Università degli Studi di Napoli Federico II
Ph.D. Program in
Information Technology and Electrical Engineering
XXXVI Cycle

Thesis for the Degree of Doctor of Philosophy

LiDAR Systems for Advanced Assisted Driving in Tramway Sector

by

Martina Guerritore

Advisor: Prof. Mauro D'Arco

Co-advisor:
Ing. Luigi Fratelli, PhD
Ing. Giuseppe Graber, PhD

Scuola Politecnica e delle Scienze di Base
Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione

"Non come chi vince sempre, ma come chi non si arrende mai."
(Frida Khalo)

Sono giunta alla fine di questo percorso accademico, terminato con tanto impegno e perseveranza. Ricordo il mio primo giorno da dottoranda, avevo gli occhi che brillavano di sogni e tanta voglia di dare il mio contributo alla ricerca. In questi anni ho condiviso gioie, successi e talvolta anche pianti ma non mi sono mai sottratta alle sfide che il percorso mi presentava; le ho sempre affrontate con tenacia, voglia di vincere ed ambizione. A questo punto non posso che ringraziare la mia famiglia per ogni singolo giorno, ora e minuto trascorso al mio fianco. Ringrazio il Prof. Mauro D'Arco per la costante presenza, la sua immensa pazienza, i suoi indispensabili consigli. Durante questo percorso di dottorato mi ha seguita passo dopo passo, trasmettendomi conoscenza, metodo di lavoro ma soprattutto correttezza ed onestà intellettuale; qualità che lo contraddistinguono e che ammiro. Ringrazio l'azienda Hitachi in Napoli e in Sophia-Antipolis, per l'opportunità professionale che mi ha concesso. Lavorare presso Hitachi mi ha permesso di mettere in pratica le conoscenze acquisite durante il mio percorso accademico e di sviluppare competenze pratiche fondamentali nel settore. Un ringraziamento speciale lo dedico a Luigi e Giuseppe, ai quali sono grata per il sostegno e l'orientamento ricevuti. Mi hanno permesso di crescere sia professionalmente che personalmente, ma soprattutto mi hanno permesso di svolgere nuove esperienze; come l'esperienza lavorativa in Francia. La Francia, che bei ricordi! In Francia, ho avuto l'opportunità di collaborare con un team eccezionale, dinamico e allegro. Grazie all'internazionalità della squadra, ho visto nuove realtà e culture diverse.

Inoltre, sono grata per l'ambiente di lavoro stimolante e inclusivo che ho trovato all'interno dell'università. Ogni giorno ho avuto l'opportunità di collaborare con colleghi fantastici, con un riferimento particolare a Lucia e Umberto, i quali mi hanno ispirata ed incoraggiata a dare il meglio di me stessa. Ora sarebbe consuetudine ringraziare gli amici, coloro che hanno reso questo viaggio non solo più sopportabile, ma anche più memorabile e gratificante. A Stefano, il mio compagno di avventure e confidente. Grazie per esserci sempre stato, e per avermi sempre ascoltata. A tutti gli altri amici che hanno condiviso con me momenti di gioia, risate, e anche di sacrificio durante questo percorso, vi ringrazio di cuore.

LiDAR Systems for Advanced Assisted Driving in Tramway Sector

Ph.D. Thesis presented
for the fulfillment of the Degree of Doctor of Philosophy
in Information Technology and Electrical Engineering
by

Martina Guerritore

January 2024

Approved as to style and content by

Mauro D'Arco

Prof. Mauro D'Arco, Advisor

Università degli Studi di Napoli Federico II
Ph.D. Program in Information Technology and Electrical Engineering
XXXVI cycle - Chairman: Prof. Stefano Russo

http://itee.dieti.unina.it

Candidate's declaration

I hereby declare that this thesis submitted to obtain the academic degree of Philosophiæ Doctor (Ph.D.) in Information Technology and Electrical Engineering is my own unaided work, that I have not used other than the sources indicated, and that all direct and indirect sources are acknowledged as references.
Parts of this dissertation have been published in international journals and/or conference articles (see list of the author's publications at the end of the thesis).

Napoli, March 18, 2024

Martina Guerritore

Martina Guerritore

Abstract

In my thesis, I focused on developing an assisted driving system for the tramway industry using Light Detection and Ranging (LiDAR). Research objectives include identifying, and localizing objects and then, estimating tram position in unknown environments. Regarding the first objective, the proposed methodology involves background subtraction, pose estimation, and 3D bounding box fitting for detailed state analysis, aiming to assess collision risks and alert drivers. An approach balancing speed and accuracy is proposed to overcome hardware limitations in processing high-performance LiDAR data. The solution is tested on real data sets collected on a city road with a 360° LiDAR. The effectiveness of this solution is compared with the most widely cited solutions. The second objective introduces two approaches for estimating tram position and orientation. For the tram position, a sensor fusion approach is suggested, enhancing accuracy and reliability and ensuring continuous localization despite sensor malfunctions. Unlike several alternatives, the considered one operates without dynamic or error models, offering a low computational burden. Performance analysis is conducted in a simulation environment replicating an architecture with four kinematic sensors. The adopted method maintains the error in position measurement, derived indirectly from an accelerometer with a 1 ppm offset on the full scale, within a few ppm of the full-scale position. Instead, estimating orientation involves integrating gyroscope measurements, but integrating over extended periods accumulates errors, often addressed using complementary filters to fuse data from accelerometers and gyroscopes, refining the orientation estimate. To avoid accelerometer-related errors, a method to fusion point clouds and gyroscope measurements is proposed. Real data is utilized to evaluate the effectiveness of all proposed strategies in a real case study.

Keywords: LiDAR, IMU, attitude, SLAM, object detection, bounding box.

Sintesi in lingua italiana

Nella mia tesi, mi sono concentrata sullo sviluppo di un sistema di guida assistita per l'industria tranviaria utilizzando il Light Detection and Ranging (LiDAR). Gli obiettivi di ricerca includono l'identificazione e la localizzazione di oggetti e successivamente la stima della posizione del tram in ambienti sconosciuti. Per quanto riguarda il primo obiettivo, la metodologia proposta prevede la sottrazione dello sfondo, la stima della posizione e l'adattamento di una riquadro 3D per un'analisi dettagliata dello stato, mirando a valutare i rischi di collisione e avvisare i conducenti. Un approccio che bilancia la velocità e l'accuratezza è proposto così da superare le limitazioni hardware nel processare dati LiDAR ad alte prestazioni. La soluzione viene testata su set di dati reali raccolti su una strada cittadina con un LiDAR a 360°. L'efficacia di questa soluzione viene confrontata con le soluzioni più ampiamente citate. Il secondo obiettivo introduce due approcci, uno per stimare la posizione e l'altro per l'orientamento del tram. Per la posizione del tram, viene suggerito un approccio di fusione dei sensori così da migliorare l'accuratezza e l'affidabilità e garantendo la localizzazione anche in presenza di eventuali malfunzionamenti di un sensore. A differenza delle alternative, la soluzione considerata opera senza modelli dinamici o di errore, offrendo un basso carico computazionale. L'analisi delle prestazioni viene effettuata in un ambiente di simulazione che replica un'architettura con quattro sensori cinematici. Il metodo adottato mostra un errore sulla posizione (ottenuta indirettamente da un accelerometro con un offset pari a 1 ppm del fondo scala) all'incirca di una ppm dalla posizione massima. Invece, la stima dell'orientamento coinvolge l'integrazione delle misurazioni del giroscopio, ma l'integrazione nel tempo porta ad accumulo di errori, spesso risolti utilizzando filtri complementari per fondere i dati dagli accelerometri e dai giroscopi, migliorando la stima dell'orientamento. Per evitare errori legati agli accelerometri, viene proposto un metodo per la fusione di misurazioni del giroscopio e dati LiDAR. Dati reali vengono utilizzati per valutare l'efficacia di tutte le strategie proposte in uno studio di caso reale.

Parole chiave: LiDAR, IMU, attitude, SLAM, object detection, bounding box.

Acknowledgements

The author's work has been supported by a PhD scholarship funded by Istituto Nazionale della Previdenza Sociale (INPS), Dottorati innovativi Intersettoriali "Industria 4.0",.

Contents

List of Acronyms

The following acronyms are used throughout the thesis.

ML	Machine Learning
LiDAR	Light Detection And Ranging
RADAR	RAdio Detection And Ranging
IMU	Inertia Measurement Unit
T-ADAS	Tram-Advanced Driver Assistance System
ADAS	Advanced Driver Assistance System
KF	Kalman Filter
eq.	equation

List of Figures

List of Tables

List of Symbols

The following symbols are used within the thesis

c	Speed of light in a vacuum
h	Planck constant

Chapter 1

Introduction

Experience is the hardest kind of teacher: it gives you the test first and the lesson afterward.

Oscar Wilde

1.1 Motivation

In the coming decades, the demand for global mobility is poised to surge predominantly within urban centers. The epicenter of this surge is anticipated to be the expansive metropolitan regions of emerging nations, notably in countries like China and India, as well as specific areas in Africa and South America. This surge, however, comes at a considerable cost, manifesting in heightened pollution levels and congestion issues [5].

Governments worldwide are increasingly gravitating towards a sustainable solution to address these challenges, with a particular emphasis on developing a public transport network centered around electric carriers. Among the available options, the tram specifically emerges as a highly sustainable solution for urban environments.

The tram system is not only environmentally friendly, characterized by its green credentials, but also boasts remarkably low operational costs. Moreover, its implementation requires significantly smaller infrastructure investments when compared to more extensive metro projects.

This makes the tram system a well-integrated and economically viable solution. Trams could be placed on the existing urban landscape, however it is important to enhance safety measures and seamless integration within urban environments. The tram operates in a mixed setting, sharing spaces with pedestrians, cyclists, various vehicles, and other road users.

Nowadays, it is beginning a transition into smart cities, characterized by the coexistence of traditional and autonomous vehicles equipped with increasingly sophisticated self-driving systems. In this futuristic urban environment, an augmentation of driver assistance systems in trams becomes a logical progression. Parallel to advancements in the automotive industry, which has defined and implemented new services and technologies to develop indispensable Advanced Driver Assistance Systems (ADAS)s for vehicular safety, a similar trajectory can be thought for the tramway sector. This involves defining functional and technological platforms for the creation of a Tram Advanced Driver Assistance System (T-ADAS), ensuring both adaptability and safety in the evolving urban landscape [6, 7].

Nevertheless, despite advanced research and prototypes dedicated to autonomous driving vehicles, including driverless ones, the tramway sector is still in its early days [8, 9, 10, 11, 12].

As a consequence, the tram industry for the development of T-ADASs should take cues from technological solutions already existing in the automotive. The calibration of these advancements to suit the distinctive needs of tramways is crucial for the integration of advanced automation.

1.2 Analysis of urban context

A growing emphasis on electric mobility and ecological attributes of tram systems has given the widespread adoption of trams in recent years [13].

Nowadays, tramways are intricately woven into the infrastructure of cities, often sharing roads with both cars and pedestrians, lacking dedicated lanes. Conversely, constructing dedicated paths for trams encounters huge architectural and urban challenges due to space constraints within cities and is often avoided [14].

The absence of dedicated lanes contributes to a growth in the frequency of accidents, predominantly minor in nature respect with to vehicle ones [15, 16].

Total	108
Intersection	59
Roundabout	3
Level crossing	2
Straight	39
Curve	5

Table 1.1. Number of accidents involving trams in Italy from 2022 report of ISTAT

Statistical evidence from The Italian National Institute of Statistics (ISTAT), in its 2022 annual report on road accidents in Italy, underscores that a substantial of incidents involving trams occur at intersections and straight sections. One of the issues, is the unforeseen behavior of cars, motorcycles, and pedestrians, leaving tram drivers with minimal reaction time. All these elements have a substantial impact on the drivers' behavior working at challenging driving scenarios [17].

The difficulties grow particularly in speed control and adjustment within the tumultuous urban scenario. Skills of tram drivers need heightened vigilance and a keen ability to anticipate potential road events [18].

A Tram Advanced Driver Assistance System (T-ADAS) should provide support in managing these driving tasks. Effective T-ADAS solutions mandate a fusion among the requirements of the railway domain to which trams belong and technologies ADAS of automotive.

1.3 Automation levels

Autonomous driving finds applications also in the railway sector, in fact, Automatic Train Operation (ATO) is a high level of automation extensively employed in driverless metro systems. ATO oversees all operations of a train, managing tasks such as acceleration, cruising, emergency braking, and precise stopping at designated locations.

The fundamental objective of integrating assisted and autonomous driving into the railway domain is to enhance train safety. This is especially critical due to the substantial braking distances required due to the adhesion coefficient between the wheel and rail, and the elevated speed and mass characteristic of trains. Modern trains are equipped with Automatic Train Protection (ATP) systems activating braking in the event of no safety. Moreover, some ones also incorporate Automatic Train Control (ATC) systems that, in addition to ATP functions, restrict speed if the train operator fails when necessary [19].

Grades of Automation (GoA) in the railway sector refer to a classification system that delineates the extent of automation in the operation of trains. The International Association of Public Transport (UITP) defines ATO, ATC, and ATP as GoA, each specifying the operational responsibilities allocated either to an automatic system or human intervention. There are five distinct GoA levels:

- GoA 0:This level involves train operations wherein the driver relies on direct line-of-sight, comparable to a tram system navigating through street traffic. The safety of the system is entirely dependent on the train driver.

- GoA 1: Known as Non-automated Train Operation (NTO), this level features a driver in the cabin while the automatic system prevents unsafe operations in violation of speed limit signals (ATP).

- GoA 2: Designated as Semi-automated Train Operation (STO), at this level, the ATO system controls train movement through acceleration and deceleration commands. The ATP system supervises the train's movement, while the driver in the cabin monitors the guideway, stopping the train in hazardous situations. The automatic opening and closing of doors may be managed either automatically or by the driver.

- GoA 3: Referred to as Driverless Train Operation (DTO), GoA 3 involves a fully automated system that handles train movement, including acceleration, deceleration, and door operations. However, a staff member may be present for customer service and emergencies.

- GoA 4: This level signifies Unattended Train Operation (UTO), where the train operates fully autonomously without any onboard staff. External control centers manage the train's entire operation, including passenger services and emergency responses.

These Grades of Automation provide a standardized framework for understanding the level of automation and human involvement in the operation of trains within the railway sector.
ADAS (Advanced Driver Assistance Systems) levels refer to a classification system that categorizes the degree of automation and autonomy in vehicles based on the capabilities of their driver-assistance features. These levels range from basic assistance systems to fully autonomous driving, delineating the extent of support provided by the technology to the driving experience. The levels are defined by the Society of Automotive Engineers (SAE) in the J3016 standard and are:

- Level 0 (No Automation): The vehicle has no automation, and the driver is solely responsible for all aspects of driving.

- Level 1 (Driver Assistance): The vehicle can assist with either steering or acceleration/deceleration, but not both simultaneously. Examples include Adaptive Cruise Control (ACC) or Lane-Keeping Assist.

- Level 2 (Partial Automation): The vehicle can control both steering and acceleration/deceleration simultaneously under certain conditions. The driver must remain engaged and monitor the driving environment. Examples include systems like Tesla's Autopilot and GM's Super Cruise.

- Level 3 (Conditional Automation): The vehicle can handle certain driving tasks under specific conditions. The driver can disengage from active control but must be ready to intervene if prompted by the system. At this level, the vehicle is capable of handling most driving scenarios within certain domains. As of my last update, there were limited deployments of Level 3 systems.

- Level 4 (High Automation): The vehicle can perform most driving tasks autonomously within predefined conditions and environments. The driver may not need to intervene in certain scenarios. Level 4 vehicles are often used in specific applications like ride-sharing or controlled environments.

- Level 5 (Full Automation): The vehicle is fully autonomous and can handle all driving tasks in all conditions without any human intervention. There is no need for a steering wheel or driver's seat. Level 5 autonomy represents full, hands-off automation.

1.4 An overview on sensors for ADAS

Advanced Driver Assistance Systems (ADAS) are currently experiencing widespread adoption in all recently manufactured vehicles. They present significant opportunities for enhancing road safety by assisting both the perceptual abilities of the driver and the decision-making process in various driving scenarios [20]. This serves as a tangible illustration of an Intelligent Transportation System (ITS). ADAS implementations dedicated to automotive safety and autonomous driving rely on standalone sensor devices (camera, RADAR, LIDAR, ultrasonic, etc.) integrated into the vehicles. These devices aim to enhance the driver's awareness of the surrounding environment [21].

1.4.1 Automotive Radar

Radar stands as an electronic device designed to detect targets through electromagnetic waves. Based on bandwidth, radar can be categorized into narrow band and wide band, with the latter providing enhanced range resolution. Two predominant frequency bands are 24 GHz and 77 GHz: the former facilitates short and medium-range detection within 5-70m, while the latter primarily supports medium to long-range detection spanning 100-250m. As precision requirements for automotive radar systems continue to rise, the ultra-high frequency band of 77 GHz is expected to dominate [22].

Radar can further be segmented into short-range radar (SRR), medium-range radar (MRR), and long-range radar (LRR) based on the covered area. SRR, characterized by higher range resolution and a broader field of view (FoV ±65° ±80°), is deployed for collision avoidance and object detection. Typical applications include blind-spot surveillance, parking assistance, and Adaptive Cruise Control (ACC) support. MRR (typical FoV ±40° ±50°) finds use in Lane Change Assist (LCA). LRR, featuring high gain and a narrow beam for extended detection range and precise direction measurement, possesses a smaller FoV of around ±10º and greater range resolution tolerance [23]. It is primarily employed for functions such as ACC, Forward Collision Warning (FCW), and Automatic Emergency Braking (AEB) .

Leading global radar sensor manufacturers encompass Bosch, Delphi, Continental, Denso, Valeo, and ZF-TRW, among others. Key chip suppliers include Freescale, Infineon, ST, Texas Instruments, and ADI [24].

1.4.2 Ultrasonic sensors

Ultrasonic sensors possess a limited detection range (0.15-6m), showcasing exceptional capabilities in very close-range 3D mapping. Their precision and resolution are heightened owing to the slow propagation of sound waves [25]. Notably, these sensors perform reliably in adverse weather conditions such as snow, fog, rain, and dust, thanks to their short-range nature. Despite being compact and cost-effective, they lack suitability for gauging speed due to their restricted range and are vulnerable to influences like temperature variations, humidity, wind, and Electromagnetic Interference (EMI) [26]. As a result, sensors necessitate compensations for temperature and humidity. Ultrasonic sensors possess a limited detection range (0.15-6m), showcasing exceptional capabilities in very close-range 3D mapping. Their precision and resolution are heightened owing to the slow propagation of sound waves [27]. Notably, these sensors perform reliably in adverse weather conditions such as snow, fog, rain, and dust, thanks to their short-range nature. Despite being compact and cost-effective, they lack suitability for gauging speed due to their restricted range and are vulnerable to influences like temperature variations, humidity, wind, and Electromagnetic Interference (EMI). As a result, sensors necessitate compensations for temperature and humidity.

Primarily utilized in applications such as space detection, parallel parking, reverse parking, blind-spot detection, and kick-to-open liftgates [28, 29]. Within this domain, Bosch Gen 6 Ultrasonic Sensors emerge as leading representatives, offering three mechanically compatible sensor variants.

1.4.3 LiDAR

LiDAR functions as an integrated optical detection and measurement system operating within the optical frequency band. The current sensor performance encompasses a range of 100 to 300 meters, with an accuracy range of ±2.0 to ±5.0 centimeters. It boasts a Horizontal Field of View (FoV) spanning 120 to 360 degrees and a Vertical FoV of 25 to 120 degrees. The angular resolution is finely tuned, ranging from 0.1 to 0.4 degrees horizontally and 0.4 to 2.0 degrees vertically.

Its strengths lie in its capacity to generate high-resolution 3D images, provide accurate depth information and target discrimination, and offer a wide FoV with high angular resolution. Additionally, LiDAR operates independently of light conditions and is adept at detecting lane markers and street signs. However, it faces limitations as it cannot transmit in adverse weather conditions such as rain, snow, fog, or dust. Presently, it falls short of meeting the combined performance, size, and cost requirements for mass-produced autonomous vehicles at the industrial level.

Typical applications for LiDAR encompass mapping and navigation, obstacle detection, and in-cabin monitoring, demonstrating its versatility in enhancing various aspects of automotive technology. See Chapter 2 for a detailed discussion of LiDAR sensors.

1.4.4 Camera

The camera serves to transform an optical image into electronic signals, employing technologies such as CCD (Charge-Coupled Device) and CMOS (Complementary Metal-Oxide-Semiconductor) [30]. CCD operates by sequentially transferring each pixel's charge packet to a shared output structure for charge-to-voltage conversion. This method results in lower noise levels and generally yields higher-quality images, especially in low-light conditions. It excels in providing a high dynamic range, enhanced depth of color, higher resolution, and improved light sensitivity [31, 32].

On the other hand, CMOS conducts charge-to-voltage conversion within each pixel. It offers advantages such as lower power consumption, high integration, reduced size, ease of manufacture, and a more budget-friendly price point [33]. The quality of images has seen improvement since the widespread adoption of CMOS technology [34].

Automotive cameras are categorized into front-view cameras, surround-view cameras, rear-view cameras, and in-cabin cameras. Major players in the Tier 1 camera manufacturing sector include Bosch, Aptiv, Continental, Denso, and Valeo. Key suppliers of image sensors include Infineon, Freescale, Texas Instruments, ST, and Panasonic, among others. Additionally, significant optics suppliers encompass Sony, Panasonic, Sharp, Konica Minolta, Omron, and various other enterprises.

1.4.5 Thesis overview

The primary objective of this thesis is to develop and evaluate the potential of ADAS in enhancing the performance, safety, and overall effectiveness of tram operations within urban environments. While ADAS has gained widespread recognition in the context of conventional automobiles, its application in the tram sector remains an underexplored and promising avenue. ADAS encompasses a range of technologies designed to assist drivers in their tasks, improve situational awareness, and mitigate the risk of accidents. More specifically in this thesis, the focus is on emerging lidar technology to develop a Advanced Assisted Driving in Tramway Sector. The proposed ADAS functions as a vigilant companion to the driver, continuously monitoring the surrounding environment. Upon detecting moving objects or potential obstacles, the system triggers timely alerts to notify the driver, providing crucial information for proactive decision-making and accident prevention.

Furthermore, Chapter 1 has introduced the motivation and existing literature on ADAS implementations in other transportation modes, drawing insights and lessons that can be applied to the tram environment. By examining successful case studies and identifying potential hurdles, this thesis endeavors to provide a roadmap for the successful integration and adoption of ADAS within tram networks.

Chapter 2 presents the general overview of LiDAR technology and investigates its role in shaping the future of autonomous transportation. Through an in-depth analysis of LiDAR's technical aspects, ranging from hardware design to internal signal processing algorithms, this introduction aims to provide a holistic understanding of its functionalities and limitations.

The primary objective of Chapter 3 is to show a LiDAR object detection algorithm delving into its underlying principles, design considerations, and real-world performance. Through a meticulous analysis of the algorithm's architecture, feature extraction methods, and decision-making processes, this research aims to elucidate the key factors contributing to its success in diverse and dynamic environments. Real-world case studies and performance assessments will be presented to showcase the algorithm's efficacy.

Chapter 4 introduces a vehicle localization algorithm, delving into its architecture, sensor fusion techniques, and decision-making processes. By scrutinizing the algorithm's adaptability to diverse scenarios, ranging from urban environments with complex infrastructure to areas with limited satellite visibility, this research seeks to unravel the algorithm's capabilities and limitations. To assess the algorithm's performance under challenging conditions, real-world case studies and simulations will be presented to showcase the algorithm's effectiveness in scenarios mirroring the complexities of everyday driving conditions.

Chapter 5 summarizes the results of the proposed methodologies. Moreover, the one seeks to unravel the algorithm's capabilities and limitations.

Chapter 2

LiDAR technology

La vita appartiene a quegli individui strani ed eccezionali, che osano essere diversi.

Oscar Wilde

This Chapter introduces the principles of LiDAR with a focus on ToF and FMCW methods. As well as a description of the LiDAR OS0 used for the experiments.

2.1 Introduction

In recent years, LiDAR (Light Detection And Ranging) is playing a large role in the automotive industry, not just in atmospheric aerosols and aerial mapping. The fuel of this widespread is in finding an adequate solution to all aspects of 3D imaging for automotive.

LiDAR is a remote sense technology based on a well-known measurement technique in the optoelectronic with established publications in the literature. It works on a simple working principle based on counting the time carried out by light to be transmitted and received to perform the distance of objects. This is reported in previous literature as the time-of-flight (TOF) principle. LiDARs have become so relevant for the capability to provide 3D high-fidelity information around a vehicle makes them invaluable tools for those tasks requiring accurate detection. Generally, cameras and RADIO Detection And Ranging (RADAR) are offered as alternative solutions. Those based on the use of cameras are the oldest and require cheap hardware; on the counterpart, they are quite sensitive to environmental conditions. They use 2D data and rely on stereo-vision to obtain distance measurements by means of triangulation [35, 36, 37, 38, 39, 40, 41]. Those solutions are optimal for indoor applications without a strong solar or night background, and although the spatial resolution of the images is limited, and depth resolution is kept at the cm level. RADAR- and LiDAR-based systems represent reliable and performing solutions that can operate both at day as well as at night time. Differently from cameras, they (i) reproduce a 3D image of the scene, (ii) perform a direct measurement of the distance, (iii) speed up the estimation of the size of the objects in the field of view, and (iv) avoid privacy issues, since the information related to the persons in the scene keeps safe their anonymity, as it consists of a mere cloud of points [2].

With respect to RADARs, which use radio-frequency or microwave signals, LiDARs use laser beams in the nanometer range to measure the time of flight of the emitted pulses; laser beams offer finer resolution and can be produced by sources characterized by size and weight much smaller than that of RADAR antennas [42, 43, 44, 45].

The main challenges for RADAR- and LiDAR-based solutions consist of trading off between amount of collected data and computational burden, to achieve real-time software execution. Actually, defining and implementing reliable and effective software solutions represents a major research topic for both Academia and Industry [46, 47].

A generic architecture of the LiDAR system is shown in Figure (2.1),[48, 49]. In particular, the LiDAR structure is divided into two parts: front-end and data processing unit. The former comprises the hardware units and the related embedded system for data pre-processing, while the latter focuses on the data processing stages. More specifically LiDAR front end typically includes the following components:

- Laser Source: LiDAR systems use laser diodes or solid-state lasers to emit short pulses of laser light. These lasers are chosen for their ability to produce highly collimated light.

- Scanning Mechanism: many LiDAR systems incorporate a scanning

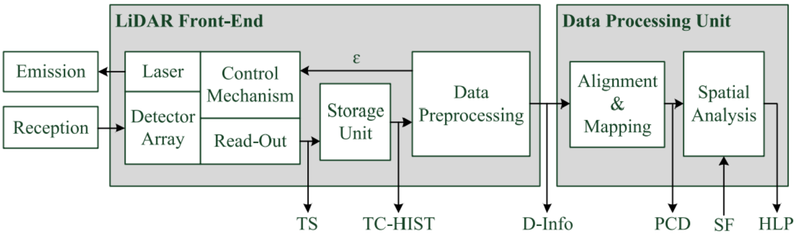


Figure 2.1. LiDAR common structure [1]

mechanism to direct the laser: flash, mechanically spinning.

- Optics: optics, including lenses and mirrors, are used to focus and shape the laser beam, ensuring that it is directed accurately and efficiently toward the target area. They also help in collecting the return signal.

- Photodetector: LiDAR system is made up of photodetectors, such as photomultiplier tubes (PMTs) or avalanche photodiodes (APDs), to detect the backscattered or reflected laser light. These detectors convert the received light into electrical signals.

- Control and Data Processing Unit: systems are controlled by a unit that coordinates laser firing, scanning, and data acquisition. This unit also processes the raw data, performing first tasks like Time-of-Flight Measurement (ToF), filtering, noise reduction, and signal processing to generate 3D point cloud data.

- GPS and IMU: many systems are equipped with Global Positioning System (GPS) receivers and Inertial Measurement Units (IMUs) to provide accurate geo-referencing and positioning data. This information is essential for creating precise 3D maps and models of the environment.

2.2 Measurement Principles

The measurement process begins by projecting a signal onto an object designated as the target. Subsequently, the reflected or backscattered signal is detected and processed to obtain the distance. This process provides a three-dimensional representation, in the form of a point cloud, related to surroundings. Hence, the range R or distance to the target could be obtained by modulating the intensity, phase, and/or frequency of the transmitted signal or measuring the time required by the signal to reach the target and to appear back at the receiver (ToF).

2.2.1 Time-of-Flight (ToF)

Time-of-flight (ToF) technology works based on the measurement of the time τ it takes for a signal to travel from a source to a target and back again. Since the speed of light c is a given constant, the range R to the target is directly proportional to the traveled time. The measured time is representative of twice the distance R to the object, as light travels to the target forth and back, and, therefore:

$$R = \frac{c\tau}{2} \tag{2.1}$$

The measurement process begins within the LiDAR front end, where an active laser source is utilized to periodically emit laser pulses. The time interval between consecutive laser pulses is denoted as a single measurement cycle. The detector array is equipped with multiple Single Photon Avalanche Diodes (SPADs). These SPADs are specialized photon diodes biased beyond their breakdown voltage, enabling a single photon to initiate the avalanche effect, ultimately leading to diode breakdown. After being triggered by a photon, a SPAD enters a temporary insensitivity phase, incapable of detecting subsequent photons for a specified duration, known as the "dead time" [50]. In practical application, a quenching circuit is used to reset the SPAD to its initial state. Signals ϵ from the embedded system are transmitted to control the laser's operation, manage the quenching circuit, and facilitate the readout process. Afterward, the first output from the LiDAR system is ToF, which is converted to a digital timestamp (TS) using a time-to-digital converter; as shown in Figure (2.1).

The resolution in the range is directly determined by the digit number of the timestamp. Ideally, the maximum measured range value is restricted only by the maximum time interval that can be measured by the time counter. In practice, both laser energy losses during the trajectory especially for diffuse targets, and electronic noise of the detection circuit, lead to a signal-to-noise ratio (SNR) a limiting factor for the effective range in pulsed LiDARs. Another aspect to be considered is SPAD's failure in distinguishing between laser photons from ambient ones. Therefore, time-correlated single photon counting (TCSPC) is typically used to solve this issue [51]. It combines multiple consecutive TSs into a time-correlated histogram (TC-HIST). The time during which TC-HISTs from all SPADs are collected and completed is called a frame. In the TC-HIST, the y-axis signifies the count number, while the x-axis denotes time. The bin width on the x-axis can be equal to or less than the TS resolution. The number of collected TS in a frame increases, and the distance information and noise floor become more intuitive. A typical TC-HIST encompasses noise features, and distance information (D-Info); the ambient photon count follows a uniform distribution. Upon completing one frame, TC-HISTs are processed by specific algorithms on embedded systems, such as peak detection, digital filters, and maximum likelihood estimation, to extract D-Info (R). More details on specific algorithms for extracting D-Info are reported in [51, 52, 1, 53, 54].

