UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

IN VEHICLE SMARTPHONE BASED POSITION ESTIMATES ON URBAN ROADS FOR LANE DEPARTURES USING ROAD LEVEL GIS INFORMATION

MÉMOIRE PRÉSENTÉ À
L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

COMME EXIGENCE PARTIELLE
DE LA

MAÎTRISE EN INGÉNIERIE

PAR
MOHAMMED ABDUL SAJEED

MAI 2022

UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

MAÎTRISE EN INGÉNIERIE
(CONCENTRATION EN GÉNIE MECANIQUE)

Direction de recherche :

Professeur Sousso Kelouwani	directeur de recherche
Professeur Kodjo Agbossou	codirecteur de recherche

Jury d'évaluation

Professeur Sousso Kelouwani	Évaluateur 1, Directeur de recherche
Professeur Martin Bolduc	Évaluateur 2, Professeur en Génie Mécanique
Professeur Hicham Chaoui	Évaluateur externe

Résumé

La localisation est essentielle pour la fonctionnalité des véhicules autonomes, dont les plus mentionnées sont les systèmes de sécurité d'avertissement de sortie de voie ou Lane Departure Warning Safety Systems (LDWSS). Les LDWSS détectent généralement les écarts de voie dans les environnements urbains en utilisant une variété de techniques de traitement de la vision ou de balayage optique. Ces LDWSS ont certaines limites; par exemple, ils peuvent être gênés par des conditions météorologiques défavorables ou des marquages au sol irréguliers. D'autres LDWSS utilisent un récepteur des systèmes de positionnement (GPS) et des cartes numériques avec une meilleure résolution au niveau de la voie pour augmenter l'efficacité du système, mais qui sont plus complexes et couteux. Pour compenser les pannes de caméra, cette étude propose un système alternatif qui ne dépend pas des marquages au sol mais les données GPS et IMU collectées à partir d'un smartphone installé dans le véhicule pour le localiser sur la route et compare sa trajectoire à une direction de référence de la route sans utiliser aucune carte numérique. Cette technique ne nécessite que des données au niveau par rapport au profil de la route provenant d'une base de données de cartographie numérique commune. De plus, des expériences démontrent que le système proposé est fiable et peut être complété par une caméra pour améliorer les estimations du positionnement du véhicules par rapport à la voie. La validation a été effectuée sur un véhicule électrique : KIA SOUL EV 2017, appartenant à l'Institut de recherche sur l'hydrogène (IRH) de l'Université du Québec à Trois-Rivières (UQTR).

Mots-clés: Systèmes avancés d'aide à la conduite (ADAS), Système d'information géographique (SIG), Systèmes de navigation inertielle, Départ de voie, Filtre de Kalman, estimation de la trajectoire.

Summary

The ego vehicle's location is critical for a variety of autonomous functions, the most notable of which are lane departure warning safety systems (LDWSS). LDWSS typically detect lane departures in urban environments by utilizing a variety of vision processing or optical scanning techniques. These LDWSS have some limitations; for example, they may be impaired by adverse weather conditions or irregular lane markings. Other LDWSS employ a global positioning system (GPS) receiver and digital maps with lane-level resolution to increase system efficiency, but at the expense of added complexity and cost. To compensate for camera failures, this paper proposes an alternate system that is not reliant on lane markings and instead uses GPS and IMU data collected from the vehicle's smartphone to locate the vehicle on the road and compare its trajectory to a reference road direction without requiring any digital maps with lane-level resolution. This technique only requires road-level data from a common digital mapping database. Additionally, experiments demonstrate that the proposed system is reliable and can be supplemented with a camera to improve vehicle estimates at the lane level.

The validation was carried out on an electric vehicle: KIA SOUL EV, belonging to the Hydrogen Research Institute (IRH) of the Université du Québec à Trois-Rivières (UQTR).

Keywords: Advanced Driver Assistance Systems (ADAS), Geographic Information System (GIS), Inertial Navigation systems, Lane departure, Kalman filter, trajectory estimation

Dedications

I dedicate this brief:

To my dear parents Mohammed Abdul Rasheed and Askari begum for all their sacrifices and support throughout my studies.

To my dear brothers, wife and friends for their constant encouragement and moral support.

Acknowledgement

It is with great pleasure that I summarise in a few lines my gratitude and appreciation to those who aided me in completing my thesis, because nothing would have been possible without their efforts and assistance.

I would like to express my gratitude to my research director, Sousso Kelouwani, Professor at the Université du Québec à Trois-Rivières, and my co-director, Kodjo Agbossou, Professor at the Université du Québec à Trois-Rivières, for agreeing to supervise me and providing me with extremely beneficial advice. I'd like to show my appreciation to them for their responsibility, availability, and assistance in finishing this project.

Additionally, I would want to express my appreciation to my mentors Ali Amamou, and Folivi and my colleagues for their recommendations and contributions to this project

Additionally, I would want to express my gratitude to the entire Hydrogen Research Institute team for their hospitality, which allowed me to gain excellent advice and discussion while also learning about various cultures.

Finally, I'd want to express my gratitude to the jury members for their time spent evaluating this work.

Table of Contents

List of figures

List of tables

List of abbreviations

AV:	Autonomous vehicles
ADAS:	Advanced driving assistance system
ACC:	Adaptive cruise control
AASHTO:	American Association of State Highway and Transportation Officials
BSM:	Blind spot monitoring
CCD:	Charging coupling device
CMOS-D:	Complementary metal oxide semiconductor device
DAS:	Driving assistance system
DOF:	Degree of Freedom
DSRC:	Dedicated Short-Range Communications
ECI:	Inertial coordinate system
ECEF:	Earth centered earth fixed
FIR:	Far Infrared system
FMCW:	Frequency modulated continuous wave
FCA:	Forward collision avoidance
GIS:	Geographic Information system
GPS:	Global positioning system
GNSS:	Global navigation satellite systems
Hz:	Hertz
IMU:	Inertial measurement unit
INS:	Inertial navigation system
ITS:	Intelligent transportation system

KF:	Kalman filter
Km:	Kilometer
LDWSS:	Lane departure warning safety systems
LKA:	Lane keeping assistance
Lidar:	Light detection and ranging
LCA:	Lane change assistance
m s2	Meters per second squared
MMSE:	Minimum mean squared error
Min:	Minutes
nm:	Nanometers
NHTSA:	National Highway Traffic administration
NHC:	Non holonomic constraints
OSM:	Open Street map
RQ:	Research questions
Radar:	Radio detection and ranging
SAE:	Society of automobile engineers
Sec:	Seconds
TJA:	Traffic jam assist
UQTR :	Université du Québec à Trois Rivières

Mathematical Foundations of aided inertial navigation

$\mathbf{z}$	True value of the vector quantity $\mathbf{z}$
$\tilde{\mathbf{z}}$	Erroneous or estimated value of $\mathbf{z}$
$\delta\mathbf{z} = \mathbf{z} - \tilde{\mathbf{z}}$	Error vector

$\hat{\mathbf{z}}$	Estimate of $\mathbf{z}$
$\mathbf{I}$	Identity matrix
$[\mathbf{a}\times]$	Skew-symmetric matrix operator, that is given vectors $\mathbf{a}$ and $\mathbf{b}$, $[\mathbf{a}\times]\cdot\mathbf{b}=\mathbf{a}\times\mathbf{b}$
$\mathbf{C}_{ab}$	A matrix that defines rotation from coordinate frame b to coordinate frame a
$\breve{\mathbf{q}}_{ab}$	A quaternion defining rotation from coordinate frame b to coordinate frame a
$\breve{\mathbf{b}}$	A quaternion, whose scalar part is zero, and vector part is equal to the vector b
$\mathbf{f}_x$	Vector of projections of specific force onto x-frame
$\boldsymbol{\omega}_{xy}$	Angular rate of y-frame relative to x-frame in projection onto y-frame
$*$	Quaternion product

Chapitre 1 - **Introduction**

1.1 Context

Peut-être qu'aucune invention n'a autant façonné le XXe siècle que l'automobile. Au quotidien, la majorité des gens utilisent l'automobile comme principal moyen de transport. En effet, plus de 70 millions d'automobiles ont été produites dans le monde en 2018. Étant donné que 87 % de la population active se déplace exclusivement en voiture, les dépenses liées à la voiture constituent la deuxième catégorie de dépenses pour une famille américaine de classe moyenne. Bien que cette invention soit un énorme succès, ses avantages ne sont pas sans prix. Des coûts économiques, environnementaux et sociaux supplémentaires sont associés à la grande mobilisation. De plus, il met en danger la sécurité des passagers en entraînant des incidents dangereux ou mortels.

Selon la National Highway Traffic Safety Administration (NHTSA), l'inattention du conducteur est la première cause d'accidents graves [1]. Cela comprend la fatigue, la somnolence, la distraction et la vitesse. Outre l'erreur du conducteur, des facteurs environnementaux tels que les conditions météorologiques ont été associés aux accidents d'automobile [1]. Le système d'aide à la conduite ou Driving Assistance System (DAS) a été introduit dans les automobiles dans le but d'augmenter l'efficacité et la sécurité de la mobilité grâce à l'incorporation de capteurs inertiels et de coussins gonflables dans le véhicule [2]. Cependant, cette assistance était initialement orientée vers l'utilisateur et donc limitée aux conducteurs et à leurs véhicules, suscitant une préoccupation croissante pour la sécurité des piétons et des véhicules dans toutes les types de conditions météorologiques. Cette préoccupation s'est traduite par des avancées dans l'assistance électronique à la conduite, les

automobiles étant désormais capables d'effectuer certaines manœuvres avec ou sans intervention manuelle. Le système de transport a été considérablement modifié par les systèmes avancés de sécurité d'assistance à la conduite ou Advanced Driving Assistance System (ADAS) et leurs applications [2-4]. Un système ADAS dans une voiture collecte des données dans les directions longitudinale et latérale du véhicule à l'aide d'un module de perception de l'environnement composé de plusieurs capteurs dont le but est de fournir les informations nécessaires pour interpréter les scènes environnantes à proximité du véhicule à l'aide d'un ordinateur central, puis d'activer une alerte ou un freinage basé sur l'analyse d'entrée de capteur. La figure 1 illustre la structure fondamentale d'un véhicule compatible ADAS.

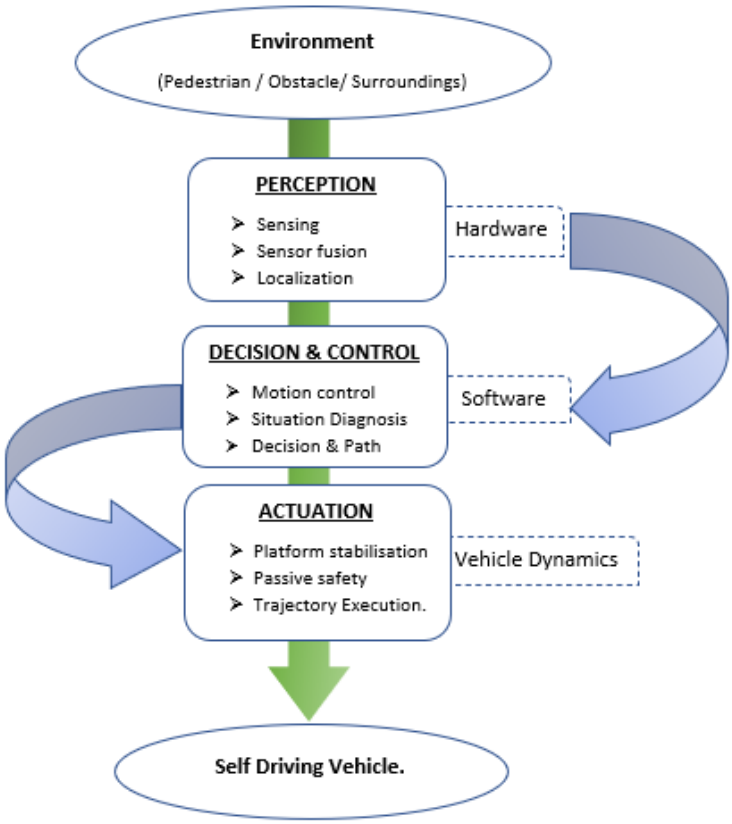


Figure 1. Diagramme de fonctionnement de l'ADAS dans les voitures autonomes

En raison des progrès technologiques, les systèmes de conduite autonome sont devenus un intérêt de recherche dans les universités et l'industrie [5, 6]. En conséquence, la Society of Automobile Engineers (SAE) a défini six niveaux d'automatisation pour ces systèmes de sécurité, allant de 0 (conduite manuelle complète) à 5 (navigation entièrement autonome) [7]. Alors que les véhicules entièrement autonomes (niveau 5) sont encore prévu dans quelques années, les véhicules utilitaires actuels sont équipés de systèmes ADAS classés 2 et 3 dans les niveaux d'autonomie de SAE. Plusieurs applications ADAS de pointe bien connues qui fonctionnent dans le sens longitudinal du véhicule incluent le régulateur de vitesse adaptatif (ACC), l'évitement de collision avant (FCA), la reconnaissance des panneaux routiers et de signalisation et l'assistance aux embouteillages (TJA), tandis que l'assistance à la sécurité latérale populaire comprend le système d'avertissement de sortie de voie (LDWSS) et l'assistance au maintien de voie (LKA) , Assistance au changement de voie (LCA) et Surveillance des angles morts (BSM). La figure 2 illustre une application ADAS courante qui utilise une variété de capteurs pour analyser l'environnement.

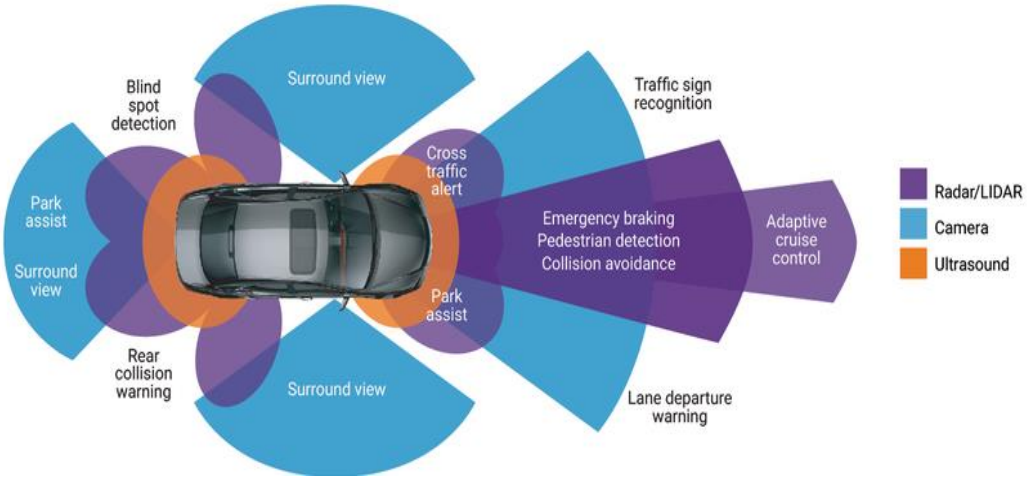


Figure 2 Divers applications et capteurs ADAS

En tant que module essentiel des véhicules autonomes, la détection de voie permet à la voiture de maintenir une position cohérente dans les voies de circulation, ce qui est nécessaire

pour le contrôle et la planification ultérieurs. Maintenir la position du véhicule dans les voies de circulation peut être une tâche fastidieuse et aggravante, en particulier lors de longs trajets urbains où les sorties de voie sont courantes. Selon l'American Association of State Highway and Transportation Official (AASHTO), près de 60 % des accidents mortels surviennent à cause d'une déviation involontaire de la voie d'un véhicule sur les routes principales [8]. La plupart de ces collisions se produisent lorsqu'un véhicule franchit une ligne de bord, une ligne médiane ou s'écarte autrement de sa voie ou de sa trajectoire [9]. Selon une étude récente comparant les accidents avec et sans LDWSS, il a été découvert qu'un LDWSS embarqué était bénéfique pour réduire les accidents de tous les niveaux de gravité de 18 %, les blessures de 24 % et les décès de 86 % sans prendre en compte les données démographiques des conducteurs [10]. La technologie de vision était initialement utilisée dans les applications LDWSS. Comme le montre la figure 3, une caméra de vision montée sur le pare-brise du véhicule surveille et calcule la distance latérale entre les lignes de chaque voie ; lorsqu'un véhicule dévie de sa voie de circulation, le processeur LDWSS calcule l'écart et avertit le conducteur par des moyens audio, visuels ou autres.

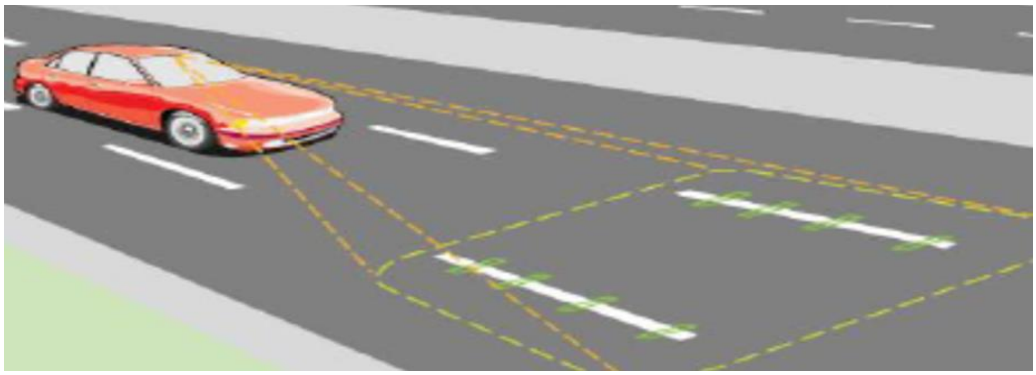


Figure 3 Caméra de vision installée sur le pare-brise pour suivre les marquages au sol.

LDWSS est un vaste sujet qui englobe plusieurs fonctions critiques exécutées par son mécanisme de fonctionnement. Ces actions critiques sont en outre classées en deux tâches : diriger le véhicule en acquérant des connaissances sur l'environnement routier (par exemple,

détecter les marquages au sol/les obstacles/les surfaces de la route) et déterminer quand alerter le conducteur lors des manœuvres. Les travaux de recherche suivants se concentreront sur la tâche préalable du LDWSS qui consiste à comprendre l'environnement routier par la navigation et le positionnement du véhicule sur la route, un composant essentiel d'un véhicule semi- ou entièrement autonome.

Perhaps no invention shaped the twentieth century more than the automobile. On a daily basis, the majority of people rely on automobiles as their primary mode of transportation. Indeed, over 70 million automobiles were produced globally in 2018. Due to the fact that 87 percent of the working population commutes exclusively by car, car-related expenses are the second largest category of spending for the average American family. While this invention has been a huge success, its benefits are not without a price. Additional economic, environmental, and social costs are associated with mass mobilization. Additionally, it endangers commuter safety by resulting in harmful or fatal incidents.

According to the National Highway Traffic Safety Administration (NHTSA), driver inattention is the primary cause of serious accidents [1]. This includes fatigue, drowsiness, distraction, and speeding. Apart from driver error, environmental factors such as weather conditions have been associated with automobile accidents [1]. The driving assistance system (DAS) was introduced in automobiles with the goal of increasing mobility efficiency and safety through the incorporation of inertial sensors and air bags within the vehicle [2]. However, this assistance was initially user-oriented and thus limited to drivers and their vehicles, prompting growing concern about pedestrian and vehicular safety in all weather conditions. This concern resulted in advancements in electronic driver assistance, with automobiles now capable of performing certain maneuvers with or without manual intervention. The transportation system has been significantly altered by advanced driver

assistance safety (ADAS) systems and their applications [2-4]. An ADAS system in a car collects data in the longitudinal and lateral directions of the vehicle using an environment perception module comprised of several sensors whose purpose is to provide the necessary information for interpreting the surrounding scenes near the vehicle using a central computer and then activating alerting or braking based on sensor input analysis. Figure 1 illustrates the fundamental structure of an ADAS-enabled vehicle.

Due to technological advancements, autonomous driving systems have become a focus of research in academia and industry [5, 6]. As a result, the Society of Automobile Engineers (SAE) has defined six levels of automation for these safety systems, ranging from 0 (complete manual driving) to 5 (Fully autonomous navigation) [7]. While fully autonomous vehicles (level 5) are still a few years away, current commercial vehicles are equipped with ADAS systems classified as SAE autonomy levels 2 and 3. Several well-known cutting-edge ADAS applications that operate in the longitudinal direction of the vehicle include Adaptive Cruise Control (ACC), Forward Collision Avoidance (FCA), Road and Traffic Sign Recognition, and Traffic Jam Assist (TJA), while popular lateral safety assistance includes Lane Departure Warning Safety System (LDWSS) and Lane Keeping Assistance (LKA), Lane Change Assistance (LCA), and Blind Spot Monitoring (BSM). Figure 2 illustrates a common ADAS application that makes use of a variety of sensors to scan the environment.

As a critical module in autonomous vehicles, ego-lane detection enables the car to maintain a consistent position within traffic lanes, which is necessary for subsequent control and planning. Maintaining vehicle position within traffic lanes can be a time-consuming and aggravating task, especially on long trips on city streets where lane departures are common. According to the American Association of State Highway and Transportation Officials (AASHTO), nearly 60% of fatal accidents occur because of a vehicle's unintentional lane

drifting on major roads [8]. Most of these collisions occur when a vehicle crosses an edge line, a center line, or otherwise departs from its intended lane or trajectory [9]. According to a recent study that compared crashes with and without an LDWSS, it was discovered that an in-vehicle LDWSS was beneficial in reducing crashes of all severity levels by 18%, injuries by 24%, and fatalities by 86% without taking driver demographics into account [10]. Vision technology was initially widely used in LDWSS applications. As shown in Figure 3, a vision camera mounted on the vehicle's windshield monitors and calculates the lateral distance between lane lines; when a vehicle deviates from its traffic lane, the LDWSS processor calculates the deviation and notifies the driver via audio, visual, or other means.

LDWSS is a broad subject that encompasses several critical functions performed by its operating mechanism. These critical actions are further classified into two tasks: navigating the vehicle by acquiring knowledge of the road environment (for example, detecting lane markings/obstacles/road surfaces) and determining when to alert the driver during maneuvers. The following research will concentrate on the LDWSS prior task of comprehending the road environment through navigation and vehicle positioning on the road, a critical component of a semi- or fully autonomous vehicle.

1.2 Problem Statement

A self-driving car must be aware of its surroundings as well as its own capabilities. Furthermore, the vehicle's position enables the estimation of its heading, which aids in the development of ADAS applications for autonomous longitudinal and lateral collision avoidance. As part of lateral collision avoidance, LDWSS application employs a critical and key function known as lane detection, which involves detecting lanes and tracking the vehicle's position in relation to the road. LDWSS currently employ a variety of technologies to determine lane markings and vehicle positioning, including vision cameras [11, 12] and

light detection and ranging (Lidar) systems [13], and in some cases, Global positioning systems (GPS) in conjunction with vision cameras [14].

However, current technology systems suffer from several flaws that contribute to the low reliability of LDWSS. For example, the majority of lane departure warning systems detect an impending lane departure using a single camera and processor [15, 16]. A closer examination of camera-based systems reveals that vision systems use a variety of image processing techniques to detect lane markings and calculate the vehicle's lateral shift on the road in response to unexpected and sudden lane departures. In normal weather conditions, the vision system performs well; however, in inclement weather, the camera's image processing techniques for determining lane markings become difficult, time consuming, and dangerous, making LDWSS ineffective [17]. Moreover, the camera relies on lane markings and boundaries to activate LDWSS; when road conditions such as missing or irregular/broken lane markings exist, the camera's performance is compromised, resulting in the accumulation of false data for lane departure detection and activation. Additionally, without a distance view, the camera is incapable of interpreting lane markings on curves, resulting in incorrect lane marking determination. The camera's performance in various weather conditions is depicted in Figure 4.

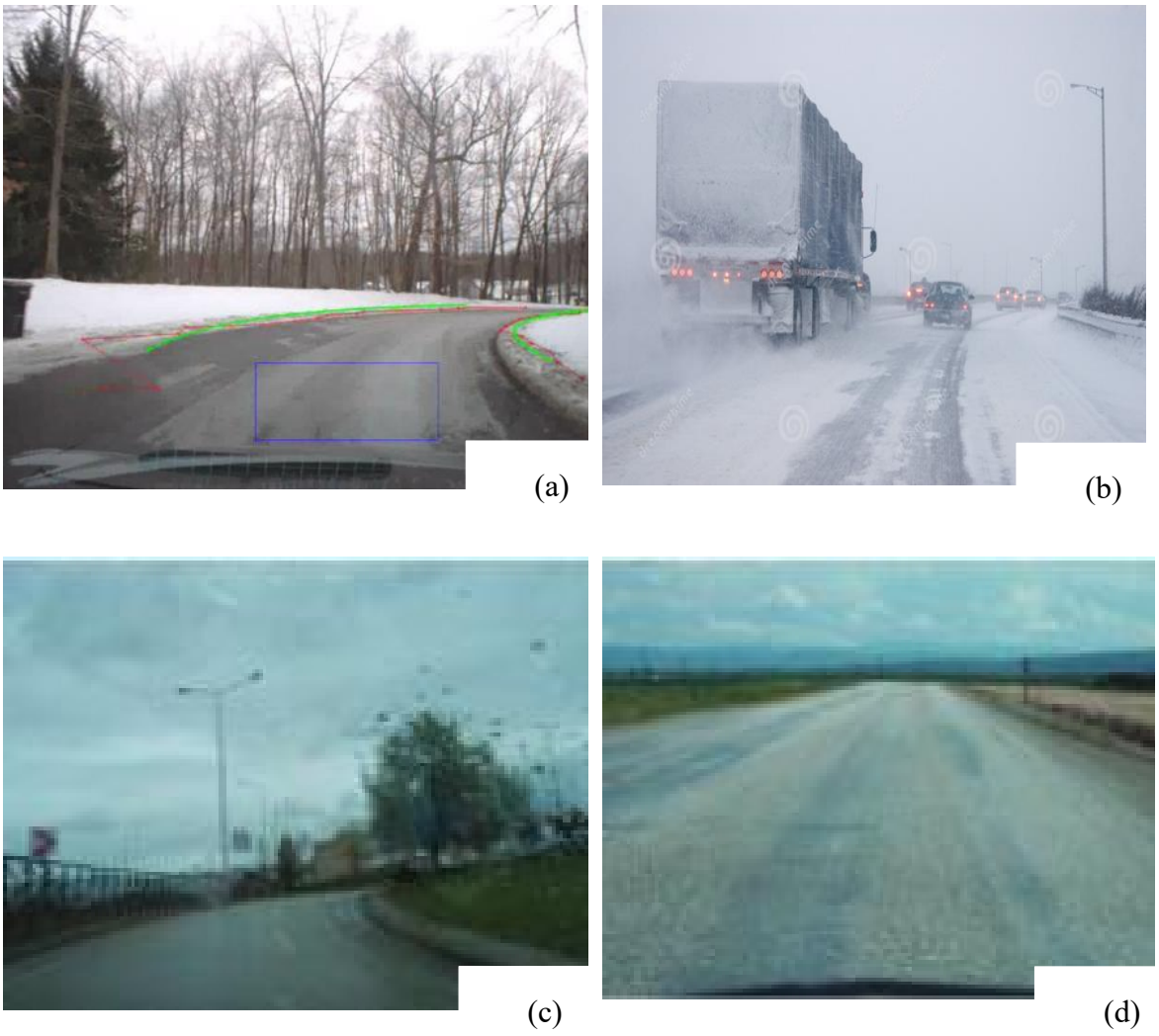


Figure 4 (a) & (c) demonstrates the absence of lane markings at curves in snow and rain weather. (b) and (d) highlights missing lane markings in snow and normal weather.

GPS-based solutions, in conjunction with cameras, aid in mapping the environment by global positioning vehicles on roads. Due to the presence of tall buildings, tunnels, and trees on urban roads, GPS receivers are limited by line-of-sight, resulting in signal delays and complicating vehicle lane positioning estimates. Although optical scanning systems, such as Lidar, are computationally simple, they are an expensive solution for activating the

imminent LDWSS. Lidar, like vision cameras, is sensitive to adverse weather conditions [18].

As a result, a backup or alternative system capable of assisting with vehicle position estimates in the event of current technology failure, such as a vision camera that is unaffected by road profile or weather conditions, is required to improve LDWSS reliability and usability in all weather conditions, and this backup system can be developed by addressing current technological constraints. The following section goes into greater detail about the research goals and objectives.

1.3 Methodology & Objectives

The overall purpose of the research was to address existing constraints in terms of data reliability, continuous data availability, and cost-performance ratio. Apart from meeting technical specifications, another hurdle was dealing with adverse climatic changes (such as snowfall or rain), which create wet pavement environments and impair sensor performance, resulting in false information and endangering commuters. The experiment was designed to develop a low-cost alternative to existing LDWSS solutions, specifically vision based LDWSS, for predicting a vehicle's lane position throughout a journey that is unaffected by climatic changes and independent of road lane markings.

To achieve the main objectives, we divided our goal into two subgoals. The first subgoal is to find a low-cost sensor for vehicle positioning that is resistant to weather changes and consistent in providing information about the vehicle's location. The second subgoal was to develop a road reference solution that is an alternative to lane markings for guiding vehicles along trajectory, so that vehicles do not have to rely on lane markings to activate

LDWSS. To achieve two sub-goals, the following methodology was developed to aid in the achievement of the overall goal and the same has been outlined in Figure 5.

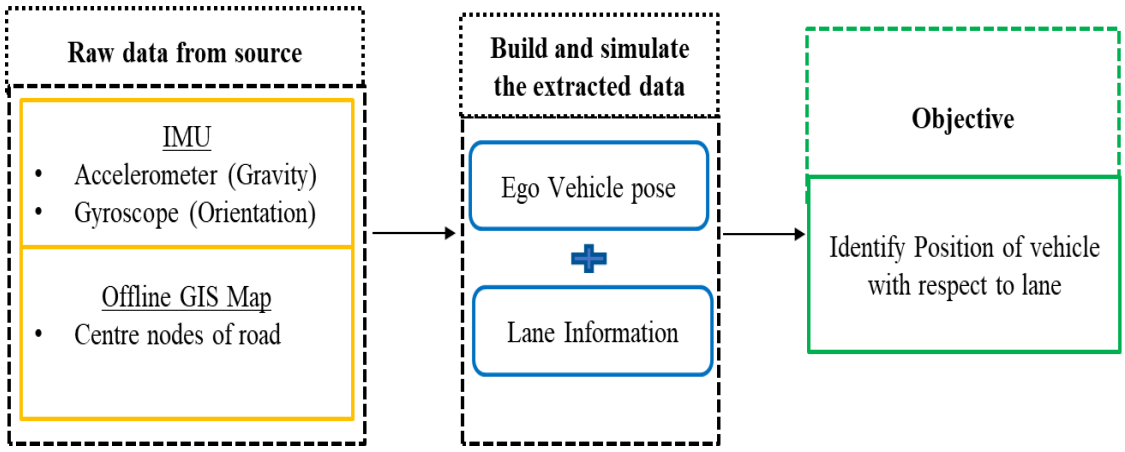


Figure 5 Outline of the methodology is presented

The first stage of methodology was to perform research on commonly used positioning technologies for LDWSS and to select relevant sensors that offer information on the vehicle's dynamic moments while remaining insensitive to external factors such as weather changes. To achieve the intended low-cost robust solution, our survey examined parameters such as sensor performance, continuous data availability, and minimal cost of sensor.

Following a brief survey, we decided on an Inertial Measurement Unit (IMU). The second step was to obtain the vehicle pose in real time. IMU readings of 6 Degree of freedom DOF accelerometer and gyroscope were recorded in real time, and a state space model was developed for error estimation. An error estimation algorithm was proposed to improve the precision of the vehicle's ego pose, where navigation errors and IMU error models were both minimised to obtain a precise estimate of position using GPS measurements. A recursive algorithm was used in MATLAB to track vehicle motion along the trajectory, and further error reduction has resulted in increased precision of ego vehicle location.

After determining the vehicle's pose, we needed to confirm its accuracy. We needed a ground truth reference for the pose results, so we used Open Street Map (OSM) geospatial data. After processing geospatial map data in QGIS software, a geospatial street network model was obtained. The street network model is made up of a large amount of information, the majority of which is related to roads. The road-related information contained hints such as nodes on the road with unique latitude and longitude coordinates. We used these latitude and longitude coordinates to create a reference lane model, which satisfied our second goal of developing alternatives to lane markings. Finally, to achieve our overall goal, we compared and evaluated vehicle position estimates derived from vehicle roll, yaw, and pitch moments as measured by an in-vehicle IMU sensor against generated global reference points from an offline map using MATLAB.

