



Politecnico di Torino

College of Computer Engineering, Cinema and Mechatronics

Master of Science in Mechatronic Engineering

Academic Year 2022/2023

MODELLING AND SIMULATION OF A SMART OBSTACLE DETECTION SENSOR FOR A BIKE OR SCOOTER WITH MODEL-BASED DESIGN

Supervisor

Prof. Massimo Violante

Candidate:

Enrico Guglielmi

Co-Supervisors

Ing. Albero Bertone

Ing. Antonio Arena

Ing. Jacopo Filippi

Table of content

Abstract.....	4
Sommario	5
Outline	6
1 Introduction	12
1.1 Micromobility	12
1.2 Electric kick scooter: state of art.....	14
1.3 Safety devices	16
1.4 POC actual state.....	21
1.4.1 Fundamentals.....	22
1.4.2 Sensor selection	24
1.4.3 Hardware design.....	26
1.4.4 Software design	27
1.4.5 Prototype tests results and conclusions	30
1.5 Feasibility study.....	31
2 Improvement of the prototype	36
2.1 Concept	36
2.2 Hardware and software improvements	37
2.2.1 STM32F4 Nucleo-144 board.....	38
2.2.2 Gimbal	38
2.2.3 Wiring and connections	39
2.2.4 Acquisition algorithm.....	46
3 Finalizing and testing	65
3.1 System's refinement	65
3.2 Calibration	67

3.3	Testing and evaluation	68
3.3.1	Simulation models	70
3.3.2	Free road	82
3.3.3	Obstacle recognition	90
3.4	Tests results and optimizations	109
3.4.1	Validating the results	109
3.4.2	Test results analysis and optimizations	114
4	Conclusions and future works	117

Abstract

This thesis has been developed within the framework of technologies for autonomous vehicles; it is connected, in particular, to the Advanced Driver-Assistance Systems (ADAS), real-time multi-sensor collision avoidance systems field of research.

Advanced Driver-Assistance Systems (ADAS) are groups of electronic technologies that assist drivers in driving actions and their aim is to improve vehicle safety during critical driving conditions by measuring external vehicle variables (such as road and traffic environment conditions). Examples of ADAS are Adaptive Cruise Control (ACC) or Lane Keeping Assist (LKA).

The main objective of the thesis is to improve the potential of an existent proof of concept (POC) sensor, for vehicles such as bike and scooter, able to correctly detect objects or road irregularities in real-time.

This task can be divided into two different but connected phases: the first one regards the improvements of the measurement system; the second one is about the device's hardware and software testing, useful to verify the theorized results previously obtained.

The measurement campaign demonstrated not only the feasibility of the project but also the potential of the new device's characteristics, which let the user to benefit from a more precise and reliable safety system.

Sommario

I contenuti di questa tesi sono stati sviluppati nell'ambito delle tecnologie per veicoli autonomi; sono connessi, in particolare, al campo dei sistemi avanzati di assistenza alla guida (ADAS), con peculiare attenzione ai sistemi real-time multi-sensore per la prevenzione delle collisioni.

I sistemi avanzati di assistenza alla guida (ADAS) sono un insieme di tecnologie atte ad assistere il guidatore durante le manovre di guida e ad migliorare la sicurezza in condizioni di guida pericolosa. Esplicano la loro azione per mezzo della misurazione di variabili esterne al veicolo (come le condizioni ambientali e il traffico stradale). Esempi di ADAS sono il Cruise control adattativo (ACC) e il Sistema di avviso di deviazione dalla corsia (LKA).

L'obiettivo principale di questa tesi è il miglioramento di un esistente prototipo (POC) di sensore per veicoli del tipo bicicletta/monopattino capace di rilevare correttamente ostacoli e irregolarità della strada in tempo reale.

Il compito quindi può essere suddiviso in due differenti ma connesse fasi: la prima riguarda il miglioramento del sistema di misurazione; la seconda concerne la fase di testing di hardware e software del dispositivo utile per verificare i risultati teorici precedentemente ottenuti.

La campagna di misurazione ha dimostrato non solo la fattibilità del progetto ma anche la potenzialità delle nuove caratteristiche del dispositivo, le quali permettono all'utilizzatore di usufruire di un sistema di sicurezza più preciso e affidabile.

Outline

Chapter 1 presents the general aspects of the micromobility scenario. Firstly, a wide overview of the characteristic vehicles of the sector will be presented, focusing in particular on the bike and scooter's current state of art. Next, the problem of safety is discussed, explaining the reasons that led to the conception of the POC in the first place. After that, the POC's hardware and software features are summarized, as well as its functioning. Finally, the feasibility study is shown.

Chapter 2 precisely describes the POC's hardware design and software programming improvements made in order to obtain a more reliable, scalable and robust product. In particular, solutions to the low computation power and vibration problems will be discussed.

Chapter 3 covers all of the stages required for testing the prototype features, from the preparation phase of the procedure through the achieved results.

Chapter 4 is a summary of the main results obtained during the thesis activities, conclusions and future work.

1 Introduction

The background and motivation for this thesis will be presented in this introductory chapter, beginning with an overview of the micromobility scenario, focusing in particular on the state of the art bike/scooter and related safety devices, and progressing to the POC's current state and feasibility.

1.1 Micromobility

Micromobility is a new method of transportation that makes use of micromobility vehicles such as e-bikes and scooters. In general, it refers to a class of lightweight vehicles that travel at speeds usually less than 25km/h and are driven by the users themselves. These devices can be manually or electrically powered, and they can be individually owned or available through a shared fleet [8].



Figure 1.0. Micromobility vehicle fleet (from left to right, e-kick scooter, e-bicycle and e-moped)

Although original forms of micromobility have existed since 1817 [8], their actual global spread has occurred in recent years as a result of two pertinent circumstances.

The first is about environmental issues caused by CO₂ emissions from urban transportation. Because of the growing issue of air pollution, many authorities have created specific urban areas where vehicles with high emissions are prohibited from driving as a temporary measure to reduce their negative effects on public health. In this manner, authorities hope to be compliant to the World Health Organization's recommended maximum emission amounts. This is the primary reason why many companies have started to invest more in the electric mobility industry, allowing for the rapid and strong implementation of eco-sustainable light mobility, particularly in large metropolitan areas. In fact, an electric vehicle is the only type of vehicle that is assured to circulate, regardless of emissions-based driving restrictions [9].

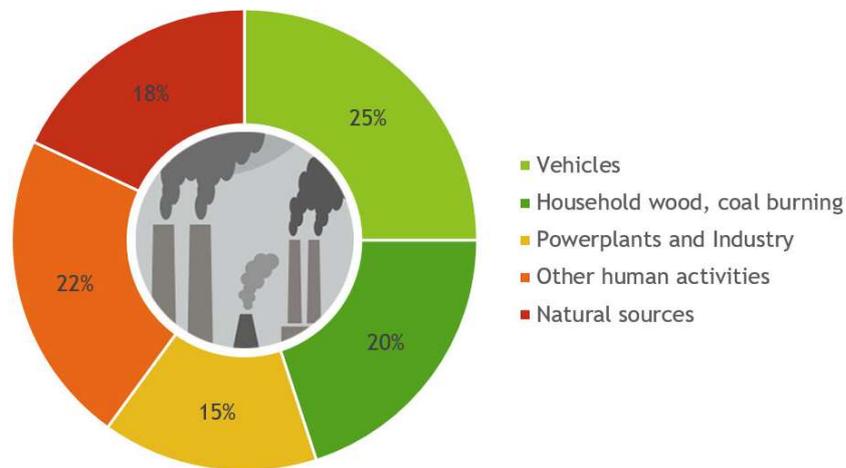


Figure 1.1. Global urban ambient source distribution PM_{2.5}. Vehicles make for 25% of total PM_{2.5} production worldwide.

