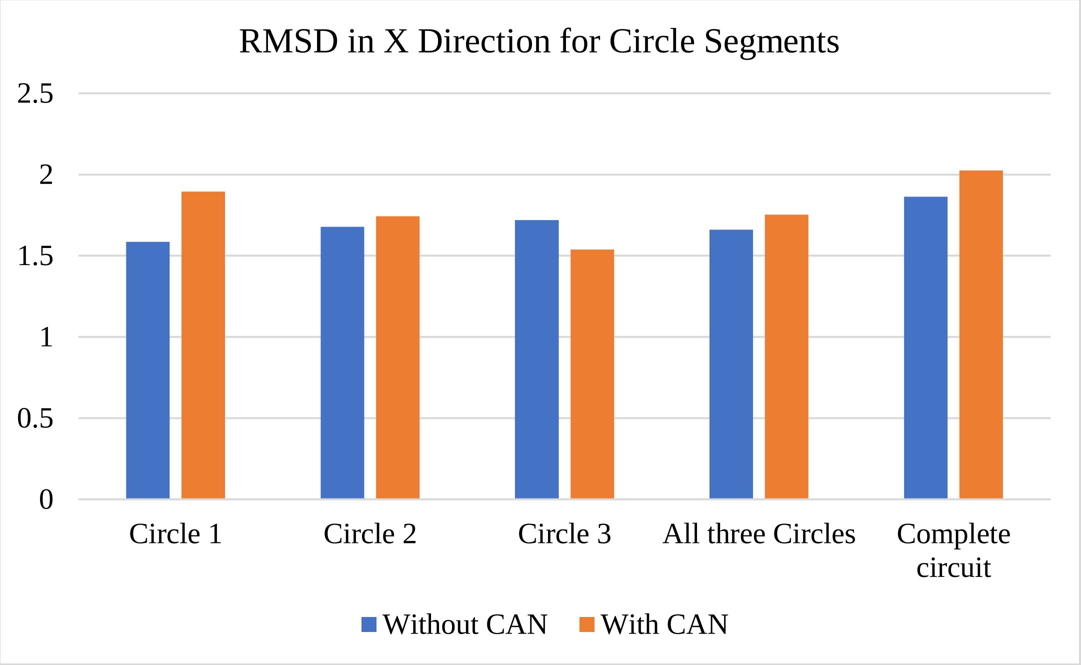


Figure 6.44 RMSD comparison of X for circle segments dataset 2.

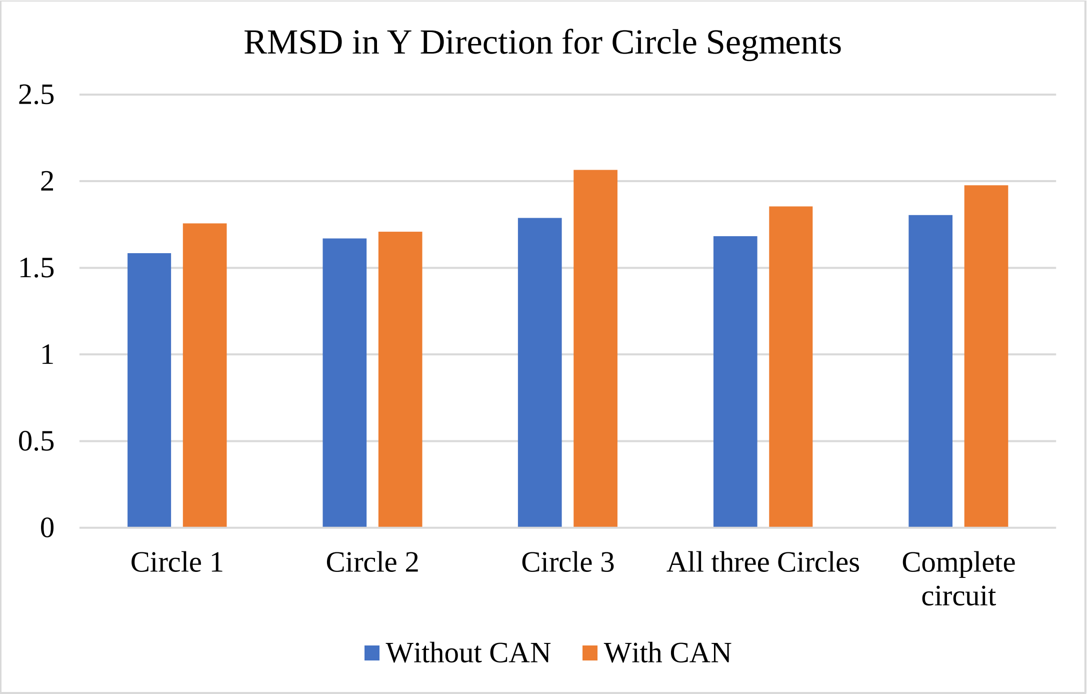


Figure 6.45 RMSD comparison of Y for circle segments dataset 2.

6.3.2.3 Dataset 3

The first major observation that can be made from Figure 6.46, Figure 6.47for the first time is that the EKFCAN RSMD is a little less than EKFNS RMSD in the X-direction. It can be deducted that dataset 3 has the maximum number of straight segments, and this could be the reason for the RMSE values for both the complete circuit in the X direction is more for EKFNS. Another deduction that can be observed is that trajectories in dataset 1 and dataset 2 are mostly spread out in the X direction, and the trajectory for dataset 3 is spread out in the Y direction. The extreme difference in the RMSD in the X direction for the turn 7 segment can be observed in Figure 6.31. The sudden slowdown of the vehicle and change in trajectory is the reason for the extreme difference in RMSD for segment turn 7.

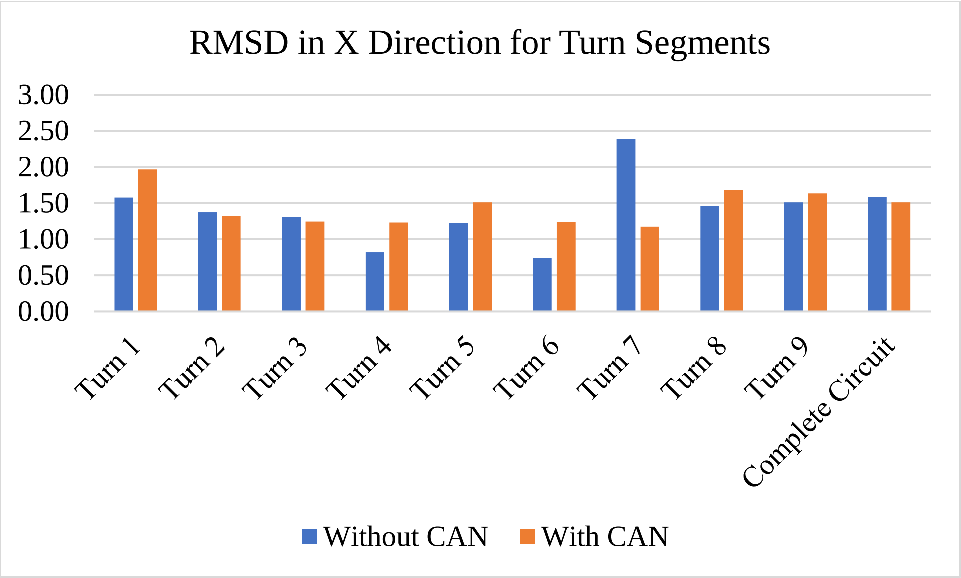


Figure 6.46 RMSD comparison of X for turn segments dataset 3.

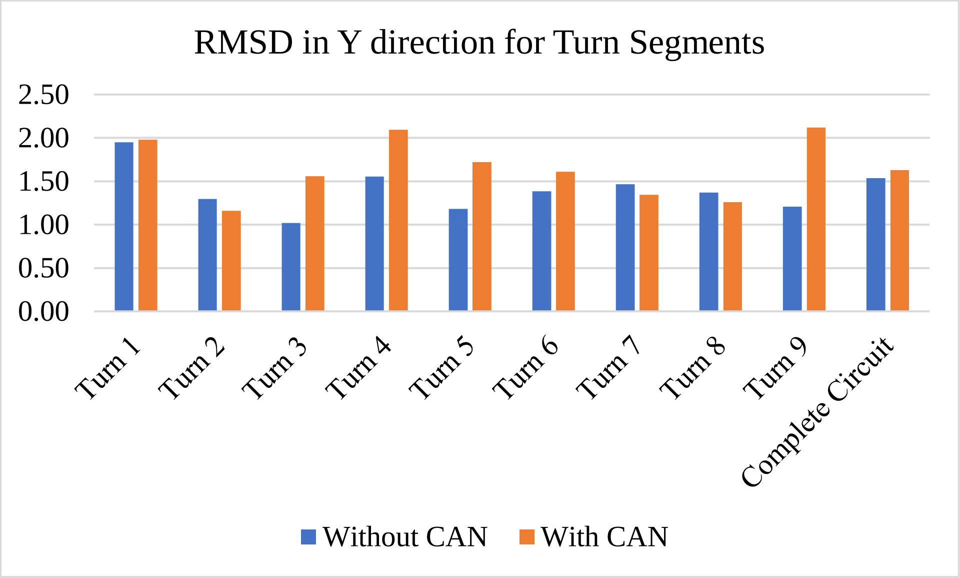


Figure 6.47 RMSD comparison of Y for turn segments dataset 3.