2.2.2 Frequency Modulation Continuous Wave (FMCW)

Frequency Modulated Continuous Wave (FMCW) is a sophisticated and versatile technology widely used, also, in radar systems. FMCW employs continuous waveform with modulating frequencies, allowing it to provide valuable information about target range and even velocity. The transmitted signal in the FMCW technique is based on the "chirp," a sinusoidal signal whose frequency is continuously varied over time. The signal is characterized by a bandwidth B, an initial frequency f_c, and a duration T_c. An easy way to represent the chirp graphically is as shown in Figure (2.2), where the slope in the frequency-time graph measures the speed of frequency of the chirp signal increases during time.

The synthesizer generates the chirp, defined in the time $[0 - T_c]$, to be transmitted to a target with the frequency that depends on the time

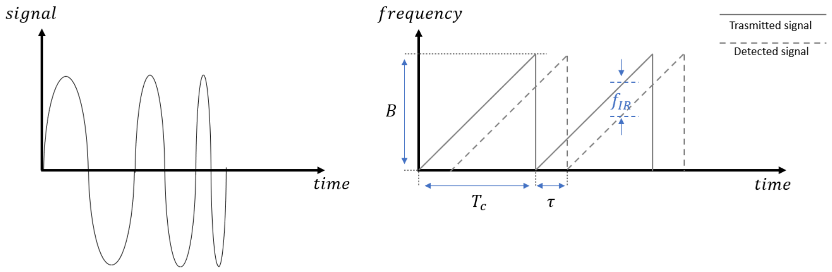


Figure 2.2. Sawtooth frequency modulation in FMCW method and beat frequency for static target.

following :

$$f_{TX}(t) = f_c + slope\ t \tag{2.2}$$

The chirp detected by the receiving antenna experiences a delay compared to the transmitted chirp, defined as the $'round\ trip\ time' \ = 2R/c$, where R is the distance from the target, and c is the speed of light. For a static target, the delay between transmitted and reflected chip/signal provides a constant frequency difference f_{IF}. In Figure (2.2)b, it is evident that the received signal is a delayed version of the transmitted signal by a time. Considering the frequency varies under linear law, f_{IF} is directly proportional to distance R of the target, according to:

$$f_{IF} = slope\ \tau = \frac{B}{T_c \tau} = \frac{B}{T_c}\frac{2R}{c} \Rightarrow R = f_{IF}\frac{cT_c}{2B} \tag{2.3}$$

where B is the bandwidth, and T_c denotes the period of the ramp. The Equation (2.3) is central to FMCW processing: the difference in frequency between the transmitted and the received signals is constant proportional to the range of the scatterer that reflected echo. When multiple objects are present, the received signal consists of various sinusoidal tones which could be separated from each other by using FFT transform.

The resolution of a LiDAR can be determined by the frequency resolution. The frequency resolution is given by the duration of the signal used to form the spectrum of the chirp signal:

$$\Delta f = 1/T_c \tag{2.4}$$

So that, the range bin width is found using the incremental form of the fundamental eq. (2.3),

$$\Delta R = \frac{c}{2B} \tag{2.5}$$

c is the speed of light in free space. To enhance spatial resolution, it is necessary to increase the duration of the observation window T_c of the chirp until the sinusoidal tones are adequately separated in frequency. However, this results in an increase in bandwidth. It is observed that the resolution range depends solely on the bandwidth swept by the chirp: the wider the bandwidth B, the greater the range resolution, and the smaller the distance between targets that the radar can discriminate.

The FMCW technique offers a noteworthy benefit by not just assessing the distance R but also determining the radial velocity v_r of the target, including its direction by means of a single echo signal.

This capability to measure velocity is a distinctive and valuable feature of FMCW, making it particularly suitable for applications such as Doppler LiDAR, traffic monitoring, and speed measurement system. If the target is moving at a certain range R, the Doppler effect occurs. So, the beat frequency will be under the effect of frequencies due to time delay τ and due to the Doppler effect f_d. The 'Sawtooth linear' frequency modulation is not suitable for separating the two frequencies. In fact, the Doppler frequency will occur only as a measurement error in the range value. With a triangular frequency shift, a target range can computed on both the rising and falling edges. An echo signal is shifted to the right in Figure (2.2)b relative to the transmitted signal, depending on the running time. Without a Doppler frequency, the frequency difference on the rising edge is equal to the measurement on the falling edge. A Doppler frequency shifts the entire echo signal in frequency upward (movement toward the sensor) or downward (movement away from the sensor). Consequently, beat frequency elements are overlaid onto:

$$f^+ = f_{IF} + f_d \, and \, f^- = f_{IF} - f_d \tag{2.6}$$

In this case, range and radial velocity can be obtained as following:

$$R = \frac{cT_c}{4B}(f^+ + f^-) \tag{2.7}$$

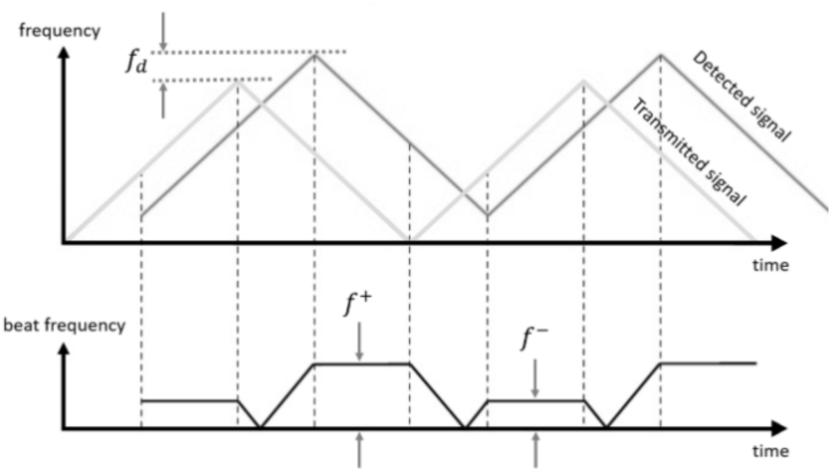


Figure 2.3. Triangular frequency modulation in FMCW method and beat frequency for moving target [2].

$$v_r = \frac{\lambda}{2} f_d = \frac{\lambda}{4}(f^+ - f^-) \tag{2.8}$$

where λ is the laser wavelength. The main benefit of autonomous vehicle applications is their ability to sense simultaneously the speed value, its direction, and the range. However, its detection scheme poses potential problems related to practical issues like the accuracy of the modulation electronics, or linearity of the intensity-voltage curve of the laser, which require advanced signal processing. Although the FMCW method shows those advantages, it is not the principal approach in lidar imaging systems for autonomous vehicles. Some teams are currently implementing lidar solutions based on ToF technology in commercial systems due to its differential advantages in implementation.

2.3 LiDAR equation

The echo signal detected by receivers is analyzed in this paragraph. The power of a received signal from a distance R, can be written as:

$$P(R) = KG(R)\beta(R)T(R). \tag{2.9}$$

The first factor K denotes the performance of the LiDAR, and the second $G(R)$ describes the range-dependent geometry. These two factors are controlled during the setup phase. The last two factors contain information to be extracted from the beam. The $\beta(R)$ is the backscatter coefficient at distance R while $T(R)$ denotes the transmission term describing the amount of radiation that is lost on the way from the LiDAR to the target and back.

More specifically, the system factor can be written as:

$$K = P_0 \frac{c\tau}{2} A\eta \tag{2.10}$$

P_0 is the average power of a single laser pulse and τ is temporal pulse length. The $P_0\tau$ product represents the energy of the impulse E_0, while the product $c\tau$ is the length of a volume illuminated by the laser pulse at a fixed time. The factor $1/2$ is justified by the fact that when a laser signal is detected at an instant t after the initial pulse edge is emitted, the same part of the pulse backscattered arrives at the receiver from the distance $R_1 = \frac{ct}{2}$. Concurrently, light generated by the trailing edge arrives from a distance $R_2 = \frac{c(t-\tau)}{2}$. Therefore, $\Delta R = R_1 - R_2 = \frac{c\tau}{2}$ represents the length of the volume from which backscattered light is received at an instantaneous moment. This length is termed the "effective (spatial) pulse length".

The factor A denotes the area of the primary receiver optics which is responsible for collecting the backscattered light, and η is the overall system efficiency, as in Figure (2.4). The energy E_0, η, and area A, are the parameters of a LiDAR system that can be optimized to obtain the best possible signal.

The geometrical factor, $G(R)$:

$$G(R) = \frac{O(R)}{R^2} \tag{2.11}$$

Incorporates the laser-beam receiver-field-of-view overlap function $O(R)$ and the term R^{-2}. R^{-2} dependence comes out from the assumption the area of the receiver telescope constitutes a portion of a sphere's surface with a radius R. As a consequence, the signal's intensity decreases drastically with distance.

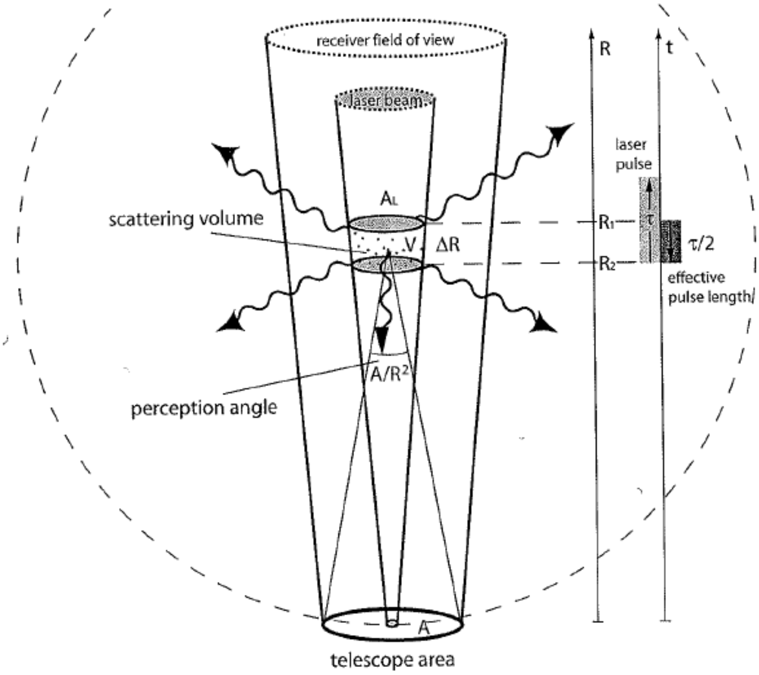


Figure 2.4. Illustration of the LiDAR geometry [3].

In concluding the LiDAR equation, it is essential to account for the portion of light that becomes attenuated during its path from the lidar to the scattering volume and back. The transmission term $T(R)$ is a variable ranging between 0 and 1, and its value is determined by:

$$T(R) = exp\left[-2\int_0^R \alpha(r,\lambda)\,dr\right] \tag{2.12}$$

This term arises from the particular expression of the Lambert–Beer–Bouguer law applicable to lidar. The integral takes into account the trajectory from the lidar to the distance R and back. The cumulative effect of all transmission losses is referred to as light extinction, and $\alpha(r,\lambda)$ represents the extinction coefficient. It is defined similarly to the backscatter coefficient, being the product of the number concentration and the extinction cross-section $\sigma_{j,ext}$ for each type of scatterer, denoted by j.

$$\alpha(r,\lambda) = \sum_{j} N_j(R)\sigma_{j,ext}(\lambda) \tag{2.13}$$

with unit in $1/m$. In summarizing the examination of individual terms, the lidar equation can be expressed in a more conventional format:

$$P(R,\lambda) = P_0 \frac{c\tau}{2} A\eta \frac{O(R)}{R^2} \beta(R;\lambda) exp\left[-2\int_0^R \alpha(r,\lambda)\,dr\right] \tag{2.14}$$

More details on the eq.(2.9) are available in [3]. It's important to note that the signal detected will invariably encompass a background component, in addition to the lidar signal described earlier. During daylight hours, the background signal is predominantly influenced by direct or scattered sunlight, while during nighttime, contributions from the moon, stars, and artificial light sources become prominent. Detector noise serves as another source of unwanted signals.

2.4 Imaging Strategies

After introducing the two primary measurement approaches employed in LiDAR imaging systems, it is noteworthy that each of them has been introduced as point-wise measurements. Nevertheless, the LiDAR images of interest consistently form 3D point clouds, providing precise depictions of expansive fields of view, extending up to 360 around the object of interest. Several strategies have been suggested for constructing LiDAR images based on the repetitive nature of point measurements. Yet, these approaches can be fundamentally categorized into two distinct families: mechanical scanners and detector arrays.

2.4.1 Mechanical scanners

LiDAR based on mechanical scanners strategies, incorporates high-quality optics and a rotating or galvanometric assembly typically featuring mirrors or prisms attached to mechanical actuators. Units with both

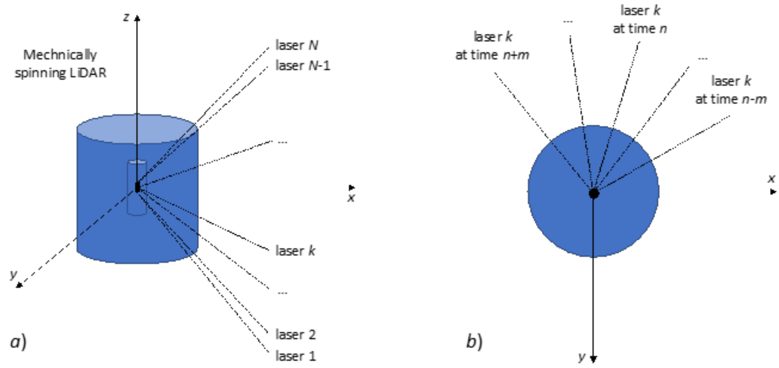


Figure 2.5. Side-view (a) and top-view (b) of the schematic of a mechanically spinning LIDAR. Each transmitter radiates at a different angle, such that the radiation pattern of the array forms a wide-aperture fan with equally distributed rays. The sensor scans the surroundings by rotating the linear array around its axis to gain a 360° horizontal field of view.

sources and detectors collectively rotate around a single axis.
In a 2D mechanical scanner lidar setup, the scanning motion of a single beam occurs along a single axis, allowing the system to capture information in a horizontal direction. This approach is commonly used for applications where a panoramic view in the horizontal plane is sufficient. However, they may be limited in capturing vertical information, which is where 3D lidar systems with additional scanning capabilities become valuable. In the latter scenario, multiple detectors may be positioned vertically along the spinning axis, enabling to capture of information in both horizontal and vertical planes, as depicted in the Figure (2.5). With this setup, the sensor achieves a 360-degree field of view (FOV) and effectively covers the entire surroundings of the vehicle. The angular resolution for a mechanical scanner LiDAR, is the second most important specification after the range resolution. The angular resolution of a LiDAR system signifies the minimum angular or linear separation between two points that the sensor can distinguish. A higher angular resolution means that the lidar system can distinguish between closely spaced angular positions, providing more

detailed information about the surroundings. On the other hand, lower angular resolution may result in a coarser representation of the scanned environment. Consequently, sensors with superior resolution can generate more densely packed point clouds. The angular resolution of a 3D mechanical scanner lidar refers to the angular resolution, both horizontal and vertical. To obtain accurate and current information on the angular resolution of a specific mechanical scanner LiDAR, one should refer to the technical documentation or specifications provided by the manufacturer. They usually provide detailed information about the LiDAR system's capabilities, including its angular resolution in both the horizontal and vertical dimensions. The angular resolution is often specified in terms of degrees or radians per unit of rotation. Sometimes, the horizontal angular resolution is expressed by using the total number of steps:

$$H_{res} = \frac{FOV_H}{d\theta} \tag{2.15}$$

where $d\theta$ is the azimuthal/horizontal angular resolution in degrees.
Despite the widespread use of these scanners, this configuration comes with several drawbacks in a consumer unit. Specifically, concerns arise regarding the reliability and maintenance of the mechanisms, the mass and inertia of the scanning unit imposing limitations on scanning speed, the inflexibility of scanning patterns, and susceptibility to misalignment due to shock and vibration.

2.4.2 Flash imagers

Considering the constraints associated with scanning lidar methods involving moving elements, alternative imaging techniques have been suggested to mitigate these effects. Optical elements for transmission illuminate an entire scene, while a linear array (or matrix) of detectors simultaneously captures signals from distinct angular subsections. This setup enables the acquisition of range data for the target in a single shot (see Figure 2.6), facilitating the seamless handling of real-time applications. The illumination can take the form of pulsed signals in flash imagers or continuous signals in FMCW lidars. A highly effective architecture for lidar imaging systems in autonomous vehicles is flash lidar, which has advanced to the point of nearing commercial deployment in short and medium-range

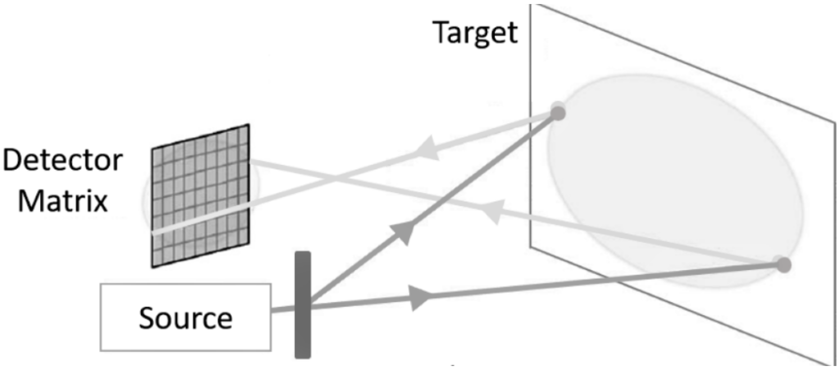


Figure 2.6. Structure of flash LiDAR [3].

applications. In flash lidar, imaging is achieved by illuminating an entire target scene or a specific portion of it with pulsed light. The backscattered light is collected by a receiver distributed across multiple detectors, as illustrated in Figure (2.6). Each detector records the distance using the conventional time-of-flight principle and in some cases, the reflected intensity. Consequently, both the optical power projected onto a 2-D array of detectors and the 3D point cloud are directly derived from a single laser pulse directed at the target [55, 56, 57]. The laser's beam divergence is typically chosen to optically align with the receiver's field of view (FOV), ensuring the simultaneous illumination of all pixels in the array. Each detector within the array is independently activated upon the return of a pulse, measuring both the intensity and range. Consequently, the spatial resolution is, determined by the density at which detectors are packed, often constrained by the patterning capabilities of technology. The illuminating region and the backscattering at the target markedly diminishes the optical power accessible, necessitating exceedingly high peak illumination power and extremely responsive detectors when compared to single-pixel scanners. Positively, flash lidars show advantages in capturing the entire scene in a single shot, enabling a rapid data capture rate. Consequently, this method proves highly robust against vibration effects and movement artifacts that might otherwise distort the image.

2.5 OUSTER OS0-128

The LiDAR employed in this thesis on the assisted driving system, is the OS0-128 model produced by OUSTER company.
The OS0 presents an unmatched blend of affordability, performance, reliability, compactness, weight, and power efficiency. Engineered for indoor and outdoor use in all weather conditions, it ensures an extended operating lifetime.
Within the OS0 sensor family, three distinct models OS0-128, OS0-64, and OS0-32—share identical mechanical dimensions while varying in vertical resolution. Throughout this section, the term "OS0" collectively denotes the model OS0-128, with specific model designations mentioned only when there are performance distinctions.

2.5.1 Mechanical structure

OS-0 falls into the category of Rotating Compact LiDAR. It provides a broad vertical field of view (FoV) of 90 degrees and a complete 360 degrees horizontal FoV.
The optical module of this particular LiDAR comprises sets of 128 transmitters (TX) and receivers (TX) arranged in an array with vertical configuration; see 1a set in Figure (2.7). Positioned on a revolving turntable (refer to 2 set in Figure 2.7), the optical module moves horizontally along the sensor's axis, enabling a comprehensive 360 degrees scanning of the scene. More specifically, the light-ranging optical module is composed of the light-ranging device and a user interface. The light-ranging device includes a ranging system controller, a light sensing module (Rx), and a light transmission module (Tx). The light sensing module incorporates a sensor array comprising individual single-photon avalanche diodes (SPADs). Each SPAD in the sensor array corresponds to a specific emitter in the emitter array, a result of the geometric arrangement of the light sensing module and the transmission module. Ranging data are generated by the light-ranging device by emitting one or more pulses from the light transmission module to objects within the Field of View (FoV). The light receiver module (Rx) detects the back-scattered energy from the objects after a certain delay time. The ranging system controller manages both the light sensing module and the light transmission module by issuing configuration

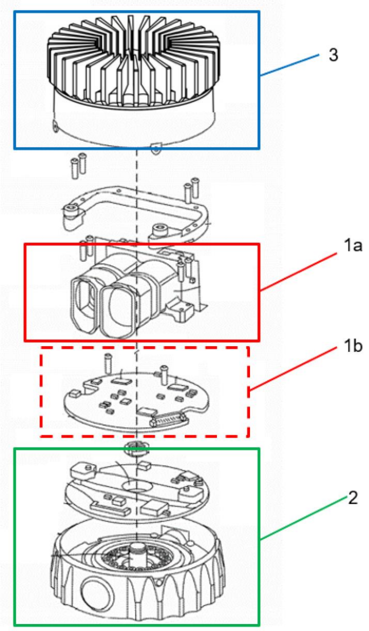


Figure 2.7. Structure of LiDAR OS0-128 from Ouster website: 1a) Light Ranging Optical Module 1b) Ranging device control board 2) Rotary actuator 3) Finned enclosure.

commands and start and stop commands.
The processing unit can employ matched filters and peak detection processing for the identification of return signals over time. Moreover, the processor can execute specific signal processing techniques, including multi-profile matched filtering, to aid in the retrieval of photon time series that are less prone to pulse shape distortion caused by SPAD saturation and quenching.

2.5.2 An Unusual Operating Wavelength

One aspect of Ouster LiDAR is their 850 *nm* operating wavelength. As a first impression, operating with wavelengths in regions of low solar flux to ease system design would be a better choice. Some lidar brands often choose operating wavelengths in regions of low solar flux to ease system design. Ouster brand operates at 850 *nm*, choice runs counter to this trend. A plot of solar photon flux versus wavelength at ground level (the amount of sunlight hitting the earth versus wavelength) shows that at 850*nm* there is almost 2x more sunlight than at 905 nm, up to 10x more sunlight than at 940*nm*, and up to 3x more sunlight than 1550*nm* – all operating wavelengths of legacy lidar systems. Several benefits of the 850*nm* wavelength exist.
Better performance in humidity can be reached in humid and foggy conditions at ground level when the laser pulses transmit through the air resulting in less laser light captured by the lidar receiver. By contrast, the 850 nm spectrum has lower atmospheric water vapor absorption in all conditions – as much as several orders of magnitude better than other popular lidar operating wavelengths like 1550 nm. This means ouster lasers are not absorbed in humid conditions and achieve more consistent operation overall. Water vapor absorption issues shouldn't be confused with liquid water in the atmosphere like rain, or dense fog which are challenging but surmountable conditions for all lidar sensors.
Moreover, an improved sensitivity in CMOS could be shown; silicon CMOS detectors are far more sensitive at 850 nm than at longer wavelengths. There is as much as a 2x reduction in sensitivity just between 850 and 905*nm*. Designing Ouster's system at 850 nm allows to detect more of the laser light reflected back toward our sensor which equates to a longer range and higher resolution.

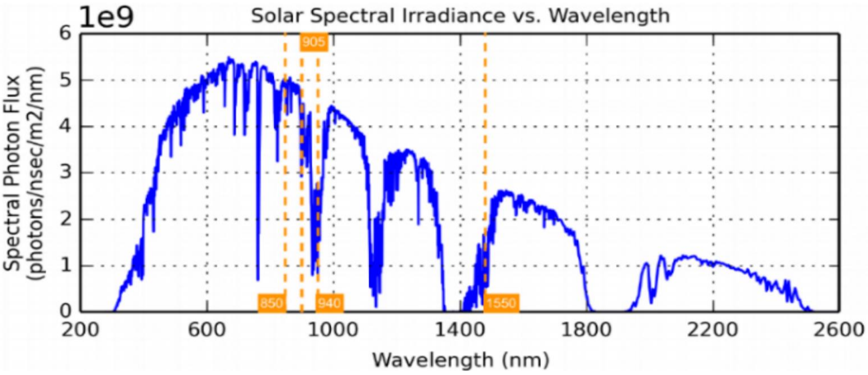


Figure 2.8. Solar Spectral Irradiance and working wavelength of LiDAR OS0-128 from Ouster website.

2.5.3 Raw Cloud-point processing

The subsequent section aims to elucidate the initial steps involved in obtaining the ultimate data for processing. The primary step involves a detailed examination of the raw data output generated by the LiDAR.The raw data output from the LiDAR consists of individual lasers that collectively form the LiDAR dataset. The transformation process of this data from the coordinate frame associated with each laser beam to the overall local Coordinate Frame of the LiDAR system is delineated.

The LiDAR Coordinate Frame adheres to the right-hand rule convention. As depicted in Figure (2.9), the LiDAR Coordinate Frame is denoted by X_L, Y_L, and Z_L. The orientation of the axes is defined with the positive x-axis aligned with the 0^o encoder angle and the red external connector. The positive y-axis is directed towards the 90^o encoder angle, and the positive z-axis points upwards, indicating the top of the sensor.
The output data packet comprises crucial information from each laser, necessitating transformation into 3D Cartesian coordinates within the Lidar Coordinate Frame. The structure of the data is contingent on the selected output mode.

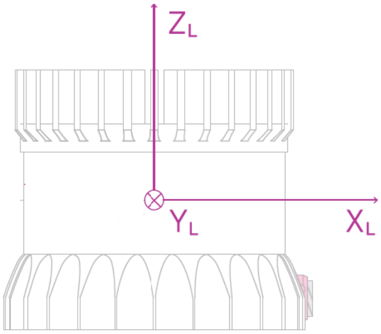


Figure 2.9. Side view of Lidar Coordinate Frame; image from the ouster brand site.

Precisely, the package can be segmented into two primary blocks: information pertinent to reconstructing a point in the Lidar Coordinate Frame and associated features.

For the reconstruction of the scene at points defined by Cartesian coordinates, the essential data includes: *r' [mm]* magnitude of distance vector from laser beam origin of the i-th channel to detected point, this value is not provided, it is just to illustrate the concept ; n [mm] - magnitude of distance vector from lidar origin to laser beam origin; *r (range)* [mm] - magnitude of vector sum $r' + n$ of the i-th channel; *beam-altitude-angles* [degree] - angle in xy plane with respect to ideal scan line of i-th channel; *beam-azimuth-angles* [degree] - angle in the plane defined by the z coordinate and the xy plane with respect to ideal scan line of i-th channel; *Encoder Count* - an azimuth angle as a raw encoder count, starting from 0 with a max value of 90,111 - incrementing 44 ticks every azimuth angle in 2048 mode, 88 ticks in 1024 mode, and 176 ticks in 512 mode. Figure (2.10) provides a graphical representation of the illustrated concepts.

Defined the angles:

$$\theta_{encoder} = 2\pi \left(1 - \frac{Encoder - Count}{90112}\right) \tag{2.16}$$

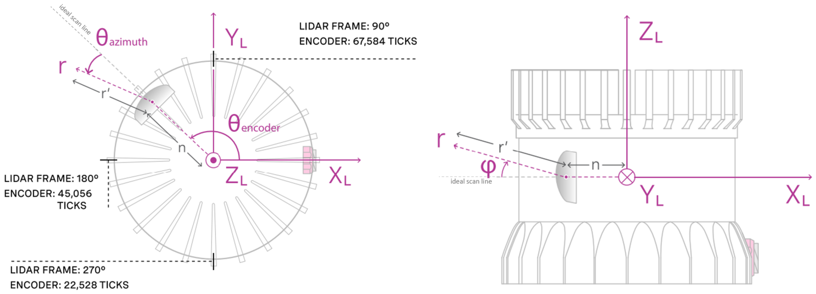


Figure 2.10. Graphical representation of the azimuth and elevation angles of a laser beam; basic image from the Ouster brand site. (a) Top view of Lidar Coordinate Frame; (b) Side view of Lidar Coordinate Frame.

$$\theta_{azimuth} = 2\pi \left(-\frac{beam - azimuth - angles}{360} \right) \tag{2.17}$$

$$\phi = 2\pi \frac{beam - altitude - angles}{360} \tag{2.18}$$

The corresponding x, y, z values for each point, are computed as following:

$$x = (r - n) * cos(\theta_{encoder} + \theta_{azimuth}) * n * cos(\phi) + n * cos(\theta_{encoder}) \tag{2.19}$$

$$y = (r - n) * sin(\theta_{encoder} + \theta_{azimuth}) * n * cos(\phi) + n * sin(\theta_{encoder}) \tag{2.20}$$

$$z = (r - n) * sin(\phi) \tag{2.21}$$

The features associated with each detected point, are the following:

- Timestamp - timestamp of the measurement in nanoseconds.
- Measurement ID - a sequentially incrementing azimuth measurement counting up from 0 to 511, or 0 to 1023, or 0 to 2047 depending on lidar mode.
- Frame ID - index of the lidar scan. Increments every time the sensor completes a rotation, crossing the zero point of the encoder.
- Signal Photons - signal intensity photons in the signal return measurement.

- Reflectivity
- Ambient Photons - photons from ambient light.

Each frame consists of several points (N) equal to the product value of the horizontal resolution (N_a) and the number of channels (N_c).
Each data packet of a frame contains N_c values of azimuth angles and elevation angles associated with each channel and N_a encoder angles. Range values are stored with blocks with length (N_c), and each block was acquired with an azimuth value $j = 1, ..., N_a$. The number of total blocks is equal to the horizontal resolution values. Figure (2.11) shows the Memory organization of the range value using a block diagram. The N points x, y, z are obtained by applying the equations (2.19-2.21) on each individual channel for a current encoder angle.