The second is about the pandemic outbreak of novel coronavirus disease 2019 (COVID-19), which has caused radical social changes around the world as a significant threat to the population's health, life, and livelihood. In such a case, the epidemic definitely had an impact on micromobility, both positively and negatively [10]. On the one hand, in order to maintain social distance, an increasing number of people chose to remain at home during the pandemic period, implying that the number of trips made by micromobility would intuitively decrease. People's concern for their health, on the other hand, may have

increased their willingness to use micromobility transportation modes for short or medium-distance journeys instead of using public transportation in order to reduce the risk of contracting coronavirus disease. For example, when states reopened following the first lockdown, people in Italy preferred individual or personal type of vehicles rather than take public transportation. Personal vehicles offered a sense of security at the time [2, 3, 11].

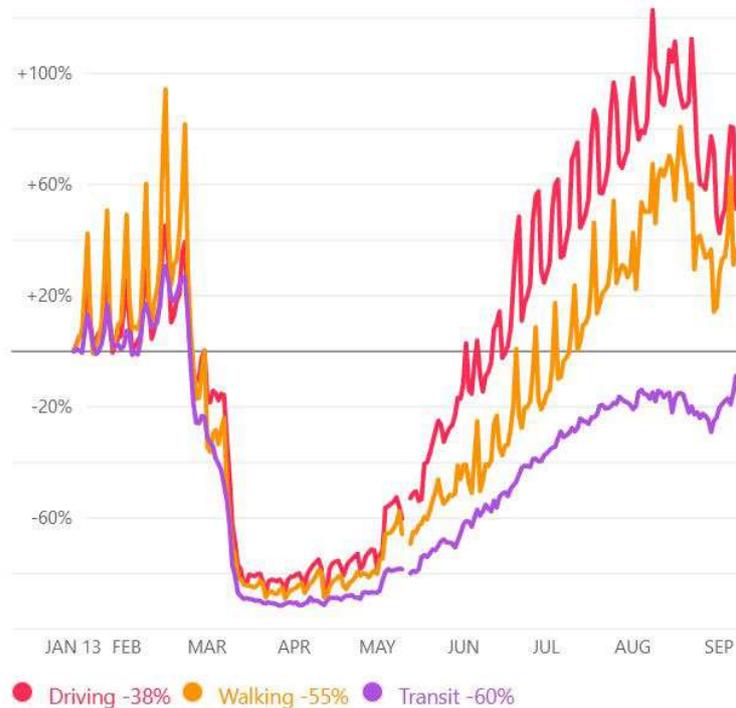


Figure 1.2. Mobility trends: changes in routing requests from January 13, 2020 to September 13, 2020

1.2 Electric kick scooter: state of art

Electric scooters are one of the pillars of post-lockdown transportation. The efficiency of these vehicles, coupled with their low maintenance costs and social distancing characteristics, increased both the rates of purchases and sharing services. Shared electric kick scooters, in particular, are thought to have one of the fastest adoption rates among micromobility transit modes [12, 13].



Figure 1.3. The Xiaomi Mi M365 is Italy's best-selling electric bike in 2019

Several key factors contributed to this [12]:

- the number of GPS-enabled smartphones has more than doubled in the last decade;
- the development of electric motors small enough to be mounted on a traditional scooter;
- the development of more efficient and smaller batteries;
- most cities are experiencing increased traffic congestion, making it faster to commute short distances on a bike or scooter;
- the total amount of private funding for this kind of project.

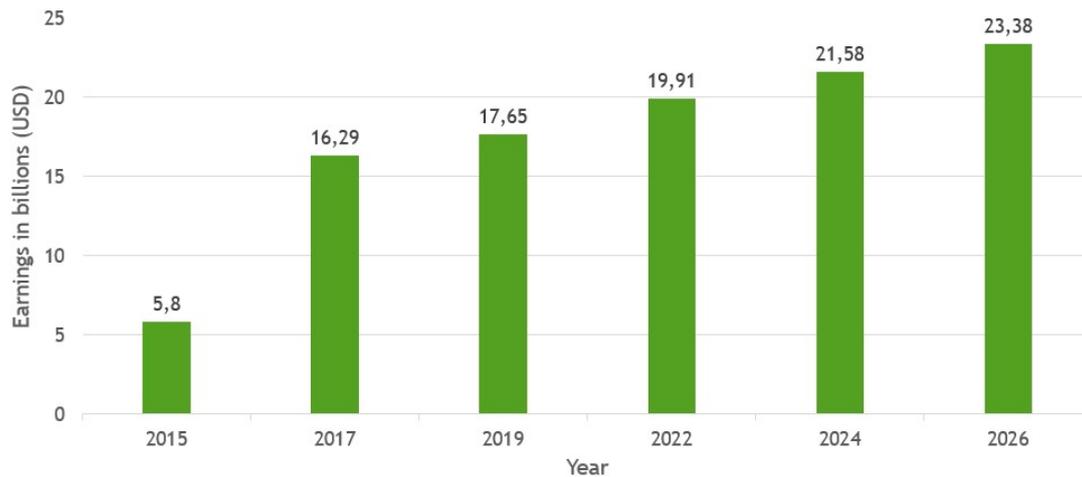


Figure 1.4. Global electric scooter industry forecast

In Italy, the electric scooter saw an initial wave of adoption in 2010, followed by a second, much more consistent wave in 2018 as a result of different bicycle sharing firms that began providing electric scooter sharing services, increasing the vehicle's popularity among the general public. The sharing service of electric scooters was made available in the rest of the world beginning in 2018, in line with the spread in Italy [14]. In particular, an electric scooter with less than 500W of power and less than 25km/h of speed is considered a bicycle under Italian legislation [17].

1.3 Safety devices

Electric scooters are an ecologically friendly alternative personal mode of transportation that appeals to city dwellers who travel short distances. However, as the popularity of electric two-wheeled vehicles has grown, so has the number of accidents involving them. These hazards range from stationary impediments like sidewalks and road pavement cracks to moving hazards like people and other vehicles, which can pose a much more unpredictable risk. Because of the unstable nature of the vehicle, also autonomous falls are a particular danger to be considered [4, 15, 16].