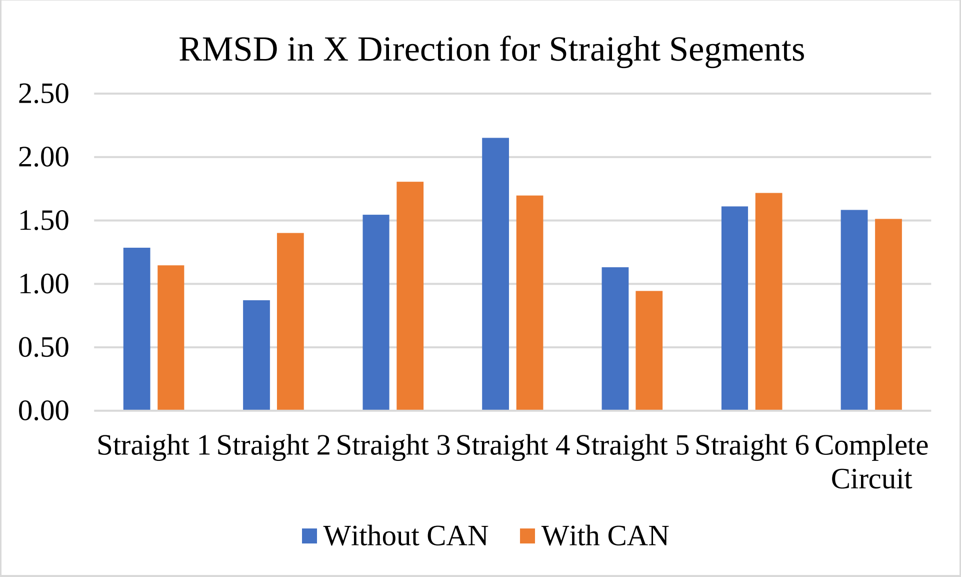


Figure 6.48 RMSD comparison of X for straight segments dataset 3.

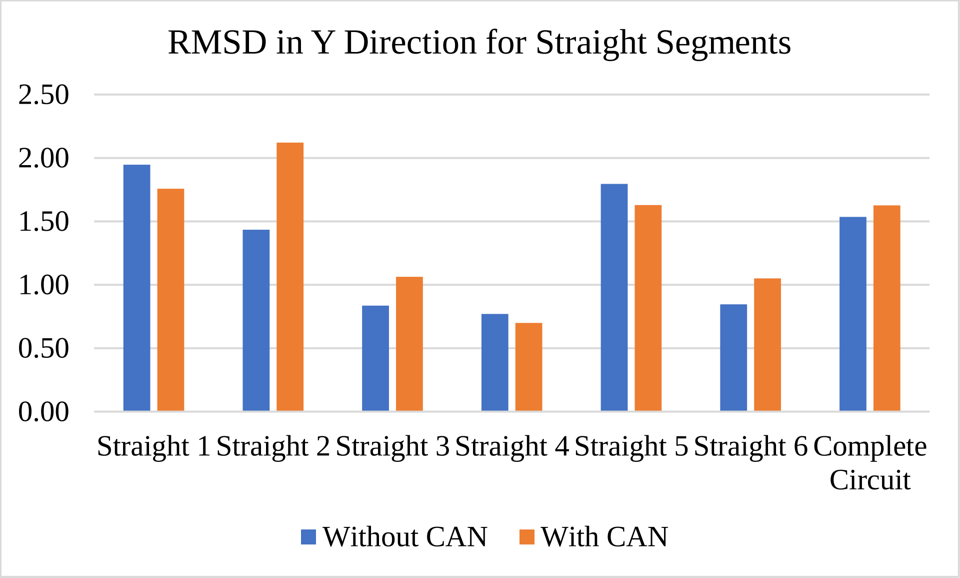


Figure 6.49 RMSD comparison of Y for straight segments dataset 3.

6.3.2.4 Overall Comparison

Table 6.1 RMSD comparison between filters.

	RMSD Comparison with GPS				**RMSD Between EKFNS and EKFCAN**	
	EKFNS		**EKFCAN**			
Dataset	**X Direction**	**Y Direction**	**X Direction**	**Y Direction**	**X Direction**	**Y Direction**
Dataset 1	1.49	1.44	1.59	1.52	1.48	1.54
Dataset 2	1.86	1.80	2.02	1.98	1.70	1.61
Dataset 3	1.58	1.53	1.51	1.63	1.57	1.64

Table 6.1 represents the RMSD between EKFNS and GPS, EKFCAN and GPS, and EKFNS and EKFCAN. The RMSDs in the comparison show how different the filters' trajectories are compared to the GPS values. Comparing the RMSD of the filters with the GPS in dataset 1, the deviation seems to be about 1.5 meters, but the EKFCAN filter has the RMSD values more than the RMSD values of EKFNS. The same trend appears to be shown in predictions from dataset 2, but both filters' RMSD values are highest in this dataset. These values can probably be attributed to the fact that dataset 2 has the lowest number of straight segments and the highest number of curve segments, including the three circle segments. dataset 3 has RMSD values slightly better than dataset 1. It must be observed that RMSD values from all 3 datasets between EKFCAN and GPS are higher than RMSD values between EKFNS and GPS except for the value in dataset 3 X direction, EKFCAN has a lower value than EKFNS.

6.3.2.5 Cost Benefit Analysis

The cost function analysis is a descriptive analysis where the cost of the solution to the value of the solution can be analyzed.

$$Cost\ Function = \frac{Value\ of\ the\ system\ -\ Cost\ of\ the\ System}{Cost\ of\ the\ System}$$

The cost function is a simple indicator of the goodness of the solutions used for the presented problems. To perform the cost benefit analysis scores were given to a certain set of attributes which either fall under the cost of the system of the value of the system. The scores were given between one and ten to represent the goodness of the system with regard to the attribute. Figure 6.50 shows an illustration of the cost function. The attributes listed under the cost of the system are cost of the solution, computational complexity and response time. The greater the number, the solution has better cost benefit to the solution provided. The cost of solution is scored zero if the implementation of the system is the least expensive and ten if the implementation of the system is the most expensive. Likewise, the system with least computational complexity is scored zero and the most computationally complex system will be scored a ten. The response time attribute refers to the availability of the system to provide solution, a slow system will be scored higher than a system that provides faster solutions.

On the other hand, the value of the system has accuracy increase and availability as its attributes. A ten score on the accuracy attributes to the system's accuracy in centimeters, and zero attribute to the system has no increase. And finally, the availability attribute refers to the availability of the solution. If a solution is always available, it will be scored a ten, and a system with low range or if the system has problems when the vehicle is in urban environments or under bridges scored less.

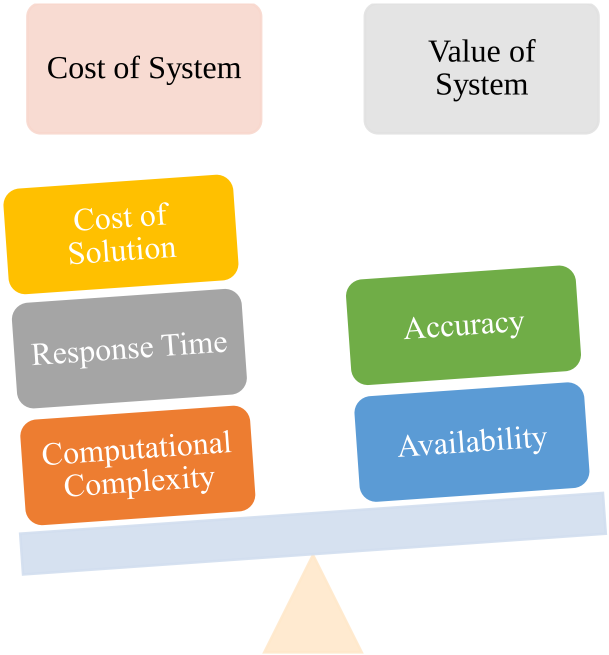


Figure 6.50 Cost benefit analysis illustration

Table 6.2 Cost benefit analysis

Solution	**Cost of solution**	**Computational complexity**	**Accuracy increase**	**Response time**	**Availability**	**Cost function**
GPS	0	0	0	6	5	**-0.2**
GPS + CAN	1	1	5	4	7	**1.0**
GPS + INS	4	1	7	4	7	**0.6**
RTK	7	8	8	5	1	**-0.6**
DGPS	6	6	7	5	3	**-0.4**
LiDAR	9	9	9	5	9	**-0.2**
Vision	8	9	8	5	8	**-0.3**

Table 6.2 shows the score for the solutions of the system. In the cost of the solution the GPS and GPS + CAN solutions have the lowest score is because it is being assumed for all the systems the vehicle has a GPS receiver and for the CAN solution the cost of the system is to add

a communication medium between the OBD II port and the GPS receiver. The highest scores for cost of the systems are LiDAR and Vision as the cost of the sensors are very expensive.