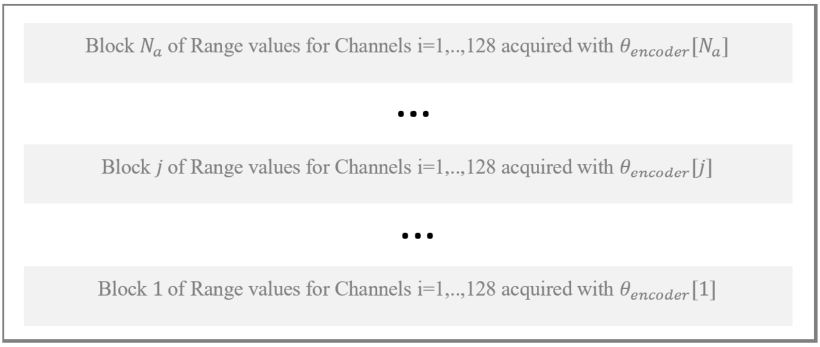

Block N_a of Range values for Channels i=1,..,128 acquired with $\theta_{encoder}[N_a]$
...
Block j of Range values for Channels i=1,..,128 acquired with $\theta_{encoder}[j]$
...
Block 1 of Range values for Channels i=1,..,128 acquired with $\theta_{encoder}[1]$

Figure 2.11. Memory organization of the range values of a single frame in a data packet for a lidar with N_c=128.

2.5.4 LIDAR specifications

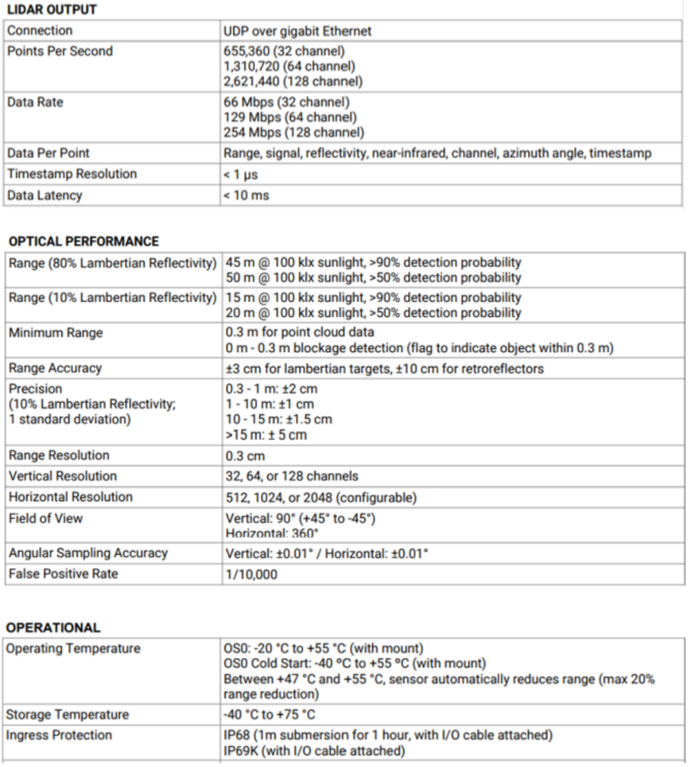
As seen before each measure, output from a channel at a certain time (timestamp), contains information about Range, Signal Photons, Reflectivity, and Ambient Photons.

The data rate depends on the number of channels: for an OS-0 128 channels, the data rate is 254 Mbps. Data rate refers to the speed at which data is transferred or processed over a communication channel or a computing system. It is typically measured in bits per second (bps), kilobits per second (kbps), megabits per second (Mbps), gigabits per second (Gbps), or higher multiples.

The OS0 128 offers a maximum range depending on the properties of the object's surface and on the sunlight. A target with property as 80% Lambertian reflectivity, and 100 klx sunlight can reach a maximum range of 45 m. Lambertian reflectivity refers to the way a surface reflects light in a diffuse and isotropic manner. This is an idealized property that is often used in various fields, such as computer graphics, computer vision, and remote sensing, for simplicity in modeling and calculations. When you see "80% Lambertian reflectivity," it means that the surface reflects only 80% of the incident light in a Lambertian manner, while the rest may be absorbed or scattered differently. In this case, reflectivity factor ρ would be 0.8 or 80%. The minimum range for a Lambertian target to 10% is 0.3 m while for distances within 0.3 m, OS0 can fail to indicate objects. Minimum range refers to the shortest distance at which the LiDAR sensor is capable of making accurate distance measurements. The range accuracy is on the order of 3 cm. Range accuracy is typically expressed as a margin of error or a percentage of the measured distance. OS0 128 shows a storage temperature from -40°C to 75°C

The "storage temperature" of a LiDAR system refers to the recommended temperature range within which the device should be stored when not in use. This specification is important because electronic and optical components within the LiDAR system may have temperature-dependent characteristics, and storing the device outside the recommended temperature range could lead to performance issues or damage. The storage temperature is distinct from the operating temperature, which is the range of temperatures within which the LiDAR system should be operated during regular use. The storage temperature is relevant when the LiDAR is not actively collecting data or in transit, such as during shipping or when the device is in storage between uses. The performance varies with operating temperature; for a temperature between +47°C and +55°C, the sensor automatically reduces the range in Figure (2.12)., details are shown.

The parameters illustrated above and in the previous sections, are shown in tabular form in Figure (2.12).



LIDAR OUTPUT

Connection	UDP over gigabit Ethernet
Points Per Second	655,360 (32 channel) 1,310,720 (64 channel) 2,621,440 (128 channel)
Data Rate	66 Mbps (32 channel) 129 Mbps (64 channel) 254 Mbps (128 channel)
Data Per Point	Range, signal, reflectivity, near-infrared, channel, azimuth angle, timestamp
Timestamp Resolution	< 1 μs
Data Latency	< 10 ms

OPTICAL PERFORMANCE

Range (80% Lambertian Reflectivity)	45 m @ 100 klx sunlight, >90% detection probability 50 m @ 100 klx sunlight, >50% detection probability
Range (10% Lambertian Reflectivity)	15 m @ 100 klx sunlight, >90% detection probability 20 m @ 100 klx sunlight, >50% detection probability
Minimum Range	0.3 m for point cloud data 0 m - 0.3 m blockage detection (flag to indicate object within 0.3 m)
Range Accuracy	±3 cm for lambertian targets, ±10 cm for retroreflectors
Precision (10% Lambertian Reflectivity; 1 standard deviation)	0.3 - 1 m: ±2 cm 1 - 10 m: ±1 cm 10 - 15 m: ±1.5 cm >15 m: ± 5 cm
Range Resolution	0.3 cm
Vertical Resolution	32, 64, or 128 channels
Horizontal Resolution	512, 1024, or 2048 (configurable)
Field of View	Vertical: 90° (+45° to -45°) Horizontal: 360°
Angular Sampling Accuracy	Vertical: ±0.01° / Horizontal: ±0.01°
False Positive Rate	1/10,000

OPERATIONAL

Operating Temperature	OS0: -20 °C to +55 °C (with mount) OS0 Cold Start: -40 °C to +55 °C (with mount) Between +47 °C and +55 °C, sensor automatically reduces range (max 20% range reduction)
Storage Temperature	-40 °C to +75 °C
Ingress Protection	IP68 (1m submersion for 1 hour, with I/O cable attached) IP69K (with I/O cable attached)

Figure 2.12. Datasheet OS0 128, from Ouster website

Chapter 3

LiDAR Object Detection

3.1 Introduction

Autonomous driving technology involves the utilization of sensors to gather real-time information about the vehicle's surroundings, allowing it to navigate the road safely and efficiently without human intervention [58]. This technology significantly alleviates the driver's burden, enhances energy efficiency, and reduces traffic accidents.

The Society of Automotive Engineers (SAE) has established the J3016 standard, categorizing driving automation into six levels, ranging from no automation (level 0) to full automation (level 5). While the sensing range and accuracy of autonomous vehicles of sensor technology on board vehicles are pivotal; achieving continuous situational awareness, especially beyond level 3, poses challenges. Autonomous vehicles heavily rely on onboard sensors like LiDAR, cameras, and radar [59]. However, factors such as sensor limitations, obstacles, lighting conditions, and adverse weather affect the vehicle's perception range, leading to blind spots and potential failures to detect imminent dangers promptly. Past incidents, like the misidentification of a tractor's white side as the sky, underline the critical need to address these limitations [60].

To overcome the deficiencies in perception and data processing beyond level 3, a holistic approach incorporating advanced sensing technology, edge computing, communication, and other technologies is essential. This synergy aims to establish an autonomous driving cooperative perception system within the Internet of Vehicles (IoV) environment. The goal is to enhance perception accuracy, extend the perception range, and reduce processing delays, ensuring a safer and more efficient autonomous driving experience. Moreover, this will furnish autonomous vehicles with more precise and comprehensive real-time environmental data, establishing a solid groundwork for achieving autonomous driving capabilities beyond level 3 [61].

Concurrently, collaborative perception within the Internet of Vehicles (IoV) environment holds the potential to decrease the reliance on onboard sensors, thereby reducing the overall cost of autonomous vehicles and expediting the commercial adoption of advanced autonomous driving functionalities. The two primary implementation approaches for cooperative perception are vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I). Vehicle-to-Infrastructure (V2I) is a communication model that allows vehicles to share information with the sensors which monitor an infrastructure system. More specifically, Intelligent infrastructure systems at base of V2I and advanced automotive applications need reliable and efficient monitoring solutions, that should be able to perform environment perception as: detecting vehicles, pedestrians, or suspicious obstacles by a section of interest, measuring the the speed of a vehicle in a lane, checking vehicles allowed to restricted zones, estimating travel times, traffic densities, etc [62]. V2I communication is typically two-way; the data can be transmitted from infrastructure sensors to the vehicle via a network and vice versa. In an intelligent transport system (ITS), V2I sensors can acquire infrastructure data and provide drivers with real-time alerts on road conditions, traffic congestion, accidents, and construction zones [63, 64, 65]. Similarly, traffic monitoring systems can use the infrastructure and vehicle information to set variable speed limits and adapt the Phasing and Timing of traffic Signals (SPaT) to increase fuel economy and traffic flow [66]. By receiving data about the current status of traffic signals (Signal Phase and Timing - SPaT information), autonomous vehicles can optimize their speed and trajectory for safe and efficient navigation, especially at intersections. This reduces the likelihood of collisions and improves traffic flow. V2I allows autonomous vehicles to receive notifications about ongoing construction activities, road maintenance, and temporary changes in road conditions. This information helps the vehicle adapt its behavior to navigate through construction zones safely.

This chapter aims at providing a wide audience with tutorial information about the use of LiDAR sensors in monitoring applications, the authors illustrate the main conceptual steps of an approach for detecting and tracking moving objects in the observed scene. The discussion is also complemented with the presentation of a short case study, which is useful to assess the performance of the proposed approach.

The contribution shows as a minor finding that even in the presence of high-throughput data, like those returned by top-end LiDARs, some tasks, namely the recognition and tracking of the objects in the scene, can be proficiently executed by means of the hardware resources offered by an up-to-date personal computer.

3.2 Related work

Environment perception refers to the ability to perceive and understand the surrounding environment. In the context of advanced technologies, such as autonomous driving, environment perception refers to the ability of a system or device to accurately perceive and interpret elements of the surrounding environment.

For example, an autonomous vehicle must be able to detect and understand the presence of other vehicles, pedestrians, road signs, obstacles, and other environmental variables to make appropriate and safe decisions during driving. This environmental perception is often based on a combination of sensors such as cameras, lidar, radar, and other devices that provide data to the artificial intelligence system responsible for autonomous driving.

Throughout this section, the term "environmental perception" generally denotes environmental perception applicable to both autonomous driving and technology V2I.

Object detection with 3D bounding boxes poses a fundamental challenge in establishing environmental perception for autonomous systems such as autonomous vehicles, robots, and unmanned aerial vehicles. Autonomous systems rely on the perception of objects in their surroundings through a variety of sensors, including cameras and LIDAR, to facilitate navigation and obstacle avoidance. Over the recent years, notable advancements have been made in 2D object detection within computer vision, driven significantly by the emergence of convolutional neural network (CNN) technology. Despite these strides, 3D object detection remains a persistent challenge, particularly in scenarios where multiple and diverse sensors are employed to acquire more comprehensive and resilient information [67, 68, 69, 70, 71, 72].

Lately, numerous researchers have concentrated on point-cloud-based techniques for 3D object detection, driven by the inherent advantages offered by this data type: accurate depth information and densely packed geometric shape features [73, 74, 75, 76].

Nevertheless, a conventional point cloud falls short in providing detailed texture information for an object, a quality that proves highly advantageous in capturing distinctive features. Conversely, images offer more details on color and texture, without the inherent depth and scale information unless employing intricate and computationally demanding algorithms (e.g., stereography). For instance, objects such as cars or pedestrians identified at extended distances yield sparse points, presenting a significant challenge in the sole reliance on point cloud data for their classification or localization. Simultaneously, within the image domain, the texture and color characteristics of objects remain discernible even at greater distances, owing to the heightened spatial density of images and are likely to be effectively captured by established and sophisticated 2D CNNs technology.

Numerous approaches have developed into detecting objects with 3D bounding boxes using point cloud data, leveraging its precision in conveying geometric details. These methods can be broadly categorized into three main groups: projection-based methods, volumetric-based methods, and PointNets-based methods.

Projection-based methods convert point clouds into 2D representation. Several studies [77, 78, 79, 80, 81] employ 2D CNNs directly on the Bird's Eye View (BEV), which is a top-down 2D representation. This approach aims to predict the 3D bounding box and orientation of an object. In a similar vein, FVNet [82] projects raw point clouds onto front view planes, to estimate the parameters of the bounding box, including object location, size, and orientation. This method allows for adapting 2D detection algorithms but limiting depth information compared to true 3D approaches.

Volumetric Grids represent the 3D space as a volumetric grid, where each grid cell contains information about the presence of objects. Lidar points are discretized into this grid, allowing for efficient processing. But a major challenge is to require careful handling of grid resolution for efficiency and accuracy [83, 84, 85].
PointNets based methods are based on PointNet neural network architecture which is designed for processing point cloud data generated by 3D sensors like Lidar or depth cameras. PointNet architectures [73, 74], demonstrate proficiency in extracting features from point cloud. PointR-CNN [86] is an exemplar in directly classifying and regressing 3D bounding boxes from dense point clouds. Although PointNet-based methods represent a powerful approach to process point clouds directly for 3D object detection, they process each point individually requiring a high computational burden.

All of these solutions are expensive in terms of computational cost, especially in the presence of high-throughput data, like those returned by top-end LiDARs. Moreover, these solutions require a large dataset for the training phase. Therefore, the aim is to propose an algorithm that offers a trade-off between accuracy and speed of execution.

3.3 Processing approaches

The proposed method is illustrated taking into account that a LiDAR consists of an array of laser transmitters, named pointers, and an array of optical receivers coupled to the transmitters, both framed in vertical support. The laser transmitters are directed at uniformly spaced angles to have a vertical field of view (FoV) of several degrees. Each couple transmitter-receiver probes the presence of a reflective obstacle along its own line of sight and determines the distance of that obstacle by measuring the time of flight of the transmitted pulse. The LiDAR performs repeated all-around scans of the neighboring scene by rotating the coupled arrays at a rate that can be often configured up to 20 Hz. The LiDAR data is organized into a sequence of frames, with each frame represented by a point cloud denoted as P_k at time k. A generic point cloud P consists of a set of points $p \in P$ with corresponding Cartesian coordinates $p_i \in \mathbf{R}^3$. Whenever it becomes necessary to precisely specify the Cartesian coordinate of a point p_i, the letters x_i, y_i, and z_i are employed. Each point p_i is associated with various parameters, such as: $r_i \in P$, typically representing the range parameter that quantifies the distance of a target point from the sensor, and $i_i \in P$, which denotes the emittance of the target along the line of sight. For instance, the range parameter of the i-th point p_i of the k-th frame is addressed as $r_i[k]$. It is worth noticing that the LiDAR operation principle naturally copes with polar coordinates, Anyway polar to Cartesian coordinate transformations are seamlessly executed through the embedded edge computing capabilities, as elucidated in Chapter 2, arranging the data frame as described. The main steps of the illustrated method are summarized in Figure (3.1), where it is highlighted that the detection of the moving objects, is followed by bounding box fitting, and hence by the final tracking.

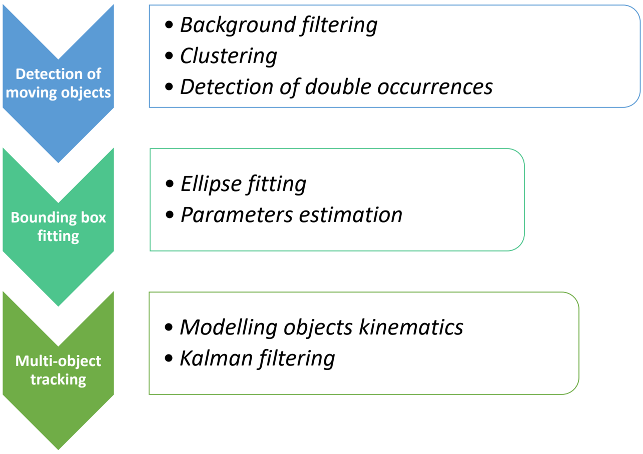


Figure 3.1. Main conceptual steps of the proposed method for detecting and tracking moving objects.

3.3.1 Background filtering

The approach adopted for the detection of the moving objects uses a couple of frames, namely the current and the previous one, and takes inspiration from the action recognition approach adopted for the analysis of RGB video images [87]. Specifically, since for a LiDAR in a static position at the road-side the range parameter changes for the objects that are moving in the scene, whereas it is ideally constant for the background points, the difference between the ranges measured by the LiDAR for the same point in a couple of subsequent frames should permit distinguishing the moving from the static objects. Considering the range value for the i-th point in the k-th frame, $P_k(i;r)$, and the corresponding value for the same point in the previous frame, $P_{k-1}(i;r)$, and introducing a threshold Th, the background points can be recognized as those for which the difference between the ranges is below the threshold. The threshold Th is fixed taking into account that variations in weather conditions and/or random errors due to vibration and instrument noise can produce some difference in the range parameter even between background points.

This method generally allows detecting of the points related to moving objects that intersect the line-of-sight or approach closer to the LiDAR. In fact, if a clear line-of-sight is recognized where an object is detected before, the difference between the range values is negative, and the target is correctly identified as a background point.

3.3.2 Clustering

Clustering allows dividing data with similar properties into homogeneous groups, and is performed by means of the DBSCAN algorithm, which identifies clusters by searching for the relative maxima of a density function. The DBSCAN algorithm is the preferred solution anytime the number of clusters to be searched for is not known in advance. It estimates the density around each point by counting the number of points in a search sphere: if the number of points is greater than a pre-selected minimum the whole set of points in the sphere is clustered as a specific group, otherwise it is marked as noise. The input parameters to DBSCAN are therefore the radius of the search sphere, ϵ, and the minimum number of points expected in the sphere in the presence of an object, $MinPts$. For a given object, the relation between the number of points and the distance is inverse of a quadratic function of the range. As an example, the histogram given in Figure (3.2) shows the number of points produced by a car in the scene at different distances from the LiDAR. The accuracy of the clustering operation critically depends on the choice of the parameters

ϵ and $MinPts$.

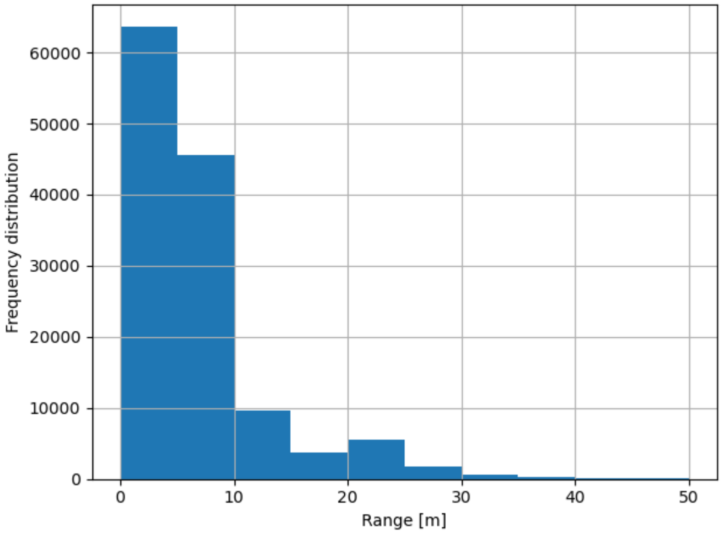


Figure 3.2. Distribution of the number of points reflected by a car as a function of its distance from the LiDAR sensor.

The adopted clustering procedure divides the scene into two partially overlapped areas. The first region includes all the points within the distance d_1 from the LiDAR, and the second region, concentric to the previous one, contains all the points with a distance greater than d_2 from the LiDAR. The distance values satisfy the inequality $d_1 \geq d_2$, which allows an overlap region containing all the points characterized by a distance within the interval (d_1, d_2). Each region is processed using a specific couple of ϵ and $MinPts$ values. Note that choosing the values ϵ and $MinPts$ requires the knowledge of the resolution of the LiDAR at distance d. The minimum vertical and horizontal distances between two points at the same distance d, under the vertical and horizontal angles θ and ϕ, are given by:

$$\delta_h = 2d\ tan(\theta/2) \tag{3.1}$$

$$\delta_w = 2d\ sin(\phi/2) \tag{3.2}$$

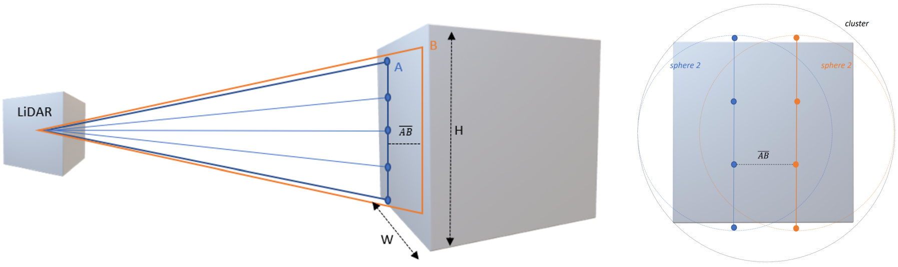
The radius ϵ has to be greater than the $max(\delta_h, \delta_w)|_{d_{max}}$ at the maximum measurable distance d_{max}, otherwise no points may fall within the search sphere.

The $MinPts$ parameter is chosen according to the equation:

$$MinPts = \frac{H}{\delta_h} + 1 \tag{3.3}$$

which estimates how many laser beams hit a target object characterized by height H standing at distance d. Figure 3.3a provides a graphic where the laser beams form a fan that hits an object of height H. It is worth noticing that a large value of $MinPts$ ensures a more robust clustering, but may exclude some smaller objects. On the other hand, a small value extracts many clusters, some of which might include noise points.

For the sake of completeness Figure (3.3)b shows the front view of an object for which the horizontal distance AB between a couple of adjacent points is less than ϵ. Although, the setting of the parameter $MinPts$ is defined by considering the vertical direction, it is worth highlighting that for an object standing at distance d and characterized by width W the number of points hit by the pointer scanning along the horizontal direction, can be estimated with eq.(3.3) by substituting W to H and δ_w to δ_h.



(a)

(b)

Figure 3.3. Laser beams striking an object of height H and width W: (a) Side view of both object and LiDAR. (b) Front view of the object.

3.3.3 Bounding Box Fitting

Ellipse fitting

Bounding box fitting is a fundamental step in the analysis of 2D and 3D images, since many tasks as object localization, multiple object detection, object tracking, and so on, are based on it. For this task, the assumption of the L-shape is not required. Briefly, the algorithm first fits an ellipse to the (x, y) couples of each cluster. The fitting problem is formulated using the standard equation of an ellipse:

$$E(x, y) = ax^2 + bxy + cy^2 + dx + ey + f = 0 \tag{3.4}$$

where $b^2 - 4ac < 0$. In particular, introducing the array $\mathbf{a} = (a, b, c, d, e, f)^T$ and the matrix $\mathbf{D} = (\mathbf{x}^2, \mathbf{xy}, \mathbf{y}^2, \mathbf{x}, \mathbf{y}, \mathbf{1})$, whose columns are defined in terms of the coordinates of the points in the cluster (bold characters are utilized for array and matrix), the problem is formulated as a linear algebra one:

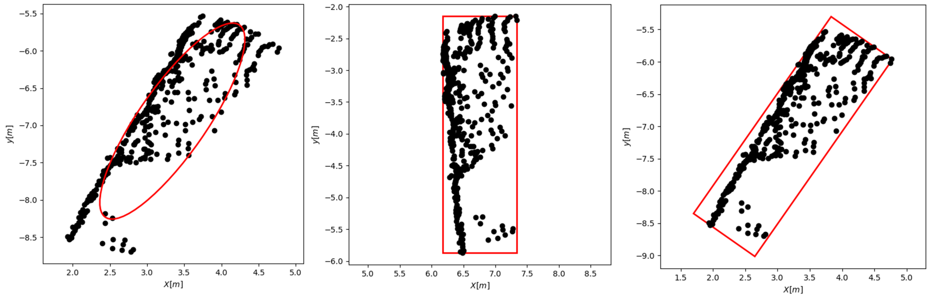
$$\begin{aligned} &\mathbf{D}^T\mathbf{D}\mathbf{a} = \lambda\mathbf{C}\mathbf{a} \\ &\textit{subject to}\ \ \mathbf{a}^T\mathbf{C}\mathbf{a} = 1 \end{aligned} \tag{3.5}$$

where $\mathbf{C}$ is a square matrix with size equal to 6, adopted to specify the constraint: $4ac - b^2 = 1$, which assures the convergence of the solution to an ellipse. Actually, according to this formulation a set of 6 solutions corresponding to the 6 different outcomes of the Lagrange multipliers λ, which can be regarded as eigenvalues of the problem, are possible. The eigenvector corresponding to the smallest eigenvalue provides the best-fitting ellipse.

Figure (3.4a) shows the points of a cluster representing a car and the fitted ellipse: the orientation of the ellipse gives an estimate of the car orientation.

Parameters estimation

The coefficients of the standard equation (a, b, c, d, e, f) are then converted to the ellipse parameters, namely the ellipse centre, the semi-major and semi-minor axes, the eccentricity, and the rotation of the semi-major axis from the x-axis. Using the last parameter, the projection of the 3D scene on the horizontal plane is rotated as if one wanted to align the axes



(a) (b) (c)

Figure 3.4. (a) The clustered points are fitted by an ellipse. (b) Rectangle bounding the cluster in the rotated frame. (c) Rectangle bounding the cluster in the original coordinate system.

of the ellipse to the reference axes. This rotation is accomplished using the rotation matrix with anchor point in the origin, that changes the coordinates of the points of the cluster (x_i, y_i) into (x_i^a, y_i^a), where the superscript a indicates that the points are aligned with the reference axes.

The length of the sides, and the coordinates of the corners and the center of the rectangle bounding the cluster are estimated after the rotation. Specifically, the l_x and l_y sides of the rectangle are obtained as:

- $l_x \;=\; |max(x_i^a) - min(x_i^a)|$
- $l_y \;=\; |max(y_i^a) - min(y_i^a)|$

the corners as:

- $(min(x_i^a),\ min(y_i^a))$
- $(l_x + min(x_i^a),\ min(y_i^a))$
- $(l_x + min(x_i^a),\ l_y + min(x_i^a))$
- $(min(x_i^a),\ l_y + min(y_i^a))$

the center as:

- $(min(x_i^a) + l_x/2, min(y_i^a) + l_y/2)$.

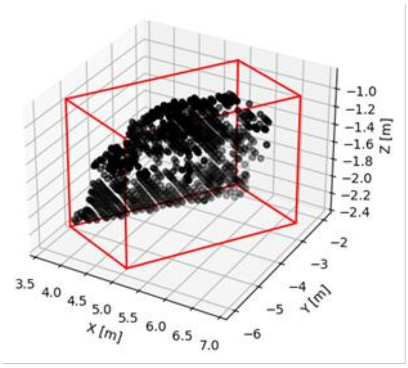


Figure 3.5. Oriented bounding box for the points representing a car.

Figure (3.4)b shows the rectangle bonding the cluster.

A back rotation is performed to display the rectangle in the native horizontal plane and augment the initial view of the scene with the inclusion of the rectangle that highlights the cluster of interest, as shown in Figure (3.4c).

Finally, a 3D bounding box is built around the cluster of interest by rising the rectangle by a height $h = |max(z_i) - min(z_i)|$, where the subscript i for the z_i coordinates means that the maximum and minimum search is restricted to the clustered z-coordinates. The 3D bounding box built on the 2D rectangle in Figure (3.4) is shown in Figure (3.5).

3.3.4 Multi-object tracking

Modelling objects kinematics

The object tracking aims at estimating through the analysis of the sequence of frames the kinematic quantities of a cluster of interest, which can be a car, a motorcycle, a bike, or a pedestrian.

The algorithm assumes that the target is stiff, such that its dynamics is completely described by the changes in orientation and the trajectory of its center, and that the motion can be observed without loss of information in the horizontal plane, i.e. the 3D bounding box always keeps parallel to the z axis.

The implementation of the kinematic model of the target requires the definition of a state vector that includes the kinematic variables of the center of the target, namely: $\mathbf{u}^k = [s_x(k)\ v_x(k)\ a_x(k)\ p_y(k)\ v_y(k)\ a_y(k)]^T$. The model can be described by the system of equations:

$$\begin{cases} s_x^k = s_x^{k-1} + v_x^{k-1}\Delta t + \frac{1}{2}a_x^{k-1}\Delta t^2 \\ v_x^k = v_x^{k-1} + a_x^{k-1}\Delta t \\ a_x^k = a_x^{k-1} \\ s_y^k = s_y^{k-1} + v_y^{k-1}\Delta t + \frac{1}{2}a_y^{k-1}\Delta t^2 \\ v_y^k = v_y^{k-1} + a_y^{k-1}\Delta t \\ a_y^k = a_y^{k-1} \end{cases} \tag{3.6}$$

where Δt is the time interval between two consecutive frames. The choice of the constant acceleration model in equation (3.6) is justified as it encompasses all three kinematic quantities while excluding the consideration of vehicle orientation, which is determined through a distinct procedure.

Kalman filtering

State Prediction: For the tracking task, the proposed approach relies on Kalman filtering, that allows predicting and estimating the dynamics of the target by combining model-based predictions and observations. According to the theory of Kalman filters, the prediction is performed by calculating:

$$\bar{\mathbf{u}}^k = \mathbf{F}\mathbf{u}^{k-1} \tag{3.7}$$

$$\bar{\mathbf{P}}^k = \mathbf{F}\mathbf{P}^{k-1}\mathbf{F}^T + \mathbf{Q} \tag{3.8}$$

where u^{k-1} and $\mathbf{P}^{k-1}$ are, respectively, the state and covariance matrix for the $k-1$ frame, $\bar{\mathbf{u}}^k$ and $\bar{\mathbf{P}}^k$ are, respectively, the predicted state and covariance matrix for the k-th frame, $\mathbf{F}$ is the state transition matrix obtained from the modeling equations (3.6), and $\mathbf{Q}$ is a matrix that takes into account the noise of the process, which is supposed Gaussian with zero mean and covariance equal to $\mathbf{Q}$.
Detection-to-Track Association: An association process then relates each target detected at the k-th frame with its version in the $k-1$-th frame.