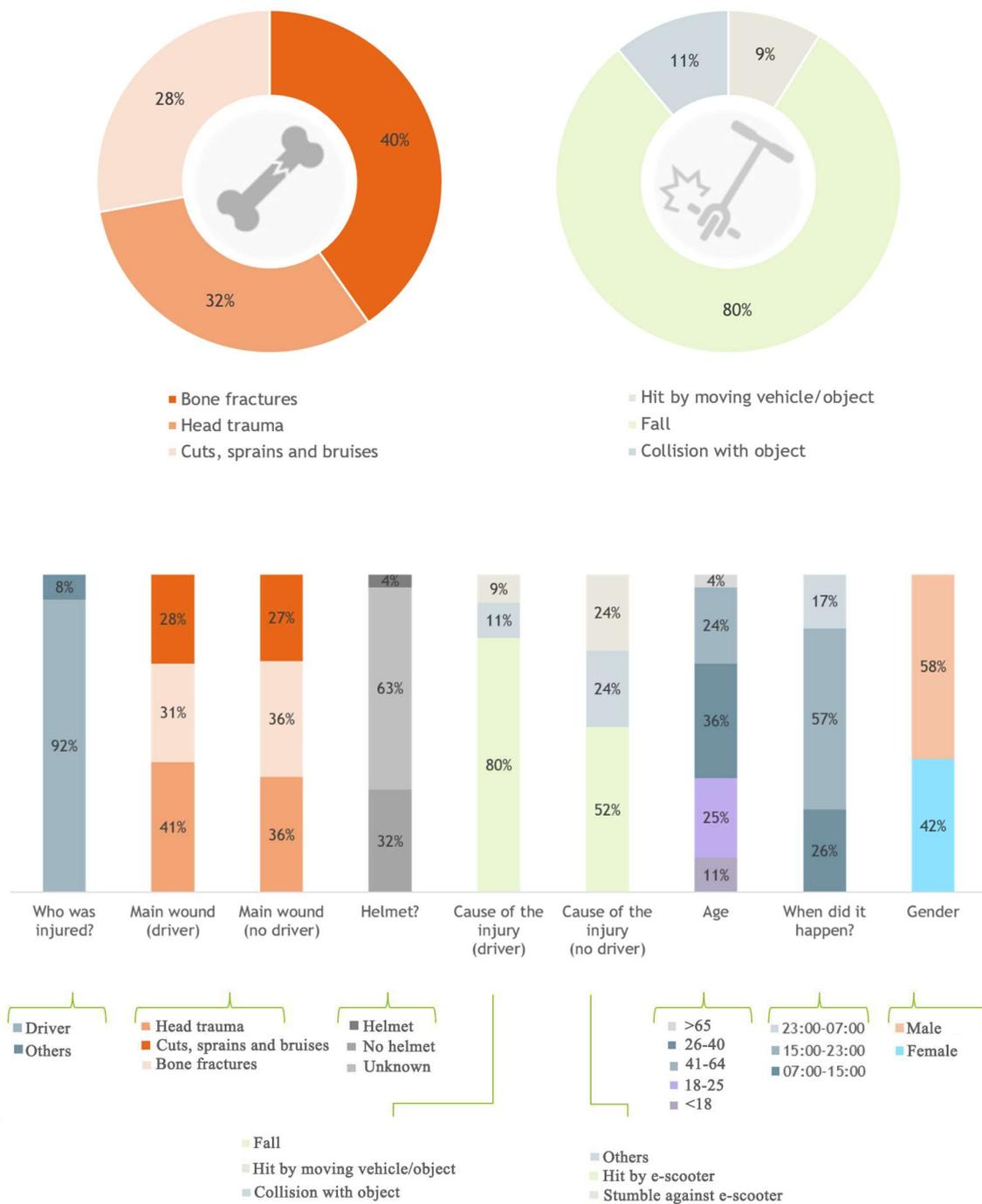


Figure 1.5 – 1.6. The main causes and injuries linked with an e-scooter accident on a sample of 249 patients in Southern California from September 2017 to August 2018

Electric scooter-related injuries are a novel phenomenon with varying degrees of severity. An intriguing point is the sparse use of helmets and the high number of accidents in patients under the age of 18, for whom the protective device is required [17]. According

to the evidence provided by these data, substantial government involvement is needed to reduce and mitigate casualties.

Manufacturers also have a moral obligation to provide these vehicles with additional and more effective safety measures. Because electric scooters are a new trend, there hasn't been much study or analysis done on their safety features to date. According to the bibliographic analysis, Garmin is the only company today that offers dedicated equipment (Varia) to private clients in order to improve the safety of two-wheeled vehicles. This is a rear light/radar device that can provide visual and audible alerts to the driver of vehicles coming from behind [19].



Figure 1.7. Varia RTL510, Garmin, frontal and side perspectives

There are numerous experimental concepts in the scientific literature in the research area. In general, the most commonly used instruments are sonar, cameras, radar, and LiDAR. The main goal of these experiments is to add various safety-related features to two-wheeled vehicles, such as collision avoidance and blind spot detection, vehicle tracking and detection, obstacle classification and road surface evaluation [5].



Figure 1.8. Illustration of an ADAS system employed on a two-wheeled vehicle (in the figure, the Blind spot detection warns the rider with a display in the rear-view mirror)

For instance, a team of scientists has developed an "active" system that can detect and track a vehicle throughout its transition, so that it accurately communicates the position to the driver on the fly. Essentially, the project is designed as a single beam laser sensor mounted on a rotationally controlled platform for rear vehicle detection and tracking in order to provide collision alerts to the two-wheeled vehicle user. Because the laser sensor could only measure one reflection at a time, the laser sensor's rotational orientation had to be controlled in real-time in order to identify and maintain focus on the tracked car as its lateral and longitudinal distances changed [6].



Figure 1.9. Bicycle equipped with an active tracking device

Another interesting example comes from the study of a group of American academic scientists. This project uses the sensor fusion signal processing concept: it is the process of combining sensor data or data drawn from disparate sources so that the resulting information is less uncertain than if these sources were used separately. The project is essentially a real-time multi-sensor detection system that combines camera and LiDAR detections for a more accurate and robust obstacles detection system for collision avoidance [7, 20].