In the computational complexity attribute again LiDAR and Vision solutions are scored the highest because they need the most computation power to implement algorithms such as computer vision. Both CAN and INS solution have the score of one because they use the same data fusion techniques to predict the states from sensors.

The response time attribute shows the speed of the solution, as GPS is the baseline and the speed of the GPS is maximum 10 Hz the solution cannot produce any faster response time. With modern computational power the LiDAR and Vision will be able to produce predictions faster than 10 Hz. The CAN and INS sensors can produce fastest response time as they rely on the speed of the sensors, which are usually better than 10 Hz.

In the accuracy increase attribute, the highest accuracy increase will be provided by the LiDAR, Vision and RTK solutions, evidently proven in the research CAN provides better accuracies than the GPS and scores moderately as it cannot provide accuracies LiDAR, Vison or RTK provide.

Finally, the availability of the solution score for the GPS is moderate because of the limitations in urban environment provided by the bridges and buildings. The same limitations are applied to the INS, CAN, RTK and DGPS because the solutions primarily depend on the availability of GPS. Finally Vision and LiDAR solutions have the highest availability because the solutions have the capability to sense the environment.

Looking at the cost function, it can be observed that CAN solution has the best value and the INS solution is the next best solution. This result is expected because of the inexpensive CAN solution and the accuracy it provides.

Chapter 7: Conclusions and Future Work

The research aimed to contribute with a novel solution based on integrating in-vehicle sensors to CV technologies in order to improve localization accuracy following a Systems Engineering approach. The design approach was achieved by implementing the CAN sensors available at the OBD II port and integrating it with the OBU of the CV technology. Another contribution is made by presenting the fusion algorithms to implement in the form of software upgrade on the OBU to adopt the in-vehicle sensors. The Python based research platform developed to simulate the trajectories can be used by other researchers to analyze localization research topics such as the fusion of other vehicle sensors with the GPS.

The research starts with the discussion about the current state of the CV technologies, followed with the advantage of CV technologies, and then highlight the application rich safety and mobility solution space. A discussion about the current pilot projects that have been deployed to test the effectiveness of the CV applications has been provided. Then, the dissertation documents the dependence of the CV technology applications of the GPS services. Illustrations on the hindrance of performance of CV applications such as FCW, EBW, Queue Warning, Blind Spot Warning to name a few, were made in detail due to current accuracy limitations provided by GPS presently.

A discussion on the accuracy errors posted by the GPS is made, and the reasons for the errors are listed. The state-of-the-art localization techniques such as DGPS, RTK, GPS+INS, LiDAR, Visual Localization are presented with the advantages and the limitations pertaining to each solution. The key limitations of the above-mentioned techniques make them hard to adapt in

CV technology because of huge infrastructure costs or the range limitations of the state-of-the-art solutions.

A brief introduction to the CAN network is made, and the network structure is discussed. The vehicle sensors data that are available on the CAN network is presented, the broadcasting nature of the network where all the nodes have information traveling on the CAN bus is discussed. Later the simplicity of the hardware requirements to convert a non-CV to make it a CV is also provided. The SAE standard for diagnostics by standardizing the OBD II port on every car manufactured after 1996 is mentioned, and the availability of the CAN network data for a plug and use is explained. System integration of the CV hardware to the CAN network via the OBD II port is proposed and illustrated.

A basic kinematic approach to predict the next state of the system is explained by adopting the Kinematic Bicycle Model, which uses the position as the inputs. To adopt more input sensors another kinematic model has been provided called the CTCV model. The CTCV model, in addition to the position information, also includes information from the vehicle speed sensor and the turn rate from the steering wheel sensor.

The A2D2 has provided with three different datasets with data obtained from GPS that provide vehicle position information, and CAN data that provide data collected from in-vehicle speed sensors and the turn rates from the steering wheel sensors.

Two EKF filters are used to predict the trajectories of the datasets. The first filter EKFNS is used to predict the trajectory by using the Kinematic Bicycle Model considering only the input from the position sensors, i.e., the GPS and predicting the position, heading, and speed of the vehicle, and the second filter EKFCAN uses the CTCV model to include the information from not only the position information from the GPS but also the CAN data which provided the information from the in-vehicle speed sensor and the steering wheel sensor.

Detailed analysis on the trajectory comparisons is made, and also the deviation of predicted trajectories by EKFNS and EKFCAN compared to GPS are evaluated in detail. In addition, the deviation of EKFNS with EKFCAN is also evaluated. In the comparison of the predicted trajectories, the robustness of the in-vehicle sensor data fused with the GPS shows a improvement of the localization accuracy. In the discussion on the predictions obtained by the EKFCAN and the EKFNS algorithms on the vehicle heading, the prediction made by the EKFCAN algorithm shows a reduction in the heading error since the algorithm fused the turn rate sensor data provided by the CAN and the GPS data. While comparing the heading errors, another observation that was made is the smooth transition of the vehicle's trajectory in the predictions made by the EKFCAN algorithm when compared to the trajectory of predictions made by the EKFNS algorithm.

In addition, a comparison of the speed predictions made by the filters is made. Here the errors in the predictions made by EKFNS are discussed in detail. The speed predictions made by EKFNS provides some negative speed values, which cannot be possible for a vehicle in motion. The elimination of these errors in the prediction of speed EKFCAN filter shows evidence of the robustness of the CAN GPS fusion technique.

A general trajectory analysis of the predicted trajectories is made, and an illustration of the predictions made by EKFNS and EKFCAN algorithms around the roundabouts is explained. Here the inability of the EKFNS filter to converge to the changing heading of the vehicle is demonstrated. Also, explaining the segments where there are turns, the inability of the EKFNS is shown, and the capability of the EKFCAN filter is shown, and how the trajectory predicted by the EKFCAN resembles the motion of a car. To support this statement, simulation results shown from few segments of the datasets where the vehicle comes to a halt, and the GPS data jumps back and forth, creating huge errors in the predictions made by the EKFNS algorithm Whereas the predictions made by EKFCAN do not jump back and forth because the algorithm detects the vehicle has come to a halt by processing the data from the speed sensor.

Another important conclusion that has been made is when comparing trajectories from localization predictions using the EKFNS algorithm, i.e., just using the GPS data, which shows that the trajectory of the vehicle negotiating the turn always seems to be very sudden. This is due that the GPS trajectories assume the vehicle is still in straight motion, and it takes a second or two to observe the shift in the trajectory, whereas the motion predicted by the EKFCAN algorithm obtaining input from the turn rate sensor detects the change in heading angle and appropriately predicts the shift in trajectory.

Since the evidence has shown the robustness of the EKFCAN algorithm predictions, evaluation of the deviations of predictions made by EKFCAN algorithm with the GPS and predictions made by EKFNS with the GPS are calculated and analyzed using the RMSD. The difference in both algorithms shows a deviation of predictions from just the GPS data of about 1.5 meters. The difference in RMSD values few of the segments of each dataset has been explained for a better understanding of the deviation of predictions from the EKFNS algorithm which has a higher value of predictions made by the EKFCAN algorithm. This was mostly in the cases where the turn segments have a prior straight segment and where the predictions form the EKFNS algorithm take more time to converge to the true heading still showing evidence that predictions made by the EKFCAN algorithm better performed in the situation provided.

From the results, it can be concluded that the addition of the in-vehicle sensors has positively impacted the trajectory predictions of the vehicle, seen by the RMSD values of predictions made by the EKFCAN algorithm which has higher values than the RMSD of predictions made by the EKFNS algorithm. The RMSD values complemented with the detailed analysis of the trajectory comparison help to prove the robustness of the EKFCAN algorithm. The adaptive nature of the EKFCAN algorithm has helped to predict the route and made the predictions smooth with respect to the GPS calculated trajectories.