1.4 Significance

In order to avoid all relevant collisions in a vehicle equipped with LDWSS technology, the system must be operational at all times, in all dynamic conditions, and economically viable. This study makes an important contribution by increasing the dependability of navigation solutions in environments where cameras or GPS are not available. Furthermore, the goal of this research is to forecast the cost-effective safety benefits of LDWSS technology for vehicle manufacturers, as well as to address existing research gaps in LDWSS technology. The developed map-assisted, low-cost, and user-friendly navigation system can be used for a variety of purposes, including in-vehicle navigation and location-based smartphone utilities. The navigation solution's accuracy has been improved, giving users a more reliable and robust navigation option. This improved method can also be applied to emergency or law enforcement navigation apps. The anticipated safety benefits of implementing the proposed

method may aid in increasing LDWSS adoption rates and influencing policymaking and regulatory approaches to improving motor vehicle safety. These policies would save lives and reduce the negative consequences of lane departure accidents.

1.5 Organization of the dissertation

This thesis is divided into seven chapters, the most recent of which introduces the study's context. The topic of the research, the problem statement, and the motivation for conducting the research are all discussed. The research objectives and methodology for moving forward with the research, as well as the significance of this research, are presented, and the section concludes with highlighting thesis organization.

Chapter two illustrates the literature review process. This section will describe the process of conducting a literature review for this thesis, followed by a discussion of the current state of the art and approaches used. It then discusses some of the general observations and limitations of the current state of the art, before presenting the approach taken in this thesis to resolve the existing problem by proposing an alternate solution.

Chapter three investigates various positioning sources and their limitations in relation to the thesis objectives to provide information for navigation sensors. Based on a review of positioning sensors, this section aids in sensor selection for this research.

Chapter four describes inertial sensors, as well as their errors and mechanisation equations, in a clear and precise but informative manner. subsequently investigates and discusses the integration of multiple mobile sensors with the algorithms and techniques used in this research to develop map-aided navigation systems

Chapter five introduces maps for navigation. Following that, geospatial data from maps is discussed, and various sources of mapping data are investigated to generate useful models for navigation applications. The generation of a geospatial model is presented, followed by the presentation of the developed outdoor geospatial model for this study.

Chapter six discusses the various field tests that were conducted and the analysis of the results. Each field test environment and scenario for outdoor and environmental testing is described in detail, as are the sensors used. The results of each field test, as well as their analysis, are discussed and summarized.

Chapter seven Chapter seven summarizes the thesis' accomplishments, describes the research findings, and highlights the thesis' shortcomings, as well as making recommendations for future research and improvements to the thesis' methodology.

Chapitre 2 - **Literature review**

2.1 Preparation of Literature review

This section conducts a systematic review of the existing literature to achieve the research objective, highlighting limitations in existing methodology, exposing research gaps, and finally presenting contributions made in this research to address research gaps.

We began our search operations by conducting an exploratory search with the end note tool. The end note tool connects to over 100 different databases using the university library's search engine, the most frequently referred to being Scopus, IEEE Xplore, Science Direct, Google Scholar, and ProQuest. The thematic terms "autonomous vehicles", "AVs", "self-driving vehicles", "intelligent transport", AND "road safety", "trajectory control", "position control", OR "lane departure" are among the search engine keywords. In order to filter and select the key papers for a comprehensive review, we defined the search criteria sequentially. Among the key criteria were keywords relevant to the research topic, work published in English, peer-reviewed papers published between January 2015 and 2021, and access to full papers. Depending on the data sources, keywords may include article titles, abstracts, and metadata. During the article search process, the following research questions (RQ) were generated:

a) RQ1: How are existing research papers structured, and what are the papers' primary objectives?

b) RQ2: What is the current state of the art and the challenges associated with vehicle positioning on the road?

c) RQ3: What are the future possibilities for combining various approaches with other emerging technologies? The sections that follow provide elaborative approaches to answering the research questions, as well as highlighting gaps in the existing literature.

2.2 Current state of art and approaches used

When semi- or fully autonomous vehicles share the road with other vehicles, it is critical to keep track of the ego vehicle's movement. From a safety standpoint, autonomous vehicles must have reliable and accurate positions in order to navigate safely. A critical component of a fully autonomous vehicle is positioning the vehicle in such a way that the road, lane, and location within the lane are known. Additionally, the navigation of an autonomous vehicle requires a precise decision within a few seconds while taking into account the dynamics of all obstacles, the road structure, and the user's safety, which complicates the task of vehicle positioning. There are three widely researched approaches to vehicle positioning on roads classified as cooperative localization, absolute localization, and sensor-based localization.

Cooperative localization: It is a technique for optimizing performance by utilizing data from adjacent sensors. For example, in [19], the authors proposed a method for determining the relative trajectories of vehicles travelling in multiple lanes toward a merging junction with an accuracy of less than half the lane width by utilizing Dedicated Short Range communication (DSRC) based vehicle-to-vehicle communication and standard GPS receivers.

Absolute localization: To perform automated driving on real roads, an autonomous vehicle must secure information about its location. Absolute localization utilizes GPS

receivers to obtain information from at least four satellites to determine the vehicle's own position, that is, the starting point to begin. Essentially, autonomous vehicles are designed to generate global paths based on data from waypoints and then follow them. The authors of [20], presented two algorithms for lane departure detection and advance curve detection using a standard GPS receiver. Extensive field testing on the interstate was conducted to determine the efficacy of both algorithms on straight and curved road segments. The results of field testing indicate that the proposed lane departure detection algorithm is capable of detecting and warning the driver of a true lane departure with an accuracy of nearly 100% on both straight and curved road segments.

Sensor based localization: Semi- or fully automated vehicles are equipped with a variety of sensors, including proprioceptive sensors that provide information about the vehicle's motion and exteroceptive sensors that provide information about the vehicle's surroundings. Odometers, velocity and steering encoders, and inertial measurement units are all examples of proprioceptive sensors (IMU). Exteroceptive sensors, on the other hand, include Lidar, Radar, and vision cameras. In the literature, it is common to combine various types of sensors to improve the vehicle's localization. For example, in [21], the host vehicle's ego lane was estimated using a Bayesian Lane estimation filter, and the vehicle was equipped with forward-looking radar and a camera. The camera makes an educated guess about the geometric shape of the lane markings. A fusion system that integrates radar and camera data performs target tracking and generates estimates of the kinematic states of vehicles ahead of the ego car. Additionally, the test vehicle was equipped with a GPS receiver to perform self-motion estimation based on gyro measurements. Likewise, combining an offline map with on-board vehicle sensors demonstrated complementary benefits. For example, in [22], the

importance of GIS information in determining automated vehicle safety functions such as efficient lane change and path planning was highlighted, and an automated vehicle was discovered to be equipped with Lidar, camera, GPS, and inertial system. In [23], authors present a novel approach for lane level localization that combines visual lane information with topological map matching via GPS and dead reckoning. When combined with the proposed map matching algorithms, the experiment outline and proposed map matching algorithms aid in achieving lane level ego vehicle localization.

2.3 General observations and limitations in current state of art

According to review articles [15, 24, 25], vision camera is the most frequently used human sense for driving. Specifically, several researchers are working on the implementation of lane departure warning systems. Lane detection is a process that involves locating road markers (white or yellow horizontal markings) on the surface of painted roads and drawing lanes boundaries. Lane detection can be achieved either from the input image using conventional image processing techniques which involves image operations like color-based extraction, gray scale image conversion, edge detection and perspective transformation or by training the algorithm with various scenarios using neural networks. However, as demonstrated in [16], developing a consistent and robust lane detection system is a difficult and time-consuming research topic; this fact may be justified by the complexity and variability of case conditions. Lane and road appearance diversity is a limiting factor in lane detection; lane markings and lane width exist in unnormalized variants. Additionally, severe occlusions may be created by nearby vehicles, which cannot be ignored. Moreover, visibility conditions constrain the systems' performance (weather conditions, lighting, haze, and night conditions). Despite significant progress in the area of lane detection and tracking using

neural networks, there is still room for improvement [25-27]. The use of Lidar optical scanning systems to measure the lateral distance between road and lane markings is an expensive solution for studying vehicle localization on a lane and is climate sensitive [14, 22, 28]. A standard GPS receiver's absolute position accuracy is in the range of 3-5m, which is insufficient to determine any lateral lane-level drift in a vehicle's trajectory required for lane departure detection. Moreover, GPS signals are obscured in urban areas by buildings, tunnels, and the effect of the atmosphere [29]. Additionally, a significant disadvantage of the DSRC approach is its limited scalability [30], It stems from the protocol's inability to provide the required time-probabilistic characteristics in dense traffic. Likewise, some systems integrate GPS data with a camera based LDWS to improve the reliability of lane departure detection in adverse road conditions [23, 31]. However, such systems require GPS technology, an inertial navigation sensor, and access to digital maps with lane-level resolution to correct the GPS position, increasing their complexity and cost of implementation.

2.4 Approach used in paper

To overcome the limitations of vision, Lidar, and GPS as revealed in previous section, an alternative approach like that of [20], is proposed, in which a GIS is used in conjunction with in-vehicle smart phone-based inertial navigation for positioning the vehicle on an urban road and determining permissible lateral distance for departure warning. To calculate the lateral distance, a road reference must be obtained from Google Maps or another navigational mapping database, such as OSM [32, 33]. Typically, any mapping database includes a street network that contains geospatial information about the attributes and geometry of streets. Roads are divided into segments of individual links, each with its own distinct characteristics

such as street name, number of lanes, bus stops, and highways. Each of these links is associated with a link identification (ID) that is globally unique in the mapping database and is used to identify and process data about any given road segment. These mapping databases can provide information at the road or lane level, depending on the quality of the survey and the services required.

The proposed technique requires only road-level information. A road-level map of a given road segment with a unique global ID is associated with a set of geographic latitude-longitude points located in the middle of the road that approximate the shape of the actual physical road with some lateral error. The distribution of these shape points along a given road segment is directly proportional to the road geometry's acuity (see figure 21 in chapter 5). Combining GIS reference points and vehicle position data can aid in achieving global positioning accuracy. In comparison to the current state of the art, the proposed method is independent of lane marking visibility and varying climatic conditions, works well in urban areas, and is a low-cost solution. Additionally, it is less complex, requires less computation, and details every small dynamic of the vehicle that is necessary for lane departure scenarios to take necessary action.

Conclusion: The procedure for conducting a literature review is illustrated, followed by a discussion of the current state of the art and methodologies used. The following section discusses some general observations and limitations of the current state of the art. Using the studies cited previously in this section, we were able to identify gaps in the other studies, and finally, a proposal is formed and presented which is used in this thesis to resolve the existing problem through an alternate solution.

Chapitre 3 - **Information source for vehicle positioning**

Positioning is the determination of a body's position but not its velocity or attitude. Many navigation technologies, even though they are strictly positioning systems, operate at a high enough rate that velocity can be calculated from the rate of change in relation to position. There are numerous technologies and data sources available for use in vehicle navigation systems and other ADAS applications that require positioning data. The challenge for in-car navigation system designers is deciding which of these information sources to use and how to combine them to meet performance requirements. This necessitates balancing the system's cost, complexity, and performance.

The following subsection focuses on various sensors used for LDWSS, emphasising key points such as working concepts and the advantages and disadvantages of sensors under various weather conditions, assisting in the selection of sensors for our goal in this thesis.

3.1 Sensors:

Car sensors are classified as either proprioceptive or exteroceptive. Proprioceptive sensors aid in measuring vehicle ego movement and dynamics. Proprioceptive sensors include wheel speed sensors, torque sensors, steering angle sensors, and IMUs. Exteroceptive sensors, on the other hand, detect obstacles and recognize the navigation scene, as well as help gather information from the surroundings and improve knowledge of the vehicle's location. Exteroceptive sensors are further subdivided into passive and active sensors. Active

sensors, such as ultrasonic, Radar and Lidar, work by emitting energy in the form of electromagnetic waves or radiation and measuring the return time to determine parameters such as distance and position. Instead of transmitting external signals or disruptions, passive sensors (infrared cameras) detect electromagnetic waves or radiation in the environment.

3.1.1 Exteroceptive sensors

3.1.1.1 Camera

Market penetration: Human driving is primarily based on an analysis of the surrounding vehicles' characteristics, such as obstacles, lane detection, and road signs. A camera can provide some of this information for automated operation. Cameras are used in almost all SAE degrees of autonomy greater than one. Cameras are the only type of imaging equipment capable of seeing colours, making them ideal for lane detection tasks. Figure 6 depicts a mono-camera from Mobileye. Mobileye is a well-known manufacturer that is currently leading monovision with smart technology, and the technical specifications of the intelligent vision-based camera can be found in [34].

Figure 6 Intelligent mono-camera by Mobileye [35]

Working principle: The camera is a digital lens imagery system that works by collecting and translating the image of an object into electrons on a pixel image sensor. Later, the camera capacitors convert electrons into voltages, which are later converted into an electronic digital signal. Two imaging sensors, the charging coupling device (CCD) and the complementary metal oxide semiconductor device (CMOS-D), are typically used in real-time applications. Brief comparisons between the CCD and CMOS-D can be found in [36, 37]. CCD cameras deliver excellent low-noise performance but are expensive and, as an alternative, the CMOS-D has been developed to reduce production costs and power consumption. Because of this advantage, the CMOS-D is widely preferred for automotive applications in the related industry. Cameras, in conjunction with computer vision and deep learning techniques, offer environmental information, such as detection of the target, their related physical descriptions (like the position of moving targets, size, and shape) and semantic descriptions (like recognizing and classifying trees, vehicles, traffic lights, and pedestrians). Camera information is easy to understand, which makes it more popular than other sensors. Various configurations exist in the camera and out of different configurations, monocular and stereo vision camera solutions are a common choice for researchers. In monocular systems, only one camera is used to detect, track, and measure longitudinal distances, based on landscape geometry. There is a downside to distance measurement in the monocular camera, since the distance is measured by using the position of the pixel in the vertical direction of the given image coordinates, which typically results in errors, due to the lack of direct depth measurements for the captured images. Compared to monocular cameras, stereo cameras with two cameras have an additional feature for measuring the distance between objects. Using stereo-view-based methods, two cameras can estimate the 3D coordinates of an object. A brief analysis of techniques for real-time obstacle detection and

classification linked with various algorithms using the stereo camera is presented in [38]. While stereo vision cameras are effective in target detection and classification, they are more expensive than mono-cameras, and they also have problems with calibration and computational complexity. In [25], vision-based techniques for activating LDW are discussed, with a focus on processing lane markings through a series of steps beginning with preprocessing of the environment, where the Region of interest (ROI) to be scanned is selected, later extracted scene from ROI is made to colour transformation process, followed by feature extraction of lane marking which shall be used for modelling lane, and then the modelled lane is tracked throughout the trajectory. The benefits and drawbacks of various vision algorithms have also been discussed. Also authors of [25], have discussed the various methods for defining lane departures. There have been many works done on lane detection using camera and interested readers are proposed to refer [39, 40], which is review covering various types of algorithms adapted for lane detection in varying conditions and advances in computer vision techniques for robust tracking of lanes on road.

Vision-based system performance degradation: While advanced methods have improved recognition techniques, small variations in weather still influence camera measurements. The camera is very sensitive when faced with adverse climatic conditions. A camera in an aerosol environment experience decreased visibility and contrast, and is unreliable in object recognition, and a camera is not recommended for environmental detection and vehicle control tasks under foggy conditions, as per [41]. In the same reference [41], a full description of the rain and fog interactions with a camera is also provided. Camera sensors have an advantage over object detection and classification and are important for automated safety systems. However, in [42], an indoor rain simulator was used to

systematically investigate the effects of rain on camera data and the results outline that the performances of the camera sensors were mainly affected by decreased gradient magnitudes, resulting in a shift in the location and size of the bounding boxes during the detection process, leading to a decline in classification scores and resulting in uncertainty. Similarly, based on indoor experiments [43], the authors experimentally researched the effects of rain and showed that raindrops lead to an increase in the average intensity of the image and a decrease in contrast. The study presented in [44], develops an approach to quantifying the vulnerabilities of the camera, based on empirical measurements and the concluding results show that camera principal output loss occurs in lighting and precipitation. Lane detection was performed using a camera for rainy conditions in [45], and the author concluded that when the vision sensor is exposed to 20mm or more of precipitation, it fails to provide information about the lane. As a result, it is necessary to develop a technology capable of maintaining the bare minimum ADAS functionality during rainfall and other adverse weather conditions.

3.1.1.2 Lidar

Market Penetration: Lidar was invented in 1960 for the study of environmental measurements (atmospheric and oceanographic parameters), and it was later adapted for topographical 3D mapping applications in the mid-1990s [46]. In the Defence Advanced Research Projects (DARPA) challenge in 2005, lidar was used on vehicles to detect and avoid obstacles [47]. Commercial Lidar manufacturers include Velodyne, Quanergy, Leddartech, Ibeo, and others. In [48], a technical description of various lidar sensors is shown. Figure 7 depicts a lidar image for an automobile.

Figure 7 Velodyne HDL 64 lidar [49]

Working principle: Just like radars use time of flight of radio wavelengths to collect target information, photodiodes are used by lidar to transmit the light pulse to the target. The optical receiver lens in the lidar system is used as a telescope for the processing of photodiode fragments of light photons. The collected reflections include 3D point clouds corresponding to the scanned environments and the strength of the reflected laser energies provides information about the range, speed, and direction of the target. Lidar manufacturers use two wavelengths: 905 nm and 1550 nm. The former is a common option for automotive manufacturers due to its reliability, eye protection, and cost-effectiveness with silicone detectors. In [50, 51], an authoritative analysis on lidar, its waveform, and market penetration strategies are discussed. Lidars provide a good physical description of the target and, due to that, lidars have been used for target detection, tracking, and motion prediction., Filtering of the ground and clustering of the target [52-55] are two methods widely used by lidar for object detection, which provide the spatial information of the target. To classify and recognize objects (like pedestrians, trees, or vehicles), lidars make use of techniques such as machine learning based on object recognition [56-59], and additional methods such as global

and local extraction of features to help in providing the structure of the target. Lidar uses the Bayesian filtering framework and data association methods for target tracking and motion prediction to provide information, such as velocity, trajectory, and object positioning [60-62]. Lidar has been widely used on a probe vehicle to extract lane-level road networks [63, 64]. In [65],camera and Lidar imagery are used to predict a dense representation of the lane boundaries in the form of a thresholded distance transform. In [22], Lidar, camera, inertial sensor, and ultrasonic sensor are used to make the vehicle suitable for lane change and path planning decisions. where the local path is generated using data from a camera, a laser scanner, an ultrawave sensor, an intertial sensor, and other vehicle features.

Degradation of lidar performance: Lidar performance in extreme weather conditions is not as strong as expected. Adverse weather conditions increase the transmission loss and decrease the reflectivity of the target. Under the perception category (fog, snow, and rain), fog has been found to have the greatest impact on the ability of lidar, due to its high expansion and backscattering properties, which are greater than in weather conditions like snowfall and rain [66]. The challenge in fog conditions is that many transmitted signals are lost, resulting in reduced power. Reduced power would alter the signal-to-noise ratio of the lidar sensor and influence its detection threshold, which leads to degraded perception performance. In [67], the depth of lidar performance in fog is studied and the observed light is scattered by fog particles, which not only reduces the detection range dramatically, but also leads to false detections. In the same study, the fog condition showed similar performance degradation to the airborne environment. In [68-70], qualitative and quantitative experimental studies of the fog effect on lidar in controlled environments outlined the loss of transmission phenomena leading to low received laser power and low target visibility. Lidar scanner capacity testing

has been carried out in the northern part of Finland at Sodankylä Airport [71], where fog creates a special problem. Results indicate that fog reduces the sensor range by 25%. In [72], the quantitative performance of lidar with varying rain intensity, with the help of a mathematical model, is presented and the results show that as rainfall intensity increases, the lidar cloud density is affected, increasing false-positive errors. The same effect is presented in [73], where the authors' analysis showed that the varying intensities, size, and shape of raindrops drastically influence the attenuation rates of lidar. The effect of snow on lidar performances, such as reflectivity and propagation through the snowy environment, is evaluated in [74], using four lidars of different manufacturers. The results observed from this experiment highlight that the receiving power levels generated by snowflakes or water droplets were high (due to false-positive errors) and tended to overload the optical receiver chain. The lidar performances are also altered by airborne particles, such as dust, which have greater characteristic wavelengths than lidar. These particles prevent the sensor from imaging its surroundings, resulting in reduced visibility and incomplete target information. In another work [75], the experiment outlines that the dust particles in the air are very often detected by laser sensors and hide obstacles behind the cloud of dust. To address these limitations, the use of lidars with a 1550 nm wavelength with a strong propagation ability is suggested. However, this solution is of limited use, because of constraints such as high cost and high energy usage. The loss of efficiency due to adverse environmental conditions was examined in [76], and a comparison was made between lidars with wavelengths 905 nm and 1550 nm. The findings of [76], have also shown that lasers with a wavelength of 1550 nm have a much higher water absorption compared to 905 nm lidar. Furthermore, due to the advantage of the higher wavelength (1550 nm), higher power can be used for the transmission of lidar signals,

which could result in an increased range of detection in adverse climatic conditions, while maintaining eye safety regulations.

3.1.1.3 Global Positioning System (GPS)

Working and Limitations: The GPS is a satellite-based global navigation satellite system (GNSS) that always provides precise location and time information in all weather conditions and, anywhere on or near the Earth. GPS was established in 1973 by the United States Department of Defense [77]. The GPS system is built around a man-made constellation of 27 Earth-orbiting satellites (24 in operation and 3 extras in case one fails). A person or object can use these satellites to determine its position in terms of latitude, longitude, and altitude. These satellites orbit the Earth at a height of 12,000 miles and rotate twice every 24 hours. These satellites' orbits are arranged in such a way that at any given time, anywhere on the planet, at least four satellites are clearly visible. A GPS receiver placed on Earth can determine its location by using any of the four visible satellites. To carry out the GPS localization process, two pieces of information are required: the distance between the GPS receiver and the satellites, and the position of each satellite in terms of latitude, longitude, and altitude. More information, such as distance calculations and related mathematical derivations, can be found in [78].

Figure 8 GPS receiver by U-Blox [79]

The primary benefit of GPS positioning is the extremely low level of positioning error. When signals from more than four satellites are received with an appropriate PDOP (Position Dilution of Precision), the positioning results can be guaranteed to be of high quality, that is errors are minimised. Additionally, the light weight and low cost are significant advantages, which can be visualized in figure 8. As a result, GPS positioning has gained popularity over the years. GPS is now ubiquitous, being integrated into cell phones, automobiles, and laptop computers.

However, GPS positioning is not without flaws. The most serious issue is signal loss. Satellite signals are frequently interfered with in urban and mountainous areas, as well as in enclosed and underground spaces. Additionally, signal acquisition and tracking are challenging for highly dynamic movements. In these instances, it is necessary to consider the availability of satellite signals. In [29], an experiment is carried out for robust navigation at the lane level localization of vehicle using camera, lidar, GPS, and inertial sensors; various scenarios are presented, which affect the navigation filter, such as satellite geometry and the number of satellites. When GPS measurements are used for lane determination, similar

drawbacks are shown in [80]. For more information on GPS and positioning errors, see [81], which discusses GPS effects such as clock offsets, pseudorange codes, estimation error of broadcast ephemeris, atmospheric effects, ionospheric effect, troposcopic delays, and relativistic delays, all of which must be considered when developing an observation model. Additionally, GPS signal security is a significant concern. As is well known, satellite signals are susceptible to interference and spoofing. Security of wireless signals, including satellite signals, has always been a critical concern for electronics engineers, even more so in communication and military applications.

3.1.2 Proprioceptive sensors

Information from vehicle-based sensors is processed to estimate position, velocity, and attitude. The dead reckoning process is used by the vast majority of vehicle-based sensors. Velocity encoders, accelerometers, and gyroscopes, for example, all provide data about the vehicle's first- or second-order derivative position and attitude. The odometer also shows the distance travelled by the navigation system. As a result, converting these sensor measurements to position and attitude estimates will be integrative in nature, requiring knowledge of the vehicle's initial state as well as the accumulation of measurement errors over time or, in the case of the odometer, with the distance travelled.

Furthermore, the information provided by the sensor mounted on the vehicle is represented in the vehicle coordinate system, with the exception of possible fixed rotations. As a result, before the sensor measurements can be processed to estimate position, velocity, and attitude, they must first be transformed into a more easily interpreted coordinate system, preferably the ECEF or the geographic coordinate system (see section 4.5). Furthermore, if sensor measurements are to be combined with data from other sources, they must be

expressed in a common coordinate system. The process of converting sensor measurements from a vehicle's mounted sensor to an estimated position and attitude of the vehicle is known as dead reckoning [82]. More information on vehicle sensor dead reckoning can be found in [83-86]. In the following section, we will look at how some of the well-known vehicle-based sensors work.

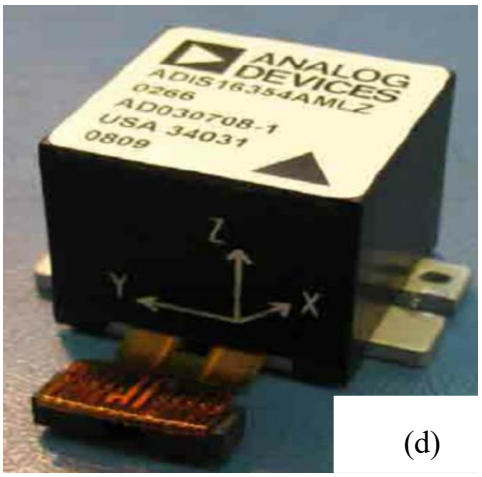
3.1.2.1 Encoders

Working: The encoders are one type of sensor that provides information about the state of the vehicle. The odometer, for example, is a meter that indicates the number of miles driven on a vehicle. An odometer calculates a vehicle's curvilinear distance by counting the number of full and fractional rotations of the wheels. This is done primarily using an encoder, which generates an integer number of pulses for each wheel revolution. The number of pulses generated during a time slot is then multiplied by a scale factor that varies with wheel radius to calculate the distance travelled during that time slot [87]. Figure 9(a) illustrates wheel encoder placed on side of wheel for distance reading.

(a)

(b)

(c)



(d)

Figure 9 The wheel encoder is shown in (a) [88], the velocity encoder is shown in (b) [89], the steering angle encoder is shown in (c) [90], and 6 DOF IMU is shown in (d) [91]

A velocity encoder, on the other hand, determines the vehicle's velocity by observing the wheel rotation rates. The difference in wheel speeds can be used to estimate the vehicle's heading change when the left and right wheels of either the rear or front wheel pair are encoded separately, or when the wheels on one side of the vehicle are encoded separately. Figure 9(b) shows velocity encoder placed on the wheel side for visualization. Sensors in the antilock braking system (ABS), frequently provide information about the speed of the various wheels. More information can be found at [92]. Even though this information is provided without incurring additional sensor costs, the generally low resolution of ABS sensors can have a significant impact on the reliability of the calculated heading estimate. As a result, additional wheel encoders may be required to accurately estimate heading changes based on wheel speed data.

A steering encoder also measures the angle of the steering wheel. As a result, it provides a measurement of the angle between the vehicle platform's front wheels and the forward

direction. The steering angle, along with data on the wheel speeds of the front wheel pair, can be used to calculate the vehicle's heading rate. Figure 9(c) depicts steering encoder.

Limitations of encoders: All these concepts for calculating the vehicle's distance travelled, velocity, and heading are predicated on the assumption that wheel revolutions can be converted to linear displacements relative to the ground. However, there are several sources of error in the translation of wheel encoder readings to the travelled distance, velocity, and heading change of the vehicle. All terrain-dependent phenomena such as vehicle wheel slips, uneven road surfaces, and skidding on frictionless roads occur in an unstructured manner. As a result, it is difficult to predict and mitigate their detrimental effect on the estimated travelled distance, velocity, and heading.

3.1.2.2 Inertial Measurement unit (IMU)

Working: An IMU is a device that provides measurement data in an inertial navigation system (INS). The specific forces and angular rates measured by accelerometers and gyroscopes are numerically integrated based on the initial values of attitude, velocity, and position to obtain the estimated navigation states [93]. Figure 9(d) presents 6 DOF accelerometer and gyroscope built IMU sensor. Due to their high cost, size, and power consumption, inertial sensors have historically been used primarily in high-end navigation systems for missile, aircraft, and marine applications.

Advances in micro-electromechanical systems (MEMS) technology, combined with electronic miniaturization, have enabled the development of chip-based inertial sensors. These chips are small, light in weight, consume little power, and have an extremely low cost and high reliability. As a result of these benefits, MEMS chips have emerged as viable

candidates for a variety of applications, including vehicle navigation systems. However, the price paid (with currently available sensors) is a lower performance characteristic [94].

Limitations: In contrast to odometers and velocity encoders, inertial sensor errors are completely independent of the terrain through which the vehicle is travelling as well as external factors such as tire slip due to wet roads. However, there are several error sources associated with inertial sensors that must be taken into account such as biases, scale factors, non-linearities and noise.

Conclusion: There are numerous sensors that provide information about vehicle positioning; however, only the most used sensors, exteroceptive and proprioceptive sensors, have been presented in the preceding section. When we were looking for a low-cost solution, exteroceptive sensors were much more expensive than proprioceptive sensors [18, 94]. In terms of performance, as previously discussed, inertial navigation has been determined to be the best alternative for vehicle pose determination, and its processing is independent of the external environment.

Chapitre 4 - **Inertial Navigation**

This chapter is intended to provide the reader with background information on past and current navigation techniques and their implementations. Coordinate transformations are shown because each sensor, such as the IMU and GPS, must be linked to a unified coordinate system. Additionally, the theory of inertial navigation is discussed, as is the distinction between tightly and loosely coupled INS/GPS navigation. Additionally, error characteristics such as noise, bias, and scale factor are discussed, as is their relative impact on sensor outputs. The Kalman filter is discussed in terms of its relevance and utility as a navigation framework.

4.1 Navigation background

People have used navigation in some form or another for thousands of years. Birds, bees, and nearly all other animals must be able to navigate between locations in space. Navigation is the process or activity of precisely determining one's position and planning and following a route. A navigation system, also known as a navigation aid, is a device that automatically calculates one's position and velocity. The navigation system or technique used determines the navigation solution. To compensate for our limited range of senses, we've been able to develop better and more accurate sensors over time. Because of technological advancements, navigation can now be accomplished using computers and sensors such as IMUs and GPS.

This thesis will discuss the research done with one of these advanced sensors, an IMU. When combined with the necessary mathematical background, this sensor is capable of detecting accelerations and angular velocities and converting them into the system's current position and orientation.

4.2 Overview on Inertial navigation

The INS is made up of two distinct components: 1) the IMU and 2) the computational unit. The former provides information on the navigation platform's accelerations and angular velocities in relation to the inertial coordinate frame of reference, preferably the ECI coordinate system (See section 4.5).

The IMU processor converts the inertial sensor outputs to units, compensates for known inertial sensor errors, and performs range checks to detect sensor failure. Closed-loop force feedback or rebalance control for accelerometers and/or gyroscope may also be included. Unit conversion converts inertial sensor outputs such as potential difference, current, or pulses into specific force and angular rate units. The angular rotations (rates) measured by the gyroscopes are used to track the relationship between the vehicle coordinate system and the navigation coordinate frame of choice, which is typically the ECEF or geographic coordinate frame (See section 4.5).

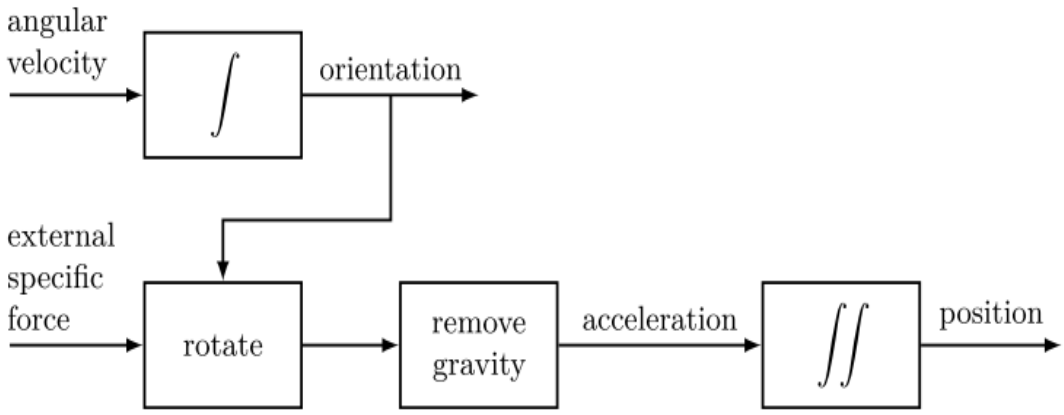


Figure 10 Dead-reckoning is depicted schematically in which accelerometer measurements (external specific force) and gyroscope measurements (angular velocity) are integrated to position and orientation [93]

This information is then used to translate the observed specific force in vehicle coordinates into the navigation frame, where the gravity acceleration is subtracted from the observed specific force, Figure 10 depicts a dead reckoning of accelerometer and gyroscope. The only thing left are the accelerations in the navigation coordinates. The accelerations are integrated twice with respect to time to determine the vehicle's position [93]. Details on inertial sensors and system terminology can be found in [95] and [96].