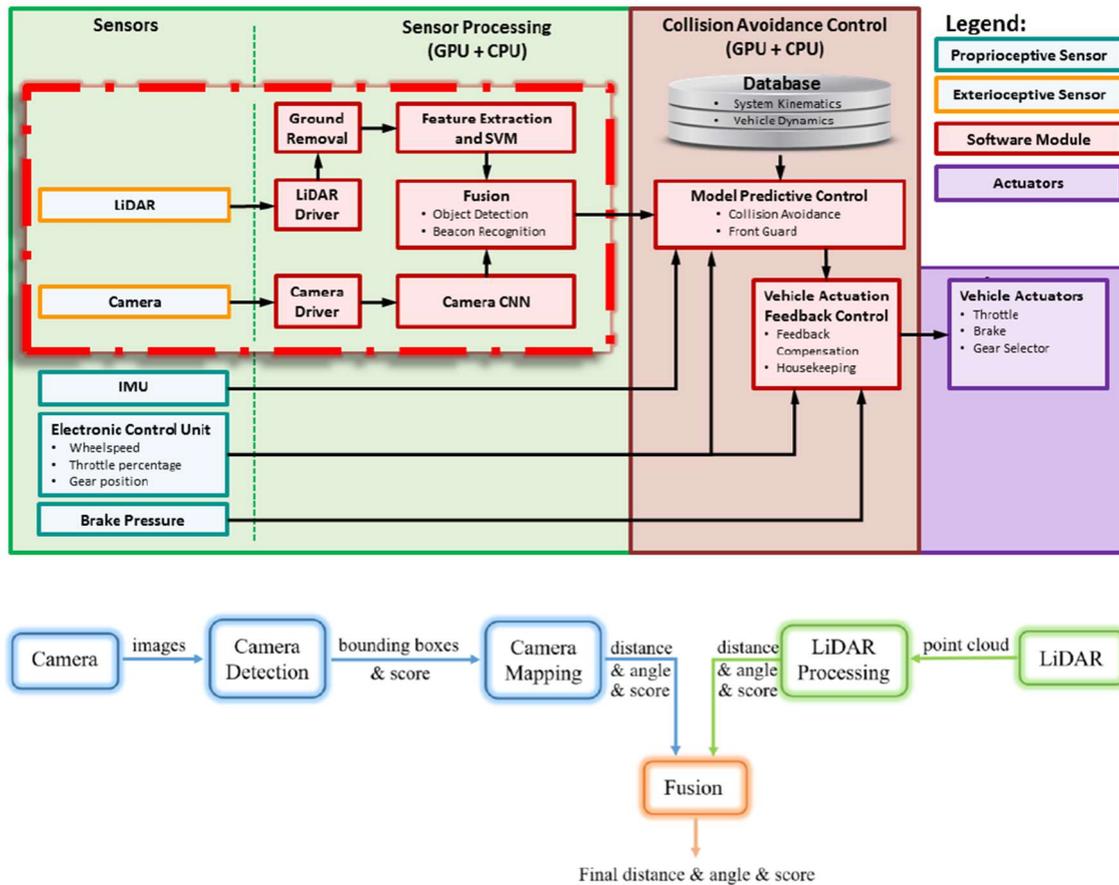


Figure 1.10. Block diagram of the system design

Despite the perceived ease of deployment in the two-wheeled vehicle operational field, the use of these devices is not without challenges. One significant issue is its unstable structure. A two-wheeled vehicle is prone to rolling, yawing, and pitching phenomena that can occur suddenly or gradually, so vibrations communicated to the sensor device are the main issue. This, in particular, impairs the collected measurements and, as a result, the functioning of the sensor to various degrees.



Figure 1.11. Examples of road surface irregularities that can cause a vibrations to a two-wheeled vehicle

The second problem is the sensing system's complexity, both in terms of hardware and software. A safety device must be able to function correctly under stress and for an extended period of time. Furthermore, it should be easy to use. A third issue, no less crucial than the other two, is the quality-to-price ratio. In the first instance, the overall cost of the sensing system must not be greater than the cost of the vehicle on which it will be installed. Second, the hardware must be able to perform the function for which it was intended.

1.4 POC actual state

According to the data, there aren't many products on the market that are relevant to safety equipment for two-wheeled vehicles, particularly electric scooters.

Teoresi Spa, a national and international engineering consulting firm based in Turin that specializes in smart mobility, artificial intelligence, safety, and security systems, is currently hosting an interesting project involving a smart obstacle detection system designed to be installed on electric scooters or bicycles. The project's specific aim is to detect dangers in real time that could endanger the driver and cause an accident [1].



Figure 1.12. Complete project structure fixed on a bicycle

A rigid attachment in the center of the handlebar was used to mount the LiDAR sensor prototype on a mountain bike. The robust metal structure was designed to keep the sensor static despite vehicle vibrations.

1.4.1 Fundamentals

The system concept, and therefore the type of sensor used, are based on four critical and fundamental aspects:

- time period of use: because the use of electric scooters and bicycles is not restricted by weather conditions, a sensor that can provide precise readings in both favorable and unfavorable weather conditions is required. This is especially

important during bad weather, such as rain, snow, or fog, as it adds additional risks and hazards to the riders;

- sensor positioning: the position of the sensor is critical in identifying possible hazards and dangers;
- nature of the identified obstacles: the goal of the project is to have a system that can identify and discriminate various obstacles of varying materials and sizes in order to provide predictive guidance to the driver;
- pointing distance: this project detects obstacles using the pointing distance, which is the average distance between the sensor and the ground. The value of this quantity is calculated to determine the minimum distance at which an obstacle must be detected in order for the driver to avoid a collision.

To ensure accurate obstacle avoidance and calculate the required stopping distance, factors such as vehicle speed and stopping capabilities must be considered. The law of the 28th of February, 2020, outlined regulations for electric scooters, including constraints on continuous nominal power (not exceeding 500W) and a maximum speed limit of 25 km/h [17]. Taking this speed limit into account, the maximum stopping distance was calculated to estimate the required lead time for obstacle detection, letting the driver enough time to completely stop the vehicle without impacting the object. The stopping distance is determined using the following formula:

$$\textit{stopping distance} = \textit{reaction space} + \textit{braking distance}$$

The reaction time is 1s, however it might vary based on the driver's concentration and reflexes. Taking the reaction space at maximum speed and braking time $t = 1s$ into account, it follows:

$$\textit{reaction space} = 25\text{km/h} * 1\text{s} = 6,94\text{m}$$

The braking distance is the distance traveled by a vehicle from the time the braking motion begins until it completely stops and is determined by a variety of factors, such as

speed, deceleration, tires-road friction coefficient and road slope. For simplicity, the following formula will be considered

$$\text{braking distance} = \frac{v^2}{2g\mu} = \frac{(6.94\text{m/s})^2}{2 * 9.8\text{m/s}^2 * 0.8} = 3,07\text{m}$$

where g is the earth gravity acceleration value, μ is the friction coefficient for optimal asphalt condition, and v is the vehicle speed [18]. It should be noted that in icy road conditions, the friction coefficient can be lowered from 0.8 to 0.05, increasing the braking distance by 16 times. It follows:

$$\text{stopping distance} = 6,94\text{m} + 3,07\text{m} = 10,01\text{m}$$

A pointing distance value of 10m is selected, since the stopping distance also represents the minimum distance at which an object must be found in order to be avoided. The inclination that the sensor must take to cover a distance of 10m is 5.68°, obtained by positioning the sensor at a height of 1m, the typical height of an electric scooter's handlebar:

$$d = \sqrt{(10\text{ m})^2 + (1\text{ m})^2} = 10,04\text{m}$$

$$\alpha = \sin^{-1}\left(\frac{1 * \sin(90)}{d}\right) = 5,68^\circ$$

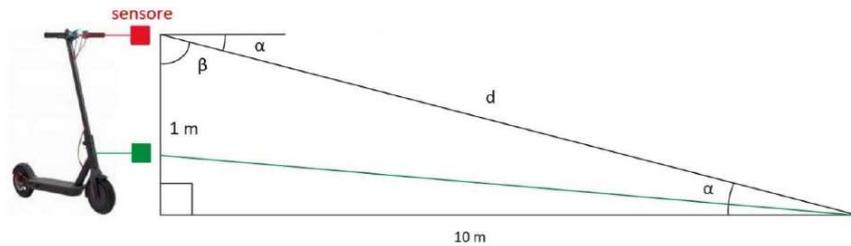


Figure 1.13. The values for pointing distance and inclination are denoted by the letters "d" and "α", respectively

1.4.2 Sensor selection

The concept of operation of distance sensors is to measure the difference between signal transmission and reception. This can be accomplished by utilizing either the duration or

the intensity of the returning signal. However, because of differences in technology and construction techniques, there may be functional differences. Three primary sensor choices were considered: ultrasonic, infrared, and LiDAR sensors. LiDAR was ultimately selected due to its high resolution, speed, and accuracy [1].