With the simplicity of hardware implementation of the CV technology and the readiness of the in-vehicle sensors in modern cars, the CAN based in-vehicle sensor data available in the form of a “plug and access” data shows a very capable solution to the current limitations of the CV technology. With the presence of the OBU of the CV, the above algorithmic approach can be integrated with CV vehicles with the addition of a communication medium between the OBD II and OBU. This will allow for a wide spectrum millions of vehicles to be able to actively adapt the CV technologies. With an accuracy difference of about 1.5 meters with just access to the CAN network, the research indicates that the proposed solution is solid approach to CV accuracy localization implementation. This can further impact reliability of safety applications and mobility applications in the V2V and V2I spectrum. Also, planning algorithms can also be built if the accuracies of the positions have increased.

7.1 Future Work

The major limitation of the proposed system is the dependency of the vehicle to first initialize the position. The algorithms are designed to simulate scenarios where the initial position of the system is potentially unknown and/or unreliable. This has been presented as a challenge when the algorithms are assuming the initial position of the vehicle uncertain, and it takes the filter a few steps to converge to the true trajectory of the vehicle. Another limitation of the present system is that the sampling of the speed of the sensors varies within the system. For example, in the dataset the speed sensor has a sampling rate of the speed of 50 Hz, while the turn rate sensor has a sampling rate of 200 Hz. The limitation that is pertained to the problem is that the system can only predict at the lower sensor sampling frequency.

The OBD II port presents another limitation. Some manufacturers have the diagnostics port encrypted. Each vehicle type and model may have a different set of codes for accessing the CAN data. This is more a policy limitation where the policymakers can standardize the automakers to provide appropriate encryption keys and code information to the CV technology adaptors to integrate the technology into every car. Another design limitation is if the OBD II port is occupied, the diagnostics of the vehicle cannot be made, this can be solved by designing a multiport plug to the OBU.

Also, for future work, is to incorporate the Dynamics model of the vehicle, such as integrating the tire slip of the vehicle or integrating any other relevant information available on the CAN network and evaluating the performance change in the system. Potential work on adapting the network triangulation or adopting loop detectors to refresh the correct position of the vehicle could also change the performance of the algorithms. Research on methods to increase the initial reliability of the location data could be a key progression in future research.

Though data is available to evaluate the applications of the CV Pilot programs, the GPS sampling of the data in the CV Pilot program is 1 Hz. The system also provides speed information gathered from the GPS data. Research on GPS outages and the performance of the algorithm will be a good research opportunity, however with the data provided by the CV Pilot project, it is hard to evaluate since there is no data available from the vehicles when the CV has GPS outages.

Further, with the advancements in edge computing, future research could evaluate the effectiveness of integrating other active and passive ego sensors.

References

[1] F. Van Diggelen and P. Enge, “The world’s first GPS MOOC and worldwide laboratory using smartphones,” in *Proceedings of the 28th international technical meeting of the satellite division of the institute of navigation (ION GNSS+ 2015)*, 2015, pp. 361–369.

[2] “GPS Accuracy.” https://www.gps.gov/systems/gps/performance/accuracy/ (accessed Oct. 04, 2021).

[3] C. Tingvall and N. Haworth, “Vision Zero-An ethical approach to safety and mobility,” 1999.

[4] B. McQueen, *Big data analytics for connected vehicles and smart cities*. Artech House, 2017.

[5] “Transportation Working Group,” *INCOSE - International Council on Systems Engineering*. https://www.incose.org/incose-member-resources/working-groups/Application/transportation (accessed Oct. 17, 2021).

[6] H. Qin, H. Li, and X. Zhao, “Development status of domestic and foreign smart city,” *Glob. Presence*, vol. 9, pp. 50–52, 2010.

[7] L. Y. H. Rongxu, “About the Sensing Layer in Internet of Things [J],” *Comput. Study*, vol. 5, 2010.

[8] A. Vasili and W. A. Moreno, “Applications and Trends in Connected Vehicles: Debates and Conclusions,” in *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*, 2019, pp. 1–5.

[9] T. Shelton, M. Zook, and A. Wiig, “The ‘actually existing smart city,’” *Cambridge J. Reg. Econ. Soc.*, vol. 8, no. 1, pp. 13–25, 2015.

[10] H. T. Mouftah, M. Erol-Kantarci, and M. H. Rehmani, *Transportation and power grid in smart cities: communication networks and services*. John Wiley & Sons, 2018.

[11] K. Su, J. Li, and H. Fu, “Smart city and the applications,” in *2011 international conference on electronics, communications and control (ICECC)*, 2011, pp. 1028–1031.

[12] D. Schrank, T. Lomax, and B. Eisele, “2012 urban mobility report,” *Texas Transp. Institute,[ONLINE]. Available http//mobility. tamu. edu/ums/report*, 2012.

[13] C. Olaverri-Monreal, P. Gomes, R. Fernandes, F. Vieira, and M. Ferreira, “The See-Through System: A VANET-enabled assistant for overtaking maneuvers,” in *2010 IEEE intelligent vehicles symposium*, 2010, pp. 123–128.

[14] J. Anda, J. LeBrun, D. Ghosal, C.-N. Chuah, and M. Zhang, “VGrid: vehicular adhoc networking and computing grid for intelligent traffic control,” in *2005 IEEE 61st Vehicular Technology Conference*, 2005, vol. 5, pp. 2905–2909.

[15] N. Liu, “Internet of Vehicles: Your next connection,” *Huawei WinWin*, vol. 11, pp. 23–28, 2011.

[16] T. H. Luan, X. S. Shen, and F. Bai, “Integrity-oriented content transmission in highway vehicular ad hoc networks,” in *2013 Proceedings IEEE INFOCOM*, 2013, pp. 2562–2570.

[17] F. Bai and B. Krishnamachari, “Exploiting the wisdom of the crowd: localized, distributed information-centric VANETs [Topics in Automotive Networking],” *IEEE Commun. Mag.*, vol. 48, no. 5, pp. 138–146, 2010.

[18] H. Hartenstein and L. P. Laberteaux, “A tutorial survey on vehicular ad hoc networks,” *IEEE Commun. Mag.*, vol. 46, no. 6, pp. 164–171, 2008.

[19] N. S. Nafi and J. Y. Khan, “A VANET based intelligent road traffic signalling system,” in *Australasian Telecommunication Networks and Applications Conference (ATNAC) 2012*, 2012, pp. 1–6.

[20] Z. H. Mir and F. Filali, "LTE and IEEE 802.11 p for vehicular networking: a performance evaluation," *EURASIP J. Wirel. Commun. Netw.*, vol. 2014, no. 1, pp. 1–15, 2014.

[21] J. B. Kenney, "Dedicated short-range communications (DSRC) standards in the United States," *Proc. IEEE*, vol. 99, no. 7, pp. 1162–1182, 2011.

[22] T. Jiang, H.-H. Chen, H.-C. Wu, and Y. Yi, "Channel modeling and inter-carrier interference analysis for V2V communication systems in frequency-dispersive channels," *Mob. Networks Appl.*, vol. 15, no. 1, pp. 4–12, 2010.

[23] X. Wu *et al.*, "Vehicular communications using DSRC: challenges, enhancements, and evolution," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 9, pp. 399–408, 2013.

[24] F. Bai, D. D. Stancil, and H. Krishnan, "Toward understanding characteristics of dedicated short range communications (DSRC) from a perspective of vehicular network engineers," in *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, 2010, pp. 329–340.

[25] V. Taliwal, D. Jiang, H. Mangold, C. Chen, and R. Sengupta, "Empirical determination of channel characteristics for DSRC vehicle-to-vehicle communication," in *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, 2004, p. 88.

[26] J. Yin *et al.*, "Performance evaluation of safety applications over DSRC vehicular ad hoc networks," in *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, 2004, pp. 1–9.

[27] S. Bharati and W. Zhuang, "CAH-MAC: cooperative ADHOC MAC for vehicular networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 9, pp. 470–479, 2013.

[28] H. A. Omar, W. Zhuang, and L. Li, "VeMAC: A TDMA-based MAC protocol for reliable broadcast in VANETs," *IEEE Trans. Mob. Comput.*, vol. 12, no. 9, pp. 1724–1736, 2012.

[29] N. Lu, Y. Ji, F. Liu, and X. Wang, "A dedicated multi-channel MAC protocol design for VANET with adaptive broadcasting," in *2010 IEEE Wireless Communication and Networking Conference*, 2010, pp. 1–6.

[30] F. Yu and S. Biswas, “Self-configuring TDMA protocols for enhancing vehicle safety with DSRC based vehicle-to-vehicle communications,” *IEEE J. Sel. areas Commun.*, vol. 25, no. 8, pp. 1526–1537, 2007.