Since navigation calculations in INSs are integrative, the systems have a low-pass filter characteristic that suppresses high-frequency sensor errors but amplifies low-frequency sensor errors. As a result, the position error grows without bounds as a function of operation time or travelled distance, and the error growth is dependent on the error characteristics of the sensors. In general, a bias in accelerometer measurements causes error growth that is proportional to the square of the operation time for an INS, and a bias in gyroscope measurements causes error growth that is proportional to the cube of the operation time [93].

In practise, drifts, noise, disturbances, misalignments, and manufacturing complications enter the equation and complicate the development of an INS. GPS is frequently supplemented with INS to overcome these errors. GPS remains the most widely used sensor in navigation systems. The combination of inertial sensors and GPS results in a system that outperforms either a GPS receiver or an INS alone. There are many experiments done using inertial navigation along with GPS. For instance, authors in [97], proposed a new vehicle cross-road turning method based on GPS/INS data. The vehicle turning can be accomplished using this method by using a predefined map generated by the line curve-fitting and predicting method based on the location and road condition provided by GPS/INS. In [31], using simulation, lane prediction was performed using INS and GPS data, as well as a vision

camera, and the results outline prediction with INS and GPS data were accurate and good in real time, compensating for camera failure. Not only have INS and GPS solutions demonstrated their robustness in vehicular approaches, but smartphones are increasingly being used to collect traffic data in some instances, and the processing power of commercial smartphones has increased significantly in recent years. The advantages of smartphone-based measurement systems include a diverse set of sensors, typically a GPS receiver and an inertial measurement unit (IMU), as well as their low cost, transparency, and ease of use. As a result of these benefits, drivers are increasingly relying on smartphones to assist them with tasks such as navigation [98], traffic estimation [99], vehicle condition monitoring [100], driver recognition [101], and manoeuvre classification [102].

4.3 Architecture of inertial navigation

When implementing GPS and IMU sensors, many factors must be considered, including dimensions, necessity, and simplifications based on the application's required accuracy. There are two general methods of GPS/IMU integration: tightly coupled integration and loosely coupled integration shown in figure 11, which deal with the structure of the filter.

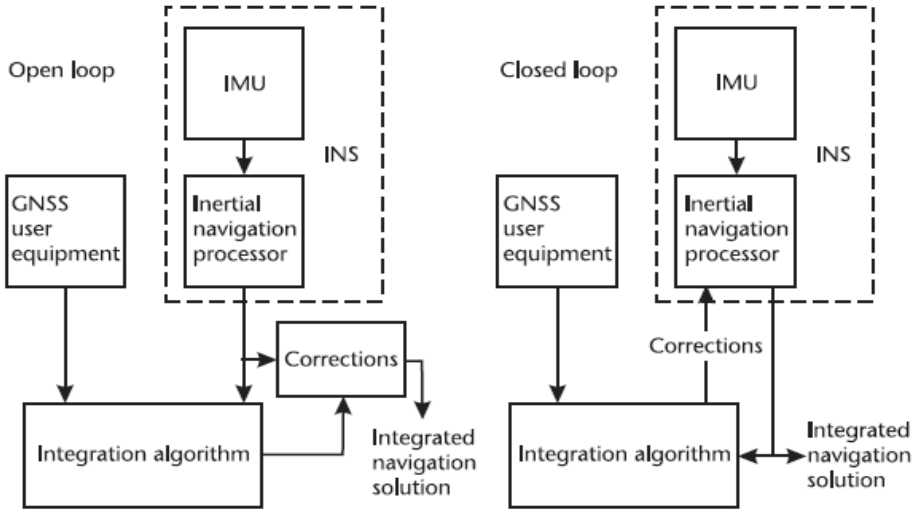


Figure 11 Open- and closed-loop INS correction architectures [103].

A loosely coupled model, in general, will have a position and velocity estimate from the GPS receiver and will use this as the vehicle states to model the error dynamics. Tightly coupled systems will augment GPS parameters such as ephemerides, error sources, and integer ambiguities directly into the Kalman filter as vehicle states, but this significantly increases the system's order [104]. Furthermore, the two main advantages of loosely coupled integration are simplicity and redundancy. The architecture is simple in that it works with any INS and any GNSS user equipment, making it ideal for retrofit applications. A separate GNSS navigation solution is usually available in addition to the integrated solution in a loosely coupled architecture. There is an independent INS solution that employs open-loop INS correction. This enables simple integrity monitoring of parallel solutions.

Positions and velocities derived from GPS are thus excellent external measurements that can be used to update the INS with position parameters, improving long-term accuracy. However, due to line of sight, GPS signal accuracy decreases on urban roads and tunnels; at that time, the INS can provide precise position and velocity data for GPS signal acquisition

and reacquisition after outages. In general, the availability of redundant measurements for determining vehicle trajectory parameters greatly improves system reliability. It has long been a fundamental truth in safety-related applications that relying on a single navigation technique is dangerous. For maximum safety, complementary navigation systems should have disjoint error mechanisms. The resulting system design is then driven by a cost-performance trade-off. The section that follows describes the most common errors encountered when using IMU.

4.4 Sources of Error

Low-cost inertial navigation is a difficult task. To some extent, all accelerometers and gyroscopes exhibit biases, scale factor and cross-coupling errors, and random noise that must be overcome. Each systematic error source has four components: a fixed contribution, a temperature-dependent variation, a run-to-run variation, and an in-run variation [103]. The fixed contribution is present every time the sensor is used and is corrected by the IMU processor using laboratory calibration data. The temperature-dependent component can also be corrected by the IMU using laboratory calibration data. The run-to-run variation results in a contribution to the error source that is different each time the sensor is used but remains constant within any run. It cannot be corrected by the IMU processor, but it can be calibrated by the INS alignment and/or integration algorithms each time the IMU is used. Finally, the contribution of in-run variation to the error source gradually changes over the course of a run. It cannot be corrected by the IMU or an alignment process. In theory, it can be corrected by integrating with other navigation sensors, but it is difficult to observe in practise. This section will go over some of these issues and propose a solution to minimize these errors.

The error estimation Kalman filter will be used to solve the problem. The necessary equations will be derived and discussed.

4.3.1 Biases

The bias is a constant error that all accelerometers and gyroscopes exhibit. It is not affected by the underlying specific force or angular rate. These are the most damaging effectors on an IMU's accuracy. The gyroscopes drift rate and accelerometer bias are small offsets that the IMU incorrectly reads and must be properly accounted for. The bias has a quadratic effect on the IMU-derived position. The biases of accelerometers and gyroscopes are not usually expressed in SI units. Accelerometer biases are expressed in terms of gravity acceleration, abbreviated to g, where 1 g = 9.80665 m s2.

Formula for error is given by; $error = \frac{1}{2} bias \cdot t^2$, Author in [105] , shows difference of error for 100 secs and 30 mins and recorded observation as shown in figure 13.

Bias (m/s^2)	Error (m) t=100 sec.	Error (m) t = 30 mins
.1	500	162000
.01	5	16200
.001	.5	1620
.0001	.05	162

Figure 12 Positional error that results from biases after a time of 100 sec and 30 mins[103]

Looking at figure 12, determining the bias is critical if any accurate measurement is to be expected. The drift rate has a similar and equally large impact on a system's position. If a drift is not properly accounted for and the IMU believes it is rotating, the navigation

equations will fail to account for gravity and the system will believe it is moving due to a maximum acceleration of 9.8 depending on how far the system has drifted.

4.3.2 Scale factors

The scale factor error is the deviation of the instrument's input-output gradient from unity after unit conversion by the IMU. Figure 13(a) shows vector-based scale factor errors in the form of an accelerometer and a gyroscope in an IMU. The scale factor error in the accelerometer output is proportional to the true specific force along the sensitive axis, whereas the scale factor error in the gyroscopes output is proportional to the true angular rate about the sensitive axis.

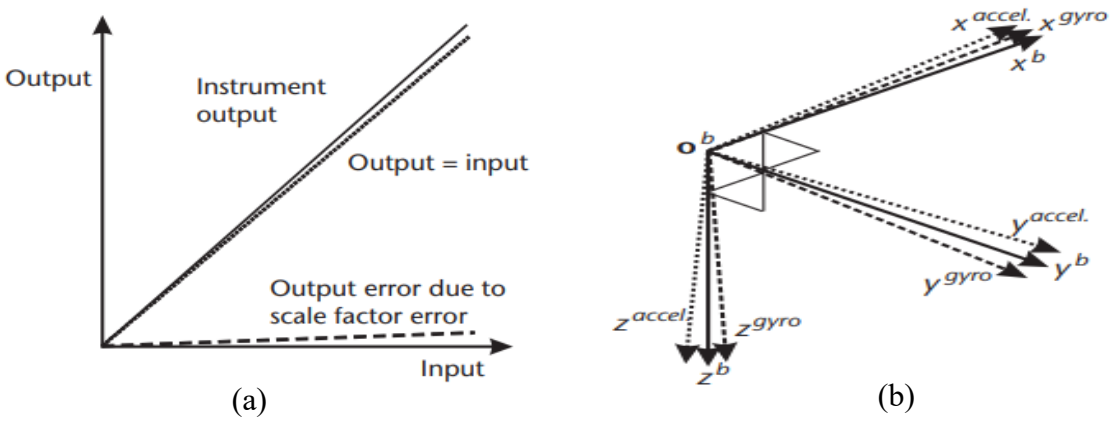


Figure 13 Scale factor error is shown in (a) & Misalignment shown in (b) [103]

4.3.3 Misalignment

Cross-coupling errors, also referred as misalignment errors, occur in all types of IMUs due to manufacturing limitations that cause the sensitive axes of the inertial sensors to be misaligned with respect to the orthogonal axes of the body frame, as illustrated in figure 13(b). Each accelerometer is sensitive to the specific force along the axes orthogonal to its sensitive axis, and each gyroscope is sensitive to the angular rate about the axes orthogonal

to its sensitive axis as a result of this. Additional scale factor errors are produced by axes misalignment, but these are typically two to four orders of magnitude smaller than cross-coupling errors. Cross-coupling errors in vibratory sensors can also occur because of sensor crosstalk.

4.3.4 Temperature

The accelerometers and gyroscope in the IMU are temperature sensitive, as demonstrated by Nebot and Durran-Whyte [10]. As a result, as the temperature of the IMU changes, so will the associated bias and drift until the temperature reaches a steady state or remains constant. This is not critical in our application; we simply wait for the IMU to stabilise before relying on the readings. This would be a problem if this system was installed in an aircraft that changed altitude and temperature.

4.5 Coordinate systems

This section describes how data from vehicle-mounted sensors is processed and converted into an estimate of the vehicle's position, velocity, and attitude, as well as how it is exchanged with the system's interfacing information sources. The coordinate system must be defined when describing a position on or near the Earth's surface. Inertial navigation systems use coordinate systems such as the inertial frame (i-frame), the conventional terrestrial frame (e-frame), the navigation frame (n-frame), and the body frame (b-frame). More information and preliminary derivations about coordinate systems can be found in [106-110].

4.5.1 Inertial coordinate system

The ECI coordinate system is the preferred inertial coordinate system for near-Earth navigation. It is the coordinate system in which Newton's laws of motion apply, according to the physical definition of inertial frame. A global inertial system is, at best, an abstraction in the real world, because any frame in the vicinity of the solar system is permeated by a gravitational field with spatially varying gradients. The ECI coordinate system's origin is at the Earth's centre of gravity (viz. a geocentric coordinate system). Its z-axis is parallel to the Earth's spin axis, its x-axis is parallel to the vernal equinox, and its y-axis completes the right-hand orthogonal coordinate system. Figure 14(a) depicts a visualisation of the ECI coordinate system, and table 1 shows the notations used in the ECI coordinate system.. The inertial sensors' accelerations and angular velocities are measured in relation to this coordinate system.

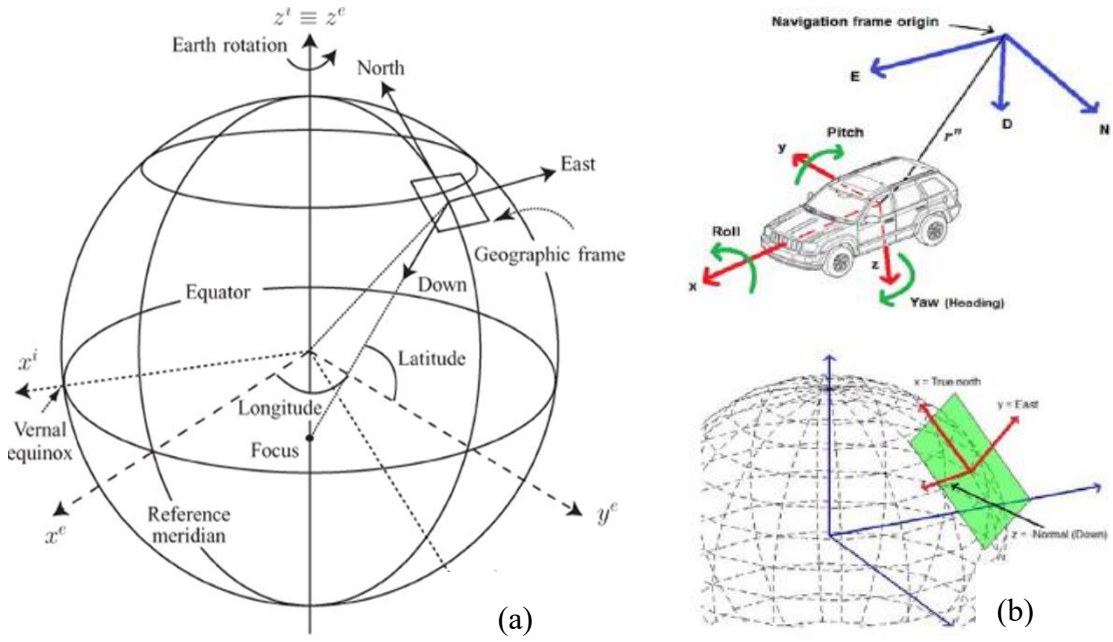


Figure 14 (a) Navigation Coordinate frames (ECI, ECEF, NED frames) and (b) shows the relationship between the vehicle frame and the coordinate system that is used as the

navigation frame (upper). The navigation frame is the local tangent frame (lower). The origin of the NED coordinate axes would be at some convenient point near the area of operation for navigation in a local tangent frame.

Table 1 Describes the axes arrangement for ECI, ECEF and NED frames

Inertial Coordinate frame (ECI coordinate frame) axes are denoted by superscript "i"	Origin - at the center of mass of the Earth z-axis - parallel to the mean spin axis of the Earth x-axis - pointing towards the mean vernal equinox y-axis - completing a right-handed orthogonal frame
Geocentric Coordinate frame (ECEF coordinate frame) axes are denoted by superscript "e"	The origin is at the Earth's centre of mass. z-axis - parallel to the mean spin axis of the Earth x-axis - pointing towards the mean meridian of Greenwich y-axis - completing a right-handed orthogonal frame
Local coordinate frame (Geographic coordinate frame) Axes are denoted by North, East and Down	Origin - at the origin of the sensor frame z-axis – aligned with the ellipsoidal normal at a point, in the down direction x-axis - pointing to north (parallel to the tangent to the meridian)

	y-axis – pointing to the east and completing a right-handed orthogonal frame.

4.5.2 Geocentric coordinate system

The world geodetic system (WGS) 84 datum defines a geodetic coordinate system, also known as the geodetic ECEF coordinate system, which is closely related to the geocentric ECEF coordinate system. A geodetic coordinate system is based on an ellipsoid that rotates around its minor axis to approximate (globally or locally) the Earth's geoid. Geocentric coordinate is depicted in Figure 14(a). A location in the coordinate system is described by the longitude and latitude angles measured with respect to the equatorial and meridional plans associated with the reference ellipsoid. The reference ellipsoid's parameters, such as shape, size, and orientation, define the coordinate system's datum. The e-frame is an Earth-Centered-Earth-Fixed Cartesian frame, and one of its realisations is the WGS-84 reference coordinate system used by GPS. Table 1 shows the notations that have been considered for geocentric coordinate systems.

4.5.3 Local coordinate system

The geographic coordinate system is a local coordinate frame whose origin is the projection of the origin of the vehicle coordinate system onto the geoid of the Earth. The x-axis points true north, the y-axis points east, and the z-axis completes the right-hand orthogonal coordinate system, pointing perpendicular to the reference ellipsoid toward the interior of the Earth. Figure 14(a) depicts the coordinate system. (It should be noted that the z-axis does not point to the centre of the Earth, but rather along the ellipsoid, normally toward

one of its foci.) When expressing the velocity components of the vehicle's motion and the attitude of the vehicle platform, the geographic coordinate system is commonly used as a reference. The three Euler angles, namely 1) roll, 2) pitch, and 3) yaw, are commonly used to describe the vehicle attitude, and they connect the vehicle and geographic coordinate systems. The n-frame is commonly used to describe a vehicle's navigation in a local coordinate frame. It is also defined as a set of Cartesian coordinate axes, typically oriented north-east-down (NED), as illustrated in Figure 14 (b). Table 1 presents notations considered for geographic coordinates.

4.5.4 *Body coordinate system*

The vehicle coordinate system, also known as body coordinates, is the coordinate system associated with the vehicle. It usually, but not always, has its origin at the vehicle's centre of gravity, and the coordinate axes are aligned with the vehicle's forward, sideways (to the right), and down directions. This coordinate system is used to express the information provided by vehicle-mounted sensors as well as the motion and dynamic constraints imposed by the vehicle model. Fig 15(b) illustrates vehicle body frame in relation to geographic coordinate frame. The b-frame refers to the vehicle to be navigated, and the conventional definition moves forward, right, and through-the-floor:

Origin - at the origin of the sensor frame

z-axis – aligned with the ellipsoidal normal at a point, in the down direction

x-axis - pointing to forward (the direction of instant motion)

y-axis – pointing to the right and completing a right-handed orthogonal frame

4.6 Navigation Equations

The navigation solution represents the navigating body's coordinate frame with respect to a reference coordinate frame (e.g., an aircraft, ship, car, or person). The Earth is a common point of reference. The vector components of the navigation solution can be resolved about the axes of a third coordinate frame (e.g., north, east, and down). The INS mechanisation equations are used to convert inertial measures (accelerations and angular rates along three orthogonal directions) from the body frame (b-fame) to navigation frame (local, or ECEF frames). A detailed description of various navigation equations in different frames can be found in [111]. However, a brief overview of inertial navigation in the ECI and ECEF frames is provided below, adapted from [112] .

4.6.1 Position and Velocity

A change in motion occurs when a force is applied to a body, according to Newton's second law of motion.

$$\frac{\boldsymbol{\mathcal{F}}}{\boldsymbol{m}} = \boldsymbol{a} = \boldsymbol{S} \tag{4.1}$$

The specific force in "(4.1)", is obtained by dividing both sides of the equation by the mass of the object. Accelerometers detect accelerations caused by forces exerted on the body in inertial navigation. These are commonly referred to as specific forces **S**. As a result, IMU readings will be referred to as specific forces that are independent of mass.

In practise, one must frequently estimate a vehicle's velocity and position with respect to a rotating reference frame, such as when navigating near the Earth. Additional apparent forces will be acting in this situation, which are functions of reference frame motion. This

yields a revised form of the navigation equation that can be directly integrated to determine the vehicle's ground speed using the Coriolis theorem [113]. According to the Coriolis theorem, the total velocity of a vehicle (v_i) in a non-rotating reference frame is equal to its ground speed (v_s) plus the added speed due to Earth rotation. The non-rotating inertial frame is given by;

$$\boldsymbol{v_i = v_s + \Omega_{ie} \, X \, r_i} \tag{4.2}$$

ECI frame is denoted by the subscript (i) in "(4.2)", and rotation of the Earth is $\Omega_{ie} = [0 \; 0 \; \omega_{ie}]^T$ and r is the vehicles position. Now differentiating this with respect to the inertial reference frame.

$$\boldsymbol{\dot{v}_i = \dot{v}_s + \dot{\Omega}_{ie} \, X \; r_i + \Omega_{ie} \, X \; v_i} \tag{4.3}$$

The rotation of the earth is constant therefore its derivative is zero. Now substituting for the velocity from the original equation we get.

$$\mathbf{\dot{v}_i = \dot{v}_s + \dot{\Omega}_{ie} \, X \; v_s + \Omega_{ie} \, X \, [\Omega_{ie} \, X \, r_i]} \tag{4.4}$$

The acceleration term on the left-hand side of the equation can be replaced with all of the accelerations in the system

$$\boldsymbol{S^i + g^i = \dot{v}_s + \dot{\omega}_{ie} \, X \; v_s + \Omega_{ie} \, X \, [\Omega_{ie} \, X \, r_i]} \tag{4.5}$$

Where S^i is the specific force (i.e. all of the accelerations seen by the IMU) and g^i is the gravity model of the system in "(4.5)". Finally solving for the acceleration of the vehicle results in the navigation equations which when integrated will turn IMU accelerations into velocities

$$\dot{\boldsymbol{v}}_{\boldsymbol{s}} = \boldsymbol{S}^{\boldsymbol{i}} - \boldsymbol{\Omega}_{\boldsymbol{ie}} \mathbf{X}\ \boldsymbol{v}_{\boldsymbol{s}} + \boldsymbol{g}^{\boldsymbol{i}} - \boldsymbol{\Omega}_{\boldsymbol{ie}} \mathbf{X} [\boldsymbol{\Omega}_{\boldsymbol{ie}} \mathbf{X}\, \boldsymbol{r}_{\boldsymbol{i}}] \tag{4.6}$$

where the term subtracted from the specific force is the Coriolis acceleration and the term subtracted from the gravity is the centripetal acceleration. Although these terms are small, as we will see, small constant acceleration offsets produce a velocity ramp when integrated. This in turn produces a quadratic position change when the velocity ramp is integrated. Now these equations, which were derived in the inertial frame, must be transformed into the ECEF (Earth) reference frame

$$\dot{\boldsymbol{v}}_{\boldsymbol{e}} = \dot{\boldsymbol{v}}_{\boldsymbol{s}} - \boldsymbol{\Omega}_{\boldsymbol{ie}} \mathbf{X}\, \boldsymbol{v}_{\boldsymbol{e}} \tag{4.7}$$

Thus, Inertial equation becomes

$$\dot{\boldsymbol{v}}_{\boldsymbol{s}} = \boldsymbol{S}_{\boldsymbol{e}} - 2\boldsymbol{\Omega}_{\boldsymbol{ie}} \mathbf{X}\ \boldsymbol{v}_{\boldsymbol{e}} + \boldsymbol{g}_{\boldsymbol{e}} - \boldsymbol{\Omega}_{\boldsymbol{ie}} \mathbf{X} [\boldsymbol{\Omega}_{\boldsymbol{ie}} \mathbf{X}\, \boldsymbol{r}_{\boldsymbol{e}}] \tag{4.8}$$

4.6.2 Orientation

Consider now how a set of gyroscopic sensors could be used to instrument a reference co-ordinate frame within a freely rotating vehicle. The vehicle's attitude with respect to the designated reference frame may be stored in the vehicle's computer as a series of numbers. The stored attitude is updated as the vehicle rotates using the gyroscopes' turn rate measurements. Numerous mathematical representations exist for defining a body's orientation in relation to a co-ordinate reference frame [106]. The parameters associated with each method can be stored in a computer and updated in real time as the vehicle rotates using the turn rate measurements provided by the gyroscopes.

Quaternions will be used to represent the attitude of the IMU (and thus the vehicle itself, as the IMU is bolted on to the vehicle). A quaternion is a way to represent the orientation of a rigid body as well as a rotation about an arbitrary axis in space. A quaternion is a complex number that has both a real and an imaginary component. The imaginary part of a quaternion, on the other hand, has three values while the real part only has one. An object's attitude is assumed to be represented by a quaternion, with the imaginary part consisting of elements 1-3 and the real part consisting of the fourth element.

$$\mathbf{q} = [\mathbf{q_{1-3}}\ \mathbf{q_4}]^{\mathbf{T}} \tag{4.9}$$

$$\mathbf{q_{1-3}} = [\mathbf{q_1}\ \mathbf{q_2}\ \mathbf{q_3}]^{\mathbf{T}} = \hat{\mathbf{n}} \cdot \mathbf{sin}\left(\frac{\boldsymbol{\theta}}{\mathbf{2}}\right) \tag{4.10}$$

$$\mathbf{q_4} = \mathbf{cos}\left(\frac{\boldsymbol{\theta}}{\mathbf{2}}\right) \tag{4.11}$$

where is a $\hat{n}$ unit vector corresponding to the axis of rotation and θ is the angle of rotation. For navigation we will need to be able to convert from roll, pitch, and yaw into quaternion space and back again. This is done by the following set of equations

$$\mathbf{q} = \begin{bmatrix} \mathbf{c_2^{\psi} c_2^{\theta} c_2^{\phi}} & - & \mathbf{s_2^{\psi} s_2^{\theta} c_2^{\phi}} \\ \mathbf{c_2^{\psi} s_2^{\theta} c_2^{\phi}} & + & \mathbf{s_2^{\psi} c_2^{\theta} s_2^{\phi}} \\ \mathbf{c_2^{\psi} c_2^{\theta} c_2^{\phi}} & - & \mathbf{s_2^{\psi} s_2^{\theta} s_2^{\phi}} \\ \mathbf{s_2^{\psi} c_2^{\theta} c_2^{\phi}} & + & \mathbf{c_2^{\psi} s_2^{\theta} s_2^{\phi}} \end{bmatrix} \tag{4.12}$$

$$\begin{bmatrix} \tan\phi \\ \sin\theta \\ \tan\psi \end{bmatrix} = \begin{bmatrix} \frac{2q_1q_2 + 2q_4q_3}{2q_4^2 + 2q_1^2 - 1} \\ 2q_4q_2 + 2q_1q_3 \\ \frac{2q_2q_3 + 2q_4q_1}{2q_4^2 + 2q_3^2 - 1} \end{bmatrix} \tag{4.13}$$

Once the attitude is put into quaternions, the kinematic differential equations for attitude rotation is given as follows

$$\dot{q} = \frac{1}{2}\Omega q \tag{4.14}$$

$$\Omega = \begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & \omega_y & -\omega_z & 0 \end{bmatrix} \tag{4.15}$$

This equation allows readings from the gyros to be used to update the current quaternion, which represents the attitude of the system.

Quaternions have the following useful properties for use in navigational systems:

- They have a high computational efficiency. Subsequent rotations (for example, an x-axis rotation followed by a y-axis rotation) are accomplished by multiplying a **4x4 by 4x1**. Two subsequent rotations in Euler space would be handled by multiplying a **3x3-by-3x3** matrix.
- Quaternions take up less memory to represent. A rotation (3x3 matrix) is represented by nine Euler angles, whereas a quaternion requires only four. This is a weakness, but some embedded applications have very limited memory requirements.
- A quaternion's time propagation always yields an orthogonal rotation matrix. An IMU will provide angular rates from its gyros, which will be noisy. If the IMU is rotated 90 degrees in a plane, the data integration from the gyros should properly reflect this rotation. Due to the noise, the IMU will report rotations on all axes. The real issue is that

the resulting 3x3 rotation matrix is no longer guaranteed to be orthogonal, which means that the rotation matrix's inverse is not equal to the transpose. To make the rotation matrix orthogonal again, extra steps must be taken at each time step.

By summarizing navigation equations derived above we have:

$$\begin{aligned} \dot{\mathbf{x}}_e &= \mathbf{v}_e \\ \dot{\mathbf{v}}_e &= -\mathbf{2}[\boldsymbol{\omega}_{ie}\times]\mathbf{v}_e + \mathbf{g}_e(\mathbf{x}_e) + \mathbf{C}_{es}(\breve{\mathbf{q}}_{es})\mathbf{f}_s \\ \dot{\breve{\mathbf{q}}}_{es} &= \frac{\mathbf{1}}{\mathbf{2}}(\breve{\mathbf{q}}_{es} * \breve{\boldsymbol{\omega}}_{is} - \breve{\boldsymbol{\omega}}_{ie} * \breve{\mathbf{q}}_{es}) \end{aligned} \tag{4.16}$$

For further understanding on inertial navigation equation readers are encouraged to refer [112].

All the quantities in "(4.16)" are assumed to be error-free. As a result, the vectors $\mathbf{v}_e$ and $\mathbf{x}_e$ represent the vehicle's true velocity and position, respectively, while the quaternion $\breve{\mathbf{q}}_{es}$ defines the rotation from the s- to the e-frame without errors.

In general, estimates of position, velocity and attitude are not error free. A bias in accelerometer measurements causes error growth proportional to the square of the operation time for an INS, whereas a bias in gyroscope measurements causes error growth proportional to the cube of the operation time. Because they have a direct reflection on the estimated attitude, gyroscope errors have a negative impact on navigation solutions. The attitude is used to compute the current gravity force in navigation coordinates and to compensate for the effect on accelerometer measurements. Because vehicle accelerations in most land vehicle applications are significantly lower than gravity accelerations, small errors in attitude can lead to large errors in estimated accelerations. These errors are added together in the velocity and position calculations. In the following sections we present our detailed derivation for linearizing error along with sensor errors in order to be used in error estimation algorithm.

4.7 Error Estimation Algorithm

4.7.1 Kalman filter

The Kalman filter is a state-space-based method for calculating the minimum mean-squared error (MMSE). R. E. Kalman, a graduate research professor in the University of Florida's electrical engineering department, invented the filter in 1960 [114]. The Kalman filter has several advantages over other estimators, including being computationally efficient by recursively processing noisy data, being a real-time estimator, being able to adapt to non-stationary signals, handling complicated time-variable multiple-input/output systems, and vector modelling random processes.

The Kalman filter a recursive digital algorithm works by calculating the minimum mean squared error (MMSE) in state space to optimise overall system performance through the integration (or fusion) of navigation sensor data. Through feedback control, the Kalman filter estimates a process. The filter makes an estimation of the state of the process at a particular point in time and then receives feedback in the form of noisy measurements. For reference, figure 15 depicts the recursive algorithm.

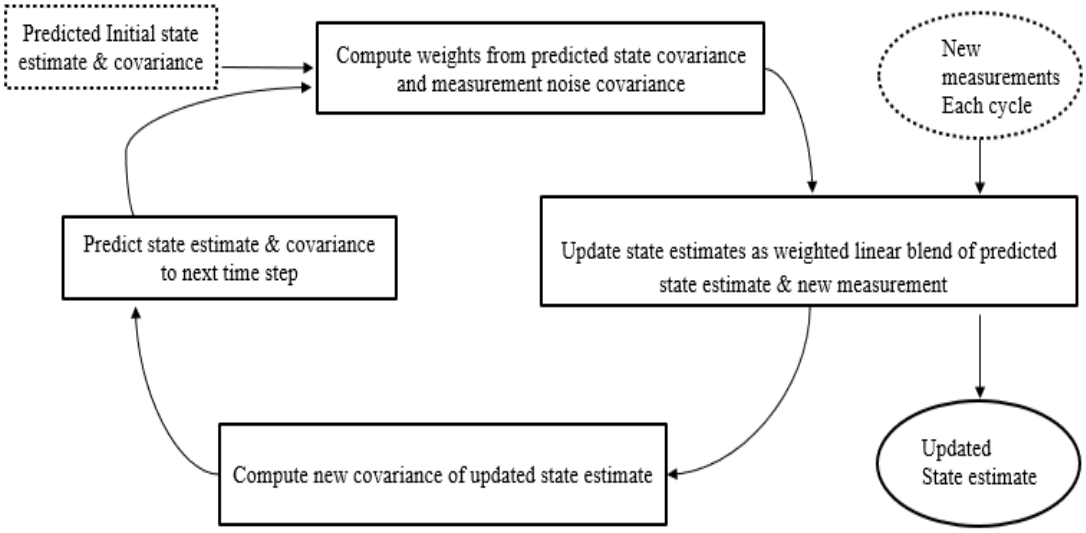


Figure 15 The Kalman filter is a recursive, linear filter. At each cycle, the state estimate is updated by combining new measurements with the predicted state estimate from previous measurements

There are two types of filter equations: time update equations and measurement update equations. The time update equations are responsible for projecting the current state and error covariance estimates forward (in time) to obtain a priori estimates for the subsequent time step. The measurement update equations are responsible for adding a new measurement to the a priori estimate to obtain a more accurate a posteriori estimate. While time update equations are referred to as predictor equations, measurement update equations are referred to as corrector equations.