The association process derives:

$$\bar{\mathbf{z}}^k = \mathbf{H}\bar{\mathbf{u}}^k \tag{3.9}$$

$$\mathbf{A}^k = \mathbf{H}\bar{\mathbf{P}}^k\mathbf{H}^T + \mathbf{R} \tag{3.10}$$

where $\bar{\mathbf{z}}^k$ and $\bar{\mathbf{A}}^k$ represent, respectively, the vector of predicted positions and the innovative matrix, $\mathbf{R}$ is the covariance matrix of the noise for uncorrelated measurements, and $\mathbf{H}$ is the observation matrix, given by:

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{3.11}$$

In the presence of a single candidate, the kinematic quantities related to the center of an object are straightforwardly updated, whereas in the presence of multiple candidates, an association process is first required. This process finds the shortest distance between the predicted positions and the current ones. When the target gets out of the scene, and no candidates are detected in the region of interest, the association process stops.

State Update: After the matching between the detection and prediction, the state update equations provide an improved *a-posteriori* estimation by combining the *a-priori* estimation with the observations.

The state and covariance matrix are given by:

$$\mathbf{P}^k = \left(\mathbf{I} - \mathbf{K}^k\mathbf{H}\right)\bar{\mathbf{P}}^k \tag{3.12}$$

and

$$\mathbf{u}^k = \bar{\mathbf{u}}^k + \mathbf{K}^k\left(\mathbf{z}^k - \bar{\mathbf{z}}^k\right) \tag{3.13}$$

where $\mathbf{K}$ is the Kalman gain, given by:

$$\mathbf{K}^k = \bar{\mathbf{P}}^k\mathbf{H}^T\left(\mathbf{R} + \mathbf{H}\bar{\mathbf{P}}^k\mathbf{H}^T\right)^{-1} \tag{3.14}$$

3.4 Instruments set-up

The LiDAR system described herein utilizes the OS0-128 model and is designed for live stream acquisition. A computer with a gigabit Ethernet connection and a 24V power supply is required for operation.

The acquisition setup, depicted in Figure (3.6), includes the OS-0 sensor, where the point clouds can be visualized using the Ouster Visual Studio or any other 3-D point cloud visualization software. Accompanying the OS-0 sensor is the Interface Box, facilitating test and evaluation purposes. This box links the OS-0 sensor cable into its two constituent connectors; power and Ethernet for a connection with the computer. Ouster provides two varieties of Interface boxes catering to both 12V and 24V requirements. The Interface Boxes (Standard) equipped with Type 1 and Type 2 cables are constructed with built-in cables for linking to Ouster's sensors. It's important to note that these cables are permanently attached to the interface box and cannot be detached due to the absence of a connector.

The Interface Boxes (12V) come with Type 3 cables and include an output connector along with an independent cable fitted with connectors for linking to Ouster's sensors. These boxes also incorporate a GPS input connector for compatibility with GPS modules.

For 24V compatibility, the Interface Box (Standard) is designed to operate the sensor for testing and evaluation. It terminates the interface cable from the sensor and provides power via a 24V DC Supply. The box allows access to the sensor's gigabit Ethernet Interface through a standard RJ45 connector.

It's essential to note that the Ouster Interface Box serves as a support tool for laboratory environments, aiding customers in evaluating Ouster's LiDAR sensor products and software development. However, the Interface Box is not suitable for outdoor use, as it lacks protection against moisture and solid particles.

Ouster Studio is a software solution developed by Ouster enabling users to configure the lidar, real-time visualization, processing, and recording of dynamic 3D LiDAR data acquired through Ouster LiDAR sensors. It is compatible across various platforms, officially supporting Windows, MacOS, and Ubuntu. Additionally, Ouster Studio has the capability to replay streamed data saved in .pcap files, which were initially recorded from live

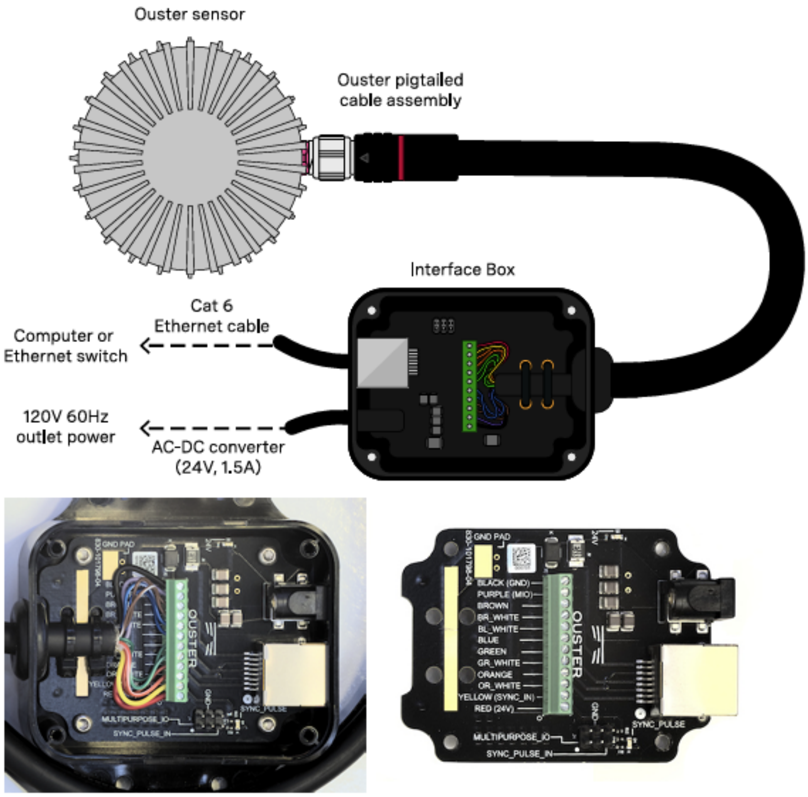
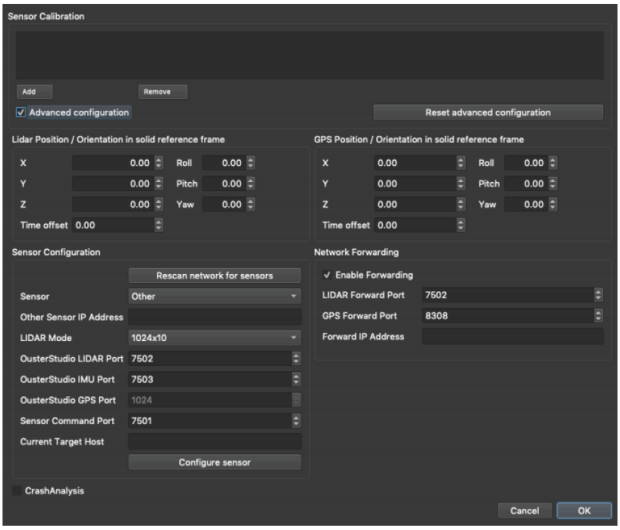


Figure 3.6. Interface box of Ouster OS0 - 128 LiDAR, from Ouster website

streams, directly within the Ouster Studio environment.
The "Sensor Configuration" section within Ouster Studio is instrumental in setting up the software's listening ports for live-stream capture and determining the Lidar mode. Specifically, the ports $udp - port - lidar$ (set to 7502) and $udp - port - imu$ (set to 7503) are designated for acquiring data from the Lidar and inertial measurement unit (IMU), respectively.

Additionally, synchronization of the LiDAR is achieved through the internal clock source. All these configurations are encapsulated in a calibration file with the *.calib* format. The calibration file specifies the LiDAR sensor properties, such as the operating mode, synchronization parameters,



Sensor Calibration

☑ Advanced configuration

Lidar Position / Orientation in solid reference frame

X 0.00 Roll 0.00
Y 0.00 Pitch 0.00
Z 0.00 Yaw 0.00
Time offset 0.00

GPS Position / Orientation in solid reference frame

X 0.00 Roll 0.00
Y 0.00 Pitch 0.00
Z 0.00 Yaw 0.00
Time offset 0.00

Sensor Configuration

Sensor: Other
Other Sensor IP Address:
LIDAR Mode: 1024x10
OusterStudio LIDAR Port: 7502
OusterStudio IMU Port: 7503
OusterStudio GPS Port: 1024
Sensor Command Port: 7501
Current Target Host:

Network Forwarding

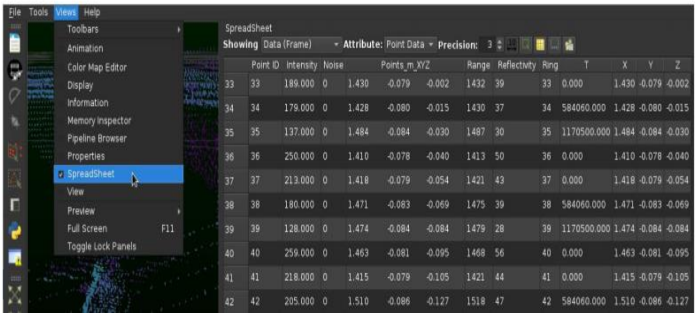
✓ Enable Forwarding
LIDAR Forward Port: 7502
GPS Forward Port: 8308
Forward IP Address:

□ CrashAnalysis

Figure 3.7. Sensor configuration through Ouster studio software.

corrections for the altitude and azimuth angles of the sensor when operating in a specific mode, and the transformation matrix for the inertial measurement unit (IMU) intrinsics. During live stream operation, the calibration file is dynamically generated and can subsequently be utilized to replay *.pcap* files produced during the live stream. This is particularly useful in scenarios where the LiDAR is not connected to the network, allowing users to replay data through the "Playback mode" button in Ouster Studio.

The Ouster software already provides the processed data for the sensor's reference system. In fact, the data returned by the ouster software consists of a sequence of frames, represented by means of a sequence of 2D arrays, each one describing a scene as a cloud of points. The Figure (3.8) provides an example. The row index identifies the point of the scene; the last 3 columns provide, respectively, the Cartesian coordinates x, y, and z of the point in a reference system with the origin in the center of the sensor. The 2D array also gives additional information in further columns; for instance, a fourth column is typically for the range parameter, which quantifies the distance of the target point from the origin of the reference system, and a fifth column gives the emittance of the target on the line of sight. It is



SpreadSheet

Showing Data (Frame) Attribute: Point Data Precision: 3

	Point ID	Intensity	Noise	Points_m_XYZ			Range	Reflectivity	Ring	T	X	Y	Z
33	33	189.000	0	1.430	-0.079	-0.002	1432	39	33	0.000	1.430	-0.079	-0.002
34	34	179.000	0	1.428	-0.080	-0.015	1430	37	34	584060.000	1.428	-0.080	-0.015
35	35	137.000	0	1.484	-0.084	-0.030	1487	30	35	1170500.000	1.484	-0.084	-0.030
36	36	250.000	0	1.410	-0.078	-0.040	1413	50	36	0.000	1.410	-0.078	-0.040
37	37	213.000	0	1.418	-0.079	-0.054	1421	43	37	0.000	1.418	-0.079	-0.054
38	38	180.000	0	1.471	-0.083	-0.069	1475	39	38	584060.000	1.471	-0.083	-0.069
39	39	128.000	0	1.474	-0.084	-0.084	1479	28	39	1170500.000	1.474	-0.084	-0.084
40	40	259.000	0	1.463	-0.081	-0.095	1468	56	40	0.000	1.463	-0.081	-0.095
41	41	218.000	0	1.415	-0.079	-0.105	1421	44	41	0.000	1.415	-0.079	-0.105
42	42	205.000	0	1.510	-0.086	-0.127	1518	47	42	584060.000	1.510	-0.086	-0.127

Figure 3.8. Spreadsheet view

worth noticing that the LiDAR operation principle naturally copes with polar coordinates as illustrated in Chapter 2, anyway polar to Cartesian coordinate transformations are executed by means of inherent edge computing capabilities to arrange the data frame as described. Finally, the spreadsheets allow for exporting data as *.csv* files.

3.5 Results

The experimental site is an urban road in the metropolitan city of Napoli, Italy, with both road and rail paths, characterized by substantial traffic congestion during office hours. The site is surrounded by lines of buildings on both sides and made up of 2 lanes, one for each direction, as shown in Figure (3.9). The monitoring of the road site is performed by means of Ouster LiDAR OS0, mounted at a height of 1.5 m on a tripod set at the side of the road in a parking slot. As illustrated in Chapter 2, the LiDAR is characterized by 128 laser transceivers, or channels, directed to gain a 90-degree vertical field of view of the site. Each channel of the LiDAR provides a (horizontal) all-around view by sampling the scene at uniformly spaced angles: 3 different configurations are available, that use 524, 1024, or 2048 angular quantization steps.

The rotation rate of the LiDAR can be chosen between 10 and 20 Hz to have either 10 or 20 frames per second; the resulting data stream can be

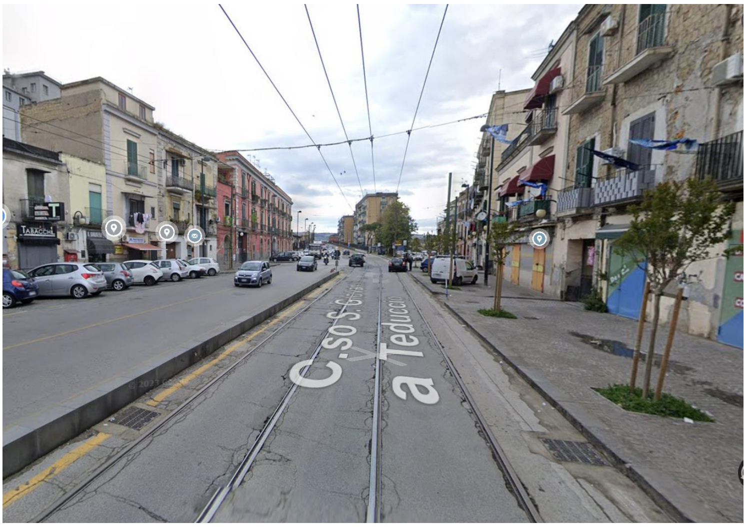


Figure 3.9. Street-view of the experimental site from Google street

reduced by decimating the number of channels, if required, provided that less time resolution is acceptable.
Finally, the mode 1024x10 is chosen for the sensor configuration which means 1024 horizontal points with an azimuth window $[0, 360]$ and a frequency rate set to $10Hz$.
Using 128 channels and scanning at 1024 angles, the frame describing the scene is made up of 131072 rows; the absence of reflected data, which can occur for rays addressed to the far horizon, produces zero values in the frame.

Subsequently, sensor configuration and data acquisition were performed using the Ouster studio software, illustrated in Section (3.4).
Background filtering: As the employed LiDAR is installed on a ground-fixed tripod, the background points remain stationary, allowing for their identification as points with a consistent range within the scene. This identification process involves the introduction of a threshold to eliminate any noise. However, the accuracy of the range measurements is susceptible to external influences, such as weather conditions, which notably affect objects situated at a distance from the LiDAR. As a consequence, a single

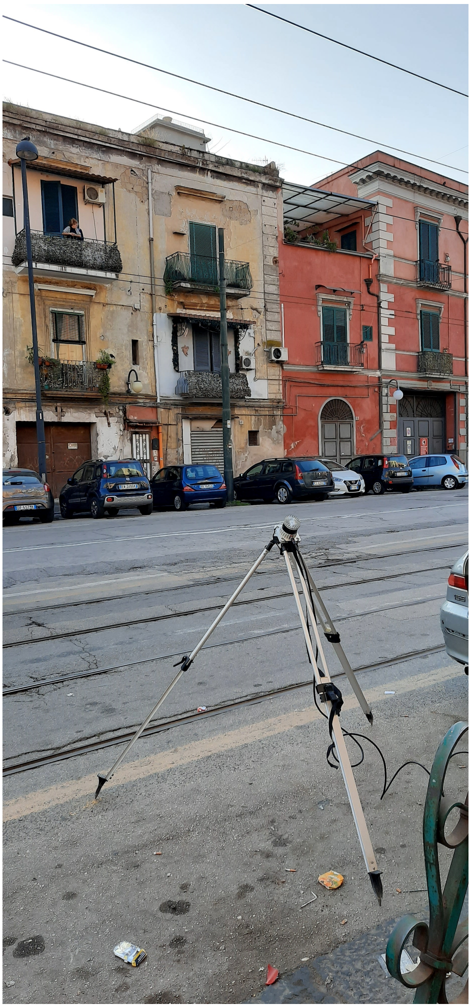


Figure 3.10. Set-up adopted to carry out the experiments

threshold value for the whole scene would reduce the accuracy of the background filtering. As the threshold value is chosen the value $Th = \alpha r_i[k]$ depending on the distance through the coefficient α. The value of the coefficient is determined at the calibration stage using a reference frame without road users. The performance in background filtering is expressed in terms of the ratio between the number of background points detected by the algorithm and the expected value in the presence of a moving object. The statistical analysis typically reveals a performance of 98.5%. An example of the results produced by the algorithm in recognizing the moving objects in an urban scene is shown in Figure (3.11). *Clustering*: For validating the clustering operation, the LiDAR field of view is divided into two regions: the near region reaches up to 8 m with a shadow portion close to the sensor up to 3 m, the far region is beyond 7 m from the LiDAR; the two regions overlap to each other in a belt centered at a distance of 7.5 m with 1 m wideness. The ϵ parameter is chosen equal to 1 m for both regions taking into account the technical specifications of the adopted LiDAR. A pedestrian is assumed to be the smallest moving object, with a rounded height, H, approaching 2 m, to be detected in road monitoring. The $MinPts$ parameter is obtained from equation (3.3) adopting for the distance parameter d the values 3 m and 7 m, to obtain respectively, 50 and 30 that are utilized for the near and far regions, respectively. The number of clustered objects detected coincides with the number of objects in the urban scene for all frames under test.

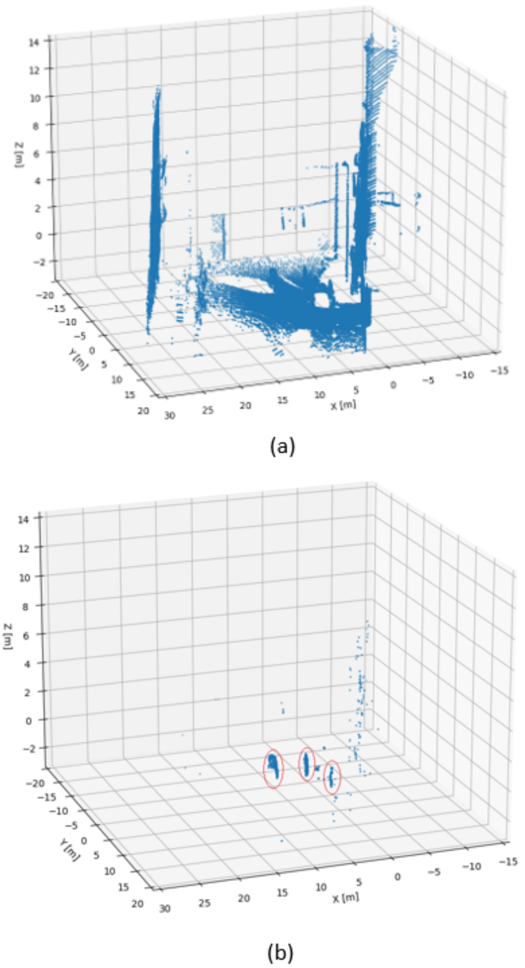
The noise points shown in the scene (3.11), are removed by the DBSCAN application and the final result is shown in the Figure (3.12).

Bounding box fitting: To demonstrate the effectiveness of the suggested approach in determining vehicle orientations, a visual representation is generated. In Figure (3.13), several examined case studies are presented. Specifically, (a) displays the ellipse fitted to the points of the cluster in the horizontal plane, (b) illustrates the bounding rectangle, and (c) showcases the parallelepiped constructed above the rectangle.

Multi-Object Tracking:

To validate the methodology, the predictions offered by the Kalman filter are expressed in terms of squared error values beneath $0.35m$.

The Figure (3.14) shows two examples of position signals. Position signal predictions are indicated in red, while measurements are indicated in green. Two vehicles are represented using two different markers. The algorithms run on a DELL PC XPS, i7 intel 11th gen, 32 GB RAM. The average computing time for each step is shown in Table (3.1). The execution time has substantial margins to be reduced by restricting the analysis to a suitably cropped region of interest. The reduced processing time is guaranteed by the simplicity of the solution which avoids frame aggrega-



(a)

(b)

Figure 3.11. Sample scene acquired by the LiDAR sensor:in (a) the cloud of points describes the urban scene in Cartesian coordinates including objects and background; in (b) the sole points resulting after background filtering are shown. The points grouped within the red borders correspond to: a pedestrian, a motorbike and a car, whereas the remaining points represent noise points.

tion and merely relies on frame comparison.

Benchmark analysis To further evaluate the performance of the method,

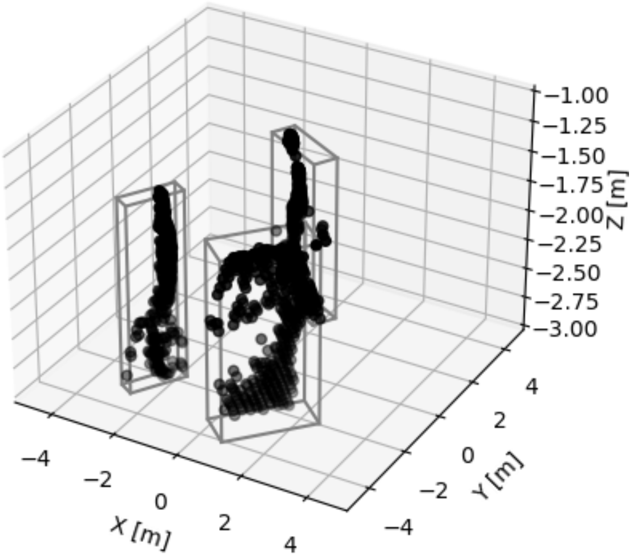


Figure 3.12. Urban scene with 3 moving objects after applying clustering and estimation of their size. The pre-segmentation scene and its residual frame are shown in Figure (3.12), as viewed from a different angle.

Table 3.1. Execution time of the proposed algorithms applied to frames with 131073 points

Steps	Time [s]
Background filtering	0.0041
Clustering	0.0398
Detection of double occurrence	0.0038
3D Orientated Bounding Box Fitting	0.0024
Multi-Object Tracking*	0.00032

* Average time for two vehicles in the scene

the proposed algorithm for the detection of moving objects is compared with the two most cited methods of state-of-the-art: the voxel density-based method proposed in [88] and referred to as 3D-DSF, and the Raster-based background filtering data, proposed in [89] and referred to as RA.

Table 3.2. Benchmark analysis

Method	Voxel size [m]	Accuracy [%]
3D-DSF	1	99,5
	0.5	99,6
	0.1	x
	0.05	x
RA	0.1	x
	0.05	x
Proposed method	-	99,7

The results are shown in Table (3.2). Here as well, the performance in background filtering is expressed in terms of the ratio between the number of background points detected by the algorithm and the expected value. It is found that the proposed method performs better than the other two methods in terms of accuracy. Table (3.2) illustrates that, particularly when the cubic voxel size is considerable, falling within the range of 0.5 meters to 1 meter, despite the effective filtering of nearly all background points, this leads to the exclusion of all object points from the scene. For small sizes, i.e. in the range $(0.1m - 0.05m)$, commercial hardware (DELL PC XPS, i7 intel 11th gen, 32 GB RAM) returns a memory error since it is unable to cope with the very high number of points: this issue is denoted by the marker x in Table (3.2). The 3D-DSF method is therefore not applicable to LiDAR with 128 channels due to the high computational effort demand. The authors of the 3D-DSF method stated the method is developed by using a LiDAR with 16 channels. In addition, for the computational load, 3D-DSF requires about 64 minutes for initialization of the 3D matrix with an aggregation of 100 frames; the required time increases as the number of frames to be aggregated increases. Similarly, the RA method based on a 3D grid as 3D-DSF, fails for small cube sizes. The tracking process is compared with [90], which shows no appreciable difference in run-time but an average error between predicted and measured position equal to about 50 cm, whereas the proposed method shows an average error of 2.7 cm.

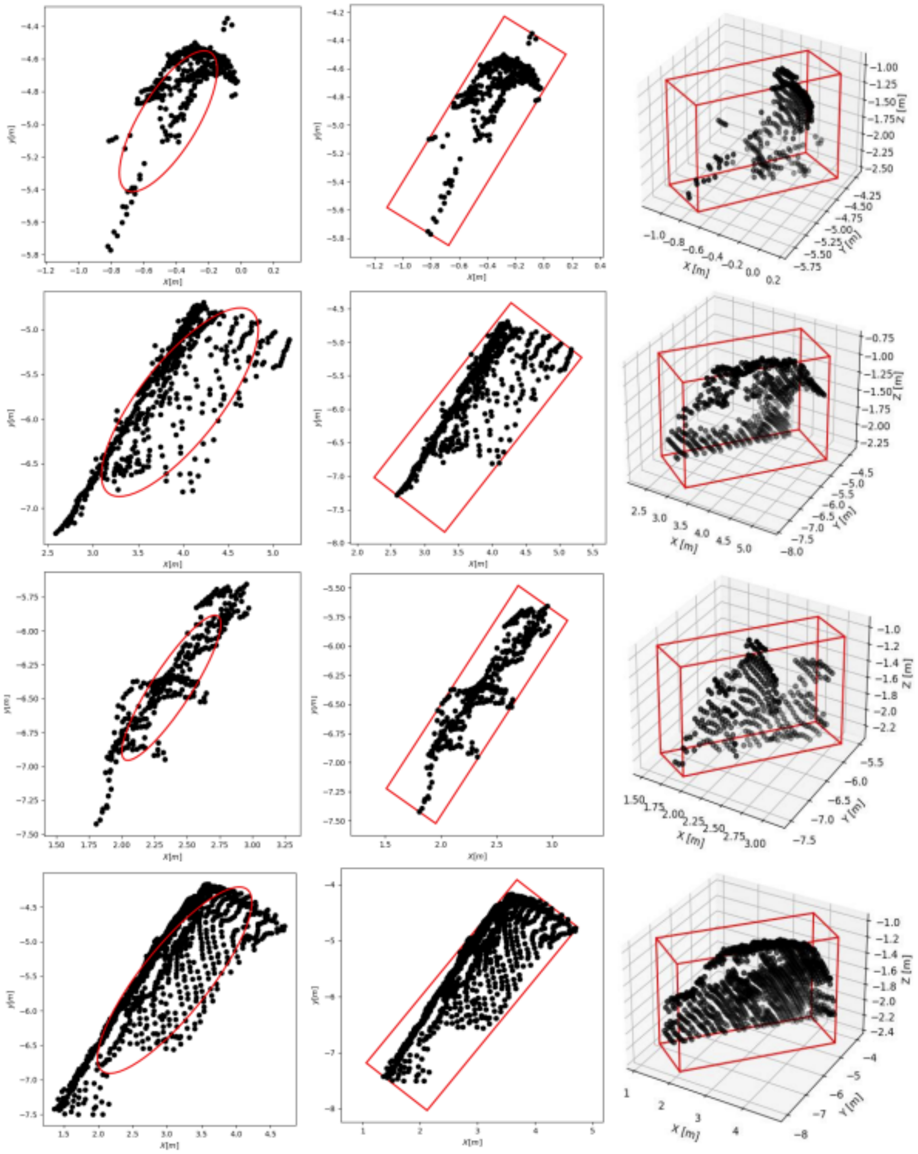
3.6 Conclusions

This study introduces a method for the real-time detection and tracking of moving objects using LiDAR sensors. The proposed approach encompasses several key steps, including background filtering, object clustering, and tracking, aimed at extracting essential information such as the presence, position, velocity, acceleration, size, and direction of objects.

The method involves a series of distinct steps. Initially, there is a real-time filtering process targeting static points like road surfaces, trees, and buildings to isolate only the points associated with moving objects. In the subsequent step, the filtered points undergo clustering to determine the number of objects present in the scene. The DBSCAN algorithm is utilized for clustering, effectively partitioning the scene into two partially overlapping regions. Special attention is given to handling the presence of objects at the scene's edge to recognize double occurrences of the same object and refine the results. The proposed approach also addresses the estimation of object orientation through ellipse fitting.

To ensure continuity and coherence in tracking objects across frames, the study suggests the application of Kalman filtering for inter-frame object association and tracking.

An experimental setup has been utilized to acquire real data in an urban site, namely in the city of Napoli, Italy. The background subtraction algorithm shows 99% of success in background point recognition. A visual estimation of the result has also been given, where the objects have been encapsulated with 3D bounding boxes. The tracking process has been evaluated in terms of MSE metrics, which have shown values in the interval $(0.0145, 0.0440)m$ for objects moving in an urban scenario. The current method does not address the classification problem, a significant aspect that warrants attention in future research endeavors. There are several noteworthy areas for exploration, including the validation of the proposed approach using diverse LiDAR sensors or varying numbers of channels.



(a) (b) (c)

Figure 3.13. Examples of bounding box selected from randomly chosen frames: (a) and (b) show the projection of the scene on the horizontal plane, highlighting the fitting ellipse and the rectangle used to feature the detected object, while (c) shows the parallelepiped built upon the rectangle.

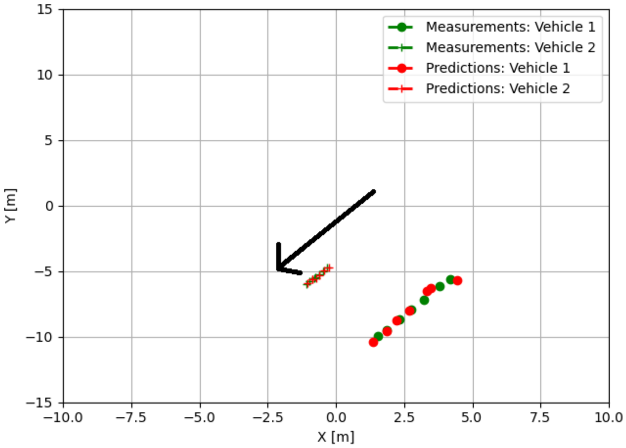


Figure 3.14. Two track position signals relating to two vehicles with different speeds. the direction of the two vehicles is indicated with the black arrow

Chapter 4

LiDAR localization

4.1 Introduction

Accurate vehicle localization is a significant challenge in achieving precise vehicular navigation. A proficient navigation system has to demonstrate effectiveness across diverse driving scenarios with varying traffic intensities and under different weather and lighting conditions. The primary goal of localization is for a vehicle to know where it is in relation to a predefined coordinate system or map. In three-dimensional space, vehicle localization refers to the process by which a vehicle determines its own position and orientation within its environment, denoted as 6 degrees of freedom (DOF) pose.