[31] P. Papadimitratos, A. De La Fortelle, K. Evenssen, R. Brignolo, and S. Cosenza, “Vehicular communication systems: Enabling technologies, applications, and future outlook on intelligent transportation,” *IEEE Commun. Mag.*, vol. 47, no. 11, pp. 84–95, 2009.

[32] E. G. Strom, “On medium access and physical layer standards for cooperative intelligent transport systems in Europe,” *Proc. IEEE*, vol. 99, no. 7, pp. 1183–1188, 2011.

[33] S. Sesia, I. Toufik, and M. Baker, *LTE-the UMTS long term evolution: from theory to practice*. John Wiley & Sons, 2011.

[34] G. Araniti, C. Campolo, M. Condoluci, A. Iera, and A. Molinaro, “LTE for vehicular networking: a survey,” *IEEE Commun. Mag.*, vol. 51, no. 5, pp. 148–157, 2013.

[35] F. S. Chaves *et al.*, “LTE UL power control for the improvement of LTE/Wi-Fi coexistence,” in *2013 IEEE 78th Vehicular Technology Conference (VTC Fall),* 2013, pp. 1–6.

[36] “Long Term Evolution Protocol Overview.” chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/viewer.html?pdfurl=https%3A%2F%2Fwww.nxp.com%2Fdocs%2Fen%2Fwhite-paper%2FLTEPTCLOVWWP.pdf&clen=438441&chunk=true (accessed Oct. 08, 2021).

[37] “LTE Protocol Stack Layers.” https://www.tutorialspoint.com/lte/lte_protocol_stack_layers.htm.

[38] J. E. Siegel, D. C. Erb, and S. E. Sarma, “A survey of the connected vehicle landscape—Architectures, enabling technologies, applications, and development areas,” *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 8, pp. 2391–2406, 2017.

[39] J. Contreras-Castillo, S. Zeadally, and J. A. Guerrero-Ibañez, “Internet of vehicles: architecture, protocols, and security,” *IEEE internet things J.*, vol. 5, no. 5, pp. 3701–3709, 2017.

[40] Z. Tian, X. Gao, S. Su, and J. Qiu, “Vcash: a novel reputation framework for identifying denial of traffic service in internet of connected vehicles,” *IEEE Internet Things J.*, vol. 7, no. 5, pp. 3901–3909, 2019.

[41] S. A. A. Shah, E. Ahmed, M. Imran, and S. Zeadally, “5G for vehicular communications,” *IEEE Commun. Mag.*, vol. 56, no. 1, pp. 111–117, 2018.

[42] B. Jo, M. J. Piran, D. Lee, and D. Y. Suh, “Efficient computation offloading in mobile cloud computing for video streaming over 5G,” *Comput. Mater. Contin.*, vol. 61, no. 1, pp. 439–463, 2019.

[43] C. Zhao, T. Wang, and A. Yang, “A heterogeneous virtual machines resource allocation scheme in slices architecture of 5g edge datacenter,” *Comput. Mater. Contin*, vol. 61, pp. 423–437, 2019.

[44] J. Chen *et al.*, “Service-oriented dynamic connection management for software-defined internet of vehicles,” *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 10, pp. 2826–2837, 2017.

[45] A. Botta, W. De Donato, V. Persico, and A. Pescapé, “Integration of cloud computing and internet of things: a survey,” *Futur. Gener. Comput. Syst.*, vol. 56, pp. 684–700, 2016.

[46] G. Premsankar, M. Di Francesco, and T. Taleb, “Edge computing for the Internet of Things: A case study,” *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1275–1284, 2018.

[47] J. Pan and J. McElhannon, “Future edge cloud and edge computing for internet of things applications,” *IEEE Internet Things J.*, vol. 5, no. 1, pp. 439–449, 2017.

[48] “CV Deployment Program Connected Vehicle Applications.” https://www.its.dot.gov/pilots/cv_pilot_apps.htm (accessed Oct. 09, 2021).

[49] F. Ahmed-Zaid *et al.*, “Vehicle safety communications–applications (vsc-a) final report,” 2011.

[50] M. Burt, R. E. Zimmer, G. J. Zink, D. A. Valentine, and W. J. Knox Jr, “Transit safety retrofit package development: architecture and design specifications.,” United States. Department of Transportation. Intelligent Transportation, 2014.

[51] D. R. Stephens, T. J. Timcho, E. Young, R. A. Klein, and J. L. Schroeder, “Accelerated Vehicle-to-Infrastructure (V2I) Safety Applications: Concept of Operations Document [2012-05].” [Online]. Available: https://rosap.ntl.bts.gov/view/dot/26496.

[52] M. Burt, R. E. Zimmer, G. J. Zink, D. A. Valentine, and W. J. J. Knox, “Transit safety retrofit package development : TRP concept of operations.” [Online]. Available: https://rosap.ntl.bts.gov/view/dot/3453.

[53] D. R. Stephens, T. J. Timcho, R. A. Klein, and J. L. Schroeder, “Accelerated Vehicle-to-Infrastructure (V2I) Safety Applications: Concept of Operations Document [2013-03].” [Online]. Available: https://rosap.ntl.bts.gov/view/dot/26500.

[54] K. Ahn, H. A. Rakha, K. Kang, and G. Vadakpat, “Multimodal intelligent traffic signal system simulation model development and assessment,” *Transp. Res. Rec.*, vol. 2558, no. 1, pp. 92–102, 2016.

[55] H. Mahmassani, H. Rakha, E. Hubbard, and D. Lukasik, “Concept Development and Needs Identification for Intelligent Network Flow Optimization (INFLO) : Concept of Operations.” [Online]. Available: https://rosap.ntl.bts.gov/view/dot/3318.

[56] B. M. Institute, Ed., “Response, Emergency Staging, Communications, Uniform Management, and Evacuation (R.E.S.C.U.M.E.): Concept of Operations.” doi: https://doi.org/10.21949/1503776.

[57] S. Mishra *et al.*, “Integrated Dynamic Transit Operations (IDTO) concept of operations.” [Online]. Available: https://rosap.ntl.bts.gov/view/dot/3451.

[58] J. D. Schneeberger, D. Hicks, R. Glassco, and L. Drumwright, “Applications for the environment : real-time information synthesis (AERIS) eco-signal operations : operational concept.” [Online]. Available: https://rosap.ntl.bts.gov/view/dot/3419.

[59] J. D. Schneeberger, D. Hicks, R. Glassco, and L. Drumwright, “Applications for the environment : real-time information synthesis (AERIS). Eco-lanes : operational concept.” [Online]. Available: https://rosap.ntl.bts.gov/view/dot/3439.

[60] T. H. E. Authority, Ed., “Connected Vehicle Pilot Deployment Program Phase 2, System Architecture Document Tampa (THEA).” [Online]. Available: https://rosap.ntl.bts.gov/view/dot/42557.

[61] D. Pape, D. Stephens, C. Stanley, K. Opie, D. Benevelli, and R. Rausch, “Connected Vehicle Pilot Deployment Program phase 1: application deployment plan: New York City: final application deployment plan.,” United States. Joint Program Office for Intelligent Transportation Systems, 2016.

[62] F. Kitchener *et al.*, “Connected Vehicle Pilot Deployment Program Phase 2, Performance Measurement and Evaluation Support Plan-ICF/Wyoming.” [Online]. Available: https://rosap.ntl.bts.gov/view/dot/31400.

[63] F. Kitchener *et al.*, “Connected Vehicle Pilot Deployment Program : Phase 2 Final System Performance Report, Baseline Conditions – WYDOT CV Pilot.” [Online]. Available: https://rosap.ntl.bts.gov/view/dot/36646.

[64] R. Bennett, R. Marsters, T. Szymkowski, and K. N. Balke, “Connected Vehicle Pilot Deployment Independent Evaluation: Mobility, Environmental, and Public Agency Efficiency (MEP) Refined Evaluation Plan-Wyoming,” United States. Department of Transportation. Intelligent Transportation, 2018.

[65] R. French, “Method and apparatus employing automatic route control system.” Google Patents, Oct. 29, 1974.

[66] C. B. Harris and E. J. Krakiwsky, “AVLN 2000 automatic vehicle location and navigation system,” in *Conference Record of papers presented at the First Vehicle Navigation and Information Systems Conference (VNIS’89)*, 1989, pp. 126–130.

[67] S. K. Honey, W. B. Zavoli, K. A. Milnes, A. C. Phillips, M. S. White, and G. E. Loughmiller, “Vehicle navigational sytem and method,” *United States Am. Pat.*, vol. 4796191, no. 3, 1989.

[68] S. K. Honey and W. B. Zavoli, “A novel approach to automotive navigation and map display,” *IEEE Trans. Ind. Electron.*, no. 1, pp. 40–43, 1987.