To provide current estimates for system variables such as position coordinates, the filter employs statistical models that weight each new measurement appropriately in relation to previous data. The formulation accounts for the second order statistics of both the process and measurement noise, both of which have a known probability distribution [61]. Using knowledge of probability distributions, an optimal, closed form solution for estimation error and minimum covariance is developed and implemented in the control state space

framework. Additionally, it determines the most recent uncertainty in estimates for real-time quality assessments or off-line system design studies. Due to its superior performance, versatility, and ease of implementation, the Kalman filter is particularly popular in GPS/inertial and GPS stand-alone devices

The following subsections are organised to provide context for derivations from peer-reviewed books. These peer-reviewed derivations are necessary to explain in order to assist the reader in understanding and following the state space representation model blocks used in our current research experiment. These model blocks are then used in the Kalman filter to estimate errors and improve position accuracy. The subsection begins by discretizing stochastic linear systems into discrete homogeneous time systems and establishing the equivalence of discrete and continuous time models. Then, a Kalman filter algorithm is presented. This research contribution is presented in which we build state space models for real-time experiments using derivations adapted from peer-reviewed books. We assume IMU errors and linearize navigation errors. Because the GPS measurement data are related, we decorrelate and transform them to simplify real-time implementation. After that, the Kalman algorithm is used to estimate the error using all of these state space model blocks.

4.7.2 Discretization of continuous systems

We begin with presenting dynamic system of equations shown below which serves as the foundation for the Kalman filter equations [115].

Consider linear stochastic system

$$\dot{\mathbf{z}} = \mathbf{A}(t)\mathbf{z} + \mathbf{G}(t)\mathbf{q} \tag{4.17}$$

$$\boldsymbol{y(t) = C(t)z + \vartheta(t)} \tag{4.18}$$

where **z** is a **n x 1** dynamic state vector in "(4.17)", **A** is a **n x n** dynamic coefficient matrix, **G** is a process noise distribution matrix, where **q** is zero-mean Gaussian white noise input with covariance $\mathbf{Q}(t)$. In general, process noise is caused by disturbances in the environment and is assumed to have a Gaussian distribution. The measured output of the system is given by "(4.18)", where **y(t)** is the **p x 1** measurement vector, $\boldsymbol{C}$ is the **p x n** measurement matrix and $\boldsymbol{\vartheta}$ is the measurement noise vector. Sensor dynamics and error introduce measurement noise into the observation model. One of the key assumptions is that the process noise and measurement noise are both Gaussian random variables.

As per [116], continuous system "(4.17)" and "(4.18)", is equivalent to the discrete system and can be rewritten into "(4.19)" and "(4.20)"

$$\mathbf{z_k = \Phi_{k+1,k} z_k + q_k^*} \tag{4.19}$$

$$\boldsymbol{y_k = C.z_k + \vartheta_k} \tag{4.20}$$

Where, $\boldsymbol{\Phi_{k+1,k} = \Phi(t_{k+1}, t_k)}$ is a state transition matrix of the system "(4.17)". In [116], the derivation for the **Φ** and theory are presented, where the unique (identity) solution of the system is shown to be a property of the matrix **Φ**, as shown in "(4.21)".

$$\boldsymbol{\frac{d\Phi(t, t_0)}{dt} = A(t)\Phi(t, t_0), \Phi(t_0, t_0) = I} \tag{4.21}$$

The estimated state and output are given as the estimated state vector (4.22) and the estimated output as (4.23)

$$\hat{\boldsymbol{z}}_{\boldsymbol{k}} = \boldsymbol{\Phi}_{\boldsymbol{k+1,k}}\mathbf{z}_{\boldsymbol{k}} + \boldsymbol{K\epsilon} \tag{4.22}$$

$$\hat{\boldsymbol{y}}_{\boldsymbol{k}} = \boldsymbol{C}.\boldsymbol{z}_{\boldsymbol{k}} \tag{4.23}$$

Where, $\hat{\mathbf{z}}_{\mathrm{k}}$ is the **n x 1** estimated state vector at time k in "(4.22)" and $\mathbf{K} \in \mathbf{R}^{\mathbf{n\,x\,p}}$ is the error gain or Kalman gain and $\boldsymbol{\epsilon}$ in "(4.22)", is called output error. $\hat{\boldsymbol{y}}_{\boldsymbol{k}}$ in "(4.23)" is the estimated output at time k.

Where error can be defined as

$$\boldsymbol{\epsilon} = \mathbf{y}_{\mathbf{k}} - \hat{\mathbf{y}}_{\mathbf{k}} \tag{4.24}$$

Similarly we see in [117], where estimation error at time k can be written as

$$\tilde{\boldsymbol{z}}_{\boldsymbol{k}} = \boldsymbol{z}_{\mathbf{k}} - \hat{\boldsymbol{z}}_{\mathbf{k}} \tag{4.25}$$

Substituting equation "(4.19)" and "(4.22)" in "(4.25)" we have,

$$\tilde{\boldsymbol{z}}_{\boldsymbol{k}} = \boldsymbol{\Phi}_{\boldsymbol{k+1,k}}\mathbf{z}_{\boldsymbol{k}} + \mathbf{q}_{\boldsymbol{k}}^{*} - (\boldsymbol{\Phi}_{\boldsymbol{k+1,k}}\mathbf{z}_{\boldsymbol{k}} + \boldsymbol{K\epsilon}) \tag{4.26}$$

where $\tilde{\boldsymbol{z}}_{\boldsymbol{k}}$ in "(4.26)" is the estimation error at time k. Similarly, we can then substitute $\hat{\mathbf{y}}_{\mathbf{k}}$ from "(4.23)" and $\boldsymbol{y}_{\mathbf{k}}$ from "(4.20)", in $\boldsymbol{\epsilon}$ in "(4.24)", and use this to obtain the expression in Equation "(4.26)". We have estimation error as,

$$\tilde{\boldsymbol{z}}_{\boldsymbol{k}} = (\mathbf{I} - \mathbf{KC})\boldsymbol{\Phi}_{\boldsymbol{k+1,k}}\mathbf{z}_{\boldsymbol{k}} + (\mathbf{I} - \mathbf{KC})\mathbf{q}_{\boldsymbol{k}}^{*} - \boldsymbol{K\vartheta}_{\boldsymbol{k}} \tag{4.27}$$

Estimation variance, is defined as

$$\boldsymbol{P}_{\boldsymbol{k}} = \boldsymbol{E}(\tilde{\boldsymbol{z}}_{\boldsymbol{k}}\tilde{\boldsymbol{z}}_{\boldsymbol{k}}^{\boldsymbol{T}}) \tag{4.28}$$

As shown in [115], based on the assumption of Gaussian noise and uncorrelated measurement and process noise we have:

$$\mathbf{E}(\boldsymbol{v_k}\boldsymbol{v_k}^{\boldsymbol{T}}) = \boldsymbol{R_k} \tag{4.29}$$

$$\mathbf{E}(\mathbf{q}_{\boldsymbol{k}}^{*}\mathbf{q}_{\boldsymbol{k}}^{*\boldsymbol{T}}) = \boldsymbol{Q_k} \tag{4.30}$$

$$\mathbf{E}(\mathbf{q}_{\boldsymbol{k}}^{*}\boldsymbol{v_k}^{\boldsymbol{T}}) = \mathbf{E}(\tilde{\boldsymbol{z}}_{\boldsymbol{k}}\boldsymbol{v_k}^{\boldsymbol{T}}) = \mathbf{E}(\tilde{\boldsymbol{z}}_{\boldsymbol{k}}\boldsymbol{v_k}^{\boldsymbol{T}}) = \mathbf{0} \tag{4.31}$$

In [118], Gaussian noise and its derivation is explained. The discrete time process noise matrix, which has a continuous time analogue **Q**, is represented by Equation "(4.30)". To implement INS error estimation in the form of a discrete Kalman filter, a system transition matrix and discrete noise covariance matrices are required. A short proof of derivation is presented in [116], where solution of "(4.19)", is shown to take form

$$\mathbf{z}(\boldsymbol{t}) = \boldsymbol{\Phi}(\boldsymbol{t}, \boldsymbol{t_0})\mathbf{z}(\boldsymbol{t_0}) + \int_{\boldsymbol{t_0}}^{\boldsymbol{t}} \boldsymbol{\Phi}(\boldsymbol{t}, \boldsymbol{\tau})\mathbf{G}(\boldsymbol{\tau})\mathbf{q}(\boldsymbol{\tau})\boldsymbol{d\tau} \tag{4.32}$$

Therefore, the term $\mathbf{q}_k^*$ in "(4.19)", can be written as

$$\mathbf{q}_{\boldsymbol{k}}^{*} = \int_{\boldsymbol{t_k}}^{\boldsymbol{t_{k+1}}} \boldsymbol{\Phi}(\boldsymbol{t_{k+1}}, \boldsymbol{\tau})\mathbf{G}(\boldsymbol{\tau})\mathbf{q}(\boldsymbol{\tau})\boldsymbol{d\tau} \tag{4.33}$$

It is possible to show that the sequence $\mathbf{q}_k^*$ presents zero-mean Gaussian white noise with covariance

$$\mathbf{Q}_{\boldsymbol{k}}^{*} = \int_{\boldsymbol{t_k}}^{\boldsymbol{t_{k+1}}} \boldsymbol{\Phi}(\boldsymbol{t_{k+1}}, \boldsymbol{\tau})\mathbf{G}(\boldsymbol{\tau})\mathbf{Q}(\boldsymbol{\tau})\mathbf{G}^{\mathbf{T}}(\boldsymbol{\tau})\boldsymbol{\Phi}^{\mathbf{T}}(\boldsymbol{t_{k+1}}, \boldsymbol{\tau})\boldsymbol{d\tau} \tag{4.34}$$

Where $\boldsymbol{\Phi(t_{k+1}, \tau)} \approx \mathbf{I} + \boldsymbol{(t_{k+1} - t_k)} \cdot \mathbf{A}(\boldsymbol{t_k})$, by approximating the integral in "(4.34)", we have,

$$\mathbf{Q}_{\boldsymbol{k}}^{*} \approx \boldsymbol{(t_{k+1} - t_k)} \cdot \mathbf{G}(\boldsymbol{t_k})\mathbf{Q}(\boldsymbol{t_k})\mathbf{G}^{\mathbf{T}}(\boldsymbol{t_k}) \tag{4.35}$$

The formula "(4.35)", is further simplified by approximating the integral in "(4.34)", by application of rectangle rule of integration. Notes on application of rectangle rule can be found in [119]. Equation "(4.35)", is also seen to relate equivalency between discrete time models to continuous time systems.

After system transition and noise covariance are discretized, next step is to derive Kalman gain. From [103], it can be seen by squaring "(4.27)", taking the expectation, and substituting equation "(4.31)", into the term for $\boldsymbol{P_k}$ "(4.28)", the covariance of the estimation error is given by,

$$\boldsymbol{P_k} = (\mathbf{I} - \mathbf{KC})\boldsymbol{M_k}\,(\mathbf{I} - \mathbf{KC})^{\boldsymbol{T}} + \boldsymbol{K R_k K^T} \tag{4.36}$$

Where $\mathbf{M_k} = \mathbf{\Phi P_{k-1} \Phi^T} + \mathbf{Q_k}$ is often termed as the priori covariance. The next step is to find K which is found through solving "(4.36)", with $\left[\frac{\boldsymbol{\partial[(P_k)]}}{\boldsymbol{\partial K}}\right] = \mathbf{0}$,

$$\mathbf{K} = \mathbf{M_k C}\,(\mathbf{C M_k C^T} + \mathbf{R_k})^{\mathbf{-1}} \tag{4.37}$$

And finally updated covariance as shown in "(4.38)", is found using the new gain from "(4.37)",

$$\boldsymbol{P_k} = (\mathbf{I} - \mathbf{KC})\boldsymbol{M_k} \tag{4.38}$$

In Table 2, Kalman estimate, and update equations are summarized for the recursive process.

Table 2 Summary of algorithm for the Kalman filter is presented in table

$\hat{\boldsymbol{z}}(\mathbf{0}) = \hat{\boldsymbol{z}}_{\mathbf{0}}, \boldsymbol{P}(\mathbf{0}) = \boldsymbol{P}_{\mathbf{0}}$	Initialize state estimate and covariance
$\hat{\boldsymbol{z}}_{\boldsymbol{k}}^{-\mathbf{1}} = \boldsymbol{\Phi}_{\boldsymbol{k+1,k}} \hat{\mathbf{Z}}_{\boldsymbol{k}}^{+}$	Predict the next state estimate
$\mathbf{M_k} = \boldsymbol{\Phi} \mathbf{P_{k-1}} \boldsymbol{\Phi}^{\mathbf{T}} + \mathbf{Q_k}$	Predict the covariance
$\mathbf{K} = \mathbf{M_k C} \left(\mathbf{C M_k C^T} + \mathbf{R_k}\right)^{-\mathbf{1}}$	Solve for the Kalman Gain
$\boldsymbol{P_k} = (\mathbf{I} - \mathbf{KC}) \boldsymbol{M_k}$	Update the covariance based on Kalman gain
$\hat{\boldsymbol{z}}_{\boldsymbol{k}}^{+} = \hat{\boldsymbol{z}}_{\boldsymbol{k}} + \boldsymbol{K}(\mathbf{y_k} - \boldsymbol{C}\hat{\boldsymbol{z}}_{\boldsymbol{k}}^{-})$	Find the new state estimate given measurement $\mathbf{y_k}$

4.7.3 Process model and real time application

We construct a state space model to estimate the scale factors and biases of the IMU, as well as the (sensor-frame) alignment of the IMU with the vehicle's coordinate system. To begin, the navigation state is supplemented with the IMU's scale factors and biases, as well as the alignment between the IMU frame and the car's fixed coordinate frame, and later, using vehicle motion constraints, a coupling between the smartphone dynamics and the smartphone's relative orientation to the vehicle is introduced. Additionally, input from the IMU's gyroscope and accelerometer provide angular velocity and specific force, which are augmented with navigation equations and propagated to yield velocity and position. We also

have external position information from GPS, which helps with calibration of the INS solution by correcting measurement drifts and enhancing smoother solutions, which are then included in the Kalman filter (KF) and integrated with the inertial navigation system (velocity and position). This process is repeated at each epoch, yielding estimates of velocity and position. Figure 16 depicts a navigation filter that was used in the study.

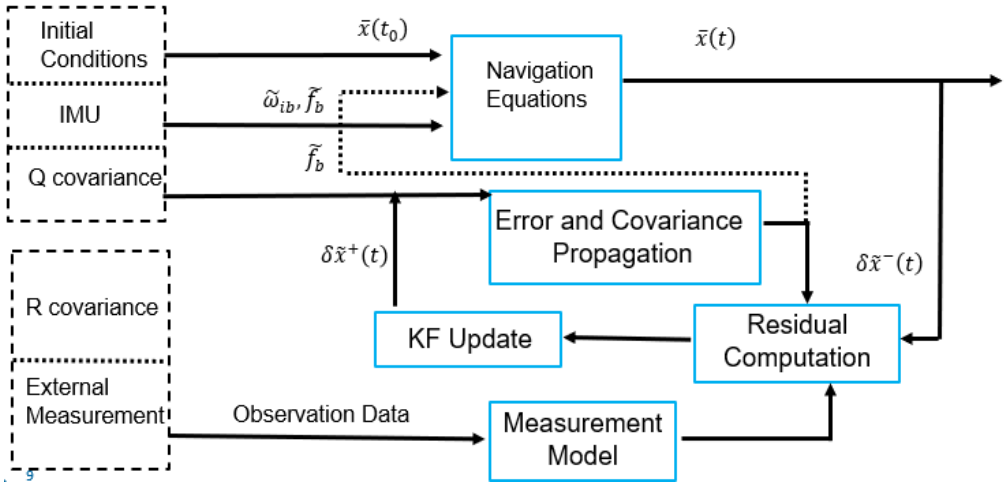


Figure 16 Applied navigation filter for error estimation

4.7.3.1 State vector

We'd like to estimate sensor compensation parameters like scale factors and biases in the navigation filter. The core INS state vector must then be supplemented with the corresponding parameters. We will also estimate the alignment between the IMU frame (s-frame) and the car-fixed coordinate frame. Finally, the state vector will take the following form:

$$\widetilde{\boldsymbol{Z}} = [(\boldsymbol{X_e} \quad \boldsymbol{V_e} \quad \boldsymbol{q_{es}})^T]^T \tag{4.39 (a)}$$

$$\mathbf{U} = [(\boldsymbol{a_{is}^s} \quad \boldsymbol{\omega_{is}^s})^T]^T \tag{4.39 (b)}$$

$$\mathbf{z} = \left((\mathbf{x_e})^{\mathbf{T}} \quad (\mathbf{v_e})^{\mathbf{T}} \quad (\mathbf{q_{es}})^{\mathbf{T}} \quad \left(\mathbf{b_f}\right)^{\mathbf{T}} \quad (\mathbf{b_\omega})^{\mathbf{T}} \quad \left(\mathbf{s_f}\right)^{\mathbf{T}} \quad (\mathbf{s_\omega})^{\mathbf{T}} \quad (\mathbf{q_{cs}})^{\mathbf{T}}\right)^{\mathbf{T}} \tag{4.39}$$

Where "Equation 4.39(a)", presents the navigation equation with $(x_e)^T$ and $(v_e)^T$ denote three-dimensional position and velocity respectively, and the $(q_{es})^T$is the quaternion defining change of frame from sensor to earth. The superscript $(.)^T$ denotes the transpose of a matrix. Further U in "4.39(b)" is the input vector, where a_{is}^{s} is the specific force and ω_{is}^{s} is angular velocity of IMU. In "(4.39)", parameters from "4.39(a)" and "4.39(b)", are combined to form an estimate vector. Where, $(q_{cs})^T$ is change of frame from sensor to car frame. And $(b_f)^T$, $(b_\omega)^T$, $(s_f)^T$, $(s_\omega)^T$ are biases and scale factor of specific force and angular velocity.

4.7.3.2 Dynamics of the enhanced states and error states

4.7.3.2.1 Linearizing error propagation for estimated state dynamics:

In real time measured by an IMU, specific force and angular rate are never ideal and deviate from true values due to measurement noise and other instrumental errors. As a result of numerically integrating these inputs, estimates of position, velocity, and orientation are slightly erroneous.The estimates are subject to

$$\begin{aligned} \dot{\tilde{\mathbf{x}}}_{\boldsymbol{e}} &= \tilde{\mathbf{v}}_{\boldsymbol{e}} \\ \dot{\tilde{\mathbf{v}}}_{\boldsymbol{e}} &= -\mathbf{2}[\boldsymbol{\omega}_{\boldsymbol{ie}} \times]\tilde{\mathbf{v}}_{\boldsymbol{e}} + \mathbf{g}_{\boldsymbol{e}}(\tilde{\mathbf{x}}_{\boldsymbol{e}}) + \mathbf{C}_{\tilde{\boldsymbol{e}}\boldsymbol{s}}(\breve{\mathbf{q}}_{\tilde{\boldsymbol{e}}\boldsymbol{s}})\tilde{\mathbf{f}}_{\boldsymbol{s}} \\ \dot{\breve{\mathbf{q}}}_{\tilde{\boldsymbol{e}}\boldsymbol{s}} &= \frac{\mathbf{1}}{\mathbf{2}}\left(\breve{\mathbf{q}}_{\tilde{\boldsymbol{e}}\boldsymbol{s}} * \breve{\tilde{\boldsymbol{\omega}}}_{\boldsymbol{is}} - \breve{\boldsymbol{\omega}}_{\boldsymbol{ie}} * \breve{\mathbf{q}}_{\tilde{\boldsymbol{e}}\boldsymbol{s}}\right) \end{aligned} \tag{4.40}$$

The only difference between "(4.16)" and "(4.40)", is the presence of tilde symbols "Tilda (~) " above the variables. These symbols emphasise that numerical integration of measured specific force and angular rate results in values for position, velocity, and orientation that deviate from reality. There are a few more words to say about the estimated

rotation from the s-frame to the e-frame. Look at the outcome of numerical integration $\widetilde{\mathbf{q}}_{es}$, all the quaternions $\widetilde{\mathbf{q}}_{es}$, $\mathbf{q}_{\tilde{e}s}$, and $\mathbf{q}_{e\tilde{s}}$ have the same value. They are denoted differently only to emphasise different physical interpretations. This rotation quaternion or the rotation matrix derived from it can be used in two ways:

To begin, this rotation can be used to convert s-frame vector coordinates to e-frame vector coordinates. However, using the estimated rotation matrix to rotate from the true sensor frame will result in coordinates being transformed into an erroneous coordinate frame $\tilde{e}$ that is turned slightly relative to the true e-frame. As a result, the quaternion $\widetilde{\mathbf{q}}_{es}$ is defined as the quaternion that defines a transformation from the true sensor frame to the computed ECEF coordinate frame. As a result, $\widetilde{\mathbf{q}}_{es} = \mathbf{q}_{\tilde{e}s}$

Second, vector coordinates from the e-frame to the s-frame can be transformed using the inverse of the quaternion $\widetilde{\mathbf{q}}_{es}$. However, using the estimated rotation matrix to rotate from the true ECEF frame will result in coordinates being transformed into an erroneous coordinate frame $\tilde{s}$ that is turned slightly relative to the true sensor frame. As a result, the quaternion $\widetilde{\mathbf{q}}_{es}$ can be defined as a quaternion that defines a transformation from the $\tilde{s}$ -frame to the true ECEF coordinate frame. As a result, $\widetilde{\mathbf{q}}_{es} = \mathbf{q}_{e\tilde{s}}$

4.7.3.2.2 Position error estimates

The linearization of INS Error state dynamic equations is demonstrated in [116]. The position and velocity error derivations below, have been adapted from the same reference.

Define position and velocity errors as

$$\begin{aligned}\mathbf{x}_e &= \tilde{\mathbf{x}}_e + \boldsymbol{\delta}\mathbf{x}_e \\ \mathbf{v}_e &= \tilde{\mathbf{v}}_e + \boldsymbol{\delta}\mathbf{v}_e\end{aligned} \tag{4.41}$$

Dynamics of the true and estimated position is given by

$$\begin{aligned}\dot{\mathbf{x}}_e &= \mathbf{v}_e \\ \dot{\tilde{\mathbf{x}}}_e &= \tilde{\mathbf{v}}_e\end{aligned} \tag{4.42}$$

Subtracting the second line in the" (4.42)", from the first yields

$$\dot{\mathbf{x}}_e - \dot{\tilde{\mathbf{x}}}_e = \mathbf{v}_e - \tilde{\mathbf{v}}_e \tag{4.43}$$

Consequently,

$$\boldsymbol{\delta}\dot{\mathbf{x}}_e = \boldsymbol{\delta}\mathbf{v}_e \tag{4.44}$$

4.7.3.2.3 Velocity error estimates

Define an error of the measured specific force as

$$\mathbf{f}_s = \tilde{\mathbf{f}}_s + \boldsymbol{\delta}\mathbf{f}_s \tag{4.45}$$

Consider the estimated rotation matrix $\mathbf{C}_{\tilde{e}s}$ (derived from quaternion). The true matrix $\mathbf{C}_{es}$ is related to the estimates as

$$\mathbf{C}_{es} = \mathbf{C}_{e\tilde{e}}\mathbf{C}_{\tilde{e}s} \tag{4.46}$$

Herein $\mathbf{C}_{e\tilde{e}}$ is the matrix that defines transformation from the estimated ECEF frame to the true one. If errors are small, then

$$\mathbf{C}_{e\tilde{e}} \approx \mathbf{I} + [\boldsymbol{\psi}_{e\tilde{e}} \times] \tag{4.47}$$

and the orientation error vector $\boldsymbol{\psi}_{e\tilde{e}}$ can be understood as a vector of "small" Euler angles between the coordinate frames $\tilde{e}$ and e. Dynamics of the true and estimated position is subject to

$$\begin{aligned} \dot{\mathbf{v}}_e &= -\mathbf{2}[\boldsymbol{\omega}_{ie} \times]\mathbf{v}_e + \mathbf{g}_e(\mathbf{x}_e) + \mathbf{C}_{es}\mathbf{f}_s \\ \dot{\tilde{\mathbf{v}}}_e &= -\mathbf{2}[\boldsymbol{\omega}_{ie} \times]\tilde{\mathbf{v}}_e + \mathbf{g}_e(\tilde{\mathbf{x}}_e) + \mathbf{C}_{\tilde{e}s}\tilde{\mathbf{f}}_s \end{aligned} \tag{4.48}$$

Subtracting the second line in the "(4.48)", from the first yields

$$\begin{aligned} \dot{\mathbf{v}}_e - \dot{\tilde{\mathbf{v}}}_e = \boldsymbol{\delta}\dot{\mathbf{v}}_e = &-\mathbf{2}[\boldsymbol{\omega}_{ie} \times]\mathbf{v}_e + \mathbf{g}_e(\mathbf{x}_e) + \mathbf{C}_{es}\mathbf{f}_s \\ &+ \mathbf{2}[\boldsymbol{\omega}_{ie} \times]\tilde{\mathbf{v}}_e - \mathbf{g}_e(\tilde{\mathbf{x}}_e) - \mathbf{C}_{\tilde{e}s}\tilde{\mathbf{f}}_s \end{aligned} \tag{4.49}$$

Employing error definitions allows to decompose true quantities in "(4.49)", into combinations of estimates/measurements and errors. Consequently,

$$\begin{aligned} \boldsymbol{\delta}\dot{\mathbf{v}}_e = &-\mathbf{2}[\boldsymbol{\omega}_{ie} \times](\tilde{\mathbf{v}}_e + \boldsymbol{\delta}\mathbf{v}_e) + \mathbf{g}_e(\tilde{\mathbf{x}}_e + \boldsymbol{\delta}\mathbf{x}_e) \\ &+ (\mathbf{I} + [\boldsymbol{\psi}_{e\tilde{e}} \times])\mathbf{C}_{\tilde{e}s}\left(\tilde{\mathbf{f}}_s + \boldsymbol{\delta}\mathbf{f}_s\right) + \mathbf{2}[\boldsymbol{\omega}_{ie} \times]\tilde{\mathbf{v}}_e \\ &- \mathbf{g}_e(\tilde{\mathbf{x}}_e) - \mathbf{C}_{\tilde{e}s}\tilde{\mathbf{f}}_s \end{aligned} \tag{4.50}$$

Doing Taylor series expansion of $\mathbf{g}_e(\mathbf{x}_e)$ and neglecting products of errors as second order terms allows to reduce "(4.50)" to

$$\boldsymbol{\delta}\dot{\mathbf{v}}_e = \left.\frac{\partial \mathbf{g}_e(\mathbf{x}_e)}{\partial(\mathbf{x}_e)^{\mathrm{T}}}\right|_{\tilde{\mathbf{x}}_e} \cdot \boldsymbol{\delta}\mathbf{x}_e - \mathbf{2}[\boldsymbol{\omega}_{ie} \times] \cdot \boldsymbol{\delta}\mathbf{v}_e - \left[\left(\mathbf{C}_{\tilde{e}s}\tilde{\mathbf{f}}_s\right) \times\right] \cdot \boldsymbol{\psi}_{e\tilde{e}} + \mathbf{C}_{\tilde{e}s} \cdot \boldsymbol{\delta}\mathbf{f}_s \tag{4.51}$$

4.7.3.2.4 Orientation error estimates

Rearrange the definition "(4.46)", we have

$$\mathbf{C}_{es}\mathbf{C}_{\tilde{e}\tilde{s}}^{\mathbf{T}} = \mathbf{C}_{e\tilde{e}} \approx \mathbf{I} + [\boldsymbol{\psi}_{e\tilde{e}} \times] \tag{4.52}$$

Differentiating the left and the right parts of the equality "(4.52)" yields

$$\left[\dot{\boldsymbol{\psi}}_{e\tilde{e}} \times\right] = \dot{\mathbf{C}}_{es}\mathbf{C}_{\tilde{e}\tilde{s}}^{\mathbf{T}} + \mathbf{C}_{es}\dot{\mathbf{C}}_{\tilde{e}\tilde{s}}^{\mathbf{T}} \tag{4.53}$$

Then, exploiting a rule of differentiation of a rotation matrix,

$$\begin{aligned}\left[\dot{\boldsymbol{\psi}}_{e\tilde{e}} \times\right] = (&\mathbf{C}_{es}[\boldsymbol{\omega}_{is} \times] - [\boldsymbol{\omega}_{ie} \times]\mathbf{C}_{es})\mathbf{C}_{\tilde{e}\tilde{s}}^{\mathbf{T}} \\ &+ \mathbf{C}_{es}\left(-[\tilde{\boldsymbol{\omega}}_{is} \times]\mathbf{C}_{\tilde{e}\tilde{s}}^{\mathbf{T}} + \mathbf{C}_{\tilde{e}\tilde{s}}^{\mathbf{T}}[\boldsymbol{\omega}_{ie} \times]\right)\end{aligned} \tag{4.54}$$

Now true quantities need to be substituted by compositions of estimated/measured quantities and errors. Thus,

$$\begin{aligned}\left[\dot{\boldsymbol{\psi}}_{e\tilde{e}} \times\right] = \big(&(\mathbf{I} + [\boldsymbol{\psi}_{e\tilde{e}} \times])\mathbf{C}_{\tilde{e}\tilde{s}}[(\tilde{\boldsymbol{\omega}}_{is} + \boldsymbol{\delta\omega}_{is}) \times] \\ &- [\boldsymbol{\omega}_{ie} \times](\mathbf{I} + [\boldsymbol{\psi}_{e\tilde{e}} \times])\mathbf{C}_{\tilde{e}\tilde{s}}\big)\mathbf{C}_{\tilde{e}\tilde{s}}^{\mathbf{T}} \\ &+ (\mathbf{I} + [\boldsymbol{\psi}_{e\tilde{e}} \times])\mathbf{C}_{\tilde{e}\tilde{s}}\left(-[\tilde{\boldsymbol{\omega}}_{is} \times]\mathbf{C}_{\tilde{e}\tilde{s}}^{\mathbf{T}}\right. \\ &\left.+ \mathbf{C}_{\tilde{e}\tilde{s}}^{\mathbf{T}}[\boldsymbol{\omega}_{ie} \times]\right)\end{aligned} \tag{4.55}$$

Expanding brackets and neglecting error products as second order terms allows to reduce "(4.55)", to the form

$$\begin{aligned}\left[\dot{\boldsymbol{\psi}}_{e\tilde{e}} \times\right] = &-[\boldsymbol{\omega}_{ie} \times][\boldsymbol{\psi}_{e\tilde{e}} \times] + [\boldsymbol{\psi}_{e\tilde{e}} \times][\boldsymbol{\omega}_{ie} \times] \\ &+ \mathbf{C}_{\tilde{e}\tilde{s}}[\boldsymbol{\delta\omega}_{is} \times]\mathbf{C}_{\tilde{e}\tilde{s}}^{\mathbf{T}}\end{aligned} \tag{4.56}$$

Finally, using Jokobi identity and properties of $[(\cdot) \times]$ operator, we transform "(4.56)", to the form

$$\left[\dot{\boldsymbol{\psi}}_{e\tilde{e}} \times\right] = -[\boldsymbol{\omega}_{ie} \times][\boldsymbol{\psi}_{e\tilde{e}} \times] + [\mathbf{C}_{\tilde{e}\tilde{s}}\boldsymbol{\delta\omega}_{is} \times] \tag{4.57}$$

This matrix equation is equivalent to

$$\dot{\boldsymbol{\psi}}_{e\tilde{e}} = -[\boldsymbol{\omega}_{ie} \times] \cdot \boldsymbol{\psi}_{e\tilde{e}} + \mathbf{C}_{\tilde{e}s} \cdot \boldsymbol{\delta\omega}_{is} \tag{4.58}$$

The summary of position (section 4.6.3.3.1), velocity (section 4.6.3.3.2) and orientation (Section 4.6.3.3.3) error propagation is given by:

$$\begin{aligned} \boldsymbol{\delta\dot{\mathbf{x}}}_e &= \boldsymbol{\delta\mathbf{v}}_e \\ \boldsymbol{\delta\dot{\mathbf{v}}}_e &= \left.\frac{\partial \mathbf{g}_e(\mathbf{x}_e)}{\partial(\mathbf{x}_e)^{\mathbf{T}}}\right|_{\tilde{\mathbf{x}}_e} \cdot \boldsymbol{\delta\mathbf{x}}_e - \mathbf{2}[\boldsymbol{\omega}_{ie} \times] \cdot \boldsymbol{\delta\mathbf{v}}_e - \left[\left(\mathbf{C}_{\tilde{e}s}\tilde{\mathbf{f}}_s\right) \times\right] \cdot \boldsymbol{\psi}_{e\tilde{e}} + \mathbf{C}_{\tilde{e}s} \cdot \boldsymbol{\delta\mathbf{f}}_s \\ \dot{\boldsymbol{\psi}}_{e\tilde{e}} &= -[\boldsymbol{\omega}_{ie} \times] \cdot \boldsymbol{\psi}_{e\tilde{e}} + \mathbf{C}_{\tilde{e}s} \cdot \boldsymbol{\delta\omega}_{is} \end{aligned} \tag{4.59}$$

4.7.3.3 Error Model for IMU state differential equation

We assume that accelerometer and gyroscope measurements are proportional to the true specific force and angular rate, respectively.

$$\begin{aligned} \mathbf{f}_s &= \mathbf{diag}\left(\mathbf{s}_f\right) \cdot \tilde{\mathbf{f}}_s + \mathbf{b}_f + \boldsymbol{\nu}_f \\ \boldsymbol{\omega}_{is} &= \mathbf{diag}(\mathbf{s}_\omega) \cdot \tilde{\boldsymbol{\omega}}_{is} + \mathbf{b}_\omega + \boldsymbol{\nu}_\omega \end{aligned} \tag{4.60}$$

The equations in "(4.60)" should be interpreted almost as identities: if we know true scale factors, offsets, and noise, we can derive true specific force and angular rate from IMU measurements. True scale factors and biases, however, are not available because sensors are never perfectly calibrated. Instead, there are estimates $\hat{\mathbf{s}}_f$, $\hat{\mathbf{b}}_f$, $\hat{\mathbf{s}}_\omega$, $\hat{\mathbf{b}}_\omega$ that deviate from the truth. Furthermore, because sensor noise is random, it cannot be compensated. As a result, a user can only make the following compensation:

$$\begin{aligned}\hat{\mathbf{f}}_s &= \mathbf{diag}(\hat{\mathbf{s}}_f)\cdot\tilde{\mathbf{f}}_s + \hat{\mathbf{b}}_f \\ \hat{\boldsymbol{\omega}}_{is} &= \mathbf{diag}(\hat{\mathbf{s}}_\omega)\cdot\tilde{\boldsymbol{\omega}}_{is} + \hat{\mathbf{b}}_\omega\end{aligned} \tag{4.61}$$

The residual errors of the compensated specific force and angular rate can be expressed in terms of noise, and the residual errors of the scale factors and offset, as

$$\begin{aligned}\delta\hat{\mathbf{f}}_s &= \mathbf{f}_s - \hat{\mathbf{f}}_s = \mathbf{diag}(\tilde{\mathbf{f}}_s)\cdot\delta\mathbf{s}_f + \delta\mathbf{b}_f + \boldsymbol{\nu}_f \\ \delta\hat{\boldsymbol{\omega}}_{is} &= \boldsymbol{\omega}_{is} - \hat{\boldsymbol{\omega}}_{is} = \mathbf{diag}(\tilde{\boldsymbol{\omega}}_{is})\cdot\delta\mathbf{s}_\omega + \delta\mathbf{b}_\omega + \boldsymbol{\nu}_\omega\end{aligned} \tag{4.62}$$

Once error navigation equation and IMU error modelling is formed, we now estimate state vector, we use IMU measurement compensated according to (4.61). As a result, the estimate of the state vector (4.23) is subject to

$$\begin{aligned}\dot{\tilde{\mathbf{x}}}_e &= \tilde{\mathbf{v}}_e \\ \dot{\tilde{\mathbf{v}}}_e &= -2[\boldsymbol{\omega}_{ie}\times]\tilde{\mathbf{v}}_e + \mathbf{g}_e(\tilde{\mathbf{x}}_e) + \mathbf{C}_{\tilde{e}s}\hat{\mathbf{f}}_s \\ \dot{\breve{\mathbf{q}}}_{\tilde{e}s} &= \frac{1}{2}\left(\breve{\mathbf{q}}_{\tilde{e}s} * \breve{\tilde{\boldsymbol{\omega}}}_{is} - \breve{\boldsymbol{\omega}}_{ie} * \breve{\mathbf{q}}_{\tilde{e}s}\right) \\ \dot{\hat{\mathbf{b}}}_f &= \mathbf{0} \\ \dot{\hat{\mathbf{b}}}_\omega &= \mathbf{0} \\ \dot{\hat{\mathbf{s}}}_f &= \mathbf{0} \\ \dot{\hat{\mathbf{s}}}_\omega &= \mathbf{0} \\ \dot{\breve{\mathbf{q}}}_{\tilde{c}s} &= \mathbf{0}\end{aligned} \tag{4.63}$$

The state vector is propagated by numerically integrating the ODE "(4.63)". Heun's method is used (2nd order Runge-Kutta method with alpha=1) [119-121]. The discrete times that make up the integration grid are derived from gyroscope measurements. Because a smartphone cannot guarantee that accelerometer and gyroscope data are collected at the same time, specific force must be interpolated or approximated at the timestamps of gyroscope measurements.