	Ultrasonic	Infrared	LiDAR
Suitability for Long Range Sensing	No	No	Yes
Resolution	High	Variable according to distance	Very High
High reading frequency	No	No	Yes
Sensitive to external conditions	Yes	No	No
Suitability to use for complex objects	No	Yes	Yes
Costs	Low	Low	High

Figure 1.14. Technology evaluation

LiDAR technology, which emits light waves rather than radio, sound, or infrared waves, is used to calculate the distance between obstacles. This is accomplished through a measurement principle known as "Time of Flight" (ToF), which involves measuring the elapsed time between the pulse's emission and the reception of the backscattered signal to compute the distance to the object.

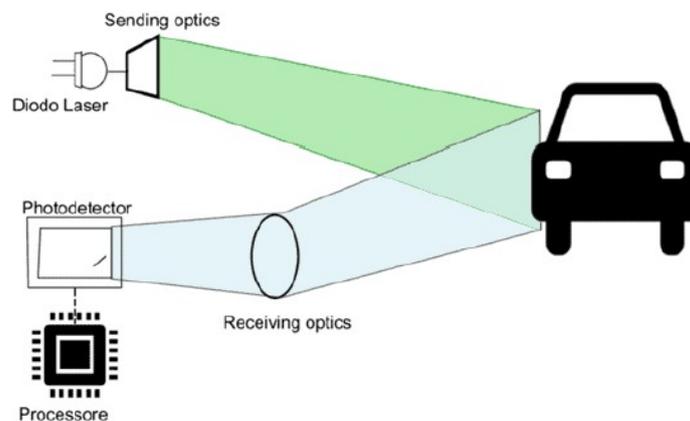


Figure 1.15. The operational technique of LiDAR

The LIDAR-Lite v3 model was selected after evaluating its characteristics in terms of distance range, cost, and performance. This sensor employs a novel signal processing technique that detects highly reflective objects while maintaining eye-safe laser power levels. It also conducts a "bias correction" process prior to the initial measurement, allowing for maximum sensitivity and accuracy, as well as a fast reading rate [21, 22, 23].



Figure 1.16. LiDAR-Lite v3

1.4.3 Hardware design

The system prototype contains several components to perform its function: the main one being the LIDAR-Lite v3 sensor for acquiring distance measurements, the Elegoo Uno R3 microcontroller for analyzing and processing data and a power bank as a power supply. In addition, for debugging reasons, the system contains secondary components such as LCD display, RGB LED, a passive buzzer and a SD card reader. To ensure the system proper operation, the physical and electrical characteristics of the sensor, as well as the communication protocol, were examined and taken into account [1].

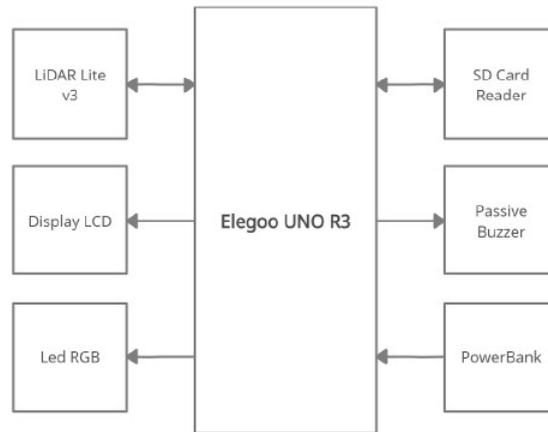


Figure 1.17. System' schematic block diagram

1.4.4 Software design

The Elegoo Uno R3 is a hardware platform comprised of a single electronic board equipped with a microcontroller that can be easily programmed using the open-source Arduino IDE. Furthermore, the board's microcontroller is pre-programmed with a bootloader that allows programs to be loaded onto the on-board flash memory via the USB connector. Particularly, the data is kept on the board until new data is written or a reset procedure is carried out. The project' script is made up of the two fundamental functions, with the following characteristics [1]:

- *Void setup()*: it is the primary source of initial instructions to be transmitted to the board prior to the program execution. It initiates the contact between the board and the sensor and configures the sensor to achieve high detection speed while maintaining measurement accuracy;
- *Void loop()*: it is in charge of supplying a set of program-related commands that assure appropriate program execution: crucial functions include *Distance()*, *Store()*, and *Obstacle()*.

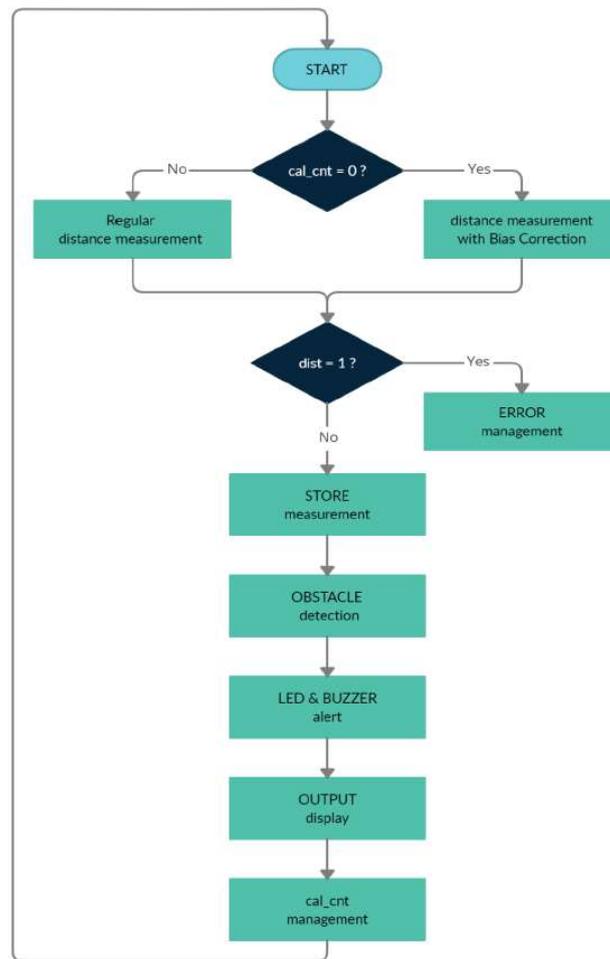


Figure 1.18. Schematic flow chart of the project's `Void loop()` function. Variables `cal_cnt` and `dist` are critical in managing the sensor recalibration process and successfully addressing measurement mistakes

The `Distance()` function is in charge of distance sensing. It incorporates bias receiver adjustment, which enables the sensor to improve the accuracy of measurements that may vary due to changes in ambient light. This ensures improved sensitivity: following the adjustment, the LiDAR undertakes a series of acquisitions in order to produce a measurement [1].

The `Store()` function manages measurement traces and filters them. Aside from the configuration of its own parameters, the accuracy of the sensor can be affected by a variety of elements such as signal reception, target distance, size, aspect and reflectivity [1].

The *Obstacle()* function integrates the essential logic for obstacle detection and discrimination. The sensor can make 36 measurements every meter, or one measurement every 3cm, at a speed of 10km/h. At 25km/h, the sensor can make 14 measurements each meter, or one measurement every 7.2cm. The number of measurements made per meter decreases as the vehicle speed increases, reducing the probability of correctly detecting obstacles. The function receives real-time distance readings from the sensor and uses a *flag* variable to determine the presence and type of obstacle. If no obstacle is detected, the *flag* variable is set to zero, otherwise it is set to a different value depending on the type of obstruction encountered. To determine the type of obstacle, the function use a finite state machine (FSM). From the default state, there are four conceivable outcomes: distance increase or decrease, radical distance increase or decrease [1].