[69] H. Oshizawa and W. C. Collier, “Description and Performance of Navmate, an In-Vehicle Route Guidance System,” in *1990 American Control Conference*, 1990, pp. 782–787.

[70] J. Tanaka, K. Hirano, H. Nobuta, T. Itoh, and S. Tsunoda, “Navigation system with map-matching method,” SAE Technical Paper, 1990.

[71] F. Morisue and K. Ikeda, “Evaluation of map-matching techniques,” in *Conference Record of papers presented at the First Vehicle Navigation and Information Systems Conference (VNIS’89)*, 1989, pp. 23–28.

[72] H. Yano, Y. Hirasa, T. Tokunaga, M. Hariguchi, and F. Ueda, “Onboard navigation system operating via GPS,” *Automob. Technol.*, vol. 45, no. 2, pp. 39–45, 1991.

[73] Y. Matsuda, Y. Fujita, and Y. Kobayashi, “Man-Machine Interface and the Control Software for Automobile Navigation System,” SAE Technical Paper, 1991.

[74] I. Skog and P. Handel, “In-car positioning and navigation technologies—A survey,” *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 1, pp. 4–21, 2009.

[75] J. M. Rowell, “Applying map databases to advanced navigation and driver assistance systems,” *J. Navig.*, vol. 54, no. 3, pp. 355–363, 2001.

[76] J. P. Löwenau, P. J. T. Venhovens, and J. H. Bernasch, “Advanced vehicle navigation applied in the BMW real time light simulation,” *J. Navig.*, vol. 53, no. 1, pp. 30–41, 2000.

[77] J. Du, J. Masters, and M. Barth, “Lane-level positioning for in-vehicle navigation and automated vehicle location (AVL) systems,” in *Proceedings. The 7th International IEEE Conference on Intelligent Transportation Systems (IEEE Cat. No. 04TH8749)*, 2004, pp. 35–40.

[78] J. Du and M. Barth, “Bayesian probabilistic vehicle lane matching for link-level in-vehicle navigation,” in *2006 IEEE Intelligent Vehicles Symposium*, 2006, pp. 522–527.

[79] M. Schlingelhof, D. Bétaille, P. Bonnifait, and K. Demaseure, “Advanced positioning technologies for co-operative systems,” *IET Intell. Transp. Syst.*, vol. 2, no. 2, pp. 81–91, 2008.

[80] D. Obradovic, H. Lenz, and M. Schupfner, “Fusion of map and sensor data in a modern car navigation system,” *J. VLSI signal Process. Syst. signal, image video Technol.*, vol. 45, no. 1, pp. 111–122, 2006.

[81] D. Obradovic, H. Lenz, and M. Schupfner, “Fusion of sensor data in Siemens car navigation system,” *IEEE Trans. Veh. Technol.*, vol. 56, no. 1, pp. 43–50, 2007.

[82] J. Marais, M. Berbineau, and M. Heddebaut, “Land mobile GNSS availability and multipath evaluation tool,” *IEEE Trans. Veh. Technol.*, vol. 54, no. 5, pp. 1697–1704, 2005.

[83] S. Godha and M. E. Cannon, “GPS/MEMS INS integrated system for navigation in urban areas,” *Gps Solut.*, vol. 11, no. 3, pp. 193–203, 2007.

[84] B. Boberg, “Robust navigation,” *Swedish J. Mil. Technol.*, no. 3, pp. 23–28, 2005.

[85] J. Huang and H.-S. Tan, “A low-order DGPS-based vehicle positioning system under urban environment,” *IEEE/ASME Trans. mechatronics*, vol. 11, no. 5, pp. 567–575, 2006.

[86] W.-W. Kao, “Integration of GPS and dead-reckoning navigation systems,” in *Vehicle Navigation and Information Systems Conference, 1991*, 1991, vol. 2, pp. 635–643.

[87] “What is GPS?” https://www.geotab.com/blog/what-is-gps/ (accessed Oct. 10, 2021).

[88] C. Jeffrey, *An introduction to GNSS: GPS, GLONASS, Galileo and other global navigation satellite systems*. NovAtel, 2010.

[89] F. Caron, E. Duflos, D. Pomorski, and P. Vanheeghe, “GPS/IMU data fusion using multisensor Kalman filtering: introduction of contextual aspects,” *Inf. fusion*, vol. 7, no. 2, pp. 221–230, 2006.

[90] S. Sukkarieh, E. M. Nebot, and H. F. Durrant-Whyte, “A high integrity IMU/GPS navigation loop for autonomous land vehicle applications,” *IEEE Trans. Robot. Autom.*, vol. 15, no. 3, pp. 572–578, 1999.

[91] R. Toledo-Moreo, M. A. Zamora-Izquierdo, B. Ubeda-Minarro, and A. F. Gómez-Skarmeta, “High-integrity IMM-EKF-based road vehicle navigation with low-cost GPS/SBAS/INS,” *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 3, pp. 491–511, 2007.

[92] D. Huang and H. Leung, “EM-IMM based land-vehicle navigation with GPS/INS,” in *Proceedings. The 7th International IEEE Conference on Intelligent Transportation Systems (IEEE Cat. No. 04TH8749)*, 2004, pp. 624–629.

[93] R. Toledo-Moreo and M. A. Zamora-Izquierdo, “IMM-based lane-change prediction in highways with low-cost GPS/INS,” *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 1, pp. 180–185, 2009.

[94] S. Wu and M. Yang, “Landmark pair based localization for intelligent vehicles using laser radar,” in *2007 IEEE Intelligent Vehicles Symposium*, 2007, pp. 209–214.

[95] T. Weiss, N. Kaempchen, and K. Dietmayer, “Precise ego-localization in urban areas using laserscanner and high accuracy feature maps,” in *IEEE Proceedings. Intelligent Vehicles Symposium, 2005.*, 2005, pp. 284–289.

[96] H. Uchiyama, D. Deguchi, T. Takahashi, I. Ide, and H. Murase, “Ego-localization using streetscape image sequences from in-vehicle cameras,” in *2009 IEEE Intelligent Vehicles Symposium*, 2009, pp. 185–190.

[97] N. Mattern, R. Schubert, and G. Wanielik, “High-accurate vehicle localization using digital maps and coherency images,” in *2010 IEEE intelligent vehicles symposium*, 2010, pp. 462–469.

[98] R. G. García-García, M. A. Sotelo, I. Parra, D. Fernández, and M. Gavilán, “3D visual odometry for GPS navigation assistance,” in *2007 IEEE Intelligent Vehicles Symposium*, 2007, pp. 444–449.

[99] D. Bétaille and R. Toledo-Moreo, “Creating enhanced maps for lane-level vehicle navigation,” *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 4, pp. 786–798, 2010.

[100] Y. Cui and S. S. Ge, “Autonomous vehicle positioning with GPS in urban canyon environments,” *IEEE Trans. Robot. Autom.*, vol. 19, no. 1, pp. 15–25, 2003.

[101] H. Badino, D. Huber, and T. Kanade, “Visual topometric localization,” in *2011 IEEE Intelligent Vehicles Symposium (IV)*, 2011, pp. 794–799.

[102] A. U. Peker, O. Tosun, and T. Acarman, “Particle filter vehicle localization and map-matching using map topology,” in *2011 IEEE Intelligent Vehicles Symposium (IV)*, 2011, pp. 248–253.

[103] V. Popescu, M. Bace, and S. Nedevschi, “Lane identification and ego-vehicle accurate global positioning in intersections,” in *2011 IEEE Intelligent Vehicles Symposium (IV)*, 2011, pp. 870–875.

[104] N. Suganuma and T. Uozumi, “Precise position estimation of autonomous vehicle based on map-matching,” in *2011 IEEE Intelligent Vehicles Symposium (IV)*, 2011, pp. 296–301.

[105] N. Cui, L. Hong, and J. R. Layne, “A comparison of nonlinear filtering approaches with an application to ground target tracking,” *Signal Processing*, vol. 85, no. 8, pp. 1469–1492, 2005.

[106] F. Chausse, J. Laneurit, and R. Chapuis, “Vehicle localization on a digital map using particles filtering,” in *IEEE Proceedings. Intelligent Vehicles Symposium, 2005.*, 2005, pp. 243–248.

[107] GeoSLAM, “What is SLAM?” https://geoslam.com/what-is-slam/ (accessed Oct. 18, 2021).

[108] M. O. A. Aqel, M. H. Marhaban, M. I. Saripan, and N. B. Ismail, “Review of visual odometry: types, approaches, challenges, and applications,” *Springerplus*, vol. 5, no. 1, pp. 1–26, 2016.

[109] A. Noureldin, T. B. Karamat, and J. Georgy, “Fundamentals of inertial navigation, satellite-based positioning and their integration,” 2013.