The state-space model describing the time development of the navigation state, is typically formulated as:

$$\boldsymbol{Z_{K+1}} = \left(\widetilde{\boldsymbol{Z}}_{\boldsymbol{k}}\,,\, \boldsymbol{U_k}\right) \tag{4.64}$$

4.7.3.4 Error State vector

An error state vector that corresponds to the state vector (4.23) is given by

$$\boldsymbol{\delta z} = \left((\boldsymbol{\delta x}_{\boldsymbol{e}})^{\mathbf{T}} \quad (\boldsymbol{\delta v}_{\boldsymbol{e}})^{\mathbf{T}} \quad (\boldsymbol{\psi}_{\boldsymbol{e\tilde{e}}})^{\mathbf{T}} \quad \left(\boldsymbol{\delta b}_{\boldsymbol{f}}\right)^{\mathbf{T}} \quad (\boldsymbol{\delta b}_{\boldsymbol{\omega}})^{\mathbf{T}} \quad \left(\boldsymbol{\delta s}_{\boldsymbol{f}}\right)^{\mathbf{T}} \quad (\boldsymbol{\delta s}_{\boldsymbol{\omega}})^{\mathbf{T}} \quad (\boldsymbol{\psi}_{\boldsymbol{c\tilde{c}}})^{\mathbf{T}}\right)^{\mathbf{T}} \tag{4.65}$$

The error state's dynamics are subject to the linear stochastic differential equation shown below in (4.66). This equation can be obtained from by modelling sensor errors as a random walk and substituting in "(4.59)" where symbols $\delta\mathbf{f}_s$ and $\delta\boldsymbol{\omega}_{is}$ can be obtain with the expression "(4.62)".

$$\boldsymbol{\delta\dot{z}} = \mathbf{A}(\boldsymbol{t}) \cdot \boldsymbol{\delta z} + \mathbf{G}(\boldsymbol{t}) \cdot \boldsymbol{\nu} \tag{4.66}$$

Herein,

$$\mathbf{A}(\boldsymbol{t}) = \begin{pmatrix} \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & -\mathbf{2}[\boldsymbol{\omega}_{ie}\times] & -\left[\left(\mathbf{C}_{\tilde{e}s}\hat{\mathbf{f}}_s\right)\times\right] & \mathbf{C}_{\tilde{e}s} & \mathbf{0}_{3\times3} & \mathbf{C}_{\tilde{e}s}\mathbf{diag}\left(\tilde{\mathbf{f}}_s\right) & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -[\boldsymbol{\omega}_{ie}\times] & \mathbf{0}_{3\times3} & \mathbf{C}_{\tilde{e}s} & \mathbf{0}_{3\times3} & \mathbf{C}_{\tilde{e}s}\mathbf{diag}(\tilde{\boldsymbol{\omega}}_{is}) & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{pmatrix} \tag{4.67}$$

Where **A(t)** is the transition matrix The system transition matrix is approximated using the rule "(4.18)". Also note that $\tilde{\boldsymbol{\omega}}_{is}$ is uncompensated gyroscope measurement and the difference between $\hat{\mathbf{f}}_s$ and $\tilde{\mathbf{f}}_s$ is that they do not have the same meaning.

$$\boldsymbol{\nu} = \left(\left(\boldsymbol{\nu_f}\right)^{\mathbf{T}} \quad (\boldsymbol{\nu_\omega})^{\mathbf{T}} \quad \left(\boldsymbol{\nu_{bf}}\right)^{\mathbf{T}} \quad (\boldsymbol{\nu_{b\omega}})^{\mathbf{T}} \quad \left(\boldsymbol{\nu_{sf}}\right)^{\mathbf{T}} \quad (\boldsymbol{\nu_{s\omega}})^{\mathbf{T}} \quad \left(\boldsymbol{\nu_{\psi_{c\tilde{c}}}}\right)^{\mathbf{T}}\right)^{\mathbf{T}} \tag{4.68}$$

$$\mathbf{G}(\boldsymbol{t}) = \begin{pmatrix} \mathbf{0_{3\times3}} & \mathbf{0_{3\times3}} & \mathbf{0_{3\times3}} & \mathbf{0_{3\times3}} & \mathbf{0_{3\times3}} & \mathbf{0_{3\times3}} & \mathbf{0_{3\times3}} \\ \mathbf{C_{\tilde{e}s}} & \mathbf{0_{3\times3}} & \mathbf{0_{3\times3}} & \mathbf{0_{3\times3}} & \mathbf{0_{3\times3}} & \mathbf{0_{3\times3}} & \mathbf{0_{3\times3}} \\ \mathbf{0_{3\times3}} & \mathbf{C_{\tilde{e}s}} & \mathbf{0_{3\times3}} & \mathbf{0_{3\times3}} & \mathbf{0_{3\times3}} & \mathbf{0_{3\times3}} & \mathbf{0_{3\times3}} \\ \mathbf{0_{3\times3}} & \mathbf{0_{3\times3}} & \mathbf{I_{3\times3}} & \mathbf{0_{3\times3}} & \mathbf{0_{3\times3}} & \mathbf{0_{3\times3}} & \mathbf{0_{3\times3}} \\ \mathbf{0_{3\times3}} & \mathbf{0_{3\times3}} & \mathbf{0_{3\times3}} & \mathbf{I_{3\times3}} & \mathbf{0_{3\times3}} & \mathbf{0_{3\times3}} & \mathbf{0_{3\times3}} \\ \mathbf{0_{3\times3}} & \mathbf{0_{3\times3}} & \mathbf{0_{3\times3}} & \mathbf{0_{3\times3}} & \mathbf{I_{3\times3}} & \mathbf{0_{3\times3}} & \mathbf{0_{3\times3}} \\ \mathbf{0_{3\times3}} & \mathbf{0_{3\times3}} & \mathbf{0_{3\times3}} & \mathbf{0_{3\times3}} & \mathbf{0_{3\times3}} & \mathbf{I_{3\times3}} & \mathbf{0_{3\times3}} \\ \mathbf{0_{3\times3}} & \mathbf{0_{3\times3}} & \mathbf{0_{3\times3}} & \mathbf{0_{3\times3}} & \mathbf{0_{3\times3}} & \mathbf{0_{3\times3}} & \mathbf{I_{3\times3}} \end{pmatrix} \tag{4.69}$$

In "(4.68)", $\boldsymbol{\nu_f}$ and $\boldsymbol{\nu_\omega}$ are noises of the accelerometers and gyroscopes, respectively. $\boldsymbol{\nu_{bf}}$, $\boldsymbol{\nu_{b\omega}}$, $\boldsymbol{\nu_{sf}}$, $\boldsymbol{\nu_{s\omega}}$, and $\boldsymbol{\nu_{\psi_{c\tilde{c}}}}$ are the noises of sensor biases and scale factors, and the noise of misalignment between IMU and a car. By applying triangle rule of integration for "(4.18)" we obtain **G(t)** and the discrete noise covariance matrix shown in "(4.70)"

$$\mathbf{Q}(\boldsymbol{t}) = \mathbf{Cov}[\boldsymbol{\nu}\boldsymbol{\nu}^{\mathbf{T}}] \tag{4.70}$$

It is worth noting that in the implementation, $\mathrm{Cov}[\boldsymbol{\nu}\boldsymbol{\nu}^{\mathrm{T}}]$ is a diagonal matrix. Its first six elements must be set in accordance with the IMU white noise parameters. Elements that correspond to misalignment noises, scale factors, and offsets are set to very small values.

4.7.3.5 Residual Computation and Measurement matrices

The state-space model describing the time development of the GNSS measurements, is given as

$$\boldsymbol{y_k} = \boldsymbol{H^{GNSS}}\left(\widetilde{\boldsymbol{Z}}_{\boldsymbol{k}}\right) + \boldsymbol{r} \tag{4.71}$$

4.7.3.5.1 External Position measurement ($\boldsymbol{H^{GNSS}}$)

The GNSS antenna and IMU in smartphones are mounted within centimetres of each other. In fact, the distance between the antenna and the IMU appears to be much smaller than the typical achievable accuracy of GNSS positioning performed by a smartphone. As a result, the lever arm between the IMU and the GNSS antenna can be ignored. A perfect GNSS receiver would provide a measurement of the position of a smartphone:

$$\mathbf{y}_{pos} = \mathbf{x}_e \tag{4.72}$$

However, because the measurement is not error-free, the measured position is related to the true position as follows:

$$\tilde{\mathbf{y}}_{pos} = \mathbf{x}_e + \boldsymbol{\delta}\mathbf{y}_{pos} \tag{4.73}$$

A residual can be calculated as follows:

$$\boldsymbol{\Delta}\mathbf{y}_{pos} = \tilde{\mathbf{y}}_{pos} - \tilde{\mathbf{x}}_e = \boldsymbol{\delta}\mathbf{x}_e + \boldsymbol{\delta}\mathbf{y}_{pos} \tag{4.74}$$

The equation (4.74) relates an INS error state to a residual (the difference between the measurement and an approximation of the measurement derived from navigation state). These equations are frequently referred to as linearized measurement equations. Measurement matrix for the (4.74), can be written as:

$$\mathbf{H}_{pos} = \frac{\boldsymbol{\partial}\boldsymbol{\Delta}\mathbf{y}_{pos}}{\boldsymbol{\partial}(\boldsymbol{\delta}\mathbf{z})^{\mathrm{T}}} = (\mathbf{I}_{3\times 3} \quad \mathbf{0}_{3\times 21}) \tag{4.75}$$

Where **I** is the identity matrix, a measurement that is linearly related to the position is compared with the same linear combination of the INS computed positions to compute a measurement residual in position aiding methods. The measurement residual is fed into a Kalman filter, which is used to estimate the INS and sensor error states.

One important property of the Kalman filter that is frequently overlooked for instance, assume that the noise covariance matrix R(k) of the vector measurement "y" is a diagonal matrix. In other words, we assume that the noises of the vector "y" components are independent (and thus uncorrelated). Then, instead of one Kalman filter update with a vector measurement "y," m consecutive updates with scalar elements of the vector "y" are possible. The outcome will be the same. This "feature" allows you to avoid numerical matrix inversion while updating the Kalman filter. As a result, computations become far simpler and more robust.

Unfortunately, in real time experimentation, the measurements observed from GPS are corelated to simplify process we can decorrelate measurements by first decorrelate original vector measurement using transformation derived from Cholesky factors of the noise covariance matrix and second do Kalman filter updates sequentially using scalar components of the transformed measurement vector.

Consider a measurement

$$\mathbf{y} = \mathbf{H} \cdot \mathbf{x} + \mathbf{r} \tag{4.76}$$

Assume that **r** is zero-mean Gaussian white noise with covariance matrix **R** that is strictly positive definite. The matrix **R** can be decomposed into Cholesky factors [122, 123] as following:

$$\mathbf{R} = \mathbf{U^T U} \tag{4.77}$$

Consider now a transformed measurement $\mathbf{y_1} = \mathbf{Ty}$,

Where,

$$\mathbf{T} = (\mathbf{U^T})^{-1} \tag{4.78}$$

Then

$$\mathbf{y_1} = \mathbf{H_1} \cdot \mathbf{x} + \mathbf{r_1} \tag{4.79}$$

Where
$$\mathbf{H_1} = \mathbf{TH} \tag{4.80}$$
$$\mathbf{r_1} = \mathbf{Tr}$$

Covariance of the transformed noise $\mathbf{r_1}$ is identity matrix, so noise of the transformed measurement $\mathbf{y_1}$ appears uncorrelated.

4.7.3.5.2 Non holonomic constraints

Under ideal conditions, a vehicle moving in a planar environment experiences no wheel slip and no motion in the direction perpendicular to the road surface. As a result, the vehicle coordinates' downward and sideways velocity components should be close to zero. When nonholonomic constraints are applied to the navigation solution, the results show a significant decrease in position error growth during GNSS outages as well as an increase in attitude accuracy. In the simulation, nonholonomic constraints were imposed by introducing pseudo-observations for the vehicle's sideways and downward velocities. When a car drives straight, the velocity vector's projections onto the second and third axes of the c-frame are zeros. This property is derived from vehicle dynamics and can be used to perform virtual

measurements. Assume that velocity projections onto the car frame can be measured. The ideal measurement will then be related to the INS state vector as follows:

$$\mathbf{y}_{\boldsymbol{vel,c}} = \mathbf{C}_{\boldsymbol{cs}}\mathbf{C}_{\boldsymbol{es}}^{\mathbf{T}}\mathbf{v}_{\boldsymbol{e}} \tag{4.81}$$

Here $\mathbf{C}_{cs}$ is a matrix that defines a rotation from the s-frame to the c-frame. Measured velocity is not error-free, thus

$$\tilde{\mathbf{y}}_{\boldsymbol{vel,c}} = \mathbf{C}_{\boldsymbol{cs}}\mathbf{C}_{\boldsymbol{es}}^{\mathbf{T}}\mathbf{v}_{\boldsymbol{e}} + \boldsymbol{\delta}\mathbf{y}_{\boldsymbol{vel,c}} \tag{4.82}$$

Assume you have an approximation or estimate of $\mathbf{C}_{cs}$. This approximation can be thought of as a transformation $\mathbf{C}_{\tilde{c}s}$ from the true sensor frame to an erroneous car frame $\tilde{c}$. Using the approach described in (4.46), the estimated rotation matrix (Ccs) can be decomposed into a combination of the true value and an error. Thus,

$$\begin{aligned} \mathbf{C}_{\boldsymbol{cs}} &= \mathbf{C}_{\boldsymbol{c\tilde{c}}}\mathbf{C}_{\boldsymbol{\tilde{c}s}} \\ \mathbf{C}_{\boldsymbol{c\tilde{c}}} &\approx \mathbf{I} + [\boldsymbol{\psi}_{\boldsymbol{c\tilde{c}}}\times] \end{aligned} \tag{4.83}$$

A residual can be formed as

$$\boldsymbol{\Delta}\mathbf{y}_{\boldsymbol{vel,c}} = \tilde{\mathbf{y}}_{\boldsymbol{vel,c}} - \mathbf{C}_{\boldsymbol{\tilde{c}s}}\mathbf{C}_{\boldsymbol{\tilde{e}s}}^{\mathbf{T}}\tilde{\mathbf{v}}_{\boldsymbol{e}} = \mathbf{C}_{\boldsymbol{cs}}\mathbf{C}_{\boldsymbol{es}}^{\mathbf{T}}\mathbf{v}_{\boldsymbol{e}} - \mathbf{C}_{\boldsymbol{\tilde{c}s}}\mathbf{C}_{\boldsymbol{\tilde{e}s}}^{\mathbf{T}}\tilde{\mathbf{v}}_{\boldsymbol{e}} + \boldsymbol{\delta}\mathbf{y}_{\boldsymbol{vel,c}} \tag{4.84}$$

Decomposing the true quantities into estimates and errors yields

$$\begin{aligned} \boldsymbol{\Delta}\mathbf{y}_{\boldsymbol{vel,c}} = (\mathbf{I} + [\boldsymbol{\psi}_{\boldsymbol{c\tilde{c}}}\times])\mathbf{C}_{\boldsymbol{\tilde{c}s}}\mathbf{C}_{\boldsymbol{\tilde{e}s}}^{\mathbf{T}}(\mathbf{I} - [\boldsymbol{\psi}_{\boldsymbol{e\tilde{e}}}\times])(\tilde{\mathbf{v}}_{\boldsymbol{e}} + \boldsymbol{\delta}\mathbf{v}_{\boldsymbol{e}}) \\ - \mathbf{C}_{\boldsymbol{\tilde{c}s}}\mathbf{C}_{\boldsymbol{\tilde{e}s}}^{\mathbf{T}}\tilde{\mathbf{v}}_{\boldsymbol{e}} + \boldsymbol{\delta}\mathbf{y}_{\boldsymbol{vel,c}} \end{aligned} \tag{4.85}$$

Finally, after brackets expansion and linearization we have,

$$\mathbf{\Delta y_{vel,c}} = \mathbf{C_{\check{c}s}C_{\check{e}s}^{T}} \cdot \boldsymbol{\delta}\mathbf{v_e} + \mathbf{C_{\check{c}s}C_{\check{e}s}^{T}}[\tilde{\mathbf{v}}_{\boldsymbol{e}} \times] \cdot \boldsymbol{\psi}_{\boldsymbol{e\check{e}}} - \left[\left(\mathbf{C_{\check{c}s}C_{\check{e}s}^{T}}\tilde{\mathbf{v}}_{\boldsymbol{e}}\right) \times\right] \cdot \boldsymbol{\psi}_{\boldsymbol{c\check{c}}} + \boldsymbol{\delta}\mathbf{y_{vel,c}} \tag{4.86}$$

And measurement matrix can be obtained as

$$\mathbf{H_{vel,c}} = \frac{\boldsymbol{\partial}\mathbf{\Delta y_{vel,c}}}{\boldsymbol{\partial}(\boldsymbol{\delta}\mathbf{z})^{\mathbf{T}}} = \left(\mathbf{0_{3\times3}} \quad \mathbf{C_{\check{c}s}C_{\check{e}s}^{T}} \quad \mathbf{C_{\check{c}s}C_{\check{e}s}^{T}}[\tilde{\mathbf{v}}_{\boldsymbol{e}} \times] \quad \mathbf{0_{3\times12}} \quad -\left[\left(\mathbf{C_{\check{c}s}C_{\check{e}s}^{T}}\tilde{\mathbf{v}}_{\boldsymbol{e}}\right) \times\right]\right) \tag{4.87}$$

After we finish building the state space model, we apply the Kalman filter algorithm shown in table 2. Prediction relies on a mathematical or stochastic model "(4.66)", whereas correction of the predicted value relies on observation "(4.71)", "(4.74)", "(4.86)".The transition model, which is represented by a matrix "(4.67)", propagates the previous state to the next, with certain amount of uncertainty. On other hand observation model "(4.71)", is a parametric model in which the addition of measurement and its noise is augmented due to the uncertainty of the observations. However, correction is available in the update process called innovation, which aids in displaying mismatches between observations and previous estimates, and this will correct the prediction, bringing it to a precise estimate. The innovation vector is weighted by gain, which will tell us whether the observations can be trusted or rejected. KF updates are done in a scalar-wise manner enhancing accuracy of the estimates

As explained above KF process, when applied to time update using "(4.64)" and "(4.66)", can be modelled as

$$\hat{\boldsymbol{z}}_{\boldsymbol{k+1/k}} = \boldsymbol{f}_{\boldsymbol{k}}\left(\hat{\boldsymbol{z}}_{\boldsymbol{k|k}}, \overline{\boldsymbol{U}}_{\boldsymbol{k}} - \boldsymbol{\delta}\widehat{\boldsymbol{U}}_{\boldsymbol{k|k}}\right) \tag{4.88}$$

$$\boldsymbol{P}_{\boldsymbol{k+1/k}} = \boldsymbol{\phi}_{\boldsymbol{k}}\boldsymbol{P}_{\boldsymbol{k}}\boldsymbol{\phi}_{\boldsymbol{k}}^{\boldsymbol{T}} + \boldsymbol{G}_{\boldsymbol{k}}\boldsymbol{Q}_{\boldsymbol{k}}\boldsymbol{G}_{\boldsymbol{k}}^{\boldsymbol{T}}$$

$$\widehat{\boldsymbol{x}}_{\boldsymbol{k+1/k}} = \left[\boldsymbol{O}_{\boldsymbol{1,12}}\left(\boldsymbol{\delta}\widehat{\boldsymbol{U}}_{\boldsymbol{k/k}}\right)^{\boldsymbol{T}}\right]^{\boldsymbol{T}}$$

The navigation state errors "(4.63)", are set to zero as the navigation is updated in the last step of the algorithm and to note: t = k. With GNSS measurements available at T(k+1), "(4.71)", is used to perform a Kalman filter measurement update utilizing GNSS measurements, and pseudo-observations of velocity and roll

$$\boldsymbol{s}_{\boldsymbol{k+1}} = \boldsymbol{H}_{\boldsymbol{k+1}}\boldsymbol{P}_{\boldsymbol{k+1/k}}\boldsymbol{H}_{\boldsymbol{k+1}}^{\boldsymbol{T}} + \boldsymbol{R}_{\boldsymbol{k+1}}$$

$$\boldsymbol{k}_{\boldsymbol{k+1}} = \boldsymbol{P}_{\boldsymbol{k+1/k}}\boldsymbol{H}_{\boldsymbol{k+1}}^{\boldsymbol{T}}\boldsymbol{s}_{\boldsymbol{k+1}}^{-\boldsymbol{1}} \tag{4.89}$$

$$\widehat{\boldsymbol{x}}_{\boldsymbol{k+1/k+1}} = \widehat{\boldsymbol{x}}_{\boldsymbol{k+1/k-k_{k+1}hp}}\boldsymbol{s}$$

$$\boldsymbol{P}_{\boldsymbol{k+1/k+1}} = \left(\boldsymbol{I}_{\boldsymbol{18}} - \boldsymbol{k}_{\boldsymbol{k+1}}\boldsymbol{H}_{\boldsymbol{k+1}}^{\boldsymbol{P_S}}\right)\boldsymbol{P}_{\boldsymbol{k+1/k}}$$

However, when there are no GNSS measurements available at T(k+1), "(4.86)", is used to perform a Kalman filter measurement update utilizing pseudo-observations of NHC

$$\boldsymbol{S}_{\boldsymbol{k+1}} = \boldsymbol{H}_{\boldsymbol{k+1}}^{\boldsymbol{ps}}\boldsymbol{P}_{\boldsymbol{k+1/k}}\left(\boldsymbol{H}_{\boldsymbol{k+1}}^{\boldsymbol{\rho_s}}\right)^{\boldsymbol{T}} + \boldsymbol{R}_{\boldsymbol{k+1}}^{\boldsymbol{ps}}$$

$$\boldsymbol{K}_{\boldsymbol{k+1}} = \boldsymbol{P}_{\boldsymbol{k+1/k}}\left(\boldsymbol{H}_{\boldsymbol{k+1}}^{\boldsymbol{\rho s}}\right)\boldsymbol{S}_{\boldsymbol{k+1}}^{-\boldsymbol{1}} \tag{4.90}$$

$$\widehat{\boldsymbol{x}}_{\boldsymbol{k+1/k+1}} = \widehat{\boldsymbol{x}}_{\boldsymbol{k+1/k}} - \boldsymbol{k}_{\boldsymbol{k+1}}\boldsymbol{h}^{\boldsymbol{ps}}\left(\hat{\boldsymbol{z}}_{\boldsymbol{k+1/k}}\right)$$

$$\boldsymbol{P}_{\boldsymbol{k+1/k+1}} = \left(\boldsymbol{I}_{\boldsymbol{18}} - \boldsymbol{k}_{\boldsymbol{k+1}}\boldsymbol{H}_{\boldsymbol{k+1}}^{\boldsymbol{ps}}\right)\boldsymbol{P}_{\boldsymbol{k+1/k}}$$

The KF algorithm recursively minimises estimation errors, resulting in accurate measurements and true estimates of vehicle pose on the road. The results of the real-time experiment are presented in section 6, which discusses error minimization in more detail.

Chapitre 5 - **Maps**

5.1 Geospatial data modelling (GIS maps)

In-car navigation systems display digital maps differently than "traditional" printed or digitalized maps. Traditional maps have images made up of pixels that humans can easily interpret. The image in traditional maps does not contain enough information for a computer to match the vehicle's location on the street network and recommend a path. Thus, digital maps used in automobile navigation systems and other intelligent transportation systems (ITS) applications are built as databases of topological and metrical (coordinates) data, as well as road class, street name, expected driving speed, and turn restrictions. This allows a computer to more easily and quickly interpret digital maps. See [3, 124] for a more detailed description of digital maps in in-car navigation systems and other ITS applications. In [125], an outline is presented that summarises the map attributes required for the development of new ADAS applications. While maps are useful for directing vehicles to their destinations, they fall short of providing accurate lane information context and lane and segment geometry. As a result, numerous self-driving vehicle maps have been created [126-129]. The authors of [17], defined two types of maps: planar maps and point-cloud maps. according to [23], Planar maps, such as high-definition (HD) maps or lane-level maps, describe the geographic world by utilising layers or planes in GIS software. Point cloud maps generate cloud maps by collecting and utilising point-cloud data collected from on-board/external sensors. TomTom,

Google, and Here, for example, have developed this type of map to function in real-world road and road surface conditions.

We place more emphasis on navigational geospatial data models because our research requires maps with road level information. Geospatial data models are nothing more than GIS maps containing the attributes required for a particular application. At various stages of the navigation process, maps and their attributes play a variety of roles in navigation applications. The most critical and widely used role is visualisation, GIS provides the capability to relate previously unrelated information, using location as the "key index variable". Locations and extents that are found in the Earth's spacetime, can be recorded through the date and time of occurrence, along with x, y, and z coordinates, representing, longitude (x), latitude (y), and elevation (z).

All Earth-based, spatial–temporal, location, and extent references, should be relatable to one another, and ultimately, to a "real" physical location or extent. The GIS map contains a wealth of useful information about roads and passageways; it can be used to assist the navigation solution by implementing a logical threshold based on the geospatial model. On digital maps, the road network is typically depicted as a planar model, whereas the street network is typically depicted as a collection of arcs (i.e., curves in R2) [130]. Each arc represents the centerline of a network road segment and is typically assumed to be piecewise linear (though splines may also be used), allowing it to be described by a finite set of points (see figure 18). The starting and ending points of the set are referred to as nodes, while the remaining points are referred to as shape points. The nodes indicate the beginning and end of the arc, indicating a beginning, a dead end, or an intersection (i.e., a point at which two arcs can connect) in the street network

The navigation map is composed of multiple layers, depending on the region and application. The choice of the required layer and its associated attributes is critical in the design of geospatial models. The geospatial model for vehicle navigation represents the geometry and attributes of street networks.

5.2 Requirement for maps in navigational applications

When using a map for navigational purposes, numerous specifications and requirements must be met. Numerous factors will influence the selection of the conceptual or geometric model and database content. Additionally, when it comes to navigation applications, the digital map's accuracy is critical [131]. If the spatial data has a higher positional accuracy than the navigation solution, particularly in GPS-denied environments, the spatial data can assist the navigation system in improving its accuracy. Numerous factors are considered when expressing map accuracy. Comparative studies on the accuracy of the primary map provider for location-based services have been conducted in [131]. However, it was determined that a provider's coverage area has a significant impact on the provider's accuracy and quality. For example, major city centres, which cover a large area, are more precise than smaller, country-side cities. Table 3 illustrates some of the key requirements for map accuracy in a navigation application.

Table 3 Map factors that influence navigation applications

Factors	**Navigation Application requirements**
Spatial Coverage	Major and minor road networks
Detail coverage	Roads, addresses, points of interest

Positional Accuracy	Ranges from (0.5m – 15m)
Currency of Map	Up-to-date maps are critical in navigation applications

Typically, spatial databases contain information about an object's spatial characteristics. Along with the various structural forms required to construct a particular feature, there may be a massive amount of data. When compiling these databases, the primary objective is to extract only the necessary information and structure it appropriately for the application at hand. For example, there are five primary functions required for navigation applications in digital road databases: map matching, address matching, path finding, and route guidance.

5.2.1 Map matching

Matching the navigation system output to the digital map's road network typically involves three steps. To begin with, a selection of potential arcs or segments is made. Then, the likelihood of the candidate arcs/segments is determined using geometric and topological information, as well as the correlation between the vehicle's trajectory history and the map's candidate paths. Finally, the vehicle's location on the most probable road segment is determined. Geometrical information includes measures such as the estimated position's proximity to the nearest road on the map, the heading difference between the navigation system's heading and the road segments in issue, and the shape of the road segments in relation to the estimated trajectory. Authors in [132], discusses and describes several commonly used geometric information extraction techniques, including point-to-point, point-to-curve, and curve-to-curve matching. The connectivity of candidate roads (arcs) is determined by the topological information criterion; for example, the vehicle cannot abruptly

change directions between road segments if there is no intersection point between them. The likelihood of road segment candidates is determined by assigning different weights to geometrical and topological information measures and combining them. Numerous approaches to weighing and combining are discussed in [133, 134], including belief theory, fuzzy network, and state machines. [125] contains a survey of state-of-the-art map matching algorithms, as well as suggestions for future research directions.