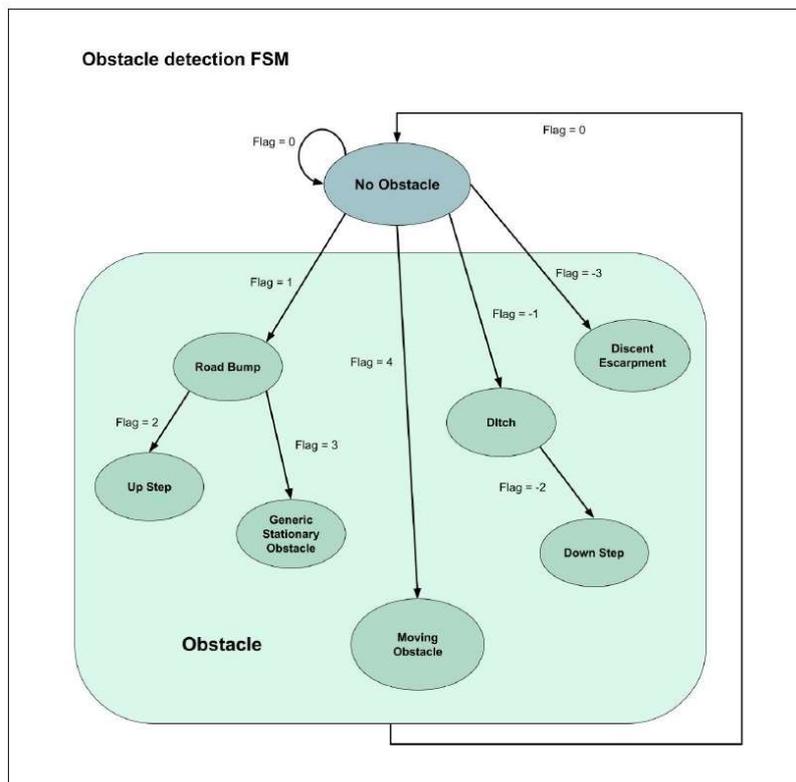


Figure 1.19. Finite-state machine for detecting the type of obstacle

1.4.5 Prototype tests results and conclusions

On asphalt roadways, the sensor prototype was tested for obstacle detection and found to be reliable. The mountain bike's shock absorbers, however, had a negative impact on the accuracy. Due to the high number of oscillations, the system is not suited for use on unpaved or uneven roads while mounted on a mountain bike. Urban regions with mostly paved roads and a speed of 10km/h are optimal. A more powerful microcontroller is necessary to maintain efficiency at greater speeds. Maintaining modest speeds enables precise obstacle detection and real-time alerts. Adverse weather conditions can also reduce the accuracy of the sensor, decreasing the quality of the measurement results [1].

Overall, the most significant challenge affecting the sensor's measurements is the instability of the two-wheeled vehicles. The sensor measurements are prone to errors of up to 15cm due to vehicle oscillations, resulting in values that fluctuate rather than being constant at the predetermined 10m distance when there are no obstacles in front of the sensor. A data filtering procedure is employed to lower the error % and provide more stable measurement readings. This improves measurement accuracy up to $\pm 3\text{cm}$ [1].

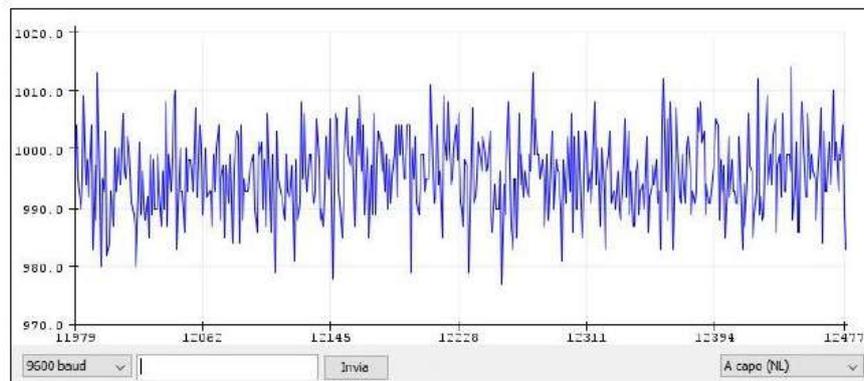


Figure 1.20. Unfiltered measurement trace trend



Figure 1.21. Filtered measurement trace trend

Overall, the system is capable of measuring distances and detecting an obstacle in front of it, warning the driver of the imminent danger, with some restrictions [1].

1.5 Feasibility study

As seen by the project's current state of the art, it is subjected to some limits, both in terms of hardware and software.

The hardware limits that have a negative impact on the system are the following:

- Elegoo Uno R3's computing power is not sufficient for a real implementation: the board is in fact built on the Arduino platform and employs an ATmega328P microcontroller. It is popular for its ease of use but it is insufficient for applications that require real-time processing and high-speed connectivity;



Figure 1.22. Elegoo Uno R3 board

- vehicle's oscillations: due to both mountain bike' shock absorbers and unpaved roads, an undulating movement of the entire sensing system can be induced. This results in a change in the height and angle of inclination of the sensor, which no longer detects the predetermined 10m distance when there are no obstacles in front of the sensor, decreasing the system's accuracy [1].

Instead, the software limitations are as follows:

- punctual acquisition is unsuitable for a real-world context: this type of acquisition, also known as single-point LiDAR, employs a LiDAR sensor that generates a single laser beam and measures the distance between the beam and the first object it meets. When compared to other methods of LiDAR acquisition, the punctual one is typically the most cost-effective, time-efficient, and space-efficient approach. This is because punctual acquisition uses only one laser beam to measure the distance to an object. Its ability to identify and map complex circumstances, as well as its measurement precision, are, nevertheless, restricted. Furthermore, this type of LiDAR acquisition can occasionally produce "ghost" reflections from surrounding objects, which can interfere with correct measurements [1];

- no software standards are employed, resulting in low scalability: when a piece of software is written without following a standard procedure, the codebase suffers from a lack of consistency and structure. As changes and upgrades get more sophisticated, this might make it difficult to maintain and scale the program over time. It is also more difficult to ensure that the software fulfills industry quality and reliability criteria. This can lead to glitches, failures and other problems.

The purpose of this thesis is to reduce the constraints that affect the POC and to improve the system's robustness and performance where it is possible and while keeping costs low. In particular, the targets are:

- to expand the range of sensed area;
- to increase the resistance to vibrations;
- to have a more responsive output;
- to have a higher level of reliability;
- to increase the capacity to distinguish obstacles.

A great amount of bibliographical research was conducted in order to comprehend the current state of the art of potentially viable solutions. Several options for achieving the selected targets were found in this respect, with each one focusing on reducing the constraints while maintaining the system's cost and complexity at minimum.

From a hardware standpoint, the following are feasible solutions:

- use of a better board: having an enhanced board is generally beneficial for any project as it may provide improved performance, stability, and compatibility with a wider range of components;
- use of a gimbal: LiDAR sensor can be installed on a stable platform that is decoupled from any vibrations by employing a gimbal, allowing it to detect distances more accurately. This is due to the gimbal's ability to compensate for vibrations and maintain a stable alignment of the sensor, resulting in more accurate and dependable results.