[110] K. Lingemann, A. Nüchter, J. Hertzberg, and H. Surmann, “High-speed laser localization for mobile robots,” *Rob. Auton. Syst.*, vol. 51, no. 4, pp. 275–296, 2005.

[111] T. Takahashi, “2D localization of outdoor mobile robots using 3D laser range data,” *Robot. Inst. Carnegie Mellon Univ. Pittsburgh, Pennsylvania*, vol. 15213, 2007.

[112] J. Horn and G. Schmidt, “Continuous localization of a mobile robot based on 3D-laser-range-data, predicted sensor images, and dead-reckoning,” *Rob. Auton. Syst.*, vol. 14, no. 2–3, pp. 99–118, 1995.

[113] E. Frontoni, *Vision based mobile robotics: mobile robot localization using vision sensors and active probabilistic approaches*. Lulu. com, 2012.

[114] W. Rone and P. Ben-Tzvi, “Mapping, localization and motion planning in mobile multi-robotic systems,” *Robotica*, vol. 31, no. 1, pp. 1–23, 2013.

[115] J. Campbell, R. Sukthankar, I. Nourbakhsh, and A. Pahwa, “A robust visual odometry and precipice detection system using consumer-grade monocular vision,” in *Proceedings of the 2005 IEEE International Conference on robotics and automation*, 2005, pp. 3421–3427.

[116] R. Gonzalez, F. Rodriguez, J. L. Guzman, C. Pradalier, and R. Siegwart, “Combined visual odometry and visual compass for off-road mobile robots localization,” *Robotica*, vol. 30, no. 6, pp. 865–878, 2012.

[117] D. Hristu-Varsakelis and W. S. Levine, *Handbook of networked and embedded control systems*, no. TK7895. E42. H29 2005. Springer, 2005.

[118] R. Currie, “Development in Car Hacking.” https://www.sans.org/white-papers/36607/ (accessed Oct. 05, 2021).

[119] “CAN lower- and higher-layer protocols.” https://www.can-cia.org/can-knowledge/ (accessed Oct. 06, 2021).

[120] J. A. Cook and J. S. Freudenberg, “Controller area network (CAN),” *EECS*, vol. 461, pp. 1–5, 2007.

[121] R. Bosch, “CAN specification 2.0, Part A and B,” *Robert Bosch GmbH, Ger.*, 1991.

[122] D. Klinedinst and C. King, “On board diagnostics: Risks and vulnerabilities of the connected vehicle,” *CERT Coord. Center, Tech. Rep*, 2016.

[123] S. Simon and B. Dale, “What is Systems Engineering ?” https://www.incose.org/docs/default-source/default-document-library/twg-se101-v11-2014-01-20.pdf?sfvrsn=e6c882c6_4.

[124] “Simple Understanding of Kinematic Bicycle Model.” https://dingyan89.medium.com/simple-understanding-of-kinematic-bicycle-model-81cac6420357.

[125] “CTRV Model A mathematical model to describe the motion of a mobile,” [Online]. Available: https://fjp.at/process models/model/bicycle-model/ctrv-model/.

[126] J. Geyer *et al.*, “A2D2: Audi autonomous driving dataset,” *arXiv Prepr. arXiv2004.06320*, 2020.

[127] “ITS DataHub.” https://www.its.dot.gov/data/ (accessed Oct. 13, 2021).

[128] D. Svensson, “Derivation of the discrete-time constant turn rate and acceleration motion model,” in *2019 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, 2019, pp. 1–5.

Appendix A: Basics of Data Fusion

A.1 Probability

The generalized explanation of probability can be defined as the possibility of an event occurring from the chances of all the events occurring.

$$p(A) = outcomes\ where\ event\ A\ could\ occur/All\ the\ possible\ outcomes$$

Further, when considering two probabilities p(A), p(B), the probability that one of the two will occur is defined as p(A) OR p(B) and called union,

$$p(A\ U\ B\ ) = p(A) + p(B)$$

and the probability of both the events occurring is defined as p(A) AND p(B) and called intersection.

$$p(A \cap B\ ) = p(A)p(B)$$

Finally, conditional probability is defined as the conditional probability of an event A to occur when there is knowledge of the event B to occur prior to the occurrence of event A. it is denoted as p(A|B)

$$p\ (A|B\ ) = p(A)p(A \cap B)/p\ (B)$$

A.1.1 Continuous Random Variables

Random variables are divided into two types of discrete random variables and continuous random variables. In the case of vehicular motion, the typical interest is to understand the randomness of continuous vehicular motion.

A random variable can be defined as a function that can map all the points from a sample space to real numbers. The random variables are usually denoted by capital letters, and since the continuous random variables are functions with respect to time, they can be denoted as follows X(t).

In the case of continuous random variables, the probability of a single discrete event happening is 0; p(A) = 0. The probability can only be calculated of the events that occurred in a set time period. This representation is commonly known as cumulative distribution function:

$$Fx(x) \ = \ p(-\infty, x]$$

Properties of the cumulative density function are:

$$fx(x) = \begin{cases} 0, & x \; tends \; to \; reach - \infty \\ 1, & x \; tends \; to \; reach + \infty \end{cases}$$

The probability density function is the more commonly used method amongst the random variables. It is the derivative of the cumulative distribution function.

$$fx(x) = \frac{d}{dx} Fx(x)$$

It can be observed that the probability of the events over a period a to b can be computed as

$$px[a, b] = \int_{a}^{b} fx(x)dx$$

A.1.2 Mean and Variance for Continuous Variables

The most popular representation of the mean is usually associated with discrete variables where the mean is defined as the average of the numbers.

$$X = \ \frac{X1 + X2 + \cdots + \ XN}{N}$$

Here the more uncommon or the mean of a continuous variable can be defined as, considering n number of outcomes.

$$X = \ \frac{(p1N)x1 + (p2N)x2 + \cdots + \ (pnN)xN}{N}$$

The definition of mean in continuous variables is defined as the Expected value E(X)

$$E(X) = \int_{-\infty}^{\infty} xfx(x)dx$$

The first statistical moment is defined as the expected variable of a random variable. Here we will be using $X^{k,}$ to represent the k^{th} moment.

$$E(\mathrm{Xk}) = \int_{-\infty}^{\infty} x^k fx(x)dx$$

By using the above equations and g(X) = X=E(X) the variance of X

$$X \ = \ E(X2) - E(X)2$$

The magnitude of the variance is the quantity of the 'noise' in the signal. Variance is usually denoted as σ^2, the square root of the variance is called the standard deviation.

A.1.3 Gaussian Distribution

Most of the random variables appear to fall on a similar curve. Under some common conditions, the sum of all variables from any type of distribution will be a normal distribution. The normal distribution with a variance σ^2 the probability density function is given as,

$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} \text{exp } (\frac{-(x-\mu)^2}{2\sigma^2}) \text{,for } x \in \mathbb{R}$$

A.2 Bayes' Rule

Independent Random Variables two variables X and Y are considered independent if the joint probabilities are equal to the product of their probabilities.

$$p(x, y) = p(x)p(y)$$

Marginal Probability is defined as the probability of an event to occur irrespective of the outcomes of other events to occur. The marginal probability of event A to happen is denoted as P(A).

Conditional probability is defined as the probability that an event A will occur, given that event B has occurred. It is commonly denoted as P(A|B).

$$P(A \mid B) \;=\; P(A, B) \;/\; P(B)$$

Joint probability is defined as the probability of two events occurring simultaneously A and B, from two different random variables X and Y. It is denoted as follows P(A and B) or P(A, B).

$$P(A, B) \;=\; P(A \mid B) \;*\; P(B)$$

The above definitions of the probabilities can help us define the basic understanding of data fusion, i.e., the Bayes' theorem.

$$P(A|B) = \frac{P(A\ \cap B)}{P(B)}$$

$$P(A\ \cap\ B) = P(B\ \cap\ A)$$

$$P(A\ \cap B) = P(A|B)P(B) =\ P(B|A)P(A)$$

$$P(A|B) = \frac{P(A)\, P(B|A)}{P(B)}$$

P(A|B) is the conditional probability of A happening when B happens. P(B|A) is the conditional probability of B happening when A happens. In the sense of other probability measurements P(A|B) is called the posterior; it is the calculation made after the event B has occurred. P(A) is called the prior, and it is the initial or known value of A

A.3 State Space Model

Modeling a system with respect to time is called dynamic system modeling. The math used to model dynamic systems are differential equations. Modeling a system equation of a moving vehicle involves questions such as distance traveled when the velocity is known. A simplified equation to encompass these types of equations can be represented as

$$x_t = G_t x_{t-1} + \omega_t$$

the above equation is known as the state equation and is formulated representing the present state x_t representing the states on the system in a column matrix. It is the linear combination of the previous state at t-1 and an additional system noise represented by ω_t. G_t represents the input states' dependency on the previous states.