In the context of autonomous vehicles, the orientation of a vehicle in three-dimensional space typically refers to the "attitude". It describes how the vehicle is positioned or aligned with respect to a reference frame. Attitude is often expressed using Euler angles (roll, pitch, and yaw) or quaternion representations. These parameters describe rotation around the three axes x, y, and z respectively.

Therefore, when the vehicle's movement can be assumed to occur on a flat surface, employing only the 2D variant with the first three DOF poses is typically sufficient.

This information is crucial for enabling the vehicle to navigate effectively and perform tasks autonomously. There are several methods and techniques used for vehicle localization, and the choice often depends on the available sensors, the complexity of the environment, and the desired level of accuracy. Numerous technologies are designed to address the localization problem. The prevalent approach for determining the position, and orientation of a vehicle is the Inertial Navigation System (INS).

An Inertial Navigation System (INS) is a navigation aid that employs Inertial Measurement Unit (IMU) tethered to the vehicle to continuously calculate the position, orientation, and velocity of a moving object without the need for external references such as GPS or landmarks. This IMU utilizes tri-axial sensors to measure acceleration and angular velocity. IMUs, being micro-electro-mechanical systems characterized by their compact size and affordability, unfortunately, exhibit suboptimal accuracy owing to the inherent manufacturing processes [91]. Strapdown equations are foundational to the functioning of Inertial Navigation Systems. They describe the relationship between the measurements obtained from inertial sensors and the essential navigation parameters—position, velocity, and attitude. The term *strapdown* denotes an IMU sensor rigidly attached to the moving platform, and the equations are expressed in the body-fixed frame. For the translational motion, these equations govern how accelerations are integrated to update velocity and position. Simultaneously, for rotational motion, the equations guide the integration of angular rates to update the orientation or attitude. Strapdown equations are crucial for the real-time, continuous estimation of the vehicle's trajectory, making INS a reliable navigation solution. Angular velocity can be employed to deduce the attitude by integrating the kinematic differential equation of a rigid body concerning Euler angles [92]. Specifically, if the attitude is initially known, it can theoretically be updated at subsequent time points over extended durations. An initial estimation of the attitude can be derived from the acceleration signal by isolating the gravitational component, referred to hereafter as internal acceleration. In the absence of external acceleration, processing the signals detected along the three axes of the accelerometer allows the identification of the static tilt of the vehicle. Unfortunately, IMU data is characterized by significant uncertainty, inherent in all micromachined devices, leading to propagated uncertainties throughout the processing stages and resulting in noisy drifts [93, 94].

In practice, to mitigate noisy drift and obtain reliable data, multiple sensors are integrated by using appropriate data fusion principles. Sensor fusion is employed for localization for several reasons, as it offers advantages that contribute to improved accuracy and reliability in determining the localization of a device or system within its environment. Several reasons exist why sensor fusion is used for localization, such as an improved accuracy and robustness to sensor failures. Combining data from multiple sensors allows for a more accurate estimation of the device's localization. Each sensor may have its strengths and weaknesses, and by fusing its data, the system can leverage the strengths of one sensor to compensate for the limitations of another. Sensor fusion enhances the system's robustness by reducing the impact of individual sensor failures. If one sensor malfunctions or provides unreliable data, other sensors in the fusion process can compensate for the deficiency, ensuring continuous and reliable localization.

Common frameworks for data fusion involve the utilization of Kalman filters (KF), extended Kalman filters (EKF), Unscented Kalman Filter (UKF), and Mahony filters [95, 96, 97, 98]. However, accelerometer measurements are susceptible to distortions caused by external accelerations during vehicle motion, resulting in a sum of external accelerations and internal acceleration [99]. However, distortions arising from external accelerations during vehicle motion present challenges in accelerometer measurements [100]. Nearly all existing methods employ an acceleration model to mitigate the impact of external acceleration, facilitating an attitude estimate based on gravitational acceleration. Regrettably, as demonstrated in [101], these strategies prove reliable in the short term but exhibit dissatisfaction over prolonged durations.

Relying on GPS signals to circumvent external acceleration estimation faces unreliability, primarily due to GPS unavailability in certain environments and the low accuracy of the resultant output [102, 103]. In contrast, more sophisticated approaches leverage computer vision techniques, particularly those grounded in artificial neural networks. However, these solutions demand substantial datasets and significant hardware resources to handle the intensive computational tasks required during the training phase [104, 105]. This study introduces novel methods aimed at enhancing the accuracy estimation of attitude, position, velocity, and acceleration for a LiDAR system. More specifically, this chapter proposes:

(1) a method for improving the estimation of the attitude of a LiDAR. The

method avoids the use of the accelerometer data which can be confounded by the external acceleration, due for instance to motion and vibration, and relies on independent data to recognize the vertical direction. In particular, it works by processing the point clouds for estimating the vector that is normal to the horizontal plane. To this end, a best-fit approach for the ground surfaces detected in subsequent frames is exploited. The normal vector is used to estimate the ground surface orientation with respect to the reference frame. Then, these orientation estimations are fused with gyroscope data by Kalman filter to improve the final attitude estimation. Moreover when the tilt of LiDAR is known, the external acceleration can be determined as additional information by removing the internal one from the acceleration readings.
(2) a theoretical framework to implement multi-sensor data fusion method for kinematic quantities. The framework allows the combination of signals obtained from position, velocity and acceleration sensors addressing the same target, and the improvement in the observation of the kinematics of the target. In different terms, it provides a general framework where it is possible to choose which kinematic signal to be improved. Differently from several alternative methods, the considered ones need no dynamic and/or error models to operate and can be implemented with low computational burden. In fact, they gain measurements by summing filtered versions of the heterogeneous kinematic quantities. In particular, in the case of position measurement, the use of filters with finite impulse responses, all characterized by finite gain throughout the bandwidth, in place of straightforward time-integrative operators, prevents the drift that is typically produced by the offset and low-frequency noise affecting velocity and acceleration data.

4.2 The reference frames

In an Inertial Navigation System (INS), various Cartesian coordinate reference frames are employed to precisely define the position, orientation, and motion of a moving object. Each frame constitutes an orthogonal, right-handed coordinate system. Here are several commonly used reference frames in INS.

The inertial frame ($i-frame$), as illustrated in Figure (4.1), has its

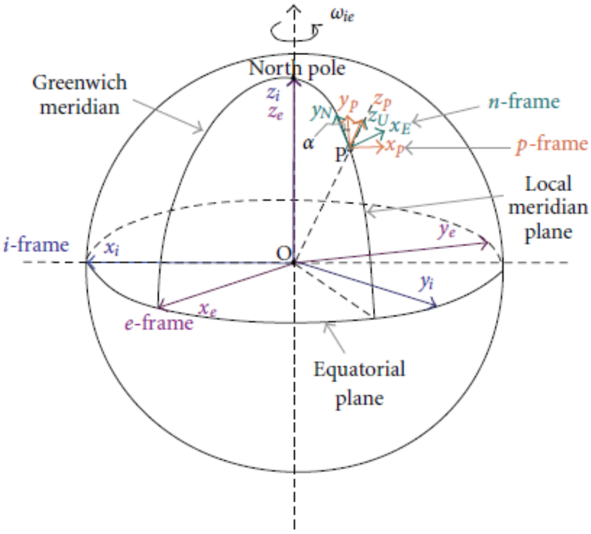


Figure 4.1. Definition of reference frames in an Inertial Navigation System [4].

point of origin situated at the Earth's center. The axes remain non-rotating relative to fixed stars, denoted as O_{x_i}, O_{y_i}, and O_{z_i}, where O_{z_i} is aligned with the Earth's polar axis. In classical physics, an inertial frame of reference remains unaccelerated over time. This frame represents a frame of reference wherein Newton's first law of motion is valid; a physical object, on which acting a zero net force, is observed to move at a constant velocity. The Earth frame ($e - frame$), as illustrated in Figure (4.1), is established with its origin at the Earth's center and axes that remain unrotated relative to the Earth. These axes are specifically labeled as O_{x_e}, O_{y_e}, and O_{z_e}. The O_{z_e} axis is aligned with the Earth's polar axis, while the O_{x_e} axis aligns with the intersection of the plane defined by the Greenwich meridian and the Earth's equatorial plane, the O_{y_e} completes the right-handed system. Unlike the previous frame, the ECEF frame rotates with the Earth around the O_{z_i} axis. The rotation angle is known as the Greenwich Mean Sidereal Time angle and is denoted by θ_{GMST} in Figure (4.1).
The navigation frame ($n - frame$), as depicted in Figure (4.1), constitutes a local geographic navigation reference frame. Its origin is situated

at the location of the navigation system, denoted as point P. Here, it is assumed the navigation system is securely fixed to the vehicle. As shown in Figure (4.1) , $n-frames$ contains two configuration. When the axes of this frame are aligned with the directions of east (PE), north (PN), and the local vertical up (PU), the $n-frame$ is commonly known as the " East-North-Up (ENU) -frame." Similar to the ENU frame, the North-East-Down NED frame contains three orthogonal axes in which the x-axis points to true North, the z-axis points towards the interior of the Earth and the y-axis completes the right-handed system pointing East. The rotation or turn rate of the navigation frame concerning the Earth-fixed frame, is influenced by the movement of point P relative to the Earth. This phenomenon is often referred to as the transport rate.

For the majority of applications, determining the position, velocity, and orientation of a system involves sensors mounted on a vehicle or platform. This vehicle often possesses its reference frame, commonly referred to as the body frame or vehicle frame. This reference frame includes an origin typically located at the center of the vehicle, point P. Usually, it features three orthogonal axes constituting a right-handed system. In certain applications, perfect alignment between the sensor frame and the body frame might not be achievable. The sensor frame is a type of reference frame that is fixed to the sensor and in such cases, a reference frame rotation becomes necessary to align the two frames effectively.

4.3 Inertial measurement unit (IMU)

4.3.1 Introduction to IMU

An Inertial Measurement Unit (IMU) is a critical component in various navigation and motion-sensing applications. It is a device that integrates multiple sensors to measure and report a body's acceleration, angular rate, and sometimes the magnetic field surrounding the device. IMUs are widely used in fields such as aerospace, robotics, automotive, and virtual reality to provide accurate and real-time information about an object's movement and orientation. Key components of a typical IMU include a three-axes accelerometer and a three-axes gyroscope.

Accelerometer

The operation principle of typical accelerometers exploits the effects caused by acceleration on a probing mass, which is encapsulated in a frame, and kept in a resting position by a system made up of linear springs. In the presence of acceleration, the mass compresses and stretches the springs, according to the mass-spring-damper equation, which, for a linearly constrained mass is:

$$m\frac{d^2x(t)}{d^2t} + k_d\frac{dx(t)}{dt} + k_b x(t) = F(t) \tag{4.1}$$

where m is the mass, $x(t)$ is the linear displacement travelled by the mass, k_b the spring constant, k_d the damping factor, and $F(t)$ the apparent force due to the reaction to acceleration. The acceleration measurement is hence performed indirectly by measuring the mass displacement by means of piezoresistive, piezoelectric, or variable capacitance devices. Figure 4.2 shows a typical architecture of an accelerometer that uses a variable capacitor to sense the mass displacement. The capacitor is formed by a planar electrode, connected to the linear springs, and coupled to a parallel electrode anchored to the bulk of the device. An *ad hoc* circuit detects the capacitance change, which is proportional to the mass position, and provides an electrical signal, which is in turn proportional to the acceleration.

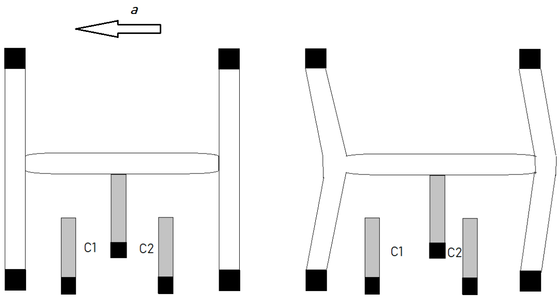


Figure 4.2. Structure of an accelerometer based on capacitive mechanism; conductive arms are indicated in gray and anchor points in black. Acceleration acts on the structure as shown on the right, thus changing the capacitance of the system.

Gyroscope

The gyroscope measures the angular velocity of body segments. Nowadays, gyroscopes are present in different technologies; mechanical, optical, and microelectronic vibrating mass gyroscopes. Figure (4.3) shows a diagram to illustrate the basics of a gyroscope that exploits a vibrating mass under the Coriolis force. The resonating mass is tethered to a polysilicon frame by springs, that let it resonate only along a given (driving) direction. To measure the strength, the frame is tethered to the substrate by springs at 90 degrees. The proof mass is excited to continuously vibrate along the drive direction. In the presence of rotation around an axis perpendicular to the direction of vibration, the mass experiences a displacement due to the effect of the Coriolis force, which is proportional to the angular velocity. On the right of Figure (4.3), the resonating mass moves toward the top along the rotation radius, whereas the frame is deflected to the left. The opposite case happens when the resonating mass moves to the bottom along the rotation radius. The figure also shows the sense fingers that measure the displacement of the frame in response to the force. From [106], the displacement S of the frame relative to the substrate is equal to:

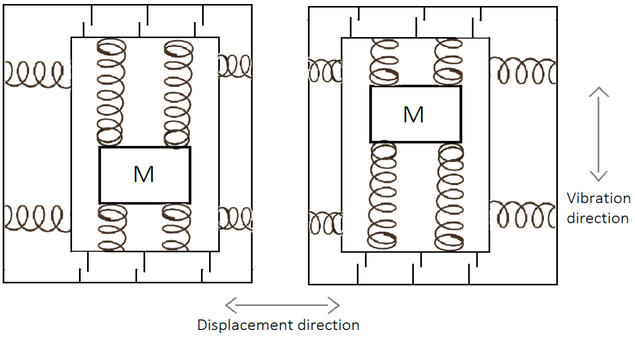


Figure 4.3. The inner frame and vibrating mass are laterally translated by Coriolis force. The displacement is proportional to the change in capacitance between the sense fingers anchored to the frame and substrate.

$$S = 2(\Omega v)m/k \tag{4.2}$$

where Ω is the angular velocity of the gyroscope, v the velocity of the resonant mass along the radius of rotation, m the mass, and k the spring constant. The angular velocity can be derived from the voltage inducted by the displacement S with good linearity.

4.3.2 IMU sensor model

An IMU sensor model is a mathematical representation describing the behavior of the IMU sensors, such as accelerometers and gyroscopes. The angular rate $\tilde{\boldsymbol{\omega}}$ and acceleration $\tilde{\boldsymbol{a}}$ measurements can be modeled as:

$$\begin{aligned}\tilde{\boldsymbol{\omega}}(t) &= \boldsymbol{\omega}(t) + \boldsymbol{b} + \boldsymbol{n}_\omega(t) \\ \tilde{\boldsymbol{a}}(t) &= \boldsymbol{a}(t) + \boldsymbol{g}(t) + \boldsymbol{n}_a(t)\end{aligned} \tag{4.3}$$

where $\tilde{\boldsymbol{\omega}}(t)$ is a vector with three components along x,y,z-axis containing the gyroscope readings at time instant t. The gyroscope reading is the sum of the true angular rate $\boldsymbol{\omega}(t)$, bias $\boldsymbol{b}$ which is assumed to be constant over time, and an uncorrelated zero-mean white Gaussian noise vector $\boldsymbol{n}_\omega(t)$. At same, the vector of accelerometer reading $\tilde{\boldsymbol{a}}(t)$ along $x, y, z-$axis could be modeled by a sum of the gravitational acceleration $\boldsymbol{g}(t)$, external acceleration $\boldsymbol{a}(t)$ and an uncorrelated zero-mean white Gaussian noise $\boldsymbol{n}_a(t)$. In this thesis, the bias is assumed to be zero. Most digital systems operate on discrete signals IMU sensor data. The discretization of IMU sensor data is crucial to make it compatible with digital processing units, allowing seamless integration into electronic systems. By discretizing IMU sensor data, the eq. (4.3) becomes:

$$\begin{aligned}\tilde{\boldsymbol{\omega}}_t &= \boldsymbol{\omega}_t + \boldsymbol{n}_{\omega,t} \\ \tilde{\boldsymbol{a}}_t &= \boldsymbol{a}_t + \boldsymbol{g}_t + \boldsymbol{n}_{a,t}\end{aligned} \tag{4.4}$$

where the subscript index t labels the discrete time stamp in the sequence.

4.3.3 Noise characterization

The measurements collected by the IMU sensors are affected by random noise. Noise in IMU sensors can arise from various sources, including systematic errors, environmental variations, electronic interferences, and intrinsic imperfections due to the miniaturization process. The random

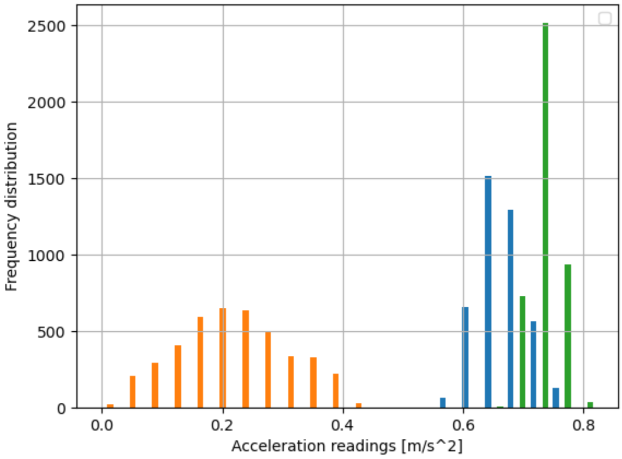


Figure 4.4. Histogram of Acceleration measurements along three axes from IMU ICM-20948. Acc-x in blue, Acc-y in orange and Acc-z in green.

noise manifests as random fluctuations in the sensor data. Previously, the measurements from the IMU sensor in the eq.(4.3) are modeled as the sum of the signal of interest, an offset, and an error. White Gaussian noise (WGN) is likely the most common stochastic model used in data processing from electronic circuits. A white Gaussian noise (WGN) $n(t)$ is characterized by a zero mean and a Gaussian distribution. The WGN exhibits a constant spectral power density for all frequencies. This attribute is advantageous in various scenarios since it suggests that the noise lacks a preference for specific frequencies or any form of correlation. White Gaussian noise in accelerometer/gyroscope data can manifest as random fluctuations around the true values. This can be observed in measurements, where the acceleration/gyroscope readings deviate slightly from the actual linear acceleration due to random noise. Figure (4.4) shows the histogram of acceleration readings obtained by the IMU sensor ICM-20948 integrated into the OS0 128 LiDAR.

The readings were recorded for condition described in Chapter (3) under normal temperature. It is apparent that the accelerations along the three axes exhibit considerable deviations from the mean values. More specifically, the mean value is 0.6699, 0.2189 and 0.7295 m/s^2 and standard deviations 0.0385, 0.0915, and 0.02246 m/s^2 for three axes x, y, z respectively. It is crucial to emphasize that the measurement error model employed for the IMU in this context originates from a sensor that remains stationary and experiences a constant temperature. Consequently, factors such as scale factor errors and bias variations induced by temperature changes, for instance, are not considered in this model.

4.3.4 Key parameters

Gyro Full Scale Range (dps)	$\pm 250; \pm 500; \pm 1000; \pm 2000$
Gyro Sensitivity (LSB/°/sec)	131; 65.5; 32.8;16.4
Accel Full Scale Range (g)	$\pm 2; \pm 4; \pm 8; \pm 16$
Accel Sensitivity (LSB/g)	16384; 8192; 4096; 2048

Table 4.1. Data sheet of IMU sensor $ICM - 20948$

The **Full-Scale Range** (FSR) refers to the maximum value that a sensor can accurately measure. It defines the range within which the gyroscope/accelerometer can provide reliable and precise measurements. The FSR of the gyroscope is typically expressed in units of degrees per second (°/s) or radians per second (rad/s), while that of the accelerometers is expressed in units of g.

The **Sensitivity** is defined as the ratio between the smallest representable value by a measuring instrument and the corresponding value of the input quantity. It is typically expressed in $LSB/unit$.

4.3.5 Calibration of IMU and LiDAR sensors

In a system where multiple sensors work, all sensors must provide data in the same reference frame. Employing the same reference frame simplifies the integration of data from different sensors. It eliminates the need to continually convert data from one coordinate system to another, streamlining sensor fusion algorithms and facilitating accurate estimates of position and orientation.

As shown in Chapter (2), the LiDAR Coordinate Frame adheres to the Right Hand Rule convention and serves as a point cloud-centric reference

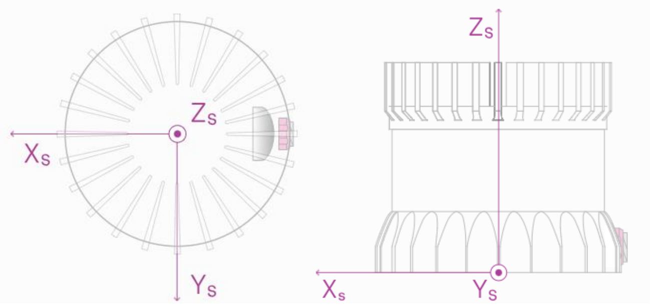


Figure 4.5. (a) Top-down view of Sensor Coordinate, (b) Side view of Sensor Coordinate Frame;from Ouster website.

frame, providing the simplest basis for the computation and manipulation of point clouds. The X-axis is oriented in the reverse direction, pointing towards the external connector, deliberately chosen so that data frames start with the azimuth angle equal to 0.

On the other hand, the Sensor Coordinate Frame also follows the Right Hand Rule convention but is a frame of reference centered on the mechanical housing, aligning with the robotics convention, and has the X-axis pointing forward; as shown in Figure (4.5). The positive X-axis of the LiDAR Coordinate Frame, marked at 0 encoder value, opposes the positive X-axis of the Sensor Coordinate Frame. This arrangement is designed to center lidar data around the positive X-axis of the Sensor Coordinate Frame. A single measurement frame initiates at the LiDAR Coordinate Frame's 0 position and concludes at the 360 position. This alignment proves advantageous for visualizing a "range image" of Ouster Sensor measurements, ensuring the "range image" is centered along the positive X-axis of the Sensor Coordinate Frame, typically oriented forward in most robotic systems.

A roto-translation matrix, also known as a rigid body transformation matrix is a $4x4$ matrix that represents both rotational and translational transformations. The roto translation matrix to transform the point cloud from the LiDAR Coordinate Frame into the sensor coordinate frame is given by:

$$M_l^s = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 38.195 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.5}$$

The IMU is slightly offset in the Sensor Coordinate Frame for practical reasons. The calibration matrix to transform imu data from imu frame into the sensor coordinate frame is given by:

$$M_i^s = \begin{bmatrix} 1 & 0 & 0 & 6.253 \\ 0 & 1 & 0 & -11.775 \\ 0 & 0 & 1 & 11.645 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.6}$$

4.4 Proposed method for attitude estimation

4.4.1 Modelling LiDAR configuration in the space

A common method describing the orientation of a rigid body in three-dimensional space involves the effective use of three angles relative to a reference frame, known as Euler angles. These angles are commonly referred to as pitch (θ), roll (ϕ), and yaw (ψ), representing the rotation around the principal axes x, y, and z. It's important to note that the yaw angle doesn't impact the LiDAR's attitude and continuously changes during a 360-degree scan around the z-axis. The LiDAR's attitude in space is solely influenced by the pitch and roll angles concerning a fixed reference system. For the purposes of this discussion, the fixed reference system convention is defined as East, North, Up (ENU).

The relationship between the coordinates $\mathbf{x}^E$ assigned to the reference system ENU and those ones assigned to a different reference system defined over the sensor S, denoted as $\mathbf{x}^S$, can be expressed as follows:

$$\mathbf{x}^E = \mathbf{R}\mathbf{x}^S \tag{4.7}$$

where $\mathbf{R}$ [1] represents the $Z-Y-X$-sequence rotation matrix. Performing various general 3D rotations in different sequences typically leads to distinct final orientations.

Notice that $\mathbf{R} \in SO(3)$, $\mathbf{R}^T\mathbf{R} = \mathbf{R}\mathbf{R}^T = \mathbf{I}$, and $det(\mathbf{R}) = 1$, where $\mathbf{I}$ is the identity matrix in the three-dimensional Euclidean space.

By utilizing the final row of matrix $\mathbf{R}$, the roll and pitch angles required to obtain the LiDAR's attitude can be represented as:

$$\phi = atan\left(\frac{R_{32}}{R_{33}}\right) \tag{4.8}$$

$$\theta = atan\left(\frac{-R_{31}}{R_{32}/sin(\phi)}\right) \tag{4.9}$$

where R_{ij} addresses the (i,j)-th element of the rotation matrix.

For the sake of convenience, the attitude of the LiDAR is preserved by using the rotation matrix form within the state vector:

$$\mathbf{x} = [R_{31}\ R_{32}\ R_{33}]^T \tag{4.10}$$

4.4.2 Kalman filter design

The employed Kalman Filter accounts for a dynamic scenario, adjusting the state vector at the time instant t based on:

$$\mathbf{x}_t = \mathbf{\Phi}_{t-1}\mathbf{x}_{t-1} + \mathbf{w}_{t-1} \tag{4.11}$$

where $\mathbf{\Phi}_{t-1}$ is the state transition matrix at time instant t-1, and $\mathbf{w}_{t-1}$ is white Gaussian noise with covariance matrix $\mathbf{Q}$.

When formulating an algorithm for rigid-body rotations, a pivotal consideration emerges: the selection of representation. Generally, Euler angles, rotation matrices, or quaternions serve as parameterizations for SO(3). However, the differential equation of a rotation matrix representa-

[1]

$$\mathbf{R} = \begin{bmatrix} cos\psi cos\theta & cos\psi sin\theta sin\phi - sin\psi cos\phi & cos\psi sin\theta cos\phi + sin\psi sin\phi \\ sin\psi cos\theta & sin\psi sin\theta sin\phi + cos\psi cos\phi & sin\psi sin\theta cos\phi - cos\psi sin\phi \\ -sin\theta & cos\theta sin\phi & cos\theta cos\phi \end{bmatrix}$$

tion is specifically chosen due to the Taylor expansion of its exponential term, aligning it with the application of a linear solution for the Kalman Filter. Additional details can be found in the reference [107]. The equation for the rotation matrix R, is given by:

$$\mathbf{R}_t = \mathbf{R}_{t-1}(\mathbf{I} + \Delta t[\boldsymbol{\omega}]_{t-1}) \tag{4.12}$$

where Δt is the sampling time, $[\,]$ denotes the skew-symmetric operator applied on $\boldsymbol{\omega}_{t-1}$ and defined as cross product matrix:

$$[\boldsymbol{\omega}]_{t-1} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}_{t-1} \tag{4.13}$$

Considering the state vector eq.(4.10) and the updated $\mathbf{R}$ matrix eq.(4.12), the eq.(4.11) becomes:

$$\mathbf{x}_t = (\mathbf{I} + \Delta t[\boldsymbol{\omega}]_{t-1})\mathbf{x}_{t-1} \tag{4.14}$$

The gyroscope-measured angular velocity is subject to noise. Therefore, by substituting (4.3) into (4.14) and considering the characteristics of the skew-matrix operator, the resulting process model equation is:

$$\mathbf{x}_t = (\mathbf{I} + \Delta t\tilde{\boldsymbol{\omega}}_{t-1})\mathbf{x}_{t-1} + \Delta t\boldsymbol{[}\mathbf{x}\boldsymbol{]}_{t-1}\boldsymbol{n}_{\omega_{t-1}} \tag{4.15}$$

So, the state transition matrix Φ_{t-1}, the white Gaussian process noise $\mathbf{w}$ and its covariance matrix $\mathbf{Q}_{t-1}$ result to be:

$$\begin{cases} \boldsymbol{\Phi}_{t-1} & = \mathbf{I}_3 + \Delta t\tilde{\boldsymbol{\omega}}_{t-1} \\ \mathbf{w}_{t-1} & = \Delta t[\mathbf{x}]_{t-1}\boldsymbol{n}_{\omega_{t-1}} \\ \mathbf{Q}_{t-1} & = -\Delta t^2[\mathbf{x}]_{t-1}\boldsymbol{\Sigma}_G^2[\mathbf{x}]_{t-1} \end{cases} \tag{4.16}$$

where $\boldsymbol{\Sigma}_G^2$ is covariance matrix of gyroscope. More details on the eq. (4.15) and (4.16) are available in the appendix section.

4.4.3 Updating measurements

The general measurement model at time step t can be expressed:

$$\mathbf{z}_t = \mathbf{H}\mathbf{x}_t + \mathbf{r}_m \tag{4.17}$$

where $\mathbf{z}_t$ is the measurement vector, $\mathbf{H}$ is the observation matrix and $\mathbf{r}_m$ is the measurement noise with a measurement noise covariance $\mathbf{M}$, assuming that this noise follows a stationary Gaussian process with zero mean.

Initially, the RANSAC algorithm is employed on the point cloud coordinates x, y, z at the specific time instant t to identify a plane and determine its corresponding normal vector $\mathbf{n}_t$, as depicted in Figure 4.6.

The RANSAC algorithm serves as a robust model-fitting technique for planes, capable of accurately fitting a model to data even in the presence of outliers. It is crucial to emphasize that the normal vector to the plane within the point cloud consistently aligns with the vertical axis of the LiDAR sensor. As illustrated in Figure (4.6), when the device has no tilt relative to E, the vertical axis of the sensor aligns with the Z-axis in E.

However, deviations in pitch (θ) or roll (ϕ) of the device concerning the reference coordinate system E result in a tilted point cloud, leading to non-zero values for all three components of the normal vector. This scenario is depicted in Figure (4.6).