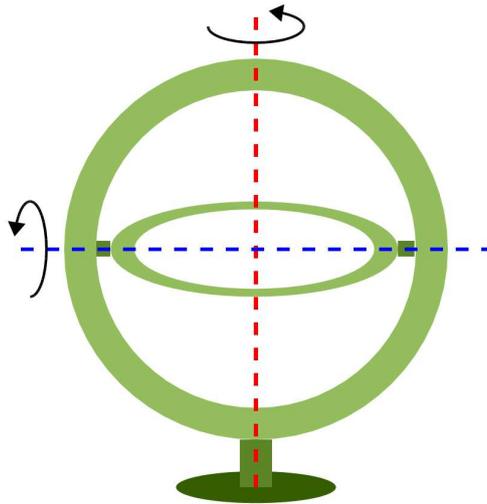


Figure 1.23. Gimbal schematic drawing

Instead, the software feasible solutions are as follows:

- implementation of spatial acquisition: it is a technique that uses a LiDAR sensor to build a 2D/3D representation of the surrounding environment by emitting numerous beams in different directions. Furthermore, "ghost" reflections have less of an impact on spatial acquisition than they do on punctual acquisition. Nevertheless, spatial acquisition comes with some drawbacks, particularly an increased demand in terms of cost and computational time;

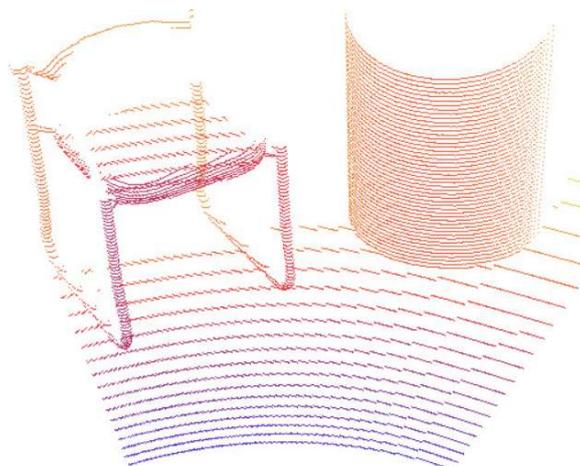


Figure 1.24. 3D spatial acquisition output (point cloud)

- improving scalability by implementing software standards: using software standards, such as Simulink and MAAB requirements, can result in better software quality, reliability, stability, and maintainability. This is particularly important in safety-critical or highly regulated areas like automotive, healthcare or finance.

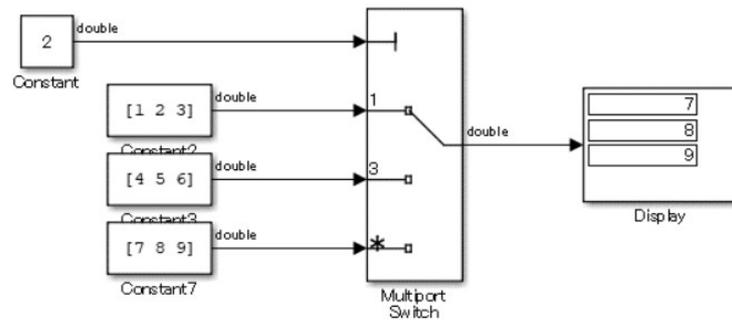


Figure 1.25. Example of a Simulink model created in accordance with the MATLAB Automotive Advisory Board (MAAB) requirements

The feasibility analysis conducted for the project's improvements has resulted in favorable outcomes, indicating that all the considered potential solutions are feasible and cost-effective.

2 Improvement of the prototype

The development of a prototype is an important phase in the design of a new product. The prototype, however, is not the final product, and revisions are frequently required to guarantee that it meets all of the project's objectives. In this chapter, the evolution of the original prototype of the project will be examined.

2.1 Concept

The table below summarizes the Teoresi staff-approved concept:

	Lidar 1D	Gimbal	IMU + fusion	STM32F4	2D/3D
Final concept	●	●	●	●	●

Figure 2.0. New proof of concept features

The system comprises several components. The LiDAR sensor, which acquires distance measurements. The gimbal, which converts the actual system from 1D to 2D/3D system. Additionally, the IMU quantifies and corrects measurement errors caused by unwanted external forces applied to the sensing system. Finally, the STM32F4 microcontroller allows the system to operate at higher speed in terms of data acquisition and elaboration rate. An exploratory market analysis was conducted to find the best deals for the components needed. The search parameters were ranked in decreasing order of importance:

- low cost;
- compatibility;
- performance;
- availability.

The research outcomes are presented in a multilevel bill of materials, which illustrates the components that make up the sensor system. The sensor support is not evaluated because it will be added later in the project and its components are not currently quantifiable.

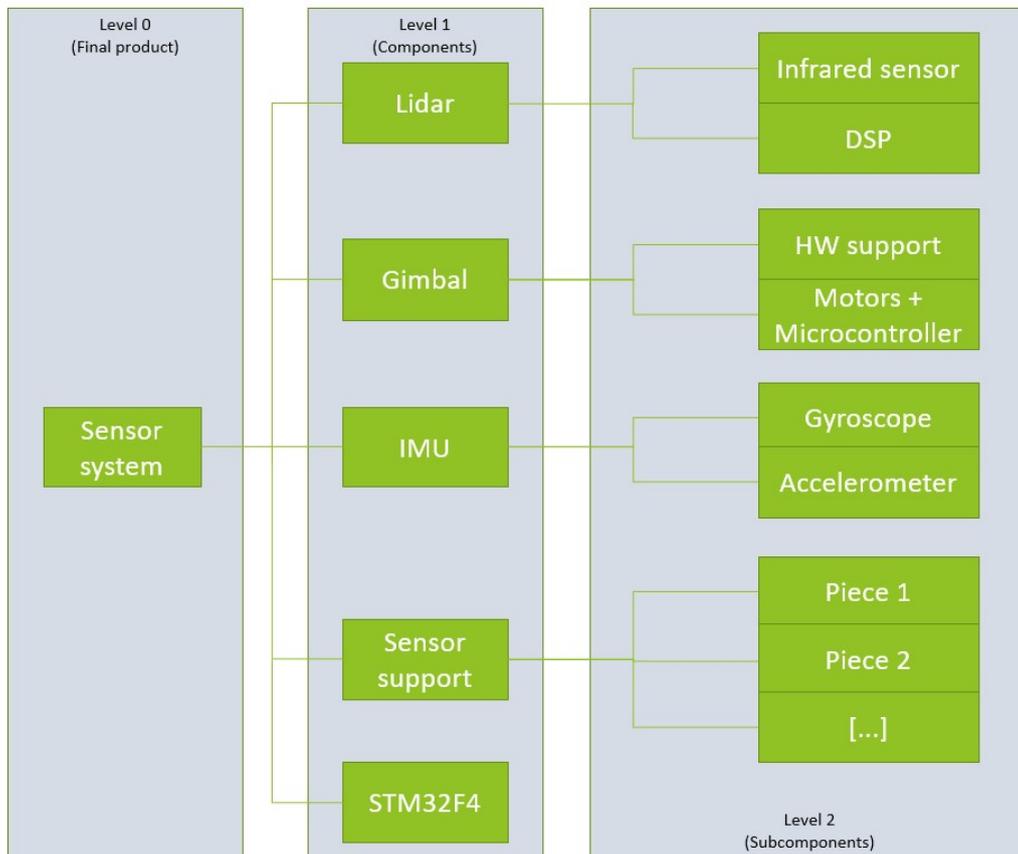


Figure 2.1. Sensing system multilevel bill of material

Secondary components such as the LCD display, RGB LED, passive buzzer and SD card reader will be excluded from the new proof of concept because they are outside the scope of this study.