5.2.2 Address Matching

Address matching is the process of associating a user-supplied address with its map location. To perform this function, addresses must be stored in a map database. Historically, the primary concern with address storage was the enormous amount of space required to run a database properly. As a result, early car navigation devices typically required the addition of addresses between nodes by range, ensuring that the link included a range of addresses. However, with the advancement of storage devices, it is no longer necessary to add each address to the precise location on the map where it is located, as the link will include a number of known addresses. Not only residential addresses are added to the databases, but also major landmarks such as schools, hospitals, road names, and shopping malls. These are included to assist the user in navigating to their destination.

5.2.3 Path Finding

Path finding, a feature found in most navigation systems, generates one or more complete paths that connect the user's starting point to the desired destination. Depending on the specific path required, this function may require multiple sets of databases to complete. For instance, determining the shortest path in the least amount of time requires traffic data and

link speed limits. A properly scaled map is required for shortest-distance path finding. Optimal path finding for increased fuel efficiency requires knowledge of both the road grade and the fastest times versus the shortest distance. Finally, finding alternate routes to avoid certain freeways necessitates road classification. In general, most of these functions will require highly accurate topology data, such as turn restrictions, direction or turn changes during rush hour, and so on.

5.2.4 Route Guidance

Other than address matching and path finding, road guidance requires no additional information. In other words, this function is very common when both functions are present. The goal of road guidance is to provide the user with turn-by-turn directions to get from one location to another.

Table 4 Each process necessitates the use of specific spatial database features.

Process	**Database Features**
Map matching	Links, nodes, coordinates of nodes, complete topology
Address matching	Links, names of links, nodes, coordinates of nodes, addresses along the links
Path finding	Link classification, connectivity between nodes, driving and turn restrictions, auxiliary attributes
Route guidance	All address matching and path-finding features

5.3 Map sources for navigation applications

Navigational maps are available in a variety of formats. These sources can be of three kinds: major map providers like Google [135] or NAVTEQ [136], community-contributed mapping solutions like OpenStreetMap [32], or maps designed and developed specifically for a specific application by mapping and surveying companies like Tele Atlas. Tele Atlas is a Dutch company that develops digital maps for use in navigation, location-based services, mobile, and web mapping applications. NAVTEQ is a US-based company that specialises in GIS data and electronic navigable base maps. Nokia Ovi map applications make use of NAVTEQ data. Additionally, NAVTEQ provides map data for navigation applications from Chrysler, Mercedes-Benz, Garmin, Magellan, Yahoo Maps, Bing Maps, and MapQuest. Bing Maps is a web mapping service provided by Microsoft that covers the entire globe. It provides road views, aerial views, street side imagery, bird's eye views (aerial images acquired from low-flying aircraft), venue maps (Bing Maps has mapped several venues throughout North America, Europe, and Asia), and 3-D maps. Google Maps is a Google-owned web mapping service. It supports a number of map-based services, including the Google Maps website, Google Transit, and the Google Maps API, which enables developers to integrate Google Maps into their websites or mobile applications. It has been used in a variety of navigation applications, including smart phones and navigators. The strengths of Google's free maps are their global coverage and dependability for navigation applications. OpenStreetMap is a free online mapping service based on the concepts of crowdsourcing and volunteer-contributed geographic data (VGI).

5.4 Designing of the Geospatial Data models

The developed geospatial models for outdoor navigation applications are presented in this section.

5.4.1 Street network for vehicle navigation application

The geometry and attributes of the street networks will be represented by a geospatial model. Polylines, curves, and traverse are the three geometrical models for street networks figure 17 provides a visualization. The geometrical model to be used will be determined by the density of the road network and the horizontal alignments of the roads.

The developed geospatial model in this research is designed for Shawinigan city, Quebec, Canada, where the model will represent the streets as polylines. Every street link will be represented by a polyline (arc) with a beginning and ending node. The intersections between streets are represented by the nodes. The attached attributes will provide primary information about navigation purposes, such as node coordinates, link start and end sites, and all possible diverged links from each link and traffic direction. The geospatial model is designed to accommodate any additional information, such as node elevation (height). To build the geospatial model, the system will use Open Street Maps to extract the road network.

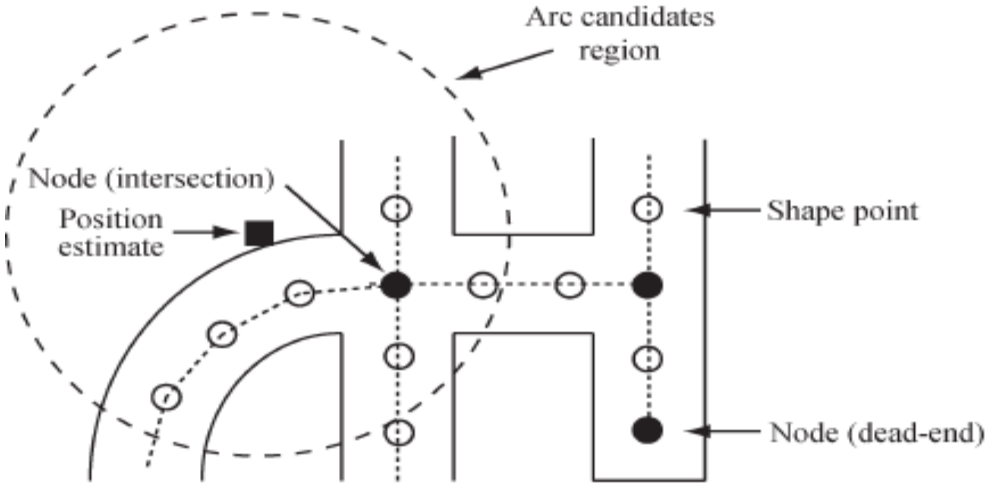


Figure 17 Rough visualization of Road network described by a planar model.

5.4.2 Geospatial model for the experiment route

A geospatial database was created and used to select the appropriate road link based on the user's destination. The geospatial database covers the study areas where the field tests will take place (refer to Chapter 6)—Shawinigan city, Quebec, Canada is an ideal location to test the developed algorithms in a GPS-denied environment due to its urban environment consisting of high buildings, trees etc. which cause GPS signal blockage. The geospatial model constructs the geometry of the road networks using Open Street Maps, with an accuracy range of 0.5m to 1m. Google Maps was used as an additional source of geometry. Figures 18 depicts the geometry of the study area's road network. Shawinigan city and region are connected by a network of roads that are mostly perpendicular to one another. Avenues are roads that run east-west, and Streets are roads that run north-south.

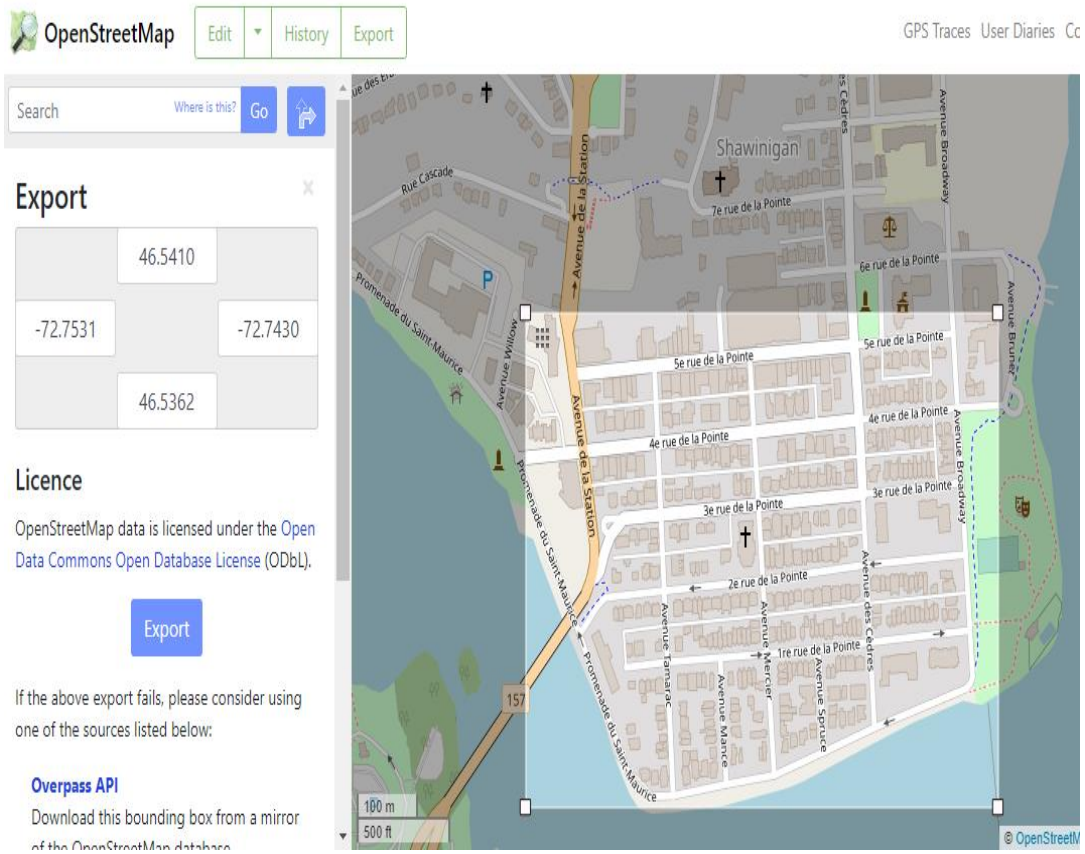


Figure 18 Open Street map displaying the road network of an experimental route that must be exported in order for QGIS software to function.

A road reference is obtained from a mapping database such as OpenStreetMap (OSM), for a selected experiment region. OSM include a street network containing geospatial data about the attributes and geometry of streets. The experiment region was downloaded from the OSM database as a text file, which contains a wealth of information about road networks such as streets, motorways, highways, bike lanes, structures, and bus stops. The saved OSM file was then exported to "QGIS software" for extraction of reference data (also known as ground truth). QGIS [137], is a free GIS application that allows users to create and export graphical maps as well as analyze and edit spatial data. Vector data is stored as point, line, or polygon features in QGIS, which supports raster and vector layers. Using the appropriate filters, the road network from OSM database was explored, and road segments with

geographic latitude-longitude points placed in the middle of the road were collected throughout the trajectory. Figure 19 illustrates QGIS software processing with multilayer filtering of street network.

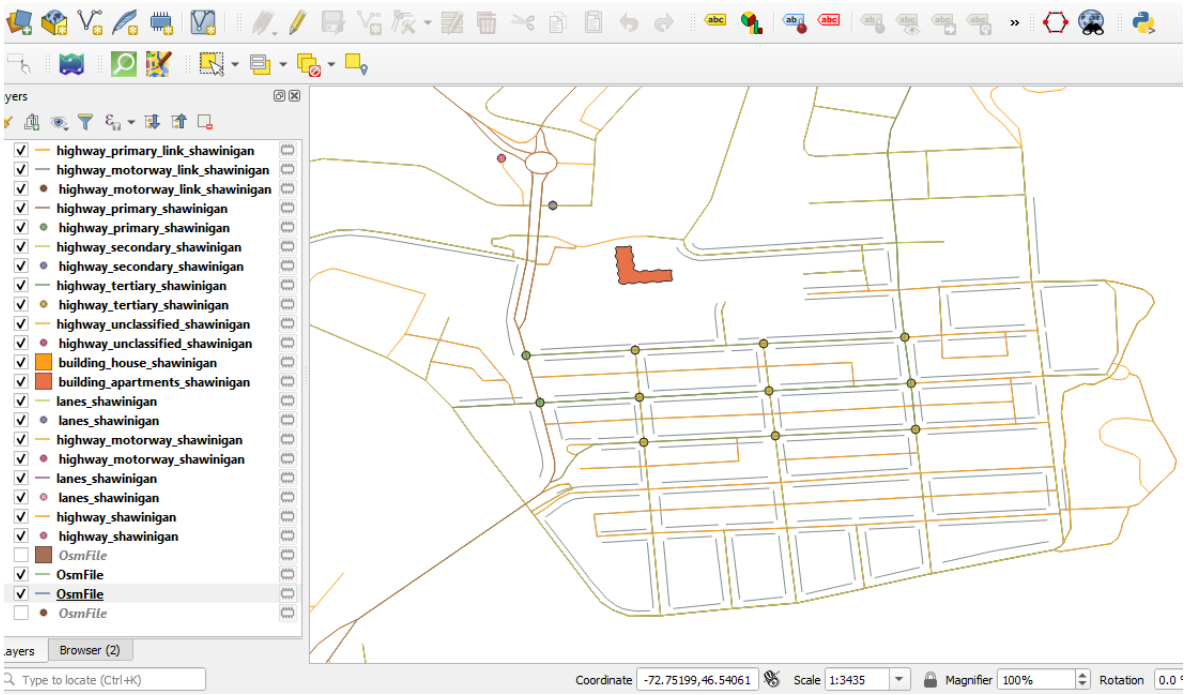
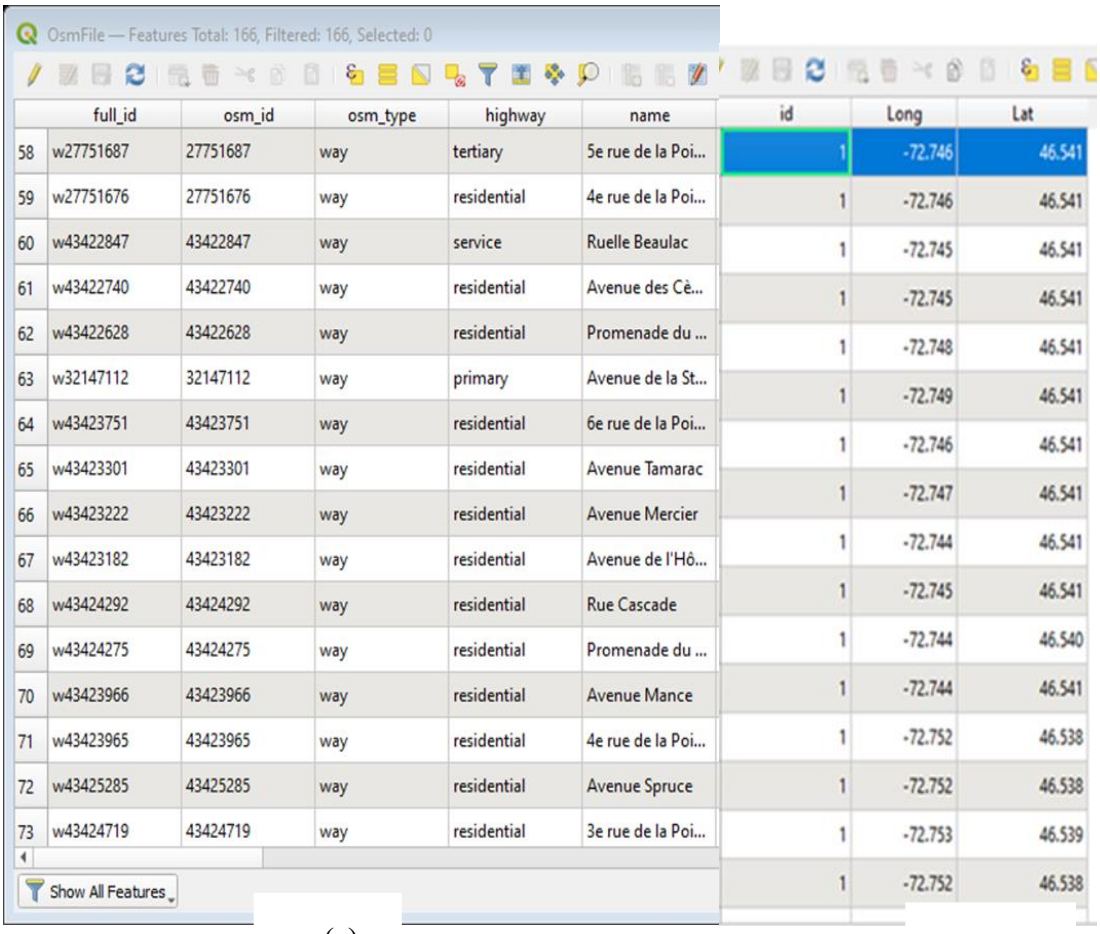


Figure 19 Image of a street network processed in QGIS software, which includes layers of step

The experimental route street network attributes have been explored using QGIS software, based on a specific selection of spatial points. An attribute table is created using a wealth of information such as street names, bus stops, highways, motorways, lanes, and buildings to outline all specific location points along the trajectory of the vehicle with latitude and longitude specifications. Figure 20 illustrates attributes table processed during QGIS processing. Based on the attribute's latitudes and longitudes of road along trajectory are collected.



	full_id	osm_id	osm_type	highway	name
58	w27751687	27751687	way	tertiary	5e rue de la Poi...
59	w27751676	27751676	way	residential	4e rue de la Poi...
60	w43422847	43422847	way	service	Ruelle Beaulac
61	w43422740	43422740	way	residential	Avenue des Cè...
62	w43422628	43422628	way	residential	Promenade du ...
63	w32147112	32147112	way	primary	Avenue de la St...
64	w43423751	43423751	way	residential	6e rue de la Poi...
65	w43423301	43423301	way	residential	Avenue Tamarac
66	w43423222	43423222	way	residential	Avenue Mercier
67	w43423182	43423182	way	residential	Avenue de l'Hô...
68	w43424292	43424292	way	residential	Rue Cascade
69	w43424275	43424275	way	residential	Promenade du ...
70	w43423966	43423966	way	residential	Avenue Mance
71	w43423965	43423965	way	residential	4e rue de la Poi...
72	w43425285	43425285	way	residential	Avenue Spruce
73	w43424719	43424719	way	residential	3e rue de la Poi...

id	Long	Lat
1	-72.746	46.541
1	-72.746	46.541
1	-72.745	46.541
1	-72.745	46.541
1	-72.748	46.541
1	-72.749	46.541
1	-72.746	46.541
1	-72.747	46.541
1	-72.744	46.541
1	-72.745	46.541
1	-72.744	46.540
1	-72.744	46.541
1	-72.752	46.538
1	-72.752	46.538
1	-72.753	46.539
1	-72.752	46.538

(a) (b)

Figure 20 (a) The image depicts an attributes table that contains all geospatial information about the road street network and (b) latitudes and longitudes of the individual places shown

Roads are divided into individual links in a street network, each with its own distinct features such as street names, lane count, bus stops, and highways with a globally unique link identification (ID) in the mapping database as can be seen figure 20. With some lateral error, a road-level map of any given road segment is linked to a set of geographic latitude-longitude points in the middle of the road. The distribution of these shape points along a specific road segment is proportional to the acuity of the road geometry; we use these points as a road reference for vehicle to guide along trajectory. Figure 21 shows a road shape based on

geographic latitude-longitude points at the centre of a road, which are represented with a round shape and red colour.

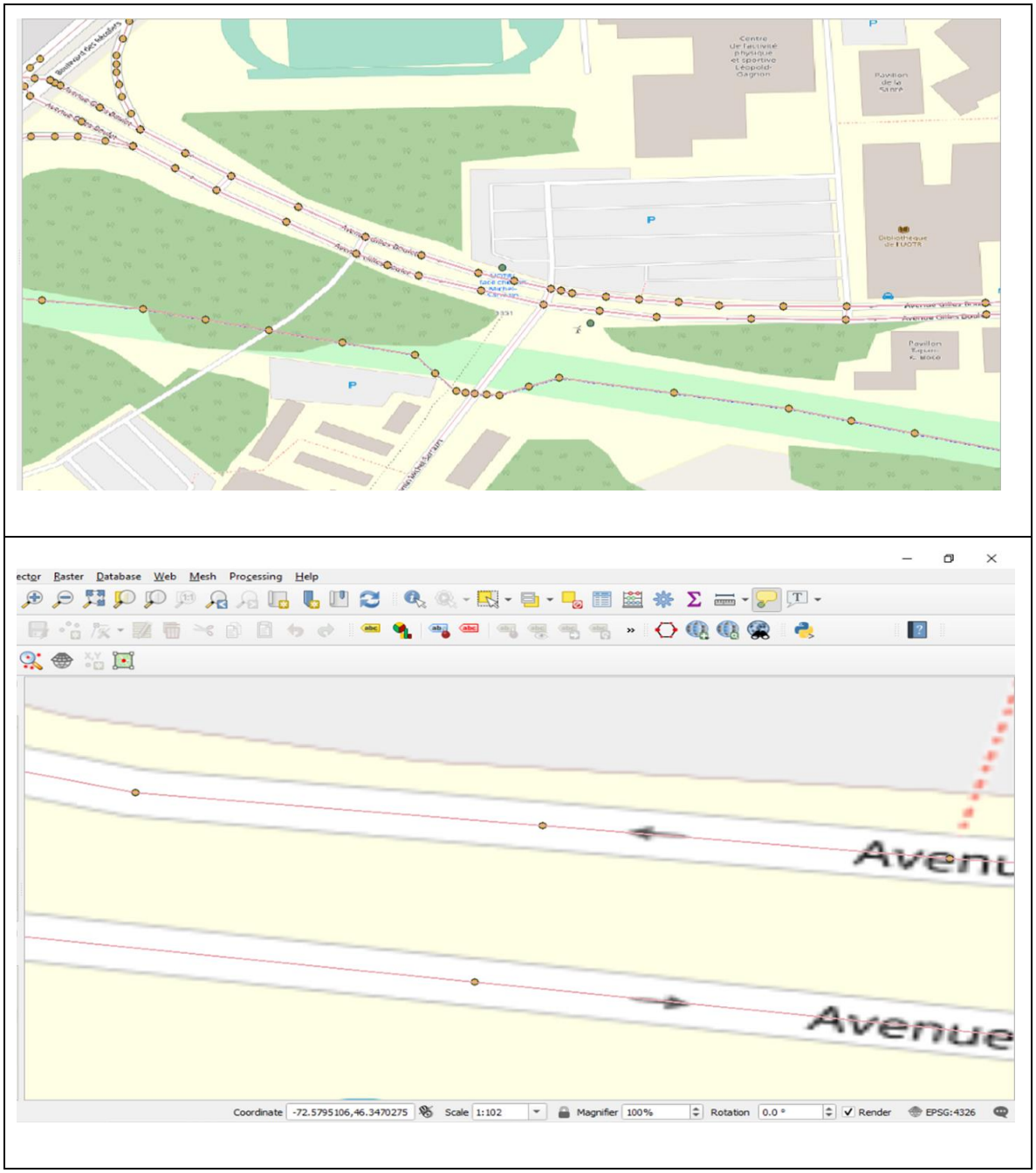


Figure 21 (a) illustrates Geographic latitude and longitude points at center of road in red color and in (b) Zoom in on node points that have unique latitude and longitude information for constructing a reference lane

Chapitre 6 - **Experiments and simulations**

In this chapter, we devised the following experimental setup to evaluate the proposed method's performance.

The experiment was carried out during the winter season, and the route was designed to simulate a variety of driving scenarios with varying climatic conditions (snowy and wet pavement roads were encountered throughout the experiment), resulting in reduced lane visibility. The experiment was carried out on Shawinigan Street in Quebec City, Canada, to cover the entire urban environment for navigation. Figure 22 depicts the experiment route, which spanned 2.1km and included straight roads, six turns, and streets with a 3m wide lane surrounded by buildings, trees, and traffic signs.

Figure 22 The region chosen for the real-time experiment is depicted using Google Maps

Figure 23 depicts real-time experiment images taken during the drive. Whereas Figure 23(a) depicts the beginning of the experiment, Figure 23(b) depicts the roads covered in snow, concealing the pavement curbs and faded lane markings that were also observed during the driving test. In addition, the experiment route included curvy roads to assess the reliability of the proposed system. Figures 23(c), 23(d), 23(e), and 23(f) depict the car on curved roads where lane markings are rarely visible to the naked eye. Furthermore, the experiment is carried out in a densely populated urban environment with numerous GPS signal-blocking elements such as tall buildings, trees, and a distributed network area with bridges. Figures 23(g) and 23(h) depict a vehicle passing through a signal-deficient area as well as beneath a bridge with varying levels of illumination. Figures 23(i) and 23(j) show a populated area with traffic signs and passing vehicles, as well as shading of lane markings caused by passing vehicles.

The experiment route was designed to cover all geographic directions, including north-south and east-west, and data was collected during a 5-minute drive using a Samsung S8+ smartphone mounted on the dashboard but in an unknown orientation. On the Android market "Play store," there are numerous applications for recording accelerometer, gyroscope, and GPS data (see Appendix A). The IMU update rate was set to 52 Hz in the application, while the GPS update rate was set to 1 Hz. The information recorded was later saved in the "CSV file" format for further editing and modification. The Samsung S8+ smartphone includes a GNSS receiver, and with aid of advanced internal settings in phone, the smartphone's ego location accuracy enhanced.

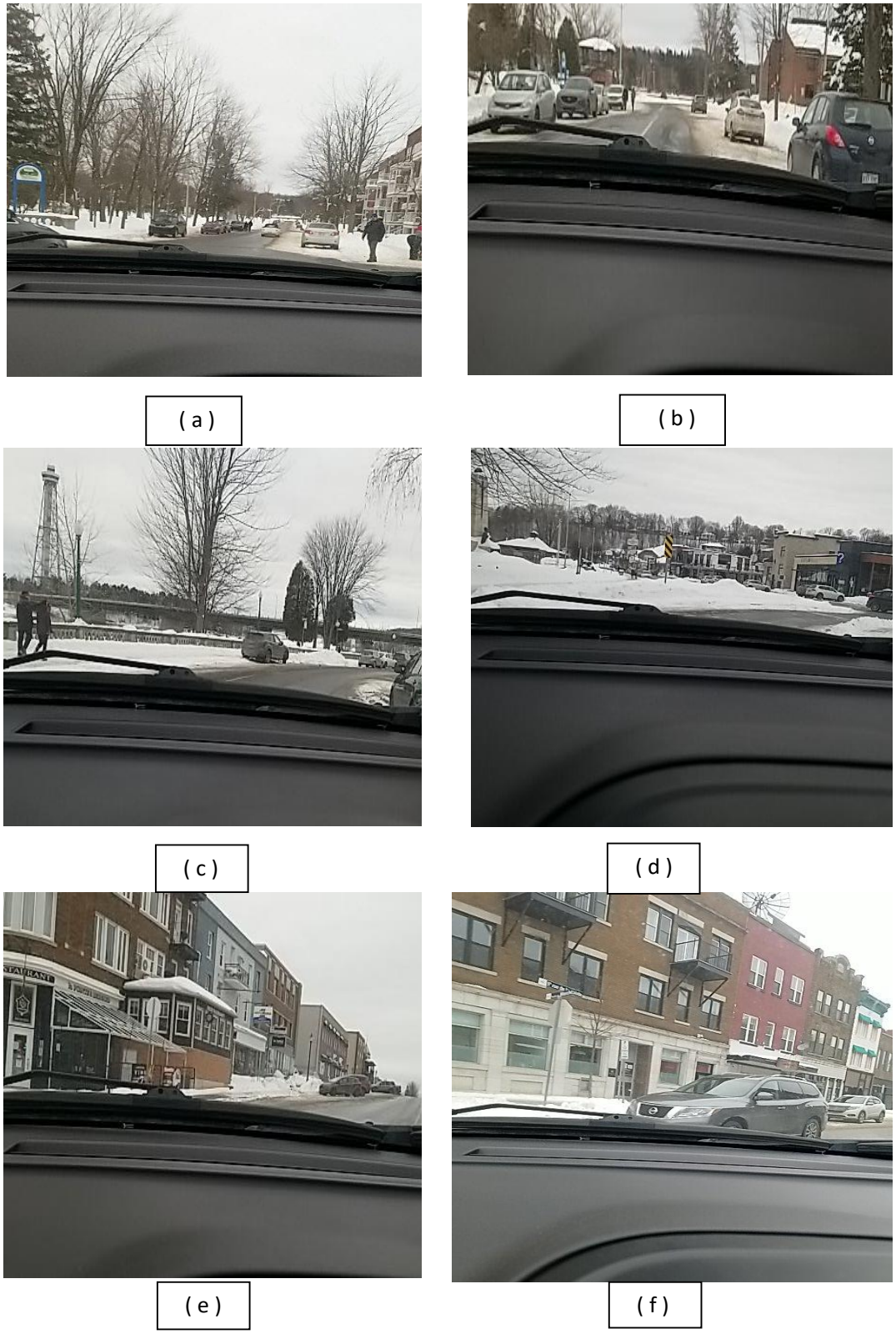


Figure 23 Several real-time driving images collected prior to the experiment's start and at random times throughout the experiment.

A road reference, on the other hand, is obtained from a mapping database such as OpenStreetMap (OSM) for a selected experiment region using the methodology described in section (5.4.2). The latitude and longitude reference road points collected from QGIS software processing were saved in Excel software for interpolation. Data from the IMU and GPS and the OSM mapping database were imported into MATLAB for simulation, and the navigation filter described in chapter 4 was applied.

6.1 Results of experiments and simulations

The results of post-processing in MATLAB simulation, depicted in Figure 24, represent the vehicle's position and orientation during the experiment. Whereas the redline and red circle points represent lane center information, the blue line represents dead reckoning using accelerometer, gyroscope, and GPS data. GPS data were not always required in our experiment, but when GPS data were available, they were used to correct IMU drift.

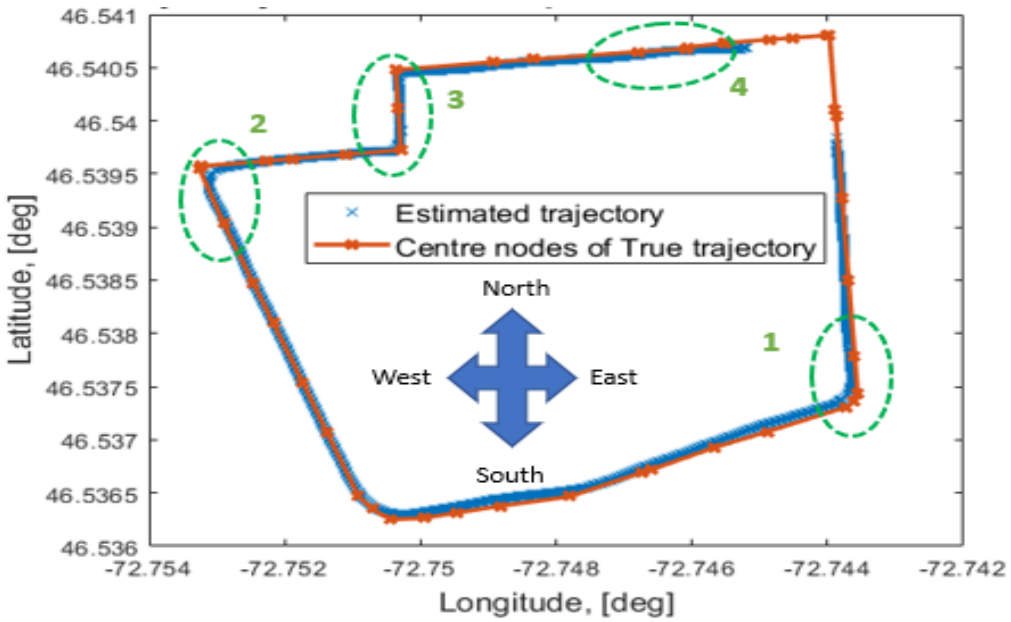


Figure 24 Vehicle trajectory during the experiment, with lane center nodes extracted from a GIS in red circles and inertial navigation dead reckoning represented in blue

The green dotted circles with numbers “1, 2, 3 and 4” in Figure 24, are magnified in Figure 25, 26, 27 and 28, focuses on trajectory behavior at turns and on straight road.