The rotation matrix $\boldsymbol{R'}$ defines the relationship between the vertical z-axis, namely $\mathbf{e}_z = [0\ 0\ 1]^T$, and the normal vector with deviations $\mathbf{n} = [n_x\ n_y\ n_z]_t^T$ at the specific time instant t. This association is expressed as follows:

$$\mathbf{e}_z = \mathbf{R'}_t \frac{\mathbf{n}_t}{||\mathbf{n}_t||} \tag{4.18}$$

where $||\mathbf{n}_t||$ denotes the norm of the vector $\mathbf{n}_t$ at time instant t. By resolving the equation (4.18) and contrasting the solution $\mathbf{R'}_t$ with the result derived from equation (4.10), the process of measurement updating can be achieved by utilizing the information contained in the last column of the rotation matrix:

$$\mathbf{z}_t = \mathbf{HR'}^T_{3,l} \tag{4.19}$$

with $\mathbf{H}$ equal to $\mathbf{I}$ and for $l = 1, 2, 3$. The IMU is synchronized with the LiDAR and both provide measurements at discrete times k, as in Figure

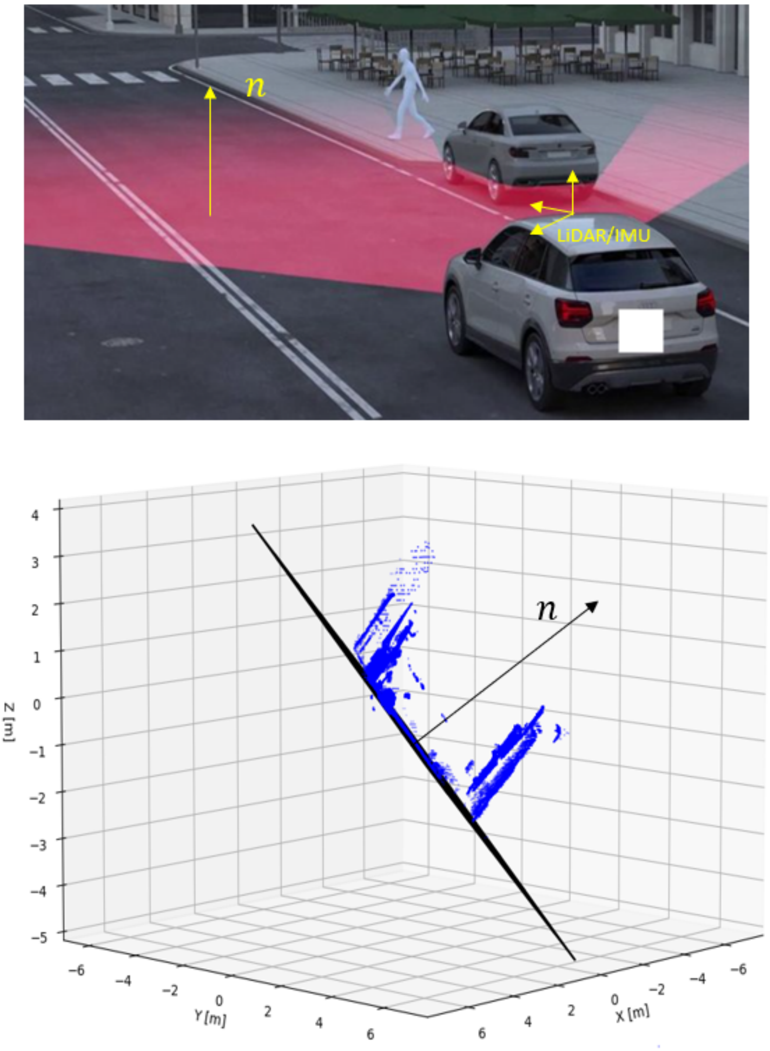


Figure 4.6. (a) For a LiDAR coordinate system S orientated as Earth reference system E in yellow, the xy plane of the S, E aligns with ground while the normal to the ground with z-axis. (b) Point cloud collected by a LiDAR on tilt: the normal to the ground shows a tilt respect with z-axis of E system.

(4.7). Typically, the IMU sensor generates a higher number of samples per second compared to the LiDAR sensor.

The measurement obtained from the LiDAR is repeatedly utilized in the filter procedure until a new point cloud is received according to the scheme in Figure (4.7).

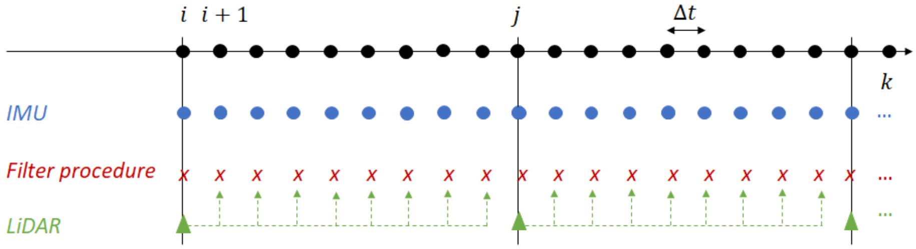


Figure 4.7. Fusion procedure for IMU and LiDAR at different rates.

4.4.4 Filtering procedure

The Kalman Filter procedure using the above process and measurement models, can be summarized as follows. The $-$ superscript denotes the *a-priori* estimate, the $+$ superscript denotes the *a-posteriori* estimate, $\mathbf{P}$ denotes the error covariance matrix, $\mathbf{K}$ the Kalman Gain, $\mathbf{x}_t$ is the state vector of the eq.(4.10) at time stamp t and, $\mathbf{\Phi}_{t-1}$ is the state transition matrix of the eq.(4.16) at time stamp $t-1$. More details of Kalman filter are outside the scope of this article. The steps are summarized as follow:

1. Compute *a-priori* state estimate

$$\mathbf{x}_t^- = \mathbf{\Phi}_{t-1}\mathbf{x}_{t-1}^+ \tag{4.20}$$

2. Compute *a-priori* error covariance matrix

$$\mathbf{P}_t^- = \mathbf{\Phi}_{t-1}\mathbf{P}_{t-1}^+\mathbf{\Phi}_{t-1}^T + \mathbf{Q}_{t-1} \tag{4.21}$$

3. Import the measurement:

   - Apply Ransac to the point cloud at time t for estimating the $||\mathbf{n}_t||$;

- Resolve the eq.(4.18) to find $\mathbf{R}'$;
- Update the $\mathbf{z}_t = \mathbf{R}'_{3,-}$ according with eq.(4.19);

4. Compute the Kalman gain

$$\mathbf{K}_t^- = \mathbf{P}_t^- \mathbf{H}^T (\mathbf{H}\mathbf{P}_t^- \mathbf{H}^T + \mathbf{M}) \tag{4.22}$$

5. Compute *a-posteriori* state estimate

$$\mathbf{x}_t^+ = \mathbf{x}_t^- + \mathbf{K}_t(\mathbf{z}_t - \mathbf{H}\mathbf{x}_t^-) \tag{4.23}$$

6. Compute *a-posteriori* error covariance matrix

$$\mathbf{P}_t^+ = (\mathbf{I}\mathbf{K}_t\mathbf{H})\mathbf{P}_t^- \tag{4.24}$$

As additional information, when the roll and pitch are known, the rotation matrix $\mathbf{R}$ shows an unknown in the *yaw* angle which can be obtained by placing it equal to $\mathbf{R}'$.

4.5 Results of attitude estimation

4.5.1 Simulation results

In order to assess the algorithm's effectiveness, sensor data are simulated utilizing the Driving Scenario Designer tool in MATLAB 2023. Driving Scenario Designer App in MATLAB is a tool designed for simulating and visualizing scenarios related to autonomous driving.
The one provides a user-friendly graphical interface, Figure (4.8), for creating and modifying scenarios that involve vehicles, sensors, and the environment. It is a part of the MATLAB Automated Driving Toolbox, which is used for developing perception, planning, and control algorithms for autonomous vehicles. Users can create realistic driving scenarios by adding roads, vehicles, traffic signs, and other objects. The app supports 2D and 3D visualization. Vehicle dynamics, such as speed, acceleration, trajectories and turning rates, can be defined to simulate realistic vehicle movements within the scenario. Finally, the app allows the placement and

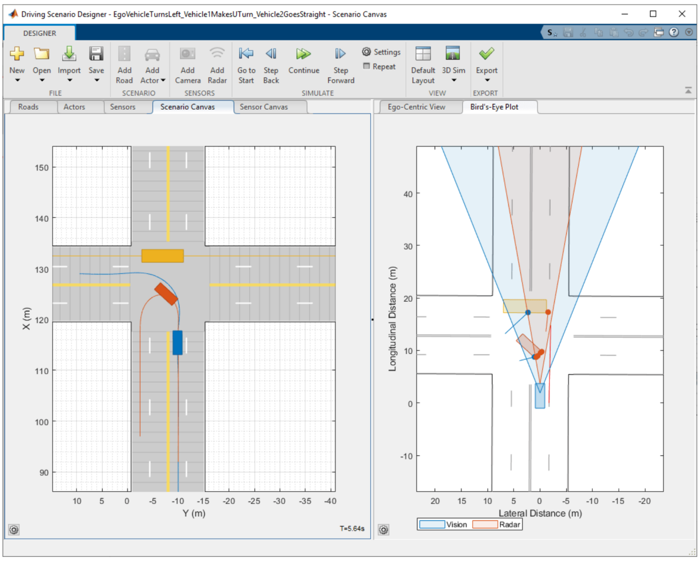


Figure 4.8. Graphical interface of Driving Scenario Designer app in Matlab

configuration of various sensors, including imu, gps, radar, lidar, and cameras, on vehicles to simulate sensor readings and data.

In all drawn scenarios, a vehicle equipped with INS and LiDAR sensors has been positioned. The sensor configuration adopts default parameters, including an IMU sampling frequency of 100 Hz, acceleration and angular velocity accuracy set at $0.2m^2/s$ and $0.2°/s$ respectively, LiDAR rate frequency at 10 Hz, azimuth resolution at 0.16, elevation resolution at 1.25°, range accuracy at $0.002m$, maximum range at $100m$, and the introduction of noise to sensor measurements.

To evaluate the accuracy of the proposed method, a total of twenty tests with an average duration of about 10 s are performed using different road scenarios. The roll and pitch angles of the LiDAR and IMU sensors mounted on a car are randomly chosen, while the speed of both sensors is varied for each test. The first two tests simulate the static sensors by setting the speed to 0 m/s, respectively. Subsequently, the synthetic sensor data are exported and processed using Python Anaconda.

The effectiveness of the proposed method is evaluated by computing the root mean square (RMS) error between the true and estimated attitude for each test. Table 4.2 shows (RMS) error obtained for each test with different speeds. The proposed method uses only a gyroscope from IMU and a LiDAR for estimation hence its estimate is independent of any movement that might be exerted by the LiDAR. Table (4.2) summarizes the root mean square (RMS) error in the attitude estimation for all performed tests. RSM error for roll takes a maximum value of 0.0439 [°] , while RSM error for pitch takes a maximum value of 0.1937 [°] . The pitch estimation accuracy also depends upon the accuracy of the roll estimate.

Table 4.2. Results of the tests under various dynamic conditions

Test	Speed [m/s]	RMSe pitch [°]	RMSe roll [°]
1	0	0.0660	0.0033
2	0	0.0111	0.0205
3	10	0.0068	0.0086
4	20	0.0587	0.0200
5	30	0.0580	0.0319
6	40	0.0270	0.0083
7	50	0.0269	0.0553
8	60	0.0220	0.0144
9	70	0.0090	0.0033
10	80	0.0017	0.0035
11	90	0.0768	0.0204
12	100	0.0091	0.0146
13	110	0.0891	0.0439
14	120	0.0746	0.0089
15	130	0.0407	0.0137
16	140	0.1937	0.0145
17	150	0.0270	0.0142
18	180	0.0118	0.0031
19	200	0.0069	0.0442
20	300	0.0589	0.0380

4.5.2 Preliminary experimental results

The Inertial Measurement Unit (IMU) sensor ICM-20948, integrated within the Ouster LiDAR, was employed. The previous section provides a detailed description and outlines the sensor characterization. Finally, calibration was performed between the mentioned IMU and LiDAR sensors, following the procedure described in Chapter (2), to align the data into a unified reference system named 'Sensor'. The experiment is carried out on an urban road in the metropolitan city of Naples (Italy), characterized by high traffic density during office hours, as illustrated in Chapter (2).

The LiDAR is mounted on a tripod at a height of 1.5 m, positioned in a parking slot at the side of the road. The one is placed randomly by hand along the x- and y-axis to simulate the roll and pitch inclinations fully, Figure (3.10).

The LiDAR Ouster is integrated with a low-cost IMU, which introduces a consistent bias in gyroscope measurements. This bias is visually evident in Figure (4.9)a, where a noticeable vertical shift is observed in the mean of a set of sensor readings compared to the zero values. Consequently, the dedicated procedure proposed in the literature by [108] is implemented to estimate and compensate for the gyroscope bias. To address this issue, the data obtained from the IMU are combined with a linear Kalman filter to accurately estimate the initial bias present in the gyroscope measurements. The vector $\boldsymbol{b}$ is initialized to zero value, and the variance of the noisy readings is used to define the covariance matrices.

By leveraging the filtering capabilities of the Kalman filter, an estimation of the bias is obtained, allowing for its subsequent subtraction from the incoming data. This process results in the generation of a corrected output that reflects the true values, Figure (4.9)b, enhancing the overall accuracy and reliability of the gyroscope system's measurements.
For the bootstrap of the proposed method, the initial value of the state vector $\boldsymbol{x}$ is chosen unitary and the noise process matrix $\boldsymbol{Q}$ is obtained according to the equation (4.16) where the gyroscope covariance matrix is chosen diagonal with value equal to 0.3 $(degrees/s)^2$ for all coefficients. Instead, the measurement noise covariance matrix $\boldsymbol{M}$ is chosen diagonal with value equal to 0.1 for all coefficients. The error covariance matrix $\boldsymbol{P}$ is chosen diagonal with a value of 0.5 for all coefficients. These values are

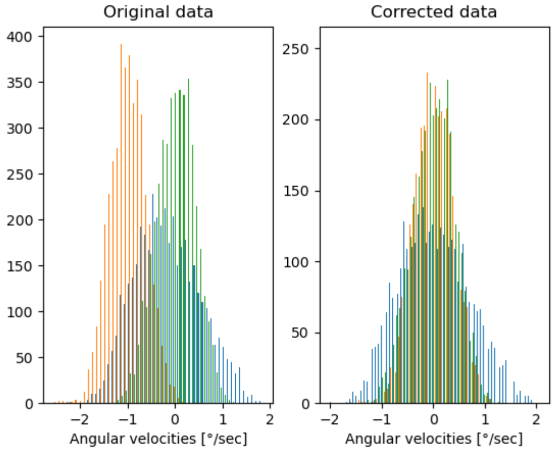


Figure 4.9. Histogram of the gyroscope reading along three axes xyz before and after bias filtering

chosen experimentally.

Since the LiDAR is set in a static position and it is therefore not affected by external acceleration, the internal IMU measures the sole internal acceleration, i.e. the gravitational contribution. The pitch and roll angles with respect to the gravitational acceleration component are used as reference in the comparison with the results offered by the proposed method. However, the acceleration measurements from IMUs are corrupted by noise, which significantly degrades the accuracy of the estimated reference parameters.

Figure (4.10a) shows the accelerations along the three axes. It is apparent that the accelerations along the three axes exhibit considerable deviations from the mean values. More specifically, the mean value 0.6699, 0.2189 and 0.7295 m/s^2 and standard deviations 0.0385, 0.0915, and 0.02246 m/s^2 for three axes x, y, z respectively. In this context, a low-pass filter can be employed to mitigate the effects of noise. Rapid signal changes are indicative of noise in the high-frequency components. As depicted in Figure (4.10b), the interesting band in the frequency spectrum of acceleration $A(f)$ lies within the range of $0.04Hz$. Butterworth low pass filter is the best choice for flat response in the passband and an ade-

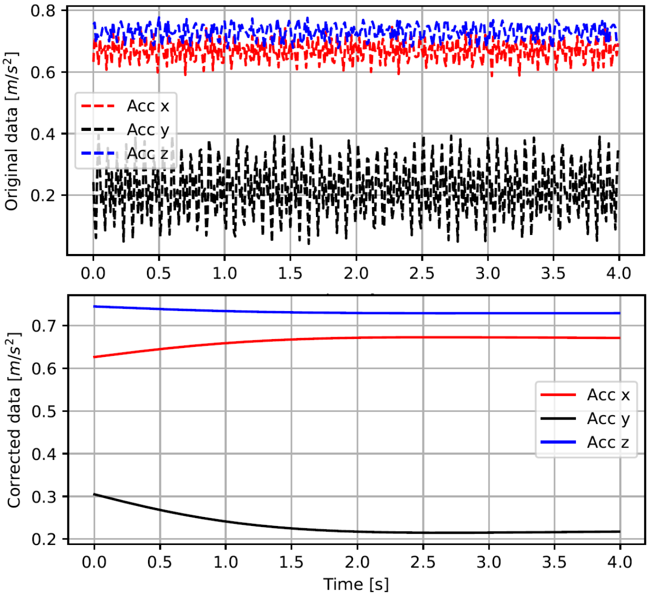


Figure 4.10. Part of the accelerometer readings before and after low-pass filtering.

quate rate of roll-off. Figure (4.10)c shows the filtered signals along three components. The cutoff frequency used for filtering is $0.1Hz$ with a filter order of 2. After the filtering, the acceleration signals are characterized by mean values of $0.6693, 0.2201$, $0.7298m/s^2$ and standard deviations of $0.0044, 0.0087$, and 0.0017 $\frac{m}{s^2}$ for three axes x, y, z respectively.

The proposed method in the presence of the experimental data produces for roll and pitch angles RMS errors equal to 1.306 degrees and 1.470 degrees respectively. The pitch estimation accuracy also depends upon the accuracy of the roll estimate. The performance of the proposed model remains constant at a steady state during very long periods and is shown in Figure (4.11).

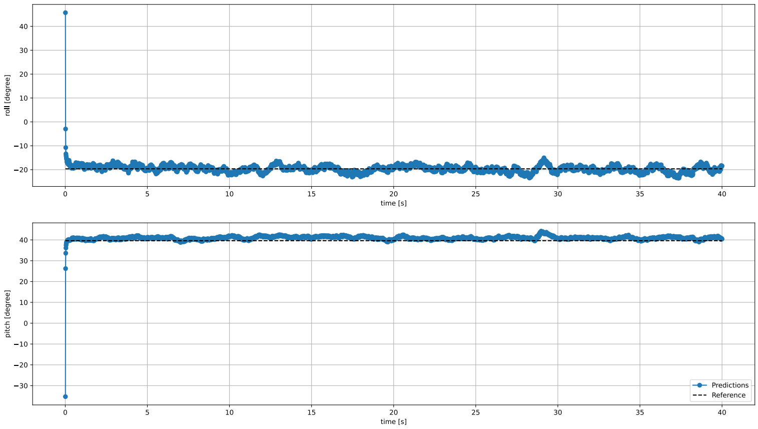


Figure 4.11. In blue, the estimation results. In black, reference values obtained from gravity acceleration using the corrected accelerometer readings

4.6 Proposed data fusion method for kinematic quantities

4.6.1 Problem Statement

In the context of Advanced Driver Assistance Systems (ADAS), determining the position of a vehicle is crucial for enhancing safety, improving the driving experience, and enabling various advanced functionalities. Several reasons why accurately knowing the position of a vehicle is important in an ADAS system, exist. Knowing the precise position of a vehicle allows the ADAS system to monitor the proximity to other vehicles, obstacles, pedestrians, and road infrastructure. This information is critical for implementing collision avoidance features, such as automatic emergency braking, lane departure warning, and forward collision warning. Another example could be Traffic Sign Recognition. Positional data is used to interpret and respond to traffic signs accurately. For example, the system needs to know the vehicle's position to determine the applicable speed limits, traffic regulations, and other road signs.

The position signal can be measured directly using multiple dedicated sensors, or, in theory, using a single sensor. Sensor fusion for position refers to the integration of data from multiple sensors to obtain a more accurate and reliable estimate of the position of an object, such as a vehicle or a robot. This approach is employed because individual sensors may have limitations, biases, or errors that can be mitigated by combining their outputs. Generally, integrating sensors that capture diverse kinematic data is a more practical approach because each sensor comes with distinct limitations. In instances where one sensor encounters yields unreliable data due to factors such as noise or environmental conditions, the presence of other different sensors becomes instrumental in preserving a dependable position estimate.
Furthermore, when position, velocity, and acceleration signals are directly measured, the results can be processed to facilitate comparisons and/or averages. For instance, if the first sensor measures the vehicle's actual position, the second measures its velocity, and the third measures its acceleration, an enhanced velocity measurement can theoretically be obtained by averaging it with the time derivative of the position and the time integral of the acceleration, incorporating the initial value of the velocity for the latter.

In a broader context, the measurement of a target's position can be measured by combining the output of more sensors, namely $y_l(n), l = 1, 2, ..., M$, where n is the discrete-time variable referring to a common time base and M is the number of sensors. This approach incorporates redundant observations of the kinematic quantities. The discrete signals representing the position $p(n)$, velocity $v(n)$, and acceleration $a(n)$, within the proposed data fusion methodology, are derived by filtering the existing digital signals, $y_l(n)$, using filters with an impulse response $q_l(n)$, and then linearly combining the resulting outputs, $w_l(n)$. These filters are tailored to the specific kinematic quantity being addressed. For instance, in the case of position measurement, the formulation would be:

$$p(n) = \sum_{l=1}^{M} c_l w_l^p(n) = \sum_{l=1}^{M} c_l \sum_{m=-\infty}^{\infty} y_l(m) q_l^p[(m-n)] \tag{4.25}$$

where the coefficients c_l adopted in the weighted average have to satisfy the constraint:

$$\sum_{l=1}^{M} c_l = 1 \tag{4.26}$$

and can be determined by considering the reliability and accuracy performance of the respective sensor. Enhanced velocity and acceleration signals can be derived directly from the position signal by applying the first-order and second-order discrete-time derivatives, respectively. Alternatively, a more sophisticated processing approach, utilizing the same model outlined in eq.(4.25) with specific filters, is also feasible and is referred to as $q_l^v(n)$ and $q_l^a(n)$.

4.6.2 System Architecture

In the analyzed scenario, kinematic quantities of the target are measured with a multi-channel system where each channel integrates a sensor capable of generating a digital output and an accompanying digital filter. Each sensor is characterized by its distinct frequency response $S_l(f)$, $l = 1, ..., M$ even when addressing the same kinematic parameter. During the calibration stage, the frequency responses of all sensors are measured, and this information becomes crucial for the formulation of the data fusion approach. To effectively illustrate the proposed data fusion methodology and specifically highlight the identification of the supplementary digital filters, it proves advantageous to introduce a reference architecture. This architecture serves as an abstraction of the actual system, maintaining analog signals throughout. The representation becomes possible by describing the analog-to-digital conversion as an analog sampling process followed by the introduction of additive noise, known as quantization noise.

Within this architectural framework, the sampled signals are filtered through analog filters characterized by impulse responses that are the analog versions of the digital filters required for the data fusion. The signals are combined by means of a summing device, which reconstructs the input signal from the samples available on the channels. Following this adopted abstraction, a consistent input representing one of the kinematic quantities is uniformly applied to all channels. The operation of channels hosting sensors targeting kinematic quantities different from the input is then described by incorporating time-integral or time-derivative operators at the front end of the sensor. Figure (4.12) depicts the block diagram of a four-channel system designed for processing a position signal as the input quantity. In addition to the position sensor, there are a velocity sensor and two acceleration sensors. To describe the system comprehensively, a frequency-domain representation is employed. In this representation, the familiar algebraic notations for the first-order and second-order time derivative operators denoted as $j2\pi f$ and $-4\pi^2 f^2$ respectively, are adopted. The sensors and analog processing filters exhibit frequency responses denoted as $S_l(f)$and $Q_l(f)$, where $l = 1, \ldots, M$, respectively, and the common input is $X(f)$. Each channel is characterized by a frequency response $H_l(f)$, formed by combining the frequency response of the operator with that of the sequentially connected sensor. Consequently, the input signals to the sampler can be expressed as $H_l(f)X_l(f)$, where l = 1, . . . , M. If the input signal corresponds to velocity or acceleration, different block diagrams need to be considered. Specifically, if the velocity signal is designated as the common input $X(f)$, the position sensor will be positioned in the diagram following the algebraic operator $\frac{1}{j2\pi f}$, and the acceleration sensors will follow $j2\pi f$. Conversely, if $X(f)$ represents the acceleration signal, then the position sensor will be situated after $-\frac{1}{4\pi^2 f^2}$, and the velocity sensor will follow $\frac{1}{j2\pi f}$.

4.6.3 Filter identification

The proposed data fusion approach necessitates identifying the frequency response of the filters $Q_l(f)$, and the implementation of the corresponding discrete-time impulse responses, $q_l(n)$. These responses can be obtained by performing the inverse discrete Fourier transform of the $2L$-length sequence, $Q_l(k)$. The identification goal can be accomplished by exploiting the generalized sampling expansion theorem, according to which a band-limited signal $x(t)$, with single-sided bandwidth B, can be acquired using a linear multi-channel system with M independent channels operating at a rate not less than $f_s = \frac{2B}{M}$. Acquiring the signals means that a digital representation of the input characterized by a sample rate at least $Mf_s = 2B$ can be gained, without any loss of information, by combining the data streams from the independent channels.

It's important to note that the generalized sampling expansion enables

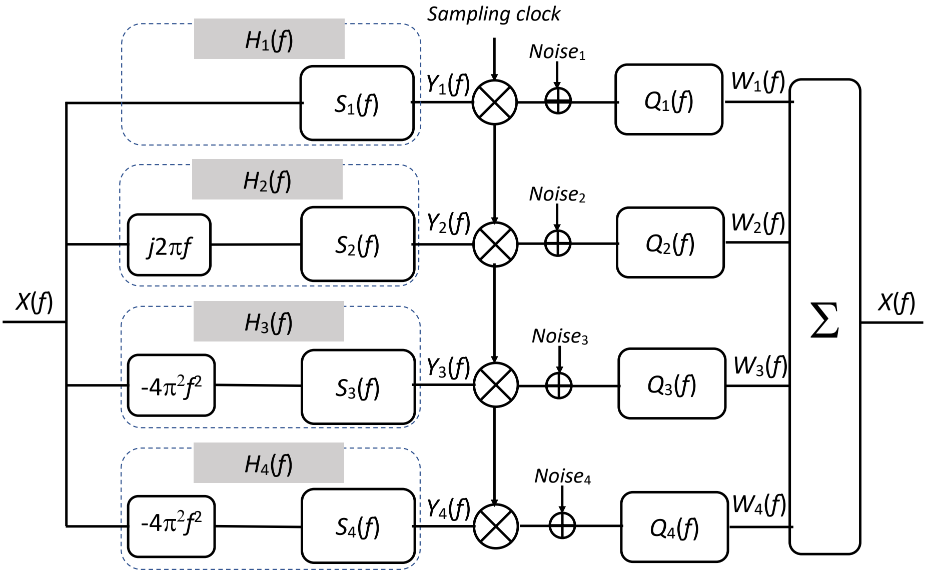


Figure 4.12. Block diagram of a multi-channel system with 4 sensors. The input quantity is a position signal, and there are a velocity sensor and 2 acceleration sensors, characterized in terms of their frequency responses $S_l(f)$. The operators $j2\pi f$ and $-4\pi^2 f^2$ stand at the front-end of the velocity and acceleration sensors to transform the position input into velocity and acceleration signals. The common input is $X(f)$ and represents the spectrum of a position signal; each channel is described in terms of a frequency response $H_l(f)$. The outputs of the sensors are sampled and degraded by (quantization) noise before entering the filters $Q_l(f)$ required to combine the outputs of the individual channels.

the integration of digital outputs from individual sensors, even if they, when considered separately, operate below the Nyquist rate. This is conditional on the cumulative number of samples per unit of time from all available sensors being equal to or greater than the Nyquist rate.

Consequently, the individual sensors within the multi-channel architecture can benefit from an extended measurement time between consecutive samples, leading to enhanced reading accuracy. Combining the outputs of all sensors results in a digital representation of the targeted kinematic parameter with a sample rate greater than or equal to the corresponding Nyquist rate.

In particular, when adopting a frequency-domain representation, the following identity can be asserted:

$$X(f) = \frac{M}{f_s} \sum_{l=1}^{M} \sum_{k=-\infty}^{\infty} H_l\left(f - k\frac{f_s}{M}\right) X\left(f - k\frac{f_s}{M}\right) Q_l(f) \tag{4.27}$$

where $X(f)$ is the input signal; $H_l(f)$ and $Q_l(f)$ with $l = 1, \ldots, M$ are the frequency responses of the channels and filters, respectively. Notice that sampling the output of the sensors $H_l(f)X(f)$ at a rate $\frac{f_s}{M}$ produces replication in the frequency domain at a pace $\frac{f_s}{M}$, as described by the infinite sum upon the index k.

This identity can be refined since the filters in (4.27) must have zero gain outside the bandwidth $(-B, B)$, because of the band-limitedness of the signal on the left side of the equation. This allows the inner sum to be limited to the terms picked by k ranging from $-M + 1$ up to $M - 1$.

Furthermore, due to the periodic structure of the argument of the sum, characterized by period $\frac{f_s}{M}$, the set of the $2M - 1$ terms can be further divided into subsets that provide equation systems for the reconstruction filters that are valid on different portions of the interval $(-B, B)$. Specifically, if the interval $(-B, B)$ is partitioned into the M adjacent intervals $\left(-B + p\frac{f_s}{M}, -B + (p+1)\frac{f_s}{M}\right)$ obtained for $p = 0, \ldots, M - 1$, and the characteristic function of each, namely $\pi_p(f) = \pi_{\left(-B+p\frac{f_s}{M}, -B+(p+1)\frac{f_s}{M}\right)}(f)$, which equals one in the subscript interval and zero outside, is used as a multiplying operator in Equation (4.27), one obtains:

$$X(f)\pi_p(f) = \frac{M}{f_s} \sum_{k=p-(M-1)}^{p} \sum_{l=1}^{M} H_l\left(f - k\frac{f_s}{M}\right) Q_{lp}(f) X\left(f - k\frac{f_s}{M}\right) \tag{4.28}$$

where the function $Q_{lp}(f)$ provides the representation of the reconstruction filter $Q_l(f)$ valid in the p-th interval.

Equation (4.28) implies that, for any fixed p, and with k ranging from

$p-(M-1)$ up to p,

$$\sum_{l=1}^{M} H_l\left(f-k\frac{f_s}{M}\right)Q_{lp}(f)=\frac{M}{f_s}\pi_p(f)\delta(k) \tag{4.29}$$

where $\delta(k)$ is the Kronecher delta function, equal to one for $k=0$ and zero for $k\neq 0$.

Equation (4.29) defines the functions $Q_{lp}(f)$, as solutions of the system:

$$\begin{bmatrix} H_1(f-(p-(M-1))\frac{f_s}{M}) & \dots & H_M(f-(p-(M-1))\frac{f_s}{M}) \\ \vdots & \ddots & \vdots \\ H_1(f-p\frac{f_s}{M}) & \dots & H_M(f-p\frac{f_s}{M}) \end{bmatrix}\begin{bmatrix} Q_{1p}(f) \\ \vdots \\ Q_{Mp}(f)\end{bmatrix}=\frac{M}{f_s}\pi_p(f)\boldsymbol{\delta}_{M-p} \tag{4.30}$$

where $\boldsymbol{\delta}_{M-p}$ is a zero vector except in the $(M-p)$-th component equal to 1.