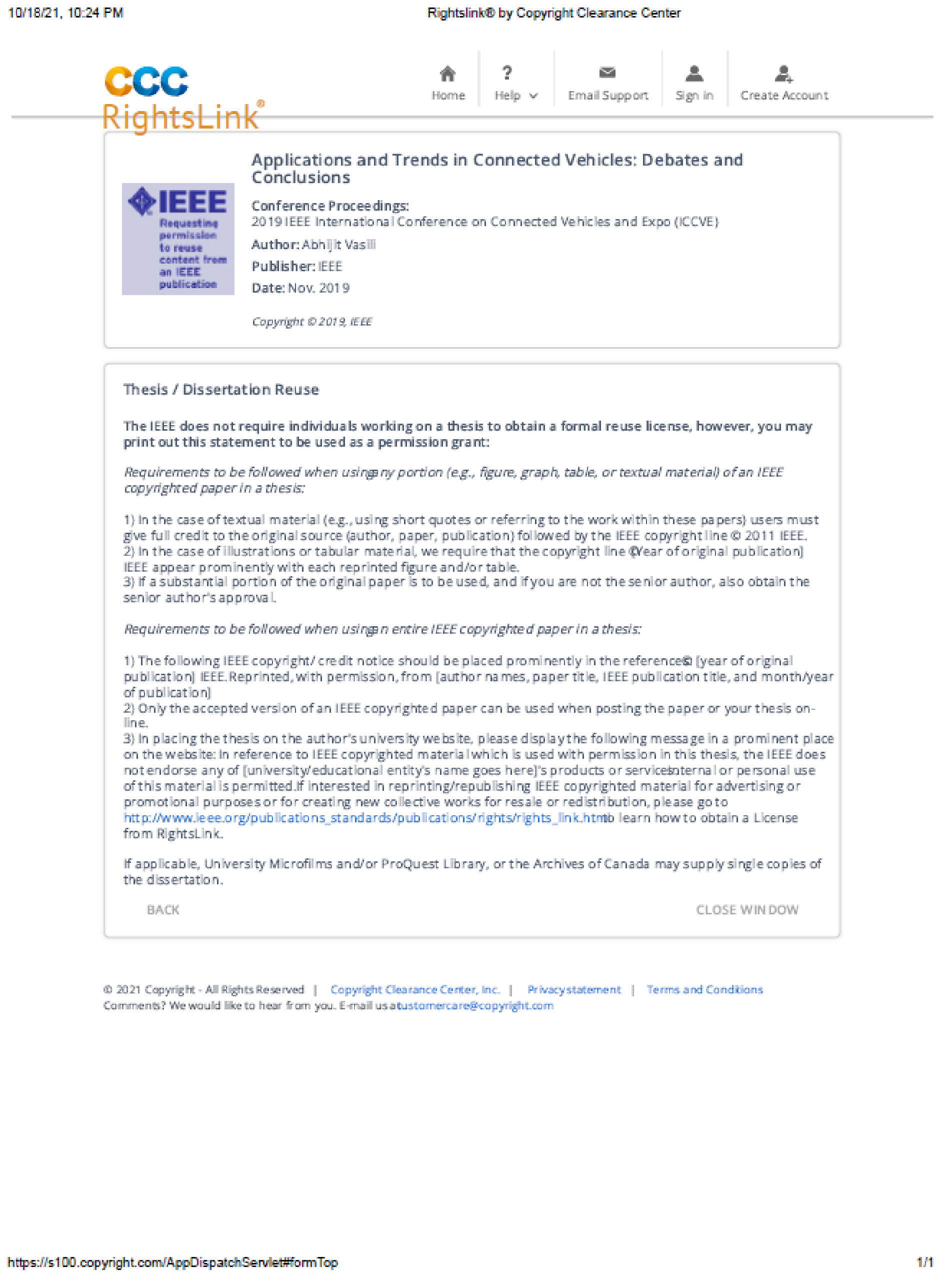
Further, if we consider the observations to be represented by y_t this can be calculated by finding the observation matrix denoted as F_t and v_t is the measurement noise. This equation is called the observation equation**.**

$$y_t = F_t^T x_t + v_t$$

The state equations and the observation can be calculated provided the first state is x_0 is known, and the system noise covariance and the measurement noise covariance are known.

Appendix B: Permission for Chapter 2

The permission for the tables and content used in the Chapter 2.



10/18/21, 10:24 PM

Rightslink® by Copyright Clearance Center

Applications and Trends in Connected Vehicles: Debates and Conclusions

Conference Proceedings:
2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)

Author: Abhijit Vasili

Publisher: IEEE

Date: Nov. 2019

Copyright © 2019, IEEE

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

© 2021 Copyright - All Rights Reserved | Copyright Clearance Center, Inc. | Privacy statement | Terms and Conditions
Comments? We would like to hear from you. E-mail us at customercare@copyright.com

https://s100.copyright.com/AppDispatchServlet#formTop

1/1

Appendix C: Flow of Python Based Framework

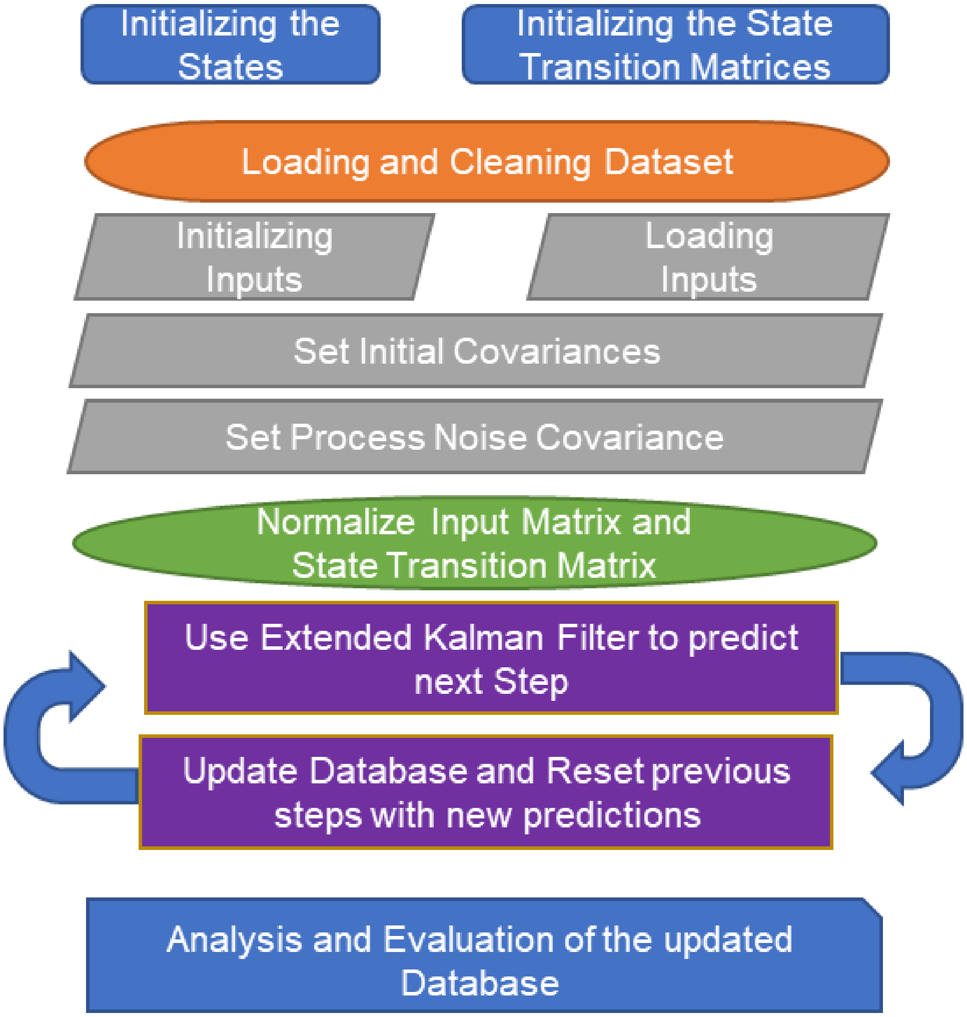

Initializing the States
Initializing the State Transition Matrices
Loading and Cleaning Dataset
Initializing Inputs
Loading Inputs
Set Initial Covariances
Set Process Noise Covariance
Normalize Input Matrix and State Transition Matrix
Use Extended Kalman Filter to predict next Step
Update Database and Reset previous steps with new predictions
Analysis and Evaluation of the updated Database

Figure C.1 Flow diagram for the Python based framework