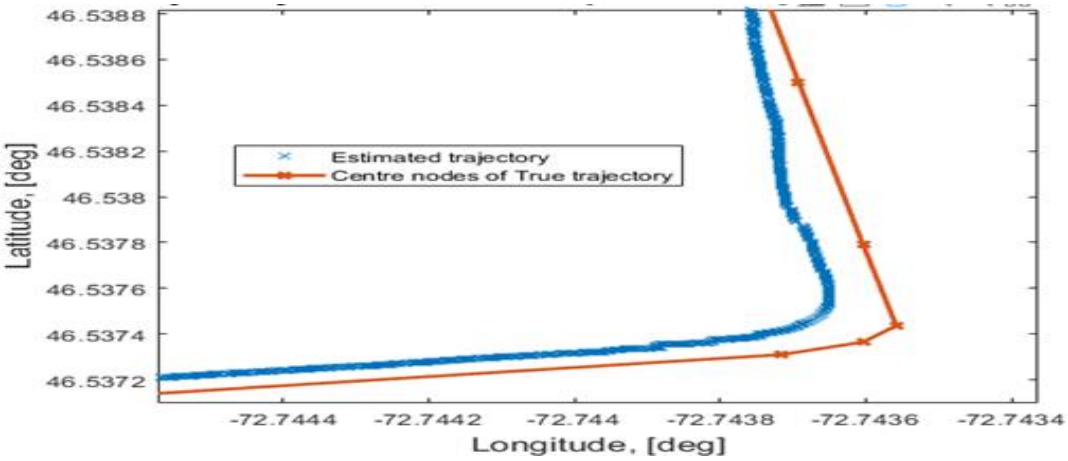


Figure 25 Zoom in of vehicle position at “1” in “Fig 24” illustrates the vehicle's position and orientation during turn. Blue color line represents ego vehicle and red color line represents the lane center markings

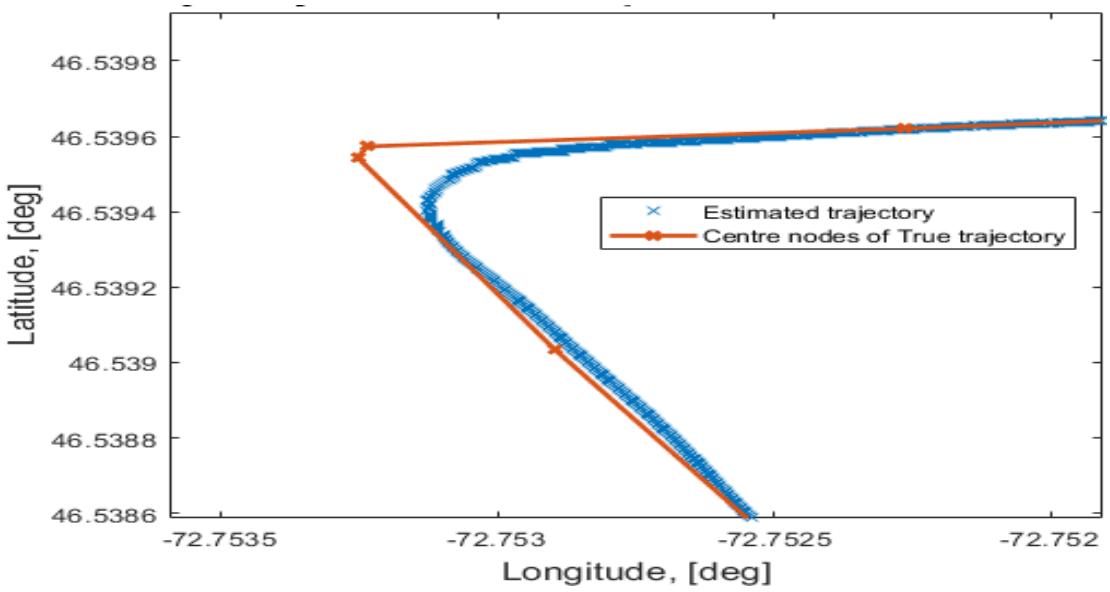


Figure 26 Zoom in position of vehicle at “2” in Figure 24, illustrate the vehicle's position and orientation during turns.

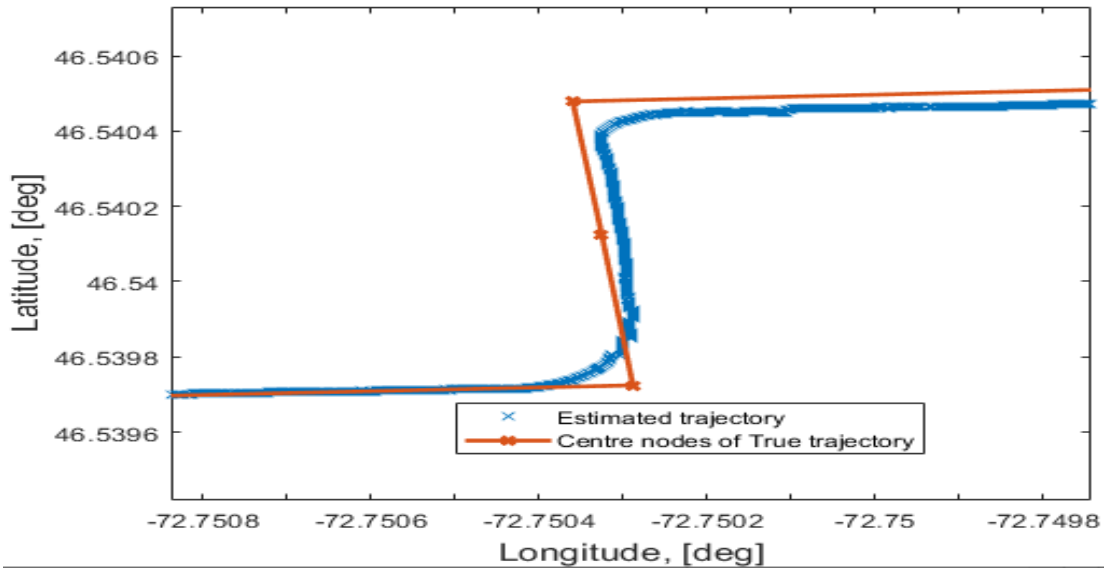


Figure 27 Zoom in position of vehicle at "3" in Figure 24, illustrate the vehicle's position and orientation during turns.

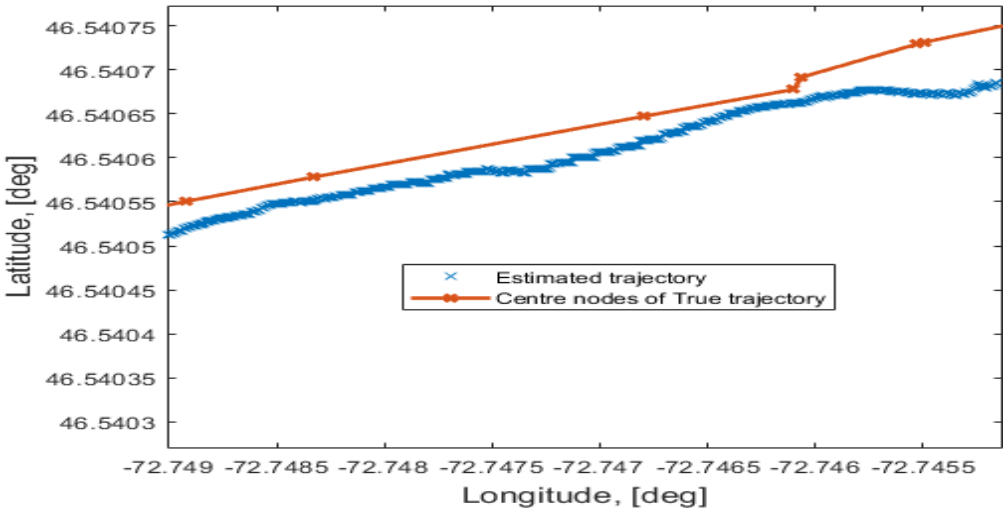


Figure 28 Zoom in position of vehicle at "4" in Figure 24, denotes the location of the vehicle on straight roads

Figure 29 illustrates the absolute error between the estimated and ground truth trajectory.

To illustrate the difference between the estimated and true trajectory, colour bars were used.

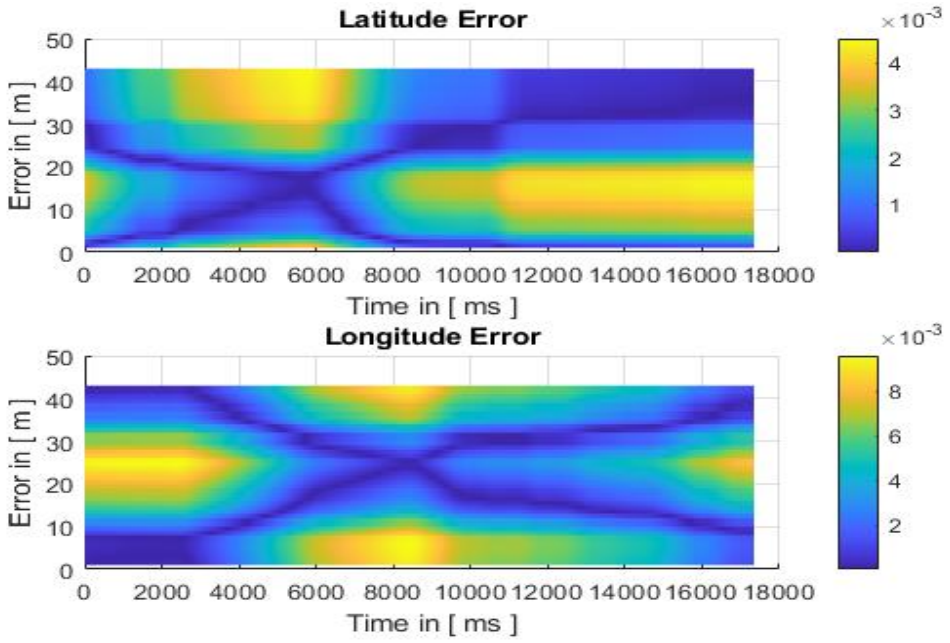


Figure 29 Error in latitude and longitude between estimated and ground truth trajectory

The term "error" refers to the absolute value of the difference between the true and estimated trajectories. The figure 29 depicts the error on the Y axis and the time at which the error occurred on the X axis. Two colours “Yellow” and “Blue” are used to make the error more visible by highlighting their intensities. In the figure 29, yellow colour in the surfaces indicates relatively high positive error (meaning Ground truth Latitude > Estimated Latitude or Ground truth Longitude > Estimated Longitude respectively) and blue colour indicates high negative error or sometimes the value is closer to 0, indicating a low error or a better match. (Blue colour difference represents Ground truth Latitude < Estimated Latitude). When comparing the figure 29 to the figure 24 "Trajectories close to lane level," we can see that wherever a curved road is encountered, as well as at the beginning and end of the trajectory, a high level of yellow colour is observed, whereas wherever straight roads are encountered, the colour blue intensities change darker.

Velocity plot of vehicle in the NED frame (North, East and Down) is depicted in figure 30, when the trajectory is north, the first plot in Figure 30, shows a spike in the graph; when the trajectory is west, the velocity in the east is seen to decrease in the second plot. The road's elevation is resembled in the third plot of Figure 30, when the road is elevated, a spike in the graph is observed; when the road is flat, the trend appears to be close to zero.

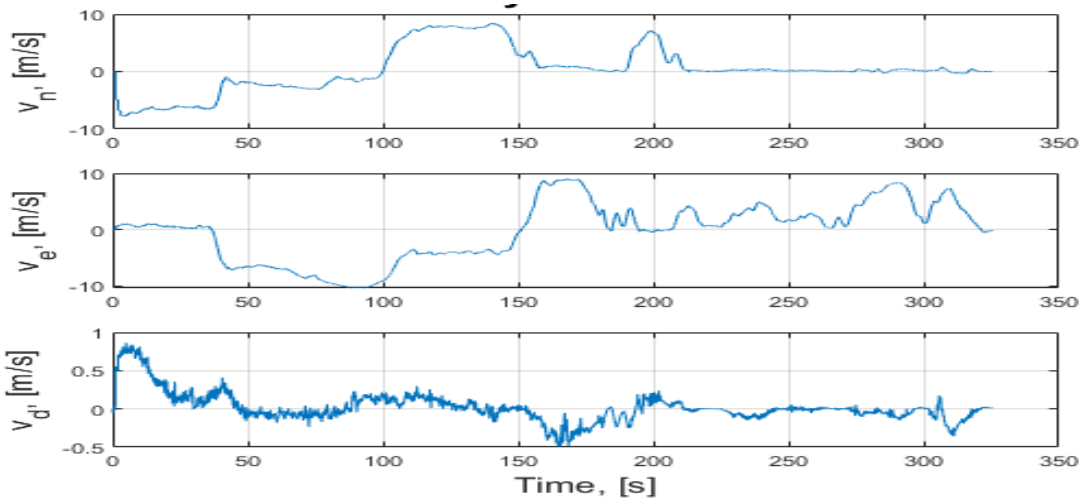


Figure 30 Velocity of the vehicle, during trajectory in North, East and Down reference frame.

Figure 31 depicts the velocity in the car frame, which represents vehicle acceleration in the "x" direction in the first plot and braking information in the "y" direction in the second plot.

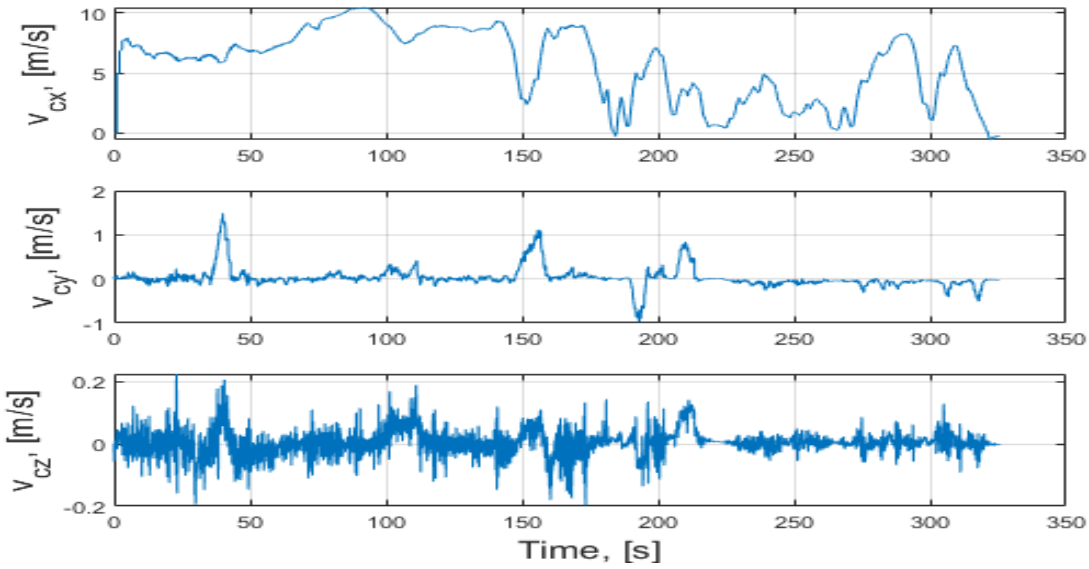


Figure 31 Velocity of the vehicle in body frame.

In Figure 32, roll, pitch, and yaw values (first, second, and third plots) recorded in the sensor frame can be seen, with roll representing the vehicle's left and right tilting movements. Pitch represents the forces acting on the vehicle as it accelerates and decelerates, whereas Yaw represents the number of turns made along the trajectory, with each right turn elevating the curve and each left turn lowering it.

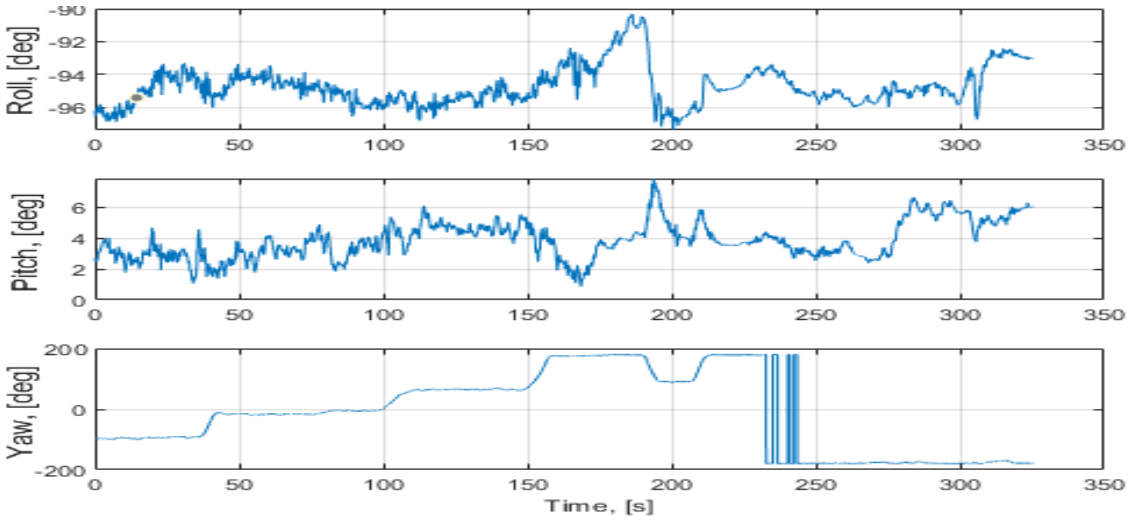


Figure 32 Euler angles in sensor frame.

Intrinsic sensor errors, such as biases and scaling factors, degrade the quality of motion estimation and jeopardize the system's integrity, as discussed in Section II. The implementation of the Kalman recursive algorithm reduces errors in 6 DOF to a minimum during trajectory, as shown in figure 33

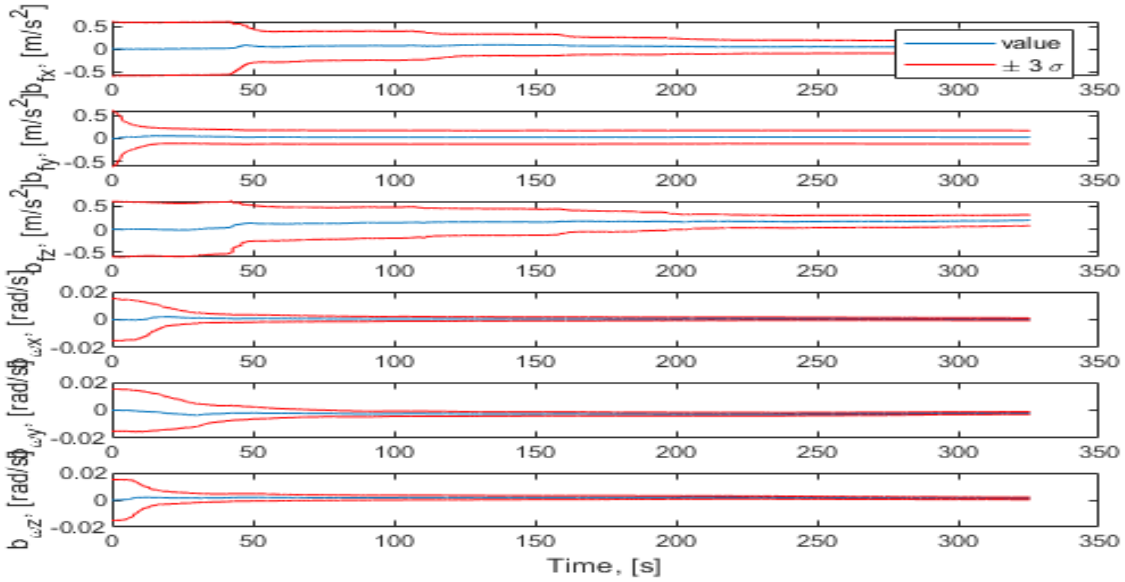


Figure 33 The output of implementing KF on sensor bias results in error minimization.

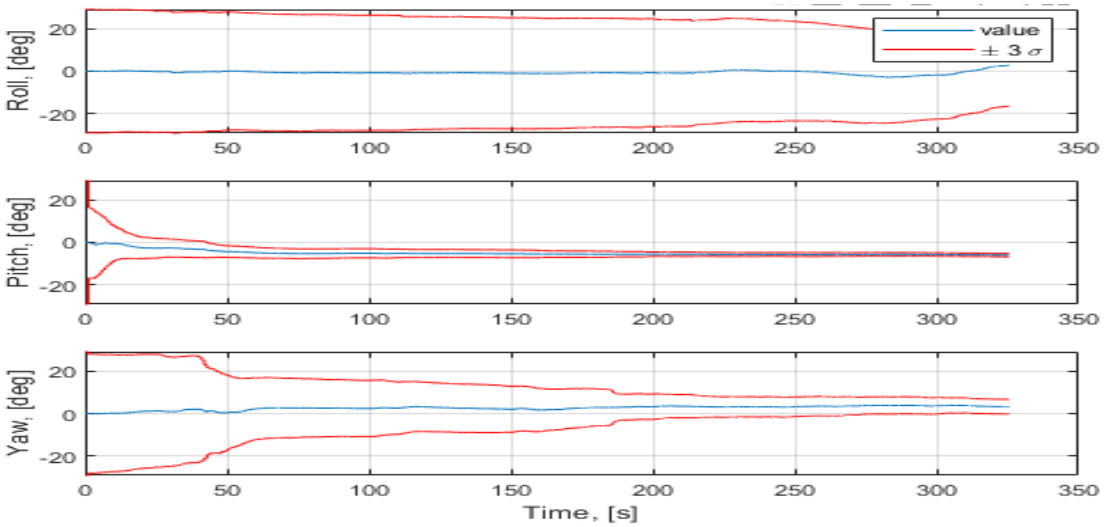


Figure 34 After implementing KF, the estimate of misalignment between IMU and car results in error minimization.

However, scale factor is difficult to estimate and is constant over time. Similarly, in figure 34, a recursive algorithm is used to minimize the misalignment between the car and the sensor.

Chapitre 7 - **Discussion de l'expérience et conclusion**

Durant l'expérience, le but était d'estimer la position du véhicule le long de la trajectoire afin qu'il reste proche du niveau de la voie, comme le montre la figure 24, Le véhicule est toujours maintenu du côté droit de la ligne centrale et n'est pas autorisé à dévier de la voie souhaitée. Les figures 25, 26 et 27 ont démontré que la technique pouvait estimer des positions que la caméra ne pouvait pas. Lorsque les nœuds en rouge sont extraits du logiciel QGIS, ils sont strictement en géométrie linéaire plutôt qu'en polynôme, ce qui fait que l'estimation du véhicule sur la Figure 24 traverse certains nœuds centraux en rouge. En raison de la difficulté d'estimer les distances latérales sur les cartes, les données quantitatives pour déterminer la distance latérale entre le véhicule et les nœuds routiers n'ont pas été présentées cette fois. Les futures travaux se concentreront sur la quantification de la distance autorisée entre un véhicule et les nœuds routiers et l'approche proposée sera combinée avec des techniques de vision pour obtenir une localisation robuste des véhicules par rapport à la voie, ce qui est nécessaire pour le système de sortie de voie.

7.1 Conclusion

Les véhicules autonomes nécessitent un positionnement précis et fiable pour naviguer en toute sécurité. Il est essentiel pour le contrôle du véhicule que le véhicule maintienne sa position sur la voie. Les LDWS actuellement disponibles reposent sur des techniques de vision pour identifier et détecter avec précision les voies. Cependant, la vision présente plusieurs inconvénients, notamment la sensibilité aux conditions météorologiques, la visibilité des marquages au sol et le manque d'information sur les virages. Le problème le

plus important était le moment où les trois désavantages se sont produits. L'événement se produit sur une courte période; par exemple, les conditions météorologiques changent de manière irrégulière et les marquages de voie visibles s'estompent en raison de l'usure de la peinture sur une courte trajectoire. De même, le virage se produit brièvement sur les routes courbes avant que la route ne devienne droite. Pour résoudre ce problème, nous avons proposé un système d'assistance pour les techniques assistées par la vision. Le système alternatif proposé est insensible à la visibilité du marquage des voies et aux conditions climatiques variable. Il est aussi efficace pour la navigation urbaine et constitue une solution rentable. De plus, il est plus simple et nécessite moins de calculs.

7.2 Limitations

Au cours de nos recherches, nous avons découvert certaines limites que nous aimerions aborder au profit des futurs chercheurs.

Compte tenu de la croissance des technologies avancées pour la localisation des véhicules sur le marché, nous avons limité cette recherche aux capteurs qui fournissent des informations sur les profils des véhicules et des routes. La deuxième limite est que, sous certaines hypothèses, le filtre de Kalman est utilisé pour estimer l’erreur afin fournir des estimations par rapport véhicule. Bien que les erreurs d'équation de navigation, les erreurs IMU et les mesures GPS aient toutes été linéarisées pour être utilisées dans le processus de Kalman, si des événements inattendus se produisent, tels qu'un freinage brusque ou des changements dans la dynamique du véhicule en raison d'un dérapage, l'ensemble du système devient non linéaire, ce qui rend le Filtre de Kalman inefficace. À l'heure actuelle, un filtre plus robuste est nécessaire pour remédier à cette lacune. La troisième limitation est que la technique de carte hors ligne, qui nécessite l'extraction de points de nœud, est une tâche

chronophage qui nécessite une sélection manuelle de chaque point de nœud le long d'une trajectoire donnée. Cependant, cette limitation peut être surmontée en utilisant des techniques d'apprentissage automatique. Enfin, seule la pose du véhicule est comparée à la position de référence au sol. Nous aurions pu développer un produit pour LDWSS en utilisant des algorithmes robustes si la distance latérale entre le véhicule et la référence réelle au sol avait été connue dans cette étude.

Chapter 7 - Discussion of experiment and conclusion

At the time of the experiment, the goal was to estimate the position of the ego vehicle along the trajectory so that it remained close to the lane level, as shown in Figure 24, The vehicle is always kept on the right side of the center node line and is not allowed to deviate from the desired lane. The observation at curves in Figure 25, 26 and 27, demonstrated that the technique could estimate positions that the camera could not. When the center nodes in red are extracted from QGIS software, they are strictly in line geometry rather than polynomial, causing the vehicle's estimation in Figure 24, to cross over some center nodes in red. Due to the difficulty of estimating lateral distances on maps, quantitative data for determining the lateral distance between the ego vehicle and road nodes were not presented this time. Future research will concentrate on quantifying the permissible distance between a vehicle and road nodes and we will combine the proposed approach with vision techniques to achieve robust lane level ego vehicle localization, which is required for lane departure system. The research study is concluded in the following paragraph.

7.1 Conclusion

Autonomous vehicles require precise and reliable positioning to navigate safely. It is critical for vehicle control that the vehicle maintains its lane position. Currently available LDWS rely on vision techniques to accurately identify and detect lanes. However, vision has several disadvantages, including weather sensitivity, visibility of lane markings, and the issue of missing information on curved roads. The most significant issue was the time in which all three disadvantages occurred. The occurrence occurs over a short period of time; for example, weather patterns change irregularly, and visible lane markings fade away due to paint wear

over a short trajectory. Similarly, turning occurs briefly on curved roads before the road becomes straight. To address this issue, we proposed an assistive system for vision-assisted techniques. The proposed alternate system is insensitive to lane marking visibility and varying climatic conditions, is effective for urban navigation, and is a cost-effective solution. Additionally, it is simpler and requires less computation.

7.2 Limitations of the study

During our research, we discovered some limitations that we would like to address for the benefit of future researchers.

Given the growth of advanced vehicle localization technologies on the market, we limited this research to sensors that provide information about the vehicle and road profiles. The second limitation is that under certain assumptions, the Kalman filter is used as an estimation error to provide vehicle estimates. Although navigation equation errors, IMU errors, and GPS measurements have all been linearized for use in the Kalman process, if any unexpected events occur, such as abrupt braking or changes in vehicle dynamics due to skidding, the entire system becomes highly nonlinear, rendering the Kalman filter ineffective. At the time, a more robust filter is required to address this shortcoming. The third limitation is that the offline map technique, which requires node point extraction, is a time-consuming task that requires manually selecting each and every node point along a given trajectory. However, this limitation can be overcome by utilising machine learning techniques. Finally, only the vehicle's pose is compared to the reference position on the ground. We could have developed a product for LDWSS using robust algorithms if the lateral distance between the vehicle and the ground truth reference had been known in this study.

References

[1] I. I. INSTITUTE. Facts + Statistics: Highway safety [Online] Available: https://www.iii.org/fact-statistic/facts-statistics-highway-safety#Motor%20vehicle%20crashes, (Accessed on 08/2019)

[2] N. K. James M. Anderson, Karlyn D. Stanley, Paul Sorensen,Constantine Samaras and Oluwatobi A. Oluwatola, "Brief History and Current State of Autonomous Vehicles," JSTOR, 2014, doi: 216.208.60.152 RAND Corporation. .

[3] J. Zhao, B. Liang, and Q. Chen, "The key technology toward the self-driving car," International Journal of Intelligent Unmanned Systems, vol. 6, no. 1, pp. 2-20, 2018, doi: 10.1108/ijius-08-2017-0008.

[4] A. Faisal, T. Yigitcanlar, M. Kamruzzaman, and G. Currie, "Understanding autonomous vehicles: A systematic literature review on capability, impact, planning and policy," Journal of Transport and Land Use, vol. 12, no. 1, 2019, doi: 10.5198/jtlu.2019.1405.

[5] W. Brenner and A. Herrmann, "An Overview of Technology, Benefits and Impact of Automated and Autonomous Driving on the Automotive Industry," in Digital Marketplaces Unleashed, 2018, ch. Chapter 39, pp. 427-442.

[6] K. Bimbraw. Autonomous Cars: Past, Present and Future :A Review of the Developments in the Last Century, the Present Scenario and the Expected Future of Autonomous Vehicle Technology

[7] Wikipedia. SAE Classification [Online] Available: https://en.wikipedia.org/wiki/Self-driving_car#SAE_Classification, (Accessed on 04/2019)

[8] A. A. A. o. S. H. a. T. Officials., "Driving down lane-departure crashes: A national priority," Washington, DC, 2008.

[9] F. H. A. (FHWA), "Roadway Departure Strategic Plan," Washington, DC, 2013.

[10] J. Cicchino, "Effects of lane departure warning on police-reported crash rates," Journal of Safety Research, pp. 61-67., 2018.

[11] T. B. Kai Li Lim, "A Methodological Review Of Visual Road Recognition Procedures For Autonomous Driving Applications," 2019, doi: arXiv:1905.01635v1.

[12] H. Zhou and H. Wang, "Vision-based lane detection and tracking for driver assistance systems: A survey," in 2017 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM), 19-21 Nov. 2017 2017, pp. 660-665, doi: 10.1109/ICCIS.2017.8274856.

[13] F. M and M. T, "Lane Detection and Tracking For Intelligent Vehicles: A Survey," in 2019 International Conference of Computer Science and Renewable Energies (ICCSRE), 22-24 July 2019 2019, pp. 1-4, doi: 10.1109/ICCSRE.2019.8807727.

[14] D. Bevly et al., "Lane Change and Merge Maneuvers for Connected and Automated Vehicles: A Survey," IEEE Transactions on Intelligent Vehicles, vol. 1, no. 1, pp. 105-120, 2016, doi: 10.1109/TIV.2015.2503342.

[15] J. E. Gayko, "Lane Departure and Lane Keeping," in Handbook of Intelligent Vehicles, 2012, ch. Chapter 26, pp. 689-708.

[16] D. Vajak, M. Vranješ, R. Grbić, and D. Vranješ, "Recent Advances in Vision-Based Lane Detection Solutions for Automotive Applications," in 2019 International Symposium ELMAR, 23-25 Sept. 2019 2019, pp. 45-50, doi: 10.1109/ELMAR.2019.8918679.

[17] L. Zheng, B. Li, B. Yang, H. Song, and Z. Lu, "Lane-Level Road Network Generation Techniques for Lane-Level Maps of Autonomous Vehicles: A Survey," Sustainability, vol. 11, no. 16, 2019, doi: 10.3390/su11164511.

[18] A. S. Mohammed, A. Amamou, F. K. Ayevide, S. Kelouwani, K. Agbossou, and N. Zioui, "The Perception System of Intelligent Ground Vehicles in All Weather Conditions: A Systematic Literature Review," Sensors, vol. 20, no. 22, 2020, doi: 10.3390/s20226532.

[19] Z. P. Shah Hussain, M.I. Hayee, "Development and Demonstration of Merge Assist System using Connected Vehicle Technology," Roadway Safety Institute Center for Transportation Studies University of Minnesota, 2019.

[20] S. H. Muhammad Faizan, M. I. Hayee, "Development and Demonstration of a cost-effective invehicle lane departure and advanced curve speed.," MN/RC 2018-34, December 2018 [Online]. Available: https://www.dot.state.mn.us/research/reports/2018/201834.pdf, (Accessed on 06/2020)

[21] D. Svensson and J. Sörstedt, "Ego lane estimation using vehicle observations and map information," in 2016 IEEE Intelligent Vehicles Symposium (IV), 19-22 June 2016 2016, pp. 909-914, doi: 10.1109/IVS.2016.7535496.

[22] C. Yong-Geon, L. Kyung-Il, and K. Jung-Ha, "Lane change and path planning of autonomous vehicles using GIS," in 2015 12th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), 28-30 Oct. 2015 2015, pp. 163-166, doi: 10.1109/URAI.2015.7358855.

[23] R. Asghar, M. Garzón, J. Lussereau, and C. Laugier, "Vehicle Localization Based on Visual Lane Marking and Topological Map Matching," in 2020 IEEE International Conference on Robotics and Automation (ICRA), 31 May-31 Aug. 2020 2020, pp. 258-264, doi: 10.1109/ICRA40945.2020.9197543.

[24] P. Penmetsa, M. Hudnall, and S. Nambisan, "Potential safety benefits of lane departure prevention technology," IATSS Research, vol. 43, no. 1, pp. 21-26, 2019, doi: 10.1016/j.iatssr.2018.08.002.

[25] S. P. Narote, P. N. Bhujbal, A. S. Narote, and D. M. Dhane, "A review of recent advances in lane detection and departure warning system," Pattern Recognition, vol. 73, pp. 216-234, 2018, doi: 10.1016/j.patcog.2017.08.014.