The system can be represented in a compact form as:

$$\mathbb{H}_p\,\mathbf{Q}_p=\boldsymbol{\delta}_{M-p} \tag{4.31}$$

where the bold character is used for the vector $\mathbf{Q}_p=(Q_{1p}(f),\quad\dots,\quad Q_{Mp}(f))^T$, and the blackboard bold character for the matrix, redefined as:

$$\mathbb{H}_p=\pi_p(f)\frac{f_s}{M}\begin{bmatrix} H_1(f-(p-(M-1))\frac{f_s}{M}) & \dots & H_M(f-(p-(M-1))\frac{f_s}{M}) \\ \vdots & \ddots & \vdots \\ H_1(f-p\frac{f_s}{M}) & \dots & H_M(f-p\frac{f_s}{M}) \end{bmatrix} \tag{4.32}$$

In order to determine $\boldsymbol{Q}_p$, the determinant of the matrix in system (4.31) must be different from zero everywhere in the interval $(-B+p\frac{f_s}{M},-B+(p+1)\frac{f_s}{M})$. (It is so clarified in which sense the channels must be independent from each other as stated by the hypothesis of the theorem.)

Named $\mathbb{G}_p$, the inverse matrix of $\mathbb{H}_p$, i.e., $\mathbb{G}_p=\mathbb{H}_p^{-1}$, the vector of the reconstruction filters for the p-th interval can be represented as:

$$\mathbf{Q}_p=\mathbb{G}_p\,\boldsymbol{\delta}_{M-p} \tag{4.33}$$

It is worth highlighting that the frequency-domain functions in matrix $\mathbb{H}_p$ are rightward shifted by f_s upon any increment in p, namely $\mathbb{H}_p = \mathscr{T}_{pf_s}[\mathbb{H}_0]$, where $\mathscr{T}_{pf_s}$ represents the translation operator by pf_s. Since translation and matrix inversion commute between each other, it also holds that $\mathbb{G}_p = \mathscr{T}_{pf_s}[\mathbb{G}_0]$. Thus, the vector of the reconstruction filters valid throughout the interval $(-B, B)$ can finally be gained by computing the elements of vector $\mathbf{Q}$ as:

$$\mathbf{Q} = \sum_{p=0}^{M-1} \mathscr{T}_{pfS}[\mathbb{G}_0]\, \boldsymbol{\delta}_{M-p} \tag{4.34}$$

Hence, the time-domain representation of the signal, $x(t)$, can be gained by first filtering the sampled signals from each channel with the functions $q_l(t)$, corresponding to the inverse Fourier transformation of $Q_l(f)$, to obtain the signals $w_l(t)$, and then linearly combining all of them, in formulas:

$$\begin{cases} w_l(t) & = \sum_{n=-\infty}^{+\infty} y_l(nT_s) q_l(t - nT_s) \\ x(t) & = c_l \sum_{l=1}^{M} w_l(t) \end{cases} \tag{4.35}$$

where, for instance, if all sensors are judged equally reliable and accurate, it can be chosen $c_l = \frac{1}{M}$, $l = 1, \ldots, M$.

4.6.4 Filter Synthesis

The implementation of the proposed method requires knowledge of the frequency responses of the sensors. In practical terms, the identification of each sensor takes place in a laboratory setting through calibration tests that imply measuring the gain and phase delay characterizing the sensor at different frequencies. In detail, a uniform grid incorporating L frequencies, ranging from zero frequency to the sensor's bandwidth, is utilized. The measured gain and phase delay are represented using complex numbers, where the magnitude corresponds to the gain, and the negative angle corresponds to the phase delay. This sequence is prolonged by mirroring the measured sequence with complex conjugate values to form a $2L$-length sequence that satisfies Hermitian symmetry concerning the L-th point. The $2L$-length sequence is finally interpreted as the discrete double-sided frequency response of the sensor. This sequence represents the input data required by the proposed method, i.e., the values $H_l(\nu_k f_c)$, $l = 1, \dots, M$, where $\nu_k = \frac{k}{2L}$, $k = 0, 1, \dots, 2L-1$, and f_c is the sample rate selected for the output result.

Using the values obtained through the identification procedure in the system of Equation (4.33), and solving the algebraic systems obtained for each frequency value $\nu_k f_c$, $k = 0, 1, \dots, L$ allows us to determine the values of the double-sided frequency responses of the filters at the same frequencies, i.e., $Q_l(\nu_k f_c)$. Taking the inverse discrete Fourier transform of the latter returns the coefficients of the finite impulse responses filters, $q_l(n)$, $n = 0, 1, \dots, 2L$, needed to perform the data fusion upon the output of the sensors, according to Equation (4.25).

The proposed method requires that sensors are synchronously clocked at the same sample rate f_s, which has to be no lower than $\frac{f_c}{M}$. If the adopted sample rate is $f_s = \frac{f_c}{K}$, with $K \ \leq \ M$, the data streams from the individual sensors are first upsampled by K by interleaving $K-1$ zeros between any couples of subsequent samples and then filtered.

Concerning the choice of the filter length, some degrees of freedom are possible, taking into account that longer filters offer higher accuracy but longer transients.

4.7 Results of data fusion method for kinematic quantities

The proposed framework provides flexibility for implementing various methods, allowing different configurations in terms of the number and types of channels, redundancy levels for each kinematic quantity, and the length of digital filters for data fusion. Each method and corresponding architecture requires a dedicated analysis to assess the performance.

Here, one example related to a particular choice is discussed with the intent of highlighting how the main configuration parameters generally impact the performance. In detail, a four-channel architecture made up of one position sensor, one velocity sensor, and two redundant acceleration sensors is configured to accurately measure a position signal (refer to Figure 4.12).

All sensors exhibit low-pass behavior, but their cutoff frequencies and in-band flatness differ from each other, as depicted in Figure (4.13) and detailed in the corresponding caption. Frequency values are normalized to the system's bandwidth, ranging from 0 to 1.

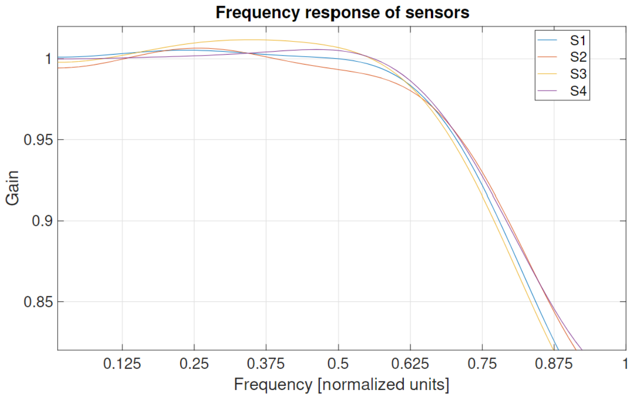


Figure 4.13. Amplitude frequency response of the adopted sensors; the values of the frequency are expressed in units normalized to the bandwidth of the system and range from 0 up to 1. All responses are characterized by 3 dB cut-off normalized frequencies that fall in the interval (0.93–0.97) with a 95% confidence level; all frequency responses grant flatness within 0.18 dB up to 0.625 normalized frequency.

Consistent with the schematic in Figure (4.12), each channel is described in terms of a frequency response incorporating the operator, inherent to the sensor's nature. Given the potential lack of independence among channels in the presence of redundant sensors, the channels' sampling operations, driven by clocks locked at the same frequency, are time-interleaved. The frequency responses of the simulated channels are presented in Figure (4.14), where the first- and second-order time derivatives are simulated using first- and second-order finite forward differences.

The simulations assume that the frequency response of each channel can be determined on a uniform frequency grid ranging from DC up to the nominal bandwidth of the channel and that the collected data are subject to uncertainty on both magnitude and phase.

The number of points on the grid and the uncertainty affecting the values play a crucial role, as they constitute the inputs required for identifying the digital filters for data fusion.

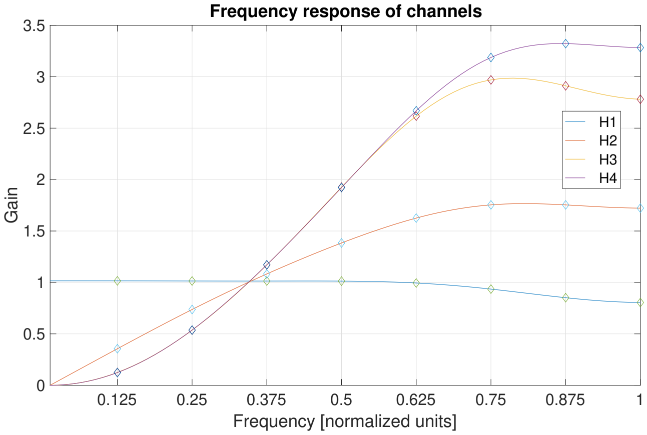


Figure 4.14. The amplitude frequency response of the 4 channels highlights the behavior of the channel hosting the position sensor (blue), the velocity sensor (red), and the acceleration sensors (yellow and magenta) within the input bandwidth. The diamond markers highlight the points considered at the identification stage.

In the simulated system, the values of the frequency response for any channel are characterized by a standard uncertainty of 0.1% for both magnitude and phase.

The set of digital filters necessary for performing data fusion is designed as outlined in the theoretical framework presented in the previous Section. These filters should ensure a unitary gain for the entire system, but a slight deviation between the actual gain and the target unitary value may occur due to the finite length of the adopted digital filters. This deviation is less significant for the frequency values corresponding to the points on the considered grid, where it is primarily a result of the uncertainty characterizing the data collected during the identification stage, rather than for the frequency values between these points.

The four digital filters utilized in the simulations have $2L = 16$ coefficients. They are designed based on the frequency response data obtained for eight equally spaced frequency points within the bandwidth of the channels. The frequency responses of the digital filters, synthesized according to the proposed approach, are illustrated in Figure (4.15).

The simulations involved 200 sinusoidal inputs with equally spaced

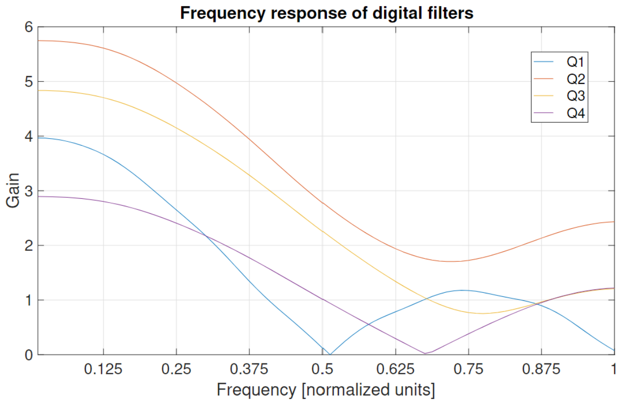


Figure 4.15. Frequency responses of the 4 digital filters that are used to perform the data fusion. All filters are characterized by $2L = 16$ coefficients.

values across the system's bandwidth. For each sinusoidal input, 1000 realizations affected by broadband noise, with a signal-to-noise ratio set at 20 dB, were analyzed.

The system's accuracy is assessed using a key performance index (KPI), defined as the ratio between the mean square value of the input and the mean square value of the signal, calculated by taking the difference between the applied input and the system's output.

Figure (4.16) illustrates the mean value of the KPI, expressed in dB units, at the test frequencies. The obtained values are interpolated for a continuous trace to infer the system's accuracy at frequencies not directly measured. The accuracy of the position measurement provided by the data fusion approach is maximized, as expected, at normalized frequency values corresponding to the points measured during the identification stage, characterized by the normalized frequency values $0.125k$, where $k = 0, \ldots, 7$. For such frequencies, the typical values of the KPI exceed 45 dB and are likely limited by the measurement uncertainty of the data used at the filter design stage.

Further details are presented in Figure (4.17), where the KPI results for a subset of analyzed frequencies are summarized using boxplots instead of mean values, as shown in Figure (4.16). In a boxplot, the central mark represents the median, the edges of the box denote the 25th and 75th per-

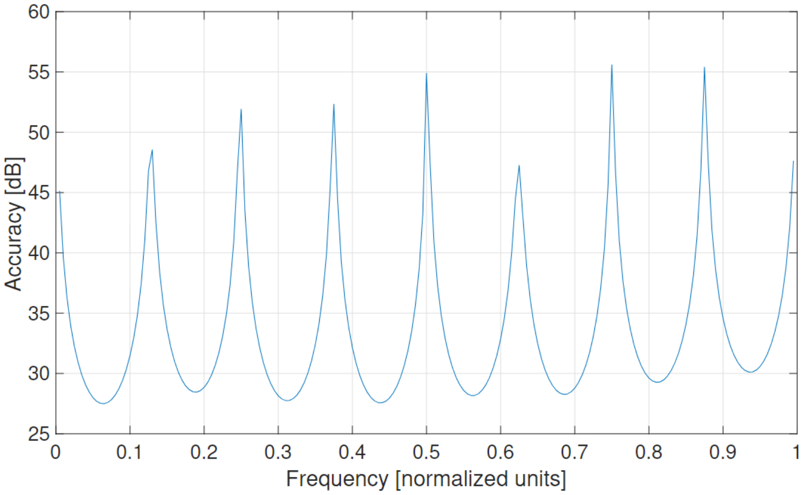


Figure 4.16. Accuracy is expressed in terms of a KPI defined as the mean value of the ratio, given in dB unit, between the mean square value of the input and the mean square value of the signal, obtained by taking the difference between the applied input and the output of the system. The values are interpolated to have a continuous trace, in order to infer the accuracy of the system at frequencies that have not been directly measured.

centiles, and the whiskers extend to the most extreme values considered non-outliers. Frequencies are selected to illustrate results for low performance (accuracy 30 dB), intermediate performance (accuracy 35 dB), and high performance (accuracy 45 dB). Lower accuracy is associated with less dispersion, while higher accuracy is accompanied by greater dispersion. These effects can be attributed to the predominant systematic error at frequencies far from the measured ones and the increasing significance of noise superimposed on the input as performance levels rise.

In the presence of non-sinusoidal test signals, such as pseudo-random trajectories with spectral contents uniformly spread throughout the sensor bandwidth, the method typically achieves an accuracy within the interval of $(33.9 - 36.2) dB$, with a mean at $35.0 dB$. This evaluation is based on Monte Carlo simulations involving 1000 pseudo-random trajectories. If the tests are repeated using the three-channel system obtained by excluding one of the acceleration sensors, the KPI estimates fall within the interval

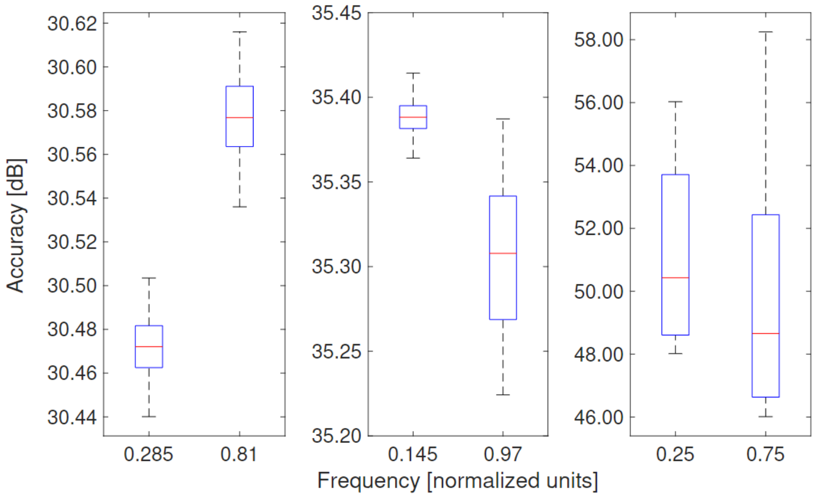


Figure 4.17. Boxplots for the KPI utilized to evaluate the accuracy of the proposed data fusion method. The values are related to a selection of the test frequencies to highlight how the dispersion of the KPI changes at different performance levels.

(31.6˘33.8)dB and exhibit a mean value equal to 32.6dB: the redundancy featuring the four-channel system allows an increment of 2.4dB for the KPI.

Intuitively, a more detailed knowledge of the frequency response of the channels, in terms of the number of measured values and their accuracy, enables the design of digital filters with more coefficients, leading to higher accuracy. For instance, in Figure (4.18), the boxplot (based on 1000 realizations) illustrates accuracy at a subset of uniformly spaced frequencies for a system employing digital filters with 400 coefficients. The design of these filters considered a set of 200 measurements characterized by 0.1% uncertainty for both magnitude and phase. The results indicate that accuracy generally improves across the bandwidth.

However, it's important to note that a filter with L coefficients introduces a response delay of LT_c seconds in the system, where T_c is the sample period. This response delay imposes limitations on the system's performance in control applications, where obtaining measurements of kinematic

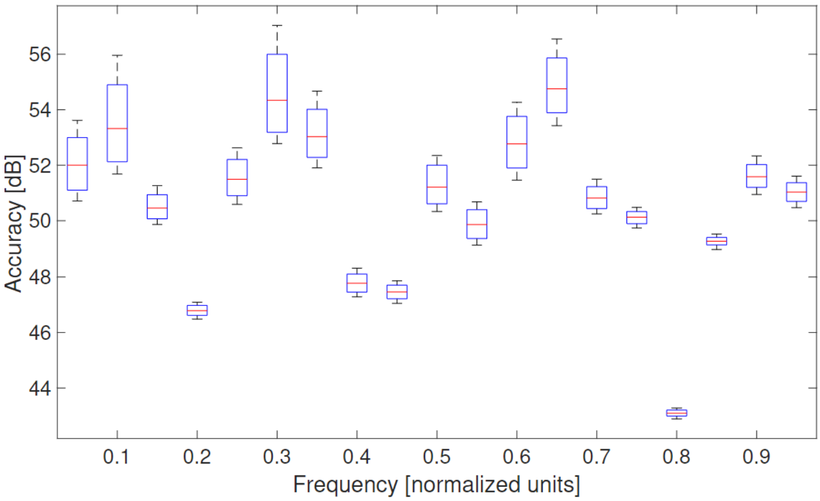


Figure 4.18. Boxplots for the KPI evaluated at normalized frequencies uniformly spaced by 0.05 units when the digital filters are characterized by length $2L = 400$.

quantities must be done swiftly to correct dynamic trajectories and minimize fluctuation effects.

4.8 Comparison with a Benchmark

The suggested approach for processing the outputs of the four sensors has been compared with a benchmark system employing the same sensors. In the benchmark system, each sensor is paired with a dedicated digital filter ensuring a calibrated acquisition channel. Each digital filter is constructed with a unitary gain at all measured frequency points. However, in contrast to the proposed method, the selection of coefficients for each filter is independent of the coefficients chosen for other sensors within the same architecture.

It is assumed that at the identification stage, the frequency response of the sensors can be measured with the same uncertainty level used to evaluate the proposed system, i.e., 0.1% for both magnitude and phase. For an equal comparison, the length of the digital filters of the benchmark system is chosen equal to $2L = 16$, namely that chosen for the proposed system. However, it is important to note that the digital filters intended for the calibration (i.e., equalization) of individual channels are distinctly different from those employed by the proposed method, as shown in Figure(4.19). The outputs of the channels of the benchmark system are further elaborated in the case of velocity and acceleration signals to obtain position signals. To this end, velocity and acceleration signals are digitally integrated once and twice, respectively; the digital integration is obtained through the cumulative sum. The four position signals are combined to obtain a single representative position signal, which is compared to that offered by the proposed system. Pseudo-random trajectories that have spectral contents uniformly spread throughout the bandwidth of the sensors are used as test signals. Figure (4.20) shows the accuracy information for the proposed method and the considered benchmark, both summarized by means of boxplots evaluated on 1000 pseudo-random trajectories. As stated, the

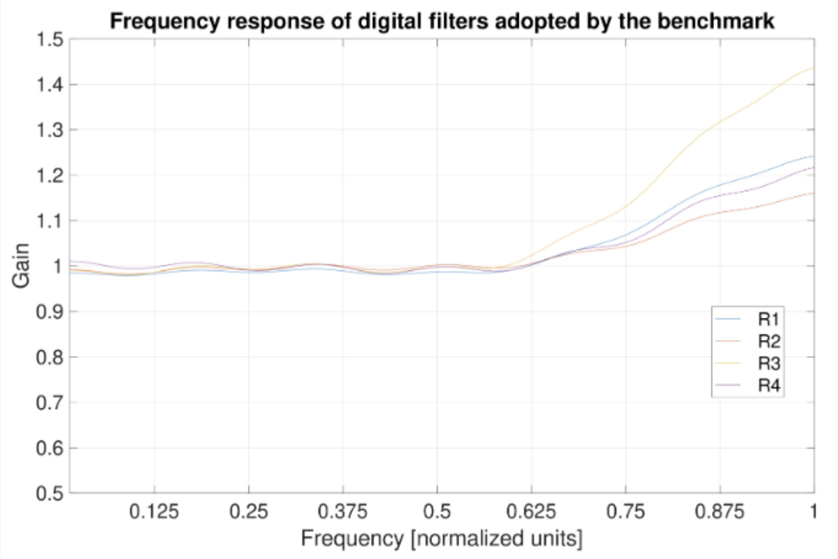


Figure 4.19. Frequency response of the digital filters adopted by the benchmark system to perform the equalization of the sensors. The frequency responses are evidently different from those used by the proposed method given in Figure (4.15).

accuracy of the proposed method is within the interval (33.9˘36.2)dB and typically sets at 35.0dB, whereas that of the benchmark is within the interval (38.1˘39.1)dB and typically sets at 38.6dB.

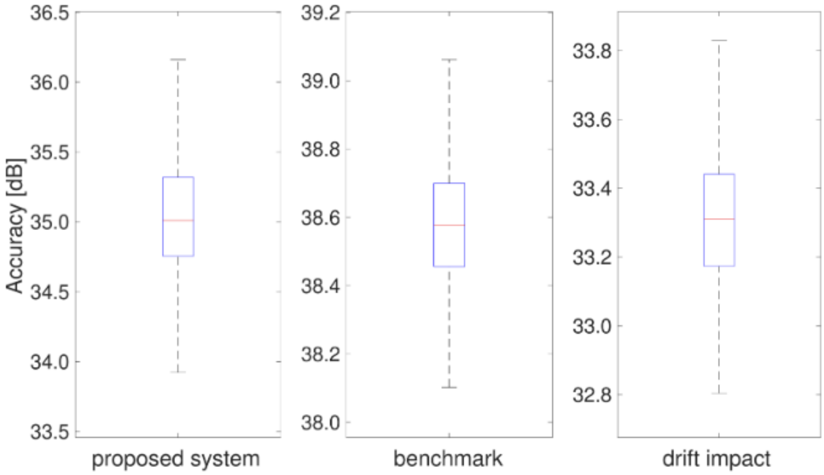


Figure 4.20. Accuracy offered by the proposed system (33.9–36.2) dB and the considered benchmark, both in ideal conditions (38.1–39.1) dB and in the presence of a constant offset equal to 1 ppm (relative to the output range of the accelerometer). The results are summarized by means of boxplots evaluated on 1000 tests performed on pseudo-random trajectories.

The benchmark system provides a theoretical limit, which is difficult, if not impossible, to achieve when stepping from simulation, where several effects can be easily controlled and neutralized, to practice. The benchmark, as well as all the systems that have to evaluate the integral of a signal, have a weak point, which consists in the use of algorithms, such as the cumulative sum, that act as unstable auto-regressive filters. These algorithms typically produce divergent outputs in the presence of offsets at the input stage. Steady or wandering offsets are very common and difficult to compensate for in kinematic sensors. At the calibration stage, it is very difficult to directly measure weak offsets that can be smaller than the resolution of the adopted meter. However, in a digital system that uses a cumulative sum, for a run that considers N consecutive samples, the offset produces a drift that reaches a peak value equal to N times the offset amount (linear drift), and a peak value equal to $\frac{1}{2}N(N+1)$ (quadratic drift) for the same run if the cumulative sum is repeated twice. To provide evidence of the issue, Figure (4.21) shows the position of a target that oscillates with respect to a rest position, as measured both by a dedicated displacement sensor and by integrating twice the output of an acceleration sensor affected by an offset equal to 1 ppm the output range of the sensor. A red trace shows the parabolic drift due to the offset, which makes the latter position measurement progressively deviate from that offered by the dedicated sensor.

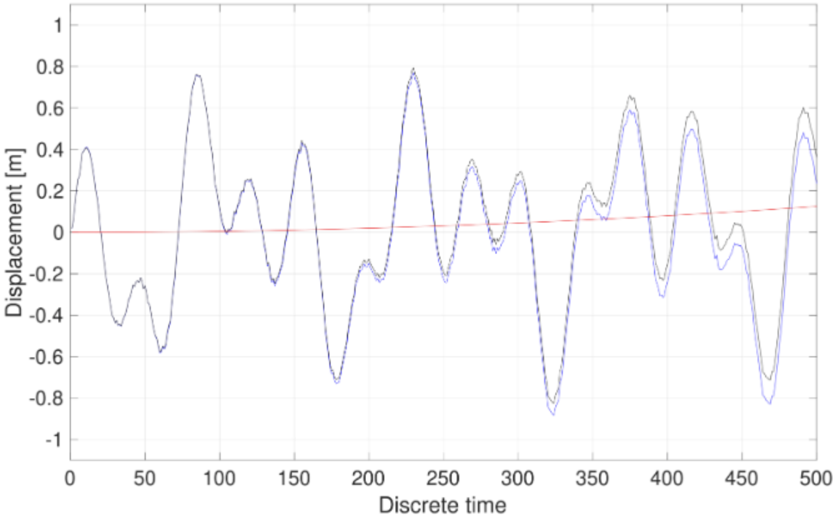


Figure 4.21. Displacement of a target around its rest position. The oscillations are simulated with a multi-frequency signal that is characterized by a peak value equal to 0.9 m, and 3 components at normalized frequencies equal to 0.0071, 0.0272, and 0.0146. The blue trace represents the output of the displacement sensor; the black trace is obtained by integrating the output of the acceleration sensor affected by an offset equal to 1 ppm (relative to the output range of the sensor); the red trace highlights the quadratic drift corresponding to the difference between the signals.

The offset can be estimated by dividing the peak deviation between the signal and the reference by the aforementioned quadratic factor. Unfortunately, static compensation is never perfect and offset residuals can show up, especially if the offset is not constant in time. The effect of a residual or unstable offset still becomes manifest as linear drift in the case of velocity sensors, and parabolic drift in the case of acceleration sensors. As discussed in the Introduction, drifts can be counteracted dynamically, if one can rely on a reliable and accurate position sensor, which does not undergo any drift. Dynamic compensation is performed seamlessly during the operation of the system, and it is repeated at a given cadence to indicate that the performance of the multi-sensor system degrades beneath an acceptance level. The choice of the duration of the time interval between the compensation actions is bounded by a trade-off: short intervals allow frequent but less effective compensation; long intervals are the basis for accurate estimations, but expose the system to poor tracking capabilities, giving the drift more time to exert its effect. To evaluate the impact of the offset on the accuracy of both multi-sensor systems, namely the proposed one and the benchmark, the results of further simulations performed in the presence of an offset at the output of the acceleration sensors are discussed. In particular, a steady offset equal to 1 ppm (relative to the output range of the acceleration sensor) is simulated. The dynamic compensation is performed any $N = 100$ samples (i.e., any 10 ms for a 10 kHz sample rate) such that the quadratic drift can reach at most a deviation equal to 0.5% the output range of the sensor, which is just detectable since it is at the level of the typical output noise. In this case, the accuracy of the benchmark degrades to values in the interval $(32.8 - 33.8)$ dB, as reported in Figure (4.20). It is worth noticing that, conversely, the proposed system adopts an inherent solution to extract position information from velocity and acceleration signals without using cumulative sums. More specifically, it uses finite impulse response filters, which are stable filters that confer finite gain to zero and low frequencies (see the frequency responses of the digital filters in Figure (4.15)), where the spectral contents of the offset reside. The impact on the accuracy of an offset equal to 1 ppm in the output range of the sensor is therefore not appreciable in the proposed system, which confirms the results shown in Figure (4.20).

4.9 Conclusion

Chapter 4 addresses the challenges of accurate vehicle localization, focusing on determining the vehicle's position and orientation within a predefined coordinate system. The discussion emphasizes the significance of attitude, introduces the Inertial Navigation System (INS) relying on Inertial Measurement Units (IMUs), and highlights the need for sensor fusion to overcome IMU data uncertainties and drifts. Moreover, the chapter explores challenges in accelerometer measurements and discusses existing methods.

A method based on the Kalman filter for determining a LiDAR's attitude (pitch and roll angles) using only a tri-axial gyroscope in an inertial measurement unit (IMU) and a point cloud, is introduced. Currently, the vehicle deploys sensor fusion solutions of gyroscope and accelerometer because of the drift problem. The accelerometer is used to correct the drift problem of gyroscope by measuring the attitude from gravitational acceleration. However, the accelerometer readings are corrupted by an external acceleration produced by the vehicle movements. The proposed solution uses the gyroscope in conjunction with the point cloud to accurately determine the attitude of the vehicle. Hence, this solution enables to avoid of the accelerometer whose measurements are corrupted by external acceleration. The proposed solution is tested by both simulations and experiments under various dynamic conditions showing accuracy in estimation even under very severe and prolonged dynamic conditions. Additionally, a theoretical framework to implement multi-sensor data fusion methods for kinematic quantities is proposed. All methods defined through the framework allow the combination of signals obtained from position, velocity, and acceleration sensors addressing the same target, and improvement in the observation of the kinematics of the target. Differently from several alternative methods, the considered ones need no dynamic and/or error models to operate and can be implemented with low computational burden. They gain measurements by summing filtered versions of the heterogeneous kinematic quantities. A simulated scenario shows that the adopted method keeps the error in a position measurement within a few ppm of the full-scale position.

Chapter 5

Conclusions

In contemporary urban landscapes, the focus on environmentally friendly public transport is gaining momentum. This shift towards sustainability is particularly pronounced in the growing interest surrounding tram systems, which emerge as zero-emission solutions complementing existing transportation modes. Trams, seamlessly integrated into urban contexts without necessitating extensive infrastructure investments, share road space with pedestrians and other vehicles, presenting a versatile and eco-friendly alternative.

However, the incorporation of trams into shared road spaces brings forth challenges, necessitating advanced technologies to ensure road safety and mitigate potential accidents. To address this, an assisted driving system tailored for trams becomes imperative. Leveraging existing automotive technology, assisted driving systems, primarily reliant on camera systems, have shown promise in enhancing safety measures.

In conclusion, this thesis has explored various aspects of LiDAR technology, focusing on its application in object detection and localization within the context of advanced driver assistance systems (ADAS) for the tramway industry. The primary research objectives are to identify, categorize, and localize objects in the surrounding environment with modern LiDAR technology and, then estimate the position of the tram relative to an unknown environment.
Chapter 1 provides an overview of the motivation behind the research, highlighting the increasing demand for sustainable transportation solu-

tions, particularly in urban centers. The focus is on the tram system as an environmentally friendly and cost-effective option. The chapter emphasizes the need for advanced driver assistance systems (ADAS) in the tram industry to enhance safety and integration within urban environments. It also discusses the current challenges faced by tram drivers, especially in complex urban scenarios.

The analysis of the urban context explores the challenges associated with tram operations, including accidents at intersections and unexpected behaviors of other road users. Statistical data on tram accidents in Italy indicates common scenarios and areas of concern. The difficulties faced by tram drivers in speed control and navigation within urban environments are discussed, emphasizing the need for effective ADAS solutions.

An overview of sensors for ADAS is presented, focusing on automotive radar, ultrasonic sensors, LiDAR, and cameras. Each sensor's characteristics, applications, and key manufacturers are discussed, providing a foundation for understanding the sensor technologies involved in ADAS.

The chapter concludes with an introduction to the thesis's objective, emphasizing the LiDAR technology and its potential application in Advanced Assisted Driving in the Tramway Sector.
Chapter 2 extensively covered LiDAR technology, elucidating its measurement principles, equation, and imaging strategies, and presenting a detailed analysis of the OUSTER OS0-128 LiDAR system.

Chapter 3 introduces the concept of autonomous driving technology, emphasizing the role of sensors in gathering real-time information for safe navigation. The chapter underscores the importance of onboard sensors such as LiDAR, cameras, and radar for autonomous vehicles but acknowledges limitations due to factors like sensor constraints, and adverse weather. To overcome these limitations, a holistic approach is proposed, integrating advanced sensing technology, edge computing, communication, and other technologies to establish a cooperative perception system within the Internet of Vehicles (IoV) environment. Collaborative perception within IoV, facilitated by vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication, is explored. V2I communication allows vehicles to share information with infrastructure sensors, enabling real-time alerts on road conditions, traffic, and more. The chapter suggests that this collaborative approach could reduce the reliance on onboard sensors, lowering the overall cost of autonomous vehicles and accelerating their commercial adoption.

Moving beyond theoretical discussions, the work of Chapter 3 transitioned into practical applications, specifically in LiDAR-based object detection. The method can detect moving objects and estimate their size, motion direction, speed, and acceleration thanks to the implementation of effective approaches, like inter-frame comparisons for background filtering, clustering for object recognition, 3D bounding box fitting for object orientation estimation, and Kalman filtering reinforced by feedback for object tracking. The method is tested on real datasets, collected on a city road with a static LiDAR mounted on a tripod, and its effectiveness is quantified with specific metrics.
Chapter 4 delves into the challenges of accurate vehicle localization, emphasizing the necessity for precision in diverse driving scenarios and environmental conditions. The primary focus is on determining the vehicle's position and orientation within a predefined coordinate system or map, involving 6 degrees of freedom (DOF) pose in three-dimensional space. The discussion extends to the significance of attitude, often represented by Euler angles or quaternions, especially in the context of autonomous vehicles. It highlights that, for motion on a flat surface, a 2D variant with the first three DOF poses is often sufficient.

It is introduced the Inertial Navigation System (INS), which relies on Inertial Measurement Units (IMUs) to continuously calculate the vehicle's position, orientation, and velocity. While INS is a reliable navigation solution, IMUs, due to their manufacturing processes, exhibit suboptimal accuracy. Strapdown equations form the foundation of INS, governing the real-time estimation of the vehicle's trajectory. However, IMU data is prone to uncertainties and noisy drifts, necessitating the integration of multiple sensors through sensor fusion techniques. The advantages of sensor fusion include enhanced accuracy, robustness to sensor failures, and mitigating the limitations of individual sensors. Common frameworks such as Kalman filters are employed for data fusion. The challenges in accelerometer measurements and the existing methods, including the limitations of relying on GPS signals, are explored. It also introduces more sophisticated approaches involving computer vision techniques and artificial neural networks, acknowledging their resource-intensive nature.

The subsequent sections of the chapter propose novel methods for improving attitude estimation using LiDAR data and introduce a theoretical framework for multi-sensor data fusion of kinematic quantities.
A methodology has been suggested to improve attitude estimation by leveraging LiDAR output, thereby circumventing the utilization of potentially uncertain acceleration data generated by the IMU. The proposed approach incorporates the *Ransac* algorithm to identify the ground plane and subsequently determine the orthogonal direction. A set of state variables crucial for attitude estimation is monitored from this information. Optimal estimations of these state variables, and consequently attitude parameters, are achieved through the application of a well-suited Kalman filter. Initial tests demonstrate promising results, particularly in terms of long-term stability.
A category of multi-sensor data fusion techniques for kinematic quantities has been illustrated. All these techniques share a common underlying framework but can be differentiated based on the number and types of available kinematic sensors. They facilitate the utilization of redundancy in homogeneous kinematic sensors and, more broadly, the amalgamation of outputs from diverse kinematic sensors.

The investigation has analyzed the potential of one such method in its fundamental implementation and has conducted a fair comparison with the most direct competing method suitable for data fusion. However, it's crucial to note that optimization was beyond the scope of this study, which aimed at defining a general framework rather than a specific method and its corresponding implementation. The four-channel architecture considered, as outlined in Section (4.6), was employed solely to illustrate how the principal configuration parameters typically influence the performance of a method derived from the proposed framework.

The most noteworthy characteristic of these methods consists in their robustness against undesired effects arising from offsets, flicker, and random walk noise. Conversely, solutions relying on time-integration processing to derive position measurements from velocity and/or acceleration signals face accuracy risks, especially in the presence of high redundancy, as even minor offsets can lead to significant errors due to cumulative effects.

In conclusion, the methods derived from the proposed framework adeptly combine direct position, velocity, and acceleration measurements. This proficiency becomes particularly evident when the position signal exhibits coarse precision, and velocity and acceleration signals encounter challenges related to offset, flicker, or random walk noise. In these scenarios, the methods within this class enable the utilization of heterogeneous sensors, ensuring position measurements with drift-free tributaries.

In summary, this thesis has made significant contributions to the understanding and practical implementation of LiDAR technology in the realm of ADAS. By addressing key aspects such as object detection and localization, the research presented here contributes to the ongoing advancements in autonomous and assisted driving systems, paving the way for enhanced safety and efficiency in future urban transportation. Future research will also delve into the exploration of data fusion algorithms, aiming to integrate information from multiple LiDARs or even heterogeneous sensors like LiDARs and RADARs. This avenue of investigation seeks to enhance the detection and tracking capabilities of the driving-assistant system.

Bibliography

[1] Gongbo Chen, Christian Wiede, and Rainer Kokozinski. Data processing approaches on spad-based d-tof lidar systems: A review. *IEEE Sensors Journal*, 21(5):5656–5667, 2020.

[2] Santiago Royo and Maria Ballesta-Garcia. An overview of lidar imaging systems for autonomous vehicles. *Applied sciences*, 9(19):4093, 2019.

[3] Ulla Wandinger. Introduction to lidar. In *Lidar: range-resolved optical remote sensing of the atmosphere*, pages 1–18. Springer, 2005.

[4] Wen Zhang, Mounir Ghogho, Baolun Yuan, et al. Mathematical model and matlab simulation of strapdown inertial navigation system. *Modelling and Simulation in Engineering*, 2012, 2012.

[5] Karima Kourtit, Peter Nijkamp, and Marina Toger. Sustainable cities, quality of life, and mobility-related happiness. In *Geography of Happiness: A Spatial Analysis of Subjective Well-Being*, pages 103–120. Springer, 2023.

[6] Lydia Habib, Ouazna Oukacha, and Simon Enjalbert. Towards tramway safety by managing advanced driver assistance systems depending on grades of automation. *IFAC-PapersOnLine*, 54(2):227–232, 2021.

[7] Lydia Habib, Ouazna Oukacha, and Simon Enjalbert. Towards tramway safety by managing advanced driver assistance systems depending on grades of automation. *IFAC-PapersOnLine*, 54(2):227–232, 2021.

[8] Yuning Wang, Zeyu Han, Yining Xing, Shaobing Xu, and Jianqiang Wang. A survey on datasets for the decision making of autonomous vehicles. *IEEE Intelligent Transportation Systems Magazine*, 2024.

[9] Chuan Liu, Xianbo Xiang, Yu Duan, Lichun Yang, and Shaolong Yang. Improved path following for autonomous marine vehicles with low-cost head-

ing/course sensors: comparative experiments. *Control Engineering Practice*, 142:105740, 2024.

[10] Anupam Chattopadhyay, Kwok-Yan Lam, and Yaswanth Tavva. Autonomous vehicle: Security by design. *IEEE Transactions on Intelligent Transportation Systems*, 22(11):7015–7029, 2020.

[11] Joanna Moody, Nathaniel Bailey, and Jinhua Zhao. Public perceptions of autonomous vehicle safety: An international comparison. *Safety science*, 121:634–650, 2020.

[12] Taşkın Dirsehan and Ceren Can. Examination of trust and sustainability concerns in autonomous vehicle adoption. *Technology in Society*, 63:101361, 2020.

[13] Jacek Szmagliński, Sławomir Grulkowski, and Krystian Birr. Identification of safety hazards and their sources in tram transport. In *MATEC Web of Conferences*, volume 231, pages 1–9, 2018.

[14] Christos N Pyrgidis. *Railway transportation systems: design, construction and operation.* CRC press, 2021.

[15] Marco Guerrieri. Correction to: Tramways in urban areas: An overview on safety at road intersections. *Urban Rail Transit*, 6(3):185–187, 2020.

[16] Laetitia Fontaine, Margarita Novales, Dominique Bertrand, and Manuel Teixeira. Safety and operation of tramways in interaction with public space. *Transportation research procedia*, 14:1114–1123, 2016.

[17] Yanchao Dong, Zhencheng Hu, Keiichi Uchimura, and Nobuki Murayama. Driver inattention monitoring system for intelligent vehicles: A review. *IEEE transactions on intelligent transportation systems*, 12(2):596–614, 2010.

[18] Anjum Naweed and Janette Rose. "it's a frightful scenario": A study of tram collisions on a mixed-traffic environment in an australian metropolitan setting. *Procedia Manufacturing*, 3:2706–2713, 2015.

[19] Catello Di Palma, Vincenzo Galdi, Vito Calderaro, and Felice De Luca. Driver assistance system for trams: Smart tram in smart cities. In *2020 IEEE International Conference on Environment and Electrical Engineering and 2020 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe)*, pages 1–6. IEEE, 2020.

[20] Salvatore Cafiso and Alessandro Di Graziano. Evaluation of the effectiveness of adas in reducing multi-vehicle collisions. *International journal of heavy vehicle systems*, 19(2):188–206, 2012.

[21] Jorge Vargas, Suleiman Alsweiss, Onur Toker, Rahul Razdan, and Joshua Santos. An overview of autonomous vehicles sensors and their vulnerability to weather conditions. *Sensors*, 21(16):5397, 2021.

[22] Jri Lee, Yi-An Li, Meng-Hsiung Hung, and Shih-Jou Huang. A fully-integrated 77-ghz fmcw radar transceiver in 65-nm cmos technology. *IEEE Journal of Solid-State Circuits*, 45(12):2746–2756, 2010.

[23] William Buller, Brian Wilson, Joseph Garbarino, Jack Kelly, Brian Thelen, Bruce M Belzowski, et al. Radar congestion study. Technical report, United States. Department of Transportation. National Highway Traffic Safety . . . , 2018.

[24] Josef Steinbaeck, Christian Steger, Gerald Holweg, and Norbert Druml. Next generation radar sensors in automotive sensor fusion systems. In *2017 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, pages 1–6. IEEE, 2017.

[25] Mubina Toa and Akeem Whitehead. Ultrasonic sensing basics. *Dallas: Texas Instruments*, pages 53–75, 2020.

[26] Seiya Nagaoka, Islam Mohammad Raihanul, Kenji Okajima, Ryoei Ito, Ken Watanabe, and Tetsu Ito. Evaluation of attenuation of ultrasonic wave in air to measure concrete roughness using aerial ultrasonic sensor. *GEOMATE Journal*, 14(42):158–163, 2018.

[27] Upma Jain, Vipashi Kansal, Ram Dewangan, Gulshan Dhasmana, and Arnav Kotiyal. Performance comparison of hc-sr04 ultrasonic sensor and tf-luna lidar for obstacle detection. In *Computer Vision and Machine Intelligence: Proceedings of CVMI 2022*, pages 631–640. Springer, 2023.

[28] Saloni Kumawat, Kuldeep Singh Rathore, Shubham Negi, and Shivangi Barthwal. Investigation and analysis of ultrasonic sensor for automatic control of garage door. In *AIP Conference Proceedings*, volume 2978. AIP Publishing, 2024.

[29] Ilaria Ciuffreda, Sara Casaccia, and Gian Marco Revel. A multi-sensor fusion approach based on pir and ultrasonic sensors installed on a robot to localise people in indoor environments. *Sensors*, 23(15):6963, 2023.

[30] Hui Zhao, Xuewu Fan, Mingyang Yang, Yue Pan, Baopeng Li, and Minrui Zhang. Super-resolution reconstruction for extremely low-light imaging by using intensified ccd or cmos camera. In *Optoelectronic Imaging and Multimedia Technology IX*, volume 12317, pages 135–145. SPIE, 2023.

[31] Ramachandra Gambheer and MS Bhat. Ccd sensor based cameras for sustainable streaming iot applications with compressed sensing. *IEEE Access*, 2023.

[32] Irina Afanasieva, Valery Murzin, Valery Ardilanov, Nikolai Ivaschenko, Maksim Pritychenko, Alexei Moiseev, Elena Shablovinskaya, and Eugene Malygin. Astronomical camera based on a ccd261-84 detector with increased sensitivity in the near-infrared. In *Photonics*, volume 10, page 774. MDPI, 2023.

[33] Hailong Liu, Hengyi Lv, Chengshan Han, and Yuchen Zhao. Vibration detection and degraded image restoration of space camera based on correlation imaging of rolling-shutter cmos. *Sensors*, 23(13):5953, 2023.

[34] Eric R Fossum. The invention of cmos image sensors: A camera in every pocket. In *2020 Pan Pacific Microelectronics Symposium (Pan Pacific)*, pages 1–6. IEEE, 2020.

[35] Eshta Ranyal, Ayan Sadhu, and Kamal Jain. Road condition monitoring using smart sensing and artificial intelligence: A review. *Sensors*, 22(8):3044, 2022.

[36] Shaohua Wan, Songtao Ding, and Chen Chen. Edge computing enabled video segmentation for real-time traffic monitoring in internet of vehicles. *Pattern Recognition*, 121:108146, 2022.

[37] Amine Kherraki and Rajae El Ouazzani. Deep convolutional neural networks architecture for an efficient emergency vehicle classification in real-time traffic monitoring. *IAES International Journal of Artificial Intelligence*, 11(1):110, 2022.

[38] Bingxu Wang, Jinhui Lan, and Jiangjiang Gao. Lidar filtering in 3d object detection based on improved ransac. *Remote Sensing*, 14(9):2110, 2022.

[39] Jie Bai, Sen Li, Han Zhang, Libo Huang, and Ping Wang. Robust target detection and tracking algorithm based on roadside radar and camera. *Sensors*, 21(4):1116, 2021.

[40] Pengfei Liu, Guizhen Yu, Zhangyu Wang, Bin Zhou, and Peng Chen. Object classification based on enhanced evidence theory: Radar–vision fusion approach for roadside application. *IEEE Transactions on Instrumentation and Measurement*, 71:1–12, 2022.

[41] Xiaobin Xu, Lei Zhang, Jian Yang, Chenfei Cao, Wen Wang, Yingying Ran, Zhiying Tan, and Minzhou Luo. A review of multi-sensor fusion slam systems based on 3d lidar. *Remote Sensing*, 14(12):2835, 2022.

[42] Ching-Pai Hsu, Boda Li, Braulio Solano-Rivas, Amar R Gohil, Pak Hung Chan, Andrew D Moore, and Valentina Donzella. A review and perspective on optical phased array for automotive lidar. *IEEE Journal of Selected Topics in Quantum Electronics*, 27(1):1–16, 2020.

[43] Kun Qian, Shilin Zhu, Xinyu Zhang, and Li Erran Li. Robust multimodal vehicle detection in foggy weather using complementary lidar and radar signals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 444–453, 2021.

[44] Bo Yang and Hua Zhang. A cfar algorithm based on monte carlo method for millimeter-wave radar road traffic target detection. *Remote Sensing*, 14(8):1779, 2022.

[45] Mahmoud Meribout, Asma Baobaid, Mohammed Ould Khaoua, Varun Kumar Tiwari, and Juan Pablo Pena. State of art iot and edge embedded systems for real-time machine vision applications. *IEEE Access*, 10:58287–58301, 2022.

[46] Luca Caltagirone, Mauro Bellone, Lennart Svensson, and Mattias Wahde. Lidar–camera fusion for road detection using fully convolutional neural networks. *Robotics and Autonomous Systems*, 111:125–131, 2019.

[47] Tianya Zhang, Peter J Jin, and Yi Ge. Multimodal gaussian mixture model for realtime roadside lidar object detection. *arXiv preprint arXiv:2204.09804*, 2022.

[48] Philipp Rosenberger, Martin Holder, Sebastian Huch, Hermann Winner, Tobias Fleck, Marc René Zofka, J Marius Zöllner, Thomas D'hondt, and Benjamin Wassermann. Benchmarking and functional decomposition of automotive lidar sensor models. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 632–639. IEEE, 2019.

[49] Brian F Aull, Andrew H Loomis, Douglas J Young, Richard M Heinrichs, Bradley J Felton, Peter J Daniels, and Deborah J Landers. Geiger-mode avalanche photodiodes for three-dimensional imaging. *Lincoln laboratory journal*, 13(2):335–349, 2002.

[50] Sergio Cova, Massimo Ghioni, Andrea Lacaita, Carlo Samori, and Franco Zappa. Avalanche photodiodes and quenching circuits for single-photon detection. *Applied optics*, 35(12):1956–1976, 1996.

[51] Desmond O'Connor. *Time-correlated single photon counting*. Academic press, 2012.

[52] Jan Nissinen, Ilkka Nissinen, Sahba Jahromi, Tuomo Talala, and Juha Kostamovaara. Time-gated cmos spad and a quantum well laser diode with a cmos driver for time-resolved diffuse optics imaging. In *2018 IEEE Nordic Circuits and Systems Conference (NORCAS): NORCHIP and International Symposium of System-on-Chip (SoC)*, pages 1–4. IEEE, 2018.

[53] KQK Nguyen, EMD Fisher, AJ Walton, and I Underwood. An experimentally verified model for estimating the distance resolution capability of direct time of flight 3d optical imaging systems. *Measurement Science and Technology*, 24(12):125001, 2013.

[54] Sung-You Tsai, Yu-Cheng Chang, and Tzu-Hsien Sang. Spad lidars: Modeling and algorithms. In *2018 14th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT)*, pages 1–4. IEEE, 2018.

[55] National Research Council et al. *Laser radar: progress and opportunities in active electro-optical sensing.* National Academies Press, 2014.

[56] Thomas E Laux and Chao-I Chen. 3d flash lidar vision systems for imaging in degraded visual environments. In *Proc. SPIE*, volume 9087, page 908704, 2014.

[57] Reuben Rohrschneider, James Masciarelli, Kevin L Miller, and Carl Weimer. An overview of ball flash lidar and related technology development. In *AIAA Guidance, Navigation, and Control (GNC) Conference*, page 4642, 2013.

[58] Keshav 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. In *2015 12th international conference on informatics in control, automation and robotics (ICINCO)*, volume 1, pages 191–198. IEEE, 2015.

[59] Henry Alexander Ignatious, Manzoor Khan, et al. An overview of sensors in autonomous vehicles. *Procedia Computer Science*, 198:736–741, 2022.

[60] Yuxiao Zhang, Alexander Carballo, Hanting Yang, and Kazuya Takeda. Perception and sensing for autonomous vehicles under adverse weather conditions: A survey. *ISPRS Journal of Photogrammetry and Remote Sensing*, 196:146–177, 2023.

[61] Faran Awais Butt, Jawwad Nasar Chattha, Jameel Ahmad, Muhammad Umer Zia, Muhammad Rizwan, and Ijaz Haider Naqvi. On the integration of enabling wireless technologies and sensor fusion for next-generation connected and autonomous vehicles. *IEEE Access*, 10:14643–14668, 2022.

[62] Darsh Parekh, Nishi Poddar, Aakash Rajpurkar, Manisha Chahal, Neeraj Kumar, Gyanendra Prasad Joshi, and Woong Cho. A review on autonomous vehicles: Progress, methods and challenges. *Electronics*, 11(14):2162, 2022.

[63] Junhyek Jang, Janghyun Baek, Kitaeg Lim, Yongcheol Ro, Sanghun Yoon, and Soohyun Jang. A study on v2i based cooperative autonomous driving. In *2023 International Conference on Electronics, Information, and Communication (ICEIC)*, pages 1–3. IEEE, 2023.

[64] Lili Miao, Shang-Fu Chen, Yu-Ling Hsu, and Kai-Lung Hua. How does c-v2x help autonomous driving to avoid accidents? *Sensors*, 22(2):686, 2022.

[65] Anastasios Giannaros, Aristeidis Karras, Leonidas Theodorakopoulos, Christos Karras, Panagiotis Kranias, Nikolaos Schizas, Gerasimos Kalogeratos, and Dimitrios Tsolis. Autonomous vehicles: Sophisticated attacks, safety issues, challenges, open topics, blockchain, and future directions. *Journal of Cybersecurity and Privacy*, 3(3):493–543, 2023.

[66] Xuting Duan, Hang Jiang, Daxin Tian, Tianyuan Zou, Jianshan Zhou, and Yue Cao. V2i based environment perception for autonomous vehicles at intersections. *China Communications*, 18(7):1–12, 2021.

[67] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[68] Bin Xue and Ningning Tong. Diod: Fast and efficient weakly semi-supervised deep complex isar object detection. *IEEE Transactions on Cybernetics*, 49(11):3991–4003, 2019.

[69] Yutian Wu, Yueyu Wang, Shuwei Zhang, and Harutoshi Ogai. Deep 3d object detection networks using lidar data: A review. *IEEE Sensors Journal*, 21(2):1152–1171, 2020.

[70] Georgios Zamanakos, Lazaros Tsochatzidis, Angelos Amanatiadis, and Ioannis Pratikakis. A comprehensive survey of lidar-based 3d object detection methods with deep learning for autonomous driving. *Computers & Graphics*, 99:153–181, 2021.

[71] Yingjuan Tang, Hongwen He, Yong Wang, Zan Mao, and Haoyu Wang. Multi-modality 3d object detection in autonomous driving: A review. *Neurocomputing*, page 126587, 2023.

[72] Danfeng Hong, Naoto Yokoya, Gui-Song Xia, Jocelyn Chanussot, and Xiao Xiang Zhu. X-modalnet: A semi-supervised deep cross-modal network

for classification of remote sensing data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 167:12–23, 2020.

[73] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[74] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

[75] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)*, 38(5):1–12, 2019.

[76] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8895–8904, 2019.

[77] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7652–7660, 2018.

[78] Martin Simony, Stefan Milzy, Karl Amendey, and Horst-Michael Gross. Complex-yolo: An euler-region-proposal for real-time 3d object detection on point clouds. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pages 0–0, 2018.

[79] Hongyang Li, Chonghao Sima, Jifeng Dai, Wenhai Wang, Lewei Lu, Huijie Wang, Jia Zeng, Zhiqi Li, Jiazhi Yang, Hanming Deng, et al. Delving into the devils of bird's-eye-view perception: A review, evaluation and recipe. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[80] Jorge Beltrán, Carlos Guindel, Francisco Miguel Moreno, Daniel Cruzado, Fernando Garcia, and Arturo De La Escalera. Birdnet: a 3d object detection framework from lidar information. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3517–3523. IEEE, 2018.

[81] Alejandro Barrera, Carlos Guindel, Jorge Beltrán, and Fernando García. Birdnet+: End-to-end 3d object detection in lidar bird's eye view. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6. IEEE, 2020.

[82] Jie Zhou, Xin Tan, Zhiwen Shao, and Lizhuang Ma. Fvnet: 3d front-view proposal generation for real-time object detection from point clouds. In *2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pages 1–8. IEEE, 2019.

[83] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4490–4499, 2018.

[84] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018.

[85] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7652–7660, 2018.

[86] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 770–779, 2019.

[87] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks for action recognition in videos. *IEEE transactions on pattern analysis and machine intelligence*, 41(11):2740–2755, 2018.

[88] Jianqing Wu, Hao Xu, Yuan Sun, Jianying Zheng, and Rui Yue. Automatic background filtering method for roadside lidar data. *Transportation Research Record*, 2672(45):106–114, 2018.

[89] Bin Lv, Hao Xu, Jianqing Wu, Yuan Tian, and Changwei Yuan. Raster-based background filtering for roadside lidar data. *IEEE Access*, 7:76779–76788, 2019.

[90] Takashi Ogawa, Hiroshi Sakai, Yasuhiro Suzuki, Kiyokazu Takagi, and Katsuhiro Morikawa. Pedestrian detection and tracking using in-vehicle lidar for automotive application. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 734–739. IEEE, 2011.

[91] Martin Brossard, Silvere Bonnabel, and Axel Barrau. Denoising imu gyroscopes with deep learning for open-loop attitude estimation. *IEEE Robotics and Automation Letters*, 5(3):4796–4803, 2020.

[92] Qiuying Wang, Kaiyue Liu, and Zhongyi Cao. System noise variance matrix adaptive kalman filter method for auv ins/dvl navigation system. *Ocean Engineering*, 267:113269, 2023.

[93] Roberta Veloso Garcia, Paula Cristiane Pinto Mesquita Pardal, HK Kuga, and MC Zanardi. Nonlinear filtering for sequential spacecraft attitude estimation with real data: Cubature kalman filter, unscented kalman filter and extended kalman filter. *Advances in Space Research*, 63(2):1038–1050, 2019.

[94] Mokhamad Nur Cahyadi, Tahiyatul Asfihani, Ronny Mardiyanto, and Risa Erfianti. Performance of gps and imu sensor fusion using unscented kalman filter for precise i-boat navigation in infinite wide waters. *Geodesy and Geodynamics*, 14(3):265–274, 2023.

[95] Jung Keun Lee and Mi Jin Choi. Effect of strapdown integration order and sampling rate on imu-based attitude estimation accuracy. *Sensors*, 18(9):2775, 2018.

[96] Hamad Ahmed and Muhammad Tahir. Accurate attitude estimation of a moving land vehicle using low-cost mems imu sensors. *IEEE Transactions on Intelligent Transportation Systems*, 18(7):1723–1739, 2016.

[97] Soyoung Park, Jungmin Park, and Chan Gook Park. Adaptive attitude estimation for low-cost mems imu using ellipsoidal method. *IEEE Transactions on Instrumentation and Measurement*, 69(9):7082–7091, 2020.

[98] Batu Candan and Halil Ersin Soken. Robust attitude estimation using imu-only measurements. *IEEE Transactions on Instrumentation and Measurement*, 70:1–9, 2021.

[99] Soyoung Park, Jungmin Park, and Chan Gook Park. Adaptive attitude estimation for low-cost mems imu using ellipsoidal method. *IEEE Transactions on Instrumentation and Measurement*, 69(9):7082–7091, 2020.

[100] Jung Keun Lee, Edward J Park, and Stephen N Robinovitch. Estimation of attitude and external acceleration using inertial sensor measurement during various dynamic conditions. *IEEE transactions on instrumentation and measurement*, 61(8):2262–2273, 2012.

[101] M Tanenhaus, D Carhoun, T Geis, E Wan, and A Holland. Miniature imu/ins with optimally fused low drift mems gyro and accelerometers for applications in gps-denied environments. In *Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium*, pages 259–264. IEEE, 2012.

[102] Sibel Canaz Sevgen and Fevzi Karsli. An improved ransac algorithm for extracting roof planes from airborne lidar data. *The Photogrammetric Record*, 35(169):40–57, 2020.

[103] Kwansik Park, Woohyun Kim, and Jiwon Seo. Effects of initial attitude estimation errors on loosely coupled smartphone gps/imu integration system. In *2020 20th International Conference on Control, Automation and Systems (ICCAS)*, pages 800–803. IEEE, 2020.

[104] Lu Xiong, Xin Xia, Yishi Lu, Wei Liu, Letian Gao, Shunhui Song, and Zhuoping Yu. Imu-based automated vehicle body sideslip angle and attitude estimation aided by gnss using parallel adaptive kalman filters. *IEEE Transactions on Vehicular Technology*, 69(10):10668–10680, 2020.

[105] Cheng Zhou, Hengcheng Xu, Lieyun Ding, Linchun Wei, and Ying Zhou. Dynamic prediction for attitude and position in shield tunneling: A deep learning method. *Automation in Construction*, 105:102840, 2019.

[106] Quentin Mourcou, Anthony Fleury, Céline Franco, Frédéric Klopcic, and Nicolas Vuillerme. Performance evaluation of smartphone inertial sensors measurement for range of motion. *Sensors*, 15(9):23168–23187, 2015.

[107] Jung Keun Lee and Mi Jin Choi. Effect of strapdown integration order and sampling rate on imu-based attitude estimation accuracy. *Sensors*, 18(9):2775, 2018.

[108] Inchara Lakshminarayan and Divya Rao. Kalman filter based estimation of constant angular rate bias for mems gyroscope. In *Proceedings of the IEEE TechSym 2014 Satellite Conference, Vellore, India*, pages 7–8, 2014.

Author's publications

1. M. D'Arco, L. Fratelli, G. Graber and M. Guerritore, "Detection and Tracking of Moving Objects Using a Roadside LiDAR System," in IEEE Instrumentation & Measurement Magazine, vol. 27, no. 1, pp. 49-56, February 2024, doi: 10.1109/MIM.2024.10423660.

2. D'Arco, Mauro, and Martina Guerritore. Multi-Sensor Data Fusion Approach for Kinematic Quantities. Energies 15.8 (2022): 2916.

3. M. D'Arco, M. Guerritore. A new method to estimate the attitude of a LiDAR through IMU and point-cloud data analysis. IEEE I2MTC – International Instrumentation and Measurement Technology Conference 2024. (Accepted)

4. D'Arco, Mauro, Martina Guerritore, and Annarita Tedesco. Application Scenarios for Gait Analysis with Wearable Sensors and Machine Learning. International Conference on 25th IMEKO TC4 Symposium and 23nd International Workshop on ADC and DAC Modelling and Testing (IWADC). DOI: 10.21014/tc4-2022.37