[26] C. L. Y. Xing, L. Chen, H. Wang, H. Wang, D. Cao, E. Velenis, and a. F.-Y. Wang, "Advances in vision-based lane detection: Algorithms, integration, assessment, and perspectives on ACP-based parallel vision," IEEE/CAA J. Autom. Sinica, vol. vol 5, no 3, pp. pp 645-661, 2018.

[27] Z. Cheng, Z. Wang, and H. Huang, "DCLane for Real-time Understanding of Lane Markings," in 2019 IEEE Intelligent Transportation Systems Conference (ITSC), 27-30 Oct. 2019 2019, pp. 3387-3392, doi: 10.1109/ITSC.2019.8917285.

[28] J. B. John Allen, Christopher Rose, "Intelligent Multi-Sensor Measurements to Enhance Vehicle Navigation and Safety Systems," 2011.

[29] C. Rose, J. Britt, J. Allen, and D. Bevly, "An Integrated Vehicle Navigation System Utilizing Lane-Detection and Lateral Position Estimation Systems in Difficult Environments for GPS," IEEE Transactions on Intelligent Transportation Systems, vol. 15, no. 6, pp. 2615-2629, 2014, doi: 10.1109/tits.2014.2321108.

[30] S. Lee and A. Lim, "An Empirical Study on Ad Hoc Performance of DSRC and Wi-Fi Vehicular Communications," International Journal of Distributed Sensor Networks, vol. 9, no. 11, p. 482695, 2013/11/01 2013, doi: 10.1155/2013/482695.

[31] Y. Li, W. Zhang, X. Ji, C. Ren, and J. Wu, "Research on Lane a Compensation Method Based on Multi-Sensor Fusion," Sensors (Basel), vol. 19, no. 7, Apr 2 2019, doi: 10.3390/s19071584.

[32] OpenStreetMap. "OSM Export." https://www.openstreetmap.org/export#map=4/21.86/98.17 (Accessed on 03/2020).

[33] M. H. a. P. Weber, "OpenStreetMap: User-generated street maps," IEEE Pervasive Comput, vol. vol 7, no. 4, pp. pp. 12-18, 2008.

[34] Mobileye. Mobileye C2-270 Technical Datasheet [Online] Available: https://itk-mdl.asutk.ru/upload/iblock/c82/Mobileye%20C2-270%20Technical%20Spec%20v1.2.pdf

[35] Autonomoustuff. Mobileye Moncamera [Online] Available: https://autonomoustuff.com/product/mobileye-camera-dev-kit/

[36] S. P. Mehta, A.; Mehta, J., "CCD or CMOS Image Sensor For Photography," in International Conference on Communications and Signal Processing (ICCSP), Melmaruvathur, India, 2015, doi: 10.1109/ICCSP.2015.7322890.

[37] T. D. Inc. CCD vs. CMOS [Online] Available: https://www.teledynedalsa.com/imaging/knowledge-center/appnotes/ccd-vs-cmos/

[38] N. B. Bernini, M.; Castangia, L.; Patander, M.; Sabbatelli, M, " Real-time obstacle detection using stereo vision for autonomous ground vehicles: A survey.," in Proceedings of the 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), Qingdao, China, 2014, doi: 10.1109/ITSC.2014.6957799. .

[39] M. V. D. Vajak, R. Grbi¢, and D. Vranje, "Recent advances in vision-based lane detection solutions for automotive applications," in in Proc. Int. Symp. ELMAR, , 2019, pp. pp. 45-50.

[40] A. M Kumar and P. Simon, "Review of Lane Detection and Tracking Algorithms in Advanced Driver Assistance System," International Journal of Computer Science and Information Technology, vol. 7, no. 4, pp. 65-78, 2015, doi: 10.5121/ijcsit.2015.7406.

[41] S. R. Hasirlioglu, A, "Introduction to rain and fog attenuation on automotive surround sensors," in In Proceedings of the IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, 2017.

[42] S. R. Hasirlioglu, A. , Maui, HI, , "A Model-Based Approach to Simulate Rain Effects on Automotive Surround Sensor Data.," in In Proceedings of the 2018 21st International

Conference on Intelligent Transportation Systems (ITSC), USA, 2018, doi: 10.1109/ITSC.2018.8569907.

[43] S. K. Hasirlioglu, A.; Doric, I.; Brandmeier, T. Rio de Janeiro, "Test methodology for rain influence on automotive surround sensors. ," in In Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC),, Brazil, 2016, doi: 10.1109/ITSC.2016.7795918.

[44] I. J. B. Xique, W.; Fard, Z.B.; Dennis, E.; Hart, B, "Evaluating Complementary Strengths and Weaknesses of ADAS Sensors," in In Proceedings of the IEEE Vehicular Technology Conference, Chicago, IL, USA,, 2018, doi: 10.1109/VTCFall.2018.8690901.

[45] C. G. Roh, J. Kim, and I. J. Im, "Analysis of Impact of Rain Conditions on ADAS," Sensors (Basel), vol. 20, no. 23, Nov 24 2020, doi: 10.3390/s20236720.

[46] Wikipedia. Lidar [Online] Available: https://en.wikipedia.org/wiki/Lidar

[47] J. HECHT. Lidar for Self Driving [Online] Available: https://www.osapublishing.org/DirectPDFAccess/4577511A-CFD1-1787 1681F42CA46FF8BC_380434/opn-29-1-26.pdf

[48] A. stuff. Lidar technical comparision [Online] Available: https://autonomoustuff.com/lidar-chart

[49] Velodyne. HDL 64 [Online] Available: https://velodynelidar.com/products/hdl-64e

[50] A. M. H. Wallace, A.; Buller, G.S. , "Full Waveform LiDAR for Adverse Weather Conditions.," IEEE Trans. Veh. Technology, 2020, doi: 10.1109/tvt.2020.2989148.

[51] Y. I.-G. Li, J. , "Lidar for Autonomous Driving: The Principles, Challenges, and Trends for Automotive Lidar and Perception Systems.," IEEE Signal Process. Mag., 2020, doi: 10.1109/msp.2020.2973615.

[52] D. I. Zermas, I.; Papanikolopoulos, N., "Fast segmentation of 3D point clouds: A paradigm on LiDAR data for autonomous vehicle applications.," in In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 2017, doi: 10.1109/ICRA.2017.7989591.

[53] I. S. Bogoslavskyi, C. , "Fast range image-based segmentation of sparse 3D laser scans for online operation. ," in In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), , Daejeon, Korea, 2016, doi: 10.1109/IROS.2016.7759050.

[54] A. P. S. Thrun, "Model based vehicle detection and tracking for autonomous urban driving.," Auton Robot, p. 26, 2009, doi: 10.1007/s10514-009-9115-1.

[55] M. M. Himmelsbach, A.; Lüttel, T.; Wünsche, H.J., "LIDAR-based 3D Object Perception," in In Proceedings of the 1st Int. Workshop on Cognition for Technical Systems, Muenchen, Germany, 2008.

[56] E. D. Capellier, F.; Cherfaoui, V.; Li, Y. , "deep learning for arbitrary LIDAR object classification in the context of autonomous driving. ," in Evidential In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 2019, pp. 1304–1311,. [Online]. Available: 10.1109/IVS.2019.8813846. [Online]. Available: 10.1109/IVS.2019.8813846

[57] W. D. P. Zeng, I.; Newman, P. , " What could move? Finding cars, pedestrians and bicyclists in 3D laser data," in In Proceedings of the IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA,, 2012, pp. 4038–4044, doi: 10.1109/ICRA.2012.6224734.

[58] J. U. B. V. Douillard, V.; Quadros, A.; Singh, , S. A Pipeline for the Segmentation and Classification of 3D Point Clouds. Berlin/Heidelberg, Germany: Springer, 2014.

[59] C. L. Premebida, O.; Nunes, U, "Exploiting LIDAR-based features on pedestrian detection in urban scenarios," in In Proceedings of the 2009 12th International IEEE Conference on Intelligent Transportation Systems,, St. Louis, MO, USA,, 2009, doi: 10.1109/ITSC.2009.5309697.

[60] S. S. Kraemer, C.; Bouzouraa, M.E., " LiDAR-Based Object Tracking and Shape Estimation Using Polylines and Free-Space Information," in In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 2018, pp. 4515–4522, doi: 10.1109/IROS.2018.8593385.

[61] X. X. Zhang, W.; Dong, C.; Dolan, J.M., " Efficient L-shape fitting for vehicle detection using laser scanners," in In Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV),, Los Angeles, CA, USA, 2017, pp. 54–59, doi: 10.1109/IVS.2017.7995698.

[62] A. S. A. Rachman. 3D-LIDAR Multi Object Tracking for Autonomous Driving. [Online] Available: https://www.semanticscholar.org/paper/3D-LIDAR-Multi-Object-Tracking-for-Autonomous-and-Rachman/bafc8fcdee9b22708491ea1293524ece9e314851

[63] Y. W. Zhang, J.;Wang, X.; Li, C.;Wang, L. , "3d lidar-based intersection recognition and road boundary detection method for unmanned ground vehicle. ," in In Proceedings of the 2015 IEEE 18th International Conference on Intelligent Transportation Systems, Las Palmas, Spain, 2015, pp. 499–504.

[64] K. S. Li, J.; Guo, D. , "A multi-feature search window method for road boundary detection based on LIDAR data. Sensors 2019, doi: Cross referenced.

[65] G. M. Min Bai, Namdar Homayounfar, Shenlong Wang, Kowshika Lakshmikanth, Shrinidhi, and Raquel Urtasun. , "Deep multi-sensor lane detection. ," IROS 2018, 2018.

[66] L. R. Hespel, N.; Huet, T.; Tanguy, B.; Ceolato, "Performance evaluation of laser scanners through the atmosphere with adverse condition," R. . Proc SPIE 2011, doi: 10.1117/12.898010.

[67] R. S. Heinzler, P.; Seekircher, J.; Ritter, W.; Stork, W. , "Weather Influence and Classification with Automotive Lidar Sensors.," in In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 2019, pp. 1527–1534, doi: 10.1109/IVS.2019.8814205.

[68] M. P. Kutila, P.; Holzhüter, H.; Colomb, M.; Duthon " Automotive LiDAR performance verification in fog and rain," in In Proceedings of the IEEE Conference on Intelligent Transportation Systems, Maui, HI, USA,, 2018, pp. 1695–1701, doi: 10.1109/ITSC.2018.8569624.

[69] M. G. Bijelic, T.; Ritter, W. , "A Benchmark for Lidar Sensors in Fog: Is Detection Breaking Down?," in In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, , 2018, doi: 10.1109/IVS.2018.8500543.

[70] Y. D. Li, P.; Colomb, M.; Ibanez-Guzman J, " What Happens for a ToF LiDAR in Fog? ," in IEEE Trans. Intell. Transp. Syst, 2020, doi: 10.1109/TITS.2020.2998077.

[71] M. P. Kutila, P.; Ritter, W.; Sawade, O.; Schäufele, B., "Automotive LIDAR sensor development scenarios for harsh weather conditions," in In Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC),, Rio de Janeiro, Brazil, 2016, pp. 265–270.

[72] C. C. Goodin, D.; Doude, M.; Hudson, C. , "Predicting the influence of rain on LIDAR in ADAS. ," Electronics pp. 8, 89., 2019.

[73] M. Y. Byeon, S.W. , "Analysis of Automotive Lidar Sensor Model Considering Scattering Effects in Regional Rain Environments. ," IEEE Access pp. 102669–102679, 2020, doi: 10.1109/access.2020.2996366.

[74] S. L. Michaud, J.-F.; Giguère, P. , "Towards Characterizing the Behavior of LiDARs in Snowy Conditions.," in In Proceedings of the 7th Workshop on Planning, Perception and Navigation for Intelligent Vehicles, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), , Hamburg, Germany, , 2015.

[75] J. H. Ryde, N. , "Performance of laser and radar ranging devices in adverse environmental conditions. ," J. Field Robot., pp. 712–727, 2009.

[76] J. Z. Wojtanowski, M.; Kaszczuk, M.; Mierczyk, Z.; Muzal, M., "Comparison of 905 nm and 1550 nm semiconductor laser rangefinders' performance deterioration due to adverse environmental conditions.," Opto-Electron. Rev. , pp. 183–190, 2014, doi: 10.2478/s11772-014-0190-2.

[77] Wikipedia. Global Positioning system (GPS) [Online] Available: https://en.wikipedia.org/wiki/Global_Positioning_System, (Accessed on 05/2020)

[78] S. A. R. Z. a. R. M. BUEHRER, Handbook of Position Location: Theory_Practice and Advances. Hoboken, New Jersey: A JOHN WILEY & SONS, , 2012.

[79] G. Images. Ublox 6 neo GPS receiever [Online] Available: https://www.rabtron.co.za/arduino/8866-ublox-neo-6-gps-module.html (Accessed on 05/2020)

[80] V. L. Knoop, P. F. d. Bakker, C. C. J. M. Tiberius, and B. v. Arem, "Lane Determination With GPS Precise Point Positioning," IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 9, pp. 2503-2513, 2017, doi: 10.1109/TITS.2016.2632751.

[81] Z. Tao, "Autonomous road vehicles localization using satellites, lane markings and vision," Information Technologies and Systems, Université de Technologie de Compiègne, 2016.

[82] I. Skog and P. Handel, "In-Car Positioning and Navigation Technologies—A Survey," IEEE Transactions on Intelligent Transportation Systems, vol. 10, no. 1, pp. 4-21, 2009, doi: 10.1109/tits.2008.2011712.

[83] H. L. D. Obradovic, and M. Schupfner,, "Fusion of sensor data in Siemens car navigation system," IEEE Trans. Veh. Technol., , vol. vol. 56, pp. 43–50, 2007.

[84] H. L. D. Obradovic, and M. Schupfner, "Fusion of map and sensor data in a modern car navigation system," VLSI Signal Process, vol. vol. 45, pp. 111–122, 2006.

[85] J. G. R. Carlson, and J. Powell, and pp. 13–27, " Error sources when land vehicle dead reckoning with differential wheelspeeds," Navigation, vol. vol. 51, 2004.

[86] J. G. R. Carlson, and J. Powell, " Practical position and yaw rate estimation with GPS and differential wheelspeeds " in in Proc. 6th Int. Symp. AVEC, Hiroshima, Japan, 2002.

[87] K. B. E. v. Puttkamer, Autonomous Land Vehicles: Steps towards Service Robots. GWV Fachverlage GmbH, Wiesbaden Vieweg+Teubner 2009.

[88] C. electronics. Wheel Encoder [Online] Available: https://core-electronics.com.au/wheel-encoder.html, (Accessed on 11/2020)

[89] Velocity encode Available: https://www.pinterest.com/pin/755338168740409573/, (Accessed on 11/2020)

[90] S. products. Steering wheel encoder, Available: https://www.skf.com/ven/products/rolling-bearings/engineered-products/sensor-bearing-units/steering-encoder-units, (Accessed on 11/2020)

[91] C. Coopmans, "Architecture, Inertial Navigation, and Payload Designs for Low-Cost Unmanned Aerial Vehicle-Based Personal Remote Sensing," Utah State University, 2010. [Online]. Available: http://digitalcommons.usu.edu/etd/692

[92] Wheel speed sensor Available: https://en.wikipedia.org/wiki/Wheel_speed_sensor (Accessed on 11/2020)

[93] T. B. Schön, J. D. Hol, and M. Kok, "Using Inertial Sensors for Position and Orientation Estimation," Foundations and Trends® in Signal Processing, vol. 11, no. 1-2, pp. 1-153, 2017, doi: 10.1561/2000000094.

[94] B. W. Abegaz and N. Shah, "Sensors based Lane Keeping and Cruise Control of Self Driving Vehicles," in 2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), 28-31 Oct. 2020 2020, pp. 0486-0491, doi: 10.1109/UEMCON51285.2020.9298141.

[95] IEEE Standard for Inertial Sensor Terminology, I. S. 528, 2001, (Accesssed on 07/2020)

[96] M. A. R. Curey, L. Thielman, and C. Barker, "Proposed IEEE inertial systems terminology standard and other inertial sensor standards " in in Proc. PLANS, 2004, pp. 83–90.

[97] B. L. a. Q. C. Jianfeng Zhao, "The key technology toward the self-driving car," International Journal of Intelligent Unmanned Systems, vol. Vol. 6 pp. 2-20, 2018, doi: 10.1108/IJIUS-08-2017-0008. Emerald Publishing Limited.

[98] Q. Z. Xiaoji Niu, You Li, Yahao Cheng, Chuang Shi, "Using Inertial Sensors of iPhone 4 for Car Navigation," IEEE Access, 2012, doi: 978-1-4673-0387-3/12.

[99] N. V. Ravi Bhoraskar, Bhaskaran Raman, Purushottam Kulkarni, "Wolverine: Traffic and road condition estimation using smartphone sensors," presented at the 2012 Fourth International Conference on Communication Systems and Networks (COMSNETS 2012), Bangalore, India, 2012.

[100] R. B. Joshua Siegel, Sanjay Sarma, Ajay Deshpande, "Smartphone-Based Vehicular Tire Pressure and Condition Monitoring," in Proceedings of SAI Intelligent Systems Conference (IntelliSys) 2016, 2016, pp. 805-824, doi: 10.1007/978-3-319-56994-9_56.

[101] E. Mantouka, E. Barmpounakis, E. Vlahogianni, and J. Golias, "Smartphone sensing for understanding driving behavior: Current practice and challenges," International Journal of

Transportation Science and Technology, 2020/07/11/ 2020, doi: https://doi.org/10.1016/j.ijtst.2020.07.001.

[102] G. Castignani, T. Derrmann, R. Frank, and T. Engel, "Smartphone-Based Adaptive Driving Maneuver Detection: A Large-Scale Evaluation Study," IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 9, pp. 2330-2339, 2017, doi: 10.1109/TITS.2016.2646760.

[103] P. D. Groves, Principles of GNSS, Inertial, and Multi-Sensor Integrated Navigation Systems (GNSS Technology and Applications) (Chapter 3: The kalman Filter : Algorithms and models). 2008.

[104] L. R. W. a. A. P. A. M. S. Grewal, Global POsitioning Systems, Inertial Navigation, and Integration. Hoboken, New Jersey Wiley, 2007.

[105] K. J. WALCHKO, "Kalman Filtering on Inertial Navigation," MASTER OF SCIENCE, UNIVERSITY OF FLORIDA, 2002, (Accessed on 08/2020)

[106] L. R. W. M. S. Grewal, and A. P. Andrews, Global Positioning Systems, Inertial Navigation, and Integration (COORDINATE TRANSFORMATIONS). John Wiley & Sons, Inc., 2007.

[107] S. e. al, The standard robotics textbooks (Representing Position and Orientation). 2006. (Accessed on 06/2020)

[108] M. S. Mark Pedley. Quaternion Algebra and Rotation Quaternions, Online (Accessed on 03/2020)

[109] M. S. Mark Pedley. Matrix and Vector Algebra, Online (Accessed on 03/2020)

[110] M. S. Mark Pedley. Aerospace, Android and Windows 8 Coordinate Systems, Online (Accessed on 03/2020)

[111] G. Falco, M. Pini, and G. Marucco, "Loose and Tight GNSS/INS Integrations: Comparison of Performance Assessed in Real Urban Scenarios," Sensors (Basel), vol. 17, no. 2, Jan 29 2017, doi: 10.3390/s17020255.

[112] D. H. T. a. J. L. weston, Strapdown Inertial navigation Technology (Strapdown system mechanisation). United Kingdom: The Institution of Electrical Engineers, 2004.

[113] Coriolis Force Available: https://en.wikipedia.org/wiki/Coriolis_force (Accessed on 05/2020)

[114] Wikipedia. Kalman filter for navigation problems [Online] Available: https://en.wikipedia.org/wiki/Kalman_filter (Accessed on 05/2020)

[115] S. A. R. Z. Shu Ting Goh, Ossama Abdelkhalik, Handbook of position location: Theory, Practice, and Advances (Chapter 5 : An Introduction to Kalman Filtering Implementation for Localization and Tracking Applications). Hoboken, New Jersey: John Wiley & Sons, Inc, 2019.

[116] J. A. FARRELL, Aided Navigation: GPS with high rate sensors (Chapter 3 : Deterministic systems and Error linearization). 2008.

[117] S. A. R. Z. a. R. M. BUEHRER, Handbook of Position Location: Theory, Practice and Advances, 1 ed. (Discrete Kalman Filter). Hoboken, New Jersey: John Wiley & Sons, Inc., 2012.

[118] P. A. Jones, "Techniques in Kalman Filtering for Autonomous Vehicle Navigation," Mechanical Engineering, Virginia Polytechnic Institute and State University, 2015.

[119] D. Simon, Optimal state estimation ; Kalman, H-infinity, and nonlinear approaches. John Wiley & Sons, Inc, 2006.

[120] Mathforcollege. Textbook notes for Runge-Kutta 2nd Order Method for Ordinary Differential order [Online] Available: http://mathforcollege.com/nm/mws/gen/08ode/mws_gen_ode_txt_runge2nd.pdf, (Accessed on 11/2020)

[121] Wikipedia. Heun's method : Two stage Runge-Kutta method [Online] Available: https://en.wikipedia.org/wiki/Heun%27s_method (Accessed on 11/2020)

[122] J. A. FARRELL, Aided Navigation: GPS with high rate sensors (Chapter 5 : Optimal State estimation) (5.6.2 Corelated Measurements and 5.6.3 Bad or Missing Data). McGraw-Hill Companies, Inc, 2008.

[123] A. P. A. Mohinder S. Grewal, Kalman Filtering: Theory and Practice with MATLAB (Section 4: Factorization Methods For “SQUARE-ROOT” Filtering). Hoboken, New Jersey: John Wiley & Sons, Inc., 2015, p. 640

[124] K. C. Tschangho J. Kim, Springer Handbook of Geographic information (GIS for Transportation). Springer, Berlin, Heidelberg: Springer-Verlag 2011.

[125] M. A. Quddus, W. Y. Ochieng, and R. B. Noland, "Current map-matching algorithms for transport applications: State-of-the art and future research directions," Transportation Research Part C: Emerging Technologies, vol. 15, no. 5, pp. 312-328, 2007, doi: 10.1016/j.trc.2007.05.002.

[126] L. W. Liu, T.; Fang, Y.; Hu, T.; Song, J. , "A smart map representation for autonomous vehicle navigation.," in In Proceedings of the 2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), Zhangjiajie, China, 2015, pp. 2308–2313.

[127] I. C. Shim, J.; Shin, S.; Oh, T.-H.; Lee, U.; Ahn, B.; Choi, D.-G.; Shim, D.H.; Kweon, I.-S. , "An autonomous driving system for unknown environments using a unified map," EEE Trans. Intell. Transp. Syst.,2015.

[128] P. Z. Bender, J.; Stiller, C. Lanelets: , "Efficient map representation for autonomous driving" in In Proceedings of the 2014 IEEE Intelligent Vehicles Symposium Proceedings, , Dearborn, MI, USA, 2014, pp. 420–425.

[129] K. O. Jetlund, E.; Huang, L. , "Information Exchange between GIS and Geospatial ITS Databases Based on a Generic Model," ISPRS Int. Geo-Inf. , 2019.

[130] S. F. Kuutti, S.; Katsaros, K.; Dianati, M.; Mccullough, F.; Mouzakitis, A. , "A survey of the state-of-the-art localization techniques and their potentials for autonomous vehicle applications.," IEEE Internet Things J. , pp. 829–846, 2018.

[131] M. A. Quddus, R. B. Noland, and W. Y. Ochieng, "The Effects of Navigation Sensors and Spatial Road Network Data Quality on the Performance of Map Matching Algorithms," GeoInformatica, vol. 13, no. 1, pp. 85-108, 2008, doi: 10.1007/s10707-007-0044-x.

[132] D. B. a. A. Kornhauser. An introduction to map matching for personal navigation assistants [Online] Available: http://www.njtide.org/reports/ (Accessed on 09/2020)

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

[134] W. O. M. Quddus, L. Zhao, and R. Noland, "A general map matching algorithm for transport telematics applications, ," GPS Solutions, vol. vol. 7, , no. 3, pp. 157–167, 2003.

[135] Google. Google Maps [Online] Available: https://www.google.co.in/maps, (Accessed on 05/2020)

[136] Wikipedia. Navteq Maps [Online] Available: https://en.wikipedia.org/wiki/Navteq, (Accessed on 06/2020)

[137] QGIS. "QGIS : A Free and Open source Geographic information system." https://www.qgis.org/en/site/ (Accessed on 06/2020).

Appendix A – Software application used

Physics Toolbox Sensor suite android application: Data used to determine the position of a vehicle. Figure 33 depicts the specifications of the sensors as well as the principles on which they operate. The tool provides information such as vehicle kinematics and positioning data:

(a) **Kinematics of the Vehicle**:

G-Force Meter - Fn/Fg ratio (x, y, z, and/or total)

Acceleration (x, y, and/or z) is measured by a accelerometer.

A gyroscope measures radial velocity (x, y, and/or z).

(b) **GPS positioning data**: latitude, longitude, altitude, speed, direction, and number of satellites.

Procedure to operate:

1.) Open application and select type of sensors you want to utilize
2.) Set update rates of sensors as per your requirement.
3.) First calibrate and collect initial data.
4.) Create a folder which will be used in saving recorded data of sensor.
5.) Push record button on screen for recording sensor data and stop for closing the recording.
6.) Finally, save the recorded data in CSV file format and use it for interpolation.

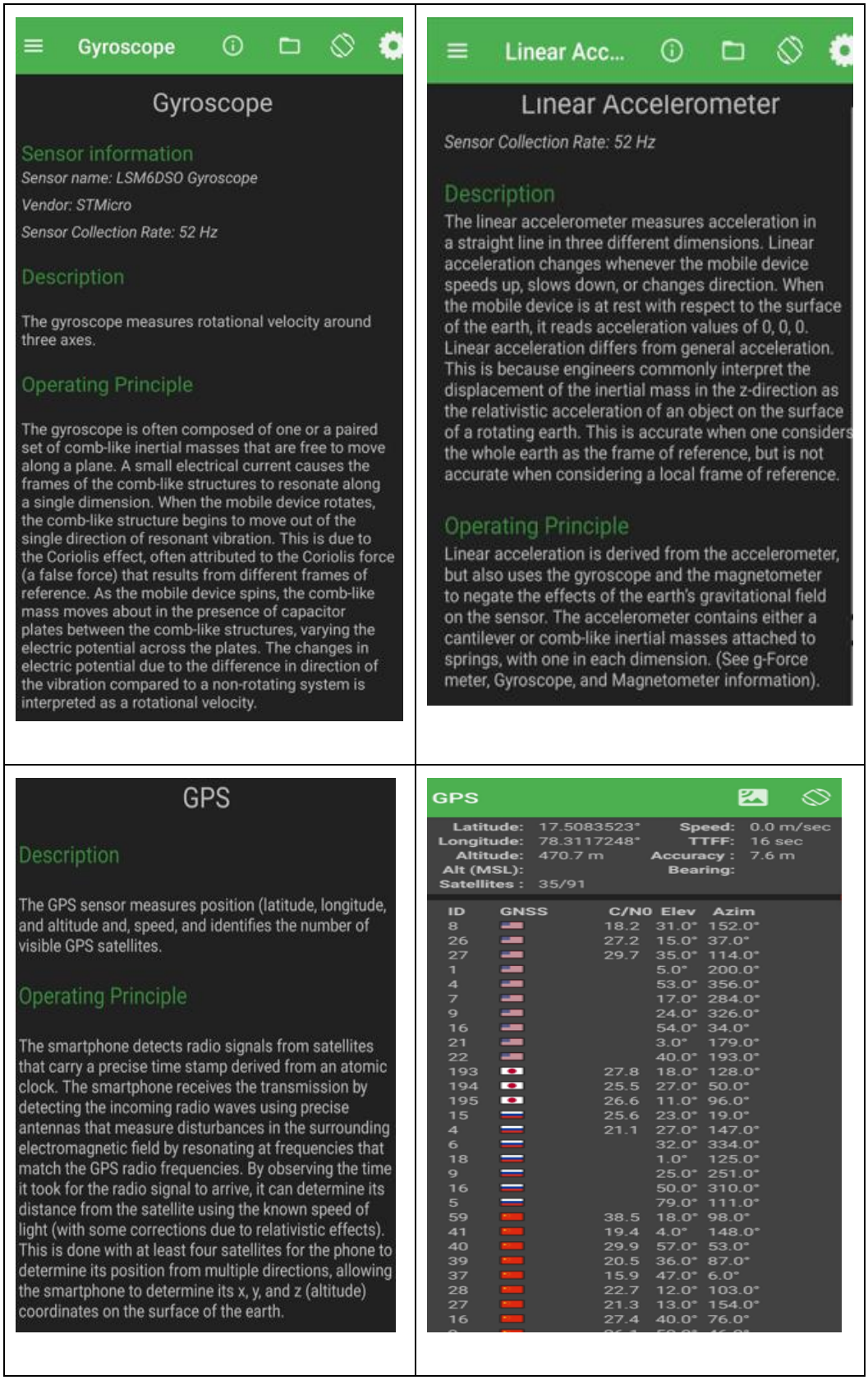


Figure 35 Sensor data about the application is displayed.

This application collects, displays, records, and exports.csv data (see figure 34) files using internal smartphone sensors. Visit www.vieyrasoftware.net to learn more about case studies in R&D. Smartphone hardware determines sensor availability, precision, and accuracy. Depending on individual requirements, a combination of sensors can be used at the same time.

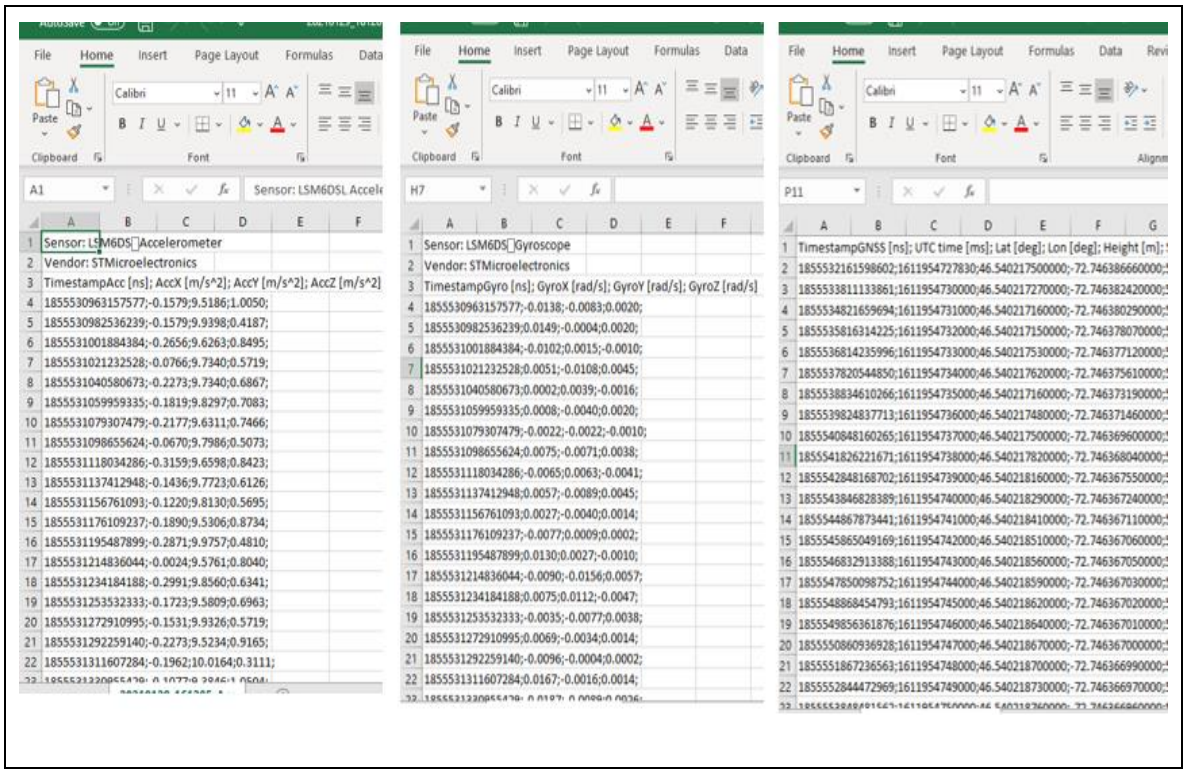


Figure 36 CSV file which consists of sensor data

Appendix B – Published Articles

1.) Sensors 2020, 20, 6532; “The Perception System of Intelligent Ground Vehicles in All Weather Conditions: A Systematic Literature Review”, doi:10.3390/s20226532. www.mdpi.com/journal/sensors

2.) 2021 IEEE Vehicle Power and Propulsion Conference (VPPC); “Vehicle Lane departure estimation on urban roads using GIS information”, doi: 10.1109/VPPC53923.2021.9699117