2.2 Hardware and software improvements

This section will delve into the specific changes that were implemented at both the hardware and software levels, along with a comprehensive explanation of the reasoning behind these modifications.

2.2.1 STM32F4 Nucleo-144 board

The first important hardware improvement is the STM32F4 Nucleo-144 board. The board features an ARM Cortex-M4 processor that can run at up to 168MHz, allowing for high-performance computing for demanding applications. It also has 1MB of Flash memory and 192KB of SRAM, which provides storage and memory for large code bases and data-intensive applications. The board includes numerous communication interfaces such as USB, Ethernet, CAN, SPI, I2C, UART, and others, making it simple to connect to a variety of external devices and sensors. The effective power management technology embedded with the board allows it to function on a low-power budget without losing performance. It is a good choice for a wide range of applications, including industrial automation and control systems, smart home devices, and Internet of Things (IoT) applications [24].

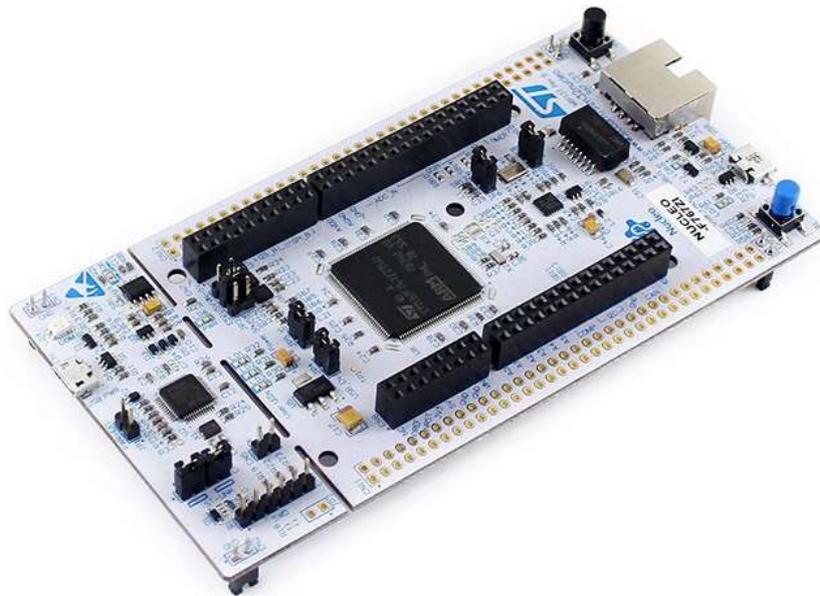


Figure 2.2. STM32F4 Nucleo-144 board

2.2.2 Gimbal

The second important hardware improvement is the Storm32-BGC. It is a type of brushless gimbal designed to stabilize cameras and other imaging equipment for smooth and stable data acquisition. It features three stabilization axes to compensate for any

undesired movement or shaking. It is based on a powerful ARM Cortex-M3 32-bit RISC CPU microcontroller running at 72MHz frequency, high-speed embedded storage (flash memory up to 512KB and SRAM up to 64KB), and a diverse set of I/Os and peripherals communication interfaces. The STorM32 project actually consists of three parts, the STorM32 controller board, the firmware “o323BGC”, and the Windows GUI “o323BGCTool”. The latter is a user-friendly interface that makes it easy to adjust parameters and fine-tune the gimbal's functionality to meet the objectives of the project [25].



Figure 2.3. Storm32bgc, brushless gimbals with 3 axes based on a STM32 32-bit microcontroller

2.2.3 Wiring and connections

The wiring and connections required for the project will be described in this chapter.

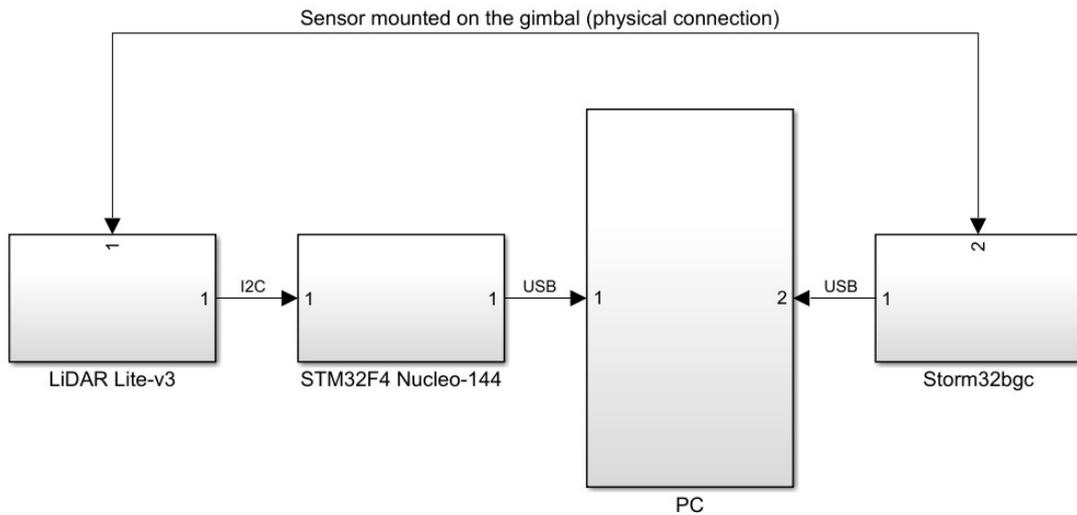


Figure 2.4. Hardware architecture schematic block diagram

2.2.3.1.1 LiDAR to STM32F4 Nucleo-144

From a hardware point of view, the LIDAR-Lite v3 interfaces with the STM32F4 Nucleo-144 through a rectangular electric latch-lock port that connects the sensor to a 6-wire cable. Each wire in the cable is identified by a unique color. Red and black cables are typically utilized for power supply, whereas green and blue wires are used for I2C connections. The remaining two wires, orange and yellow, enable power supply via an internal pull-up resistor and control the connection mode, respectively [22].

Wire Color	Function
Red	5 V DC (+)
Orange	Power enable
Yellow	Mode control
Green	I ² C SCL
Blu	I ² C SDA
Black	Ground (-)

Figure 2.5: The color of a wire is related to its function.

The connections of the LiDAR to the previous project's board, the Elegoo Uno R3 microcontroller, are shown in the image below. There were four connections: the red wire of the LiDAR was linked to the microcontroller's 5V pin, the green wire to the SCL pin, the blue wire to the SDA pin, and the black wire to the GND pin [1]. The sensor's yellow and orange cables were unnecessary and hence were not used. For the STM32F4 Nucleo-144, the same connection strategy will be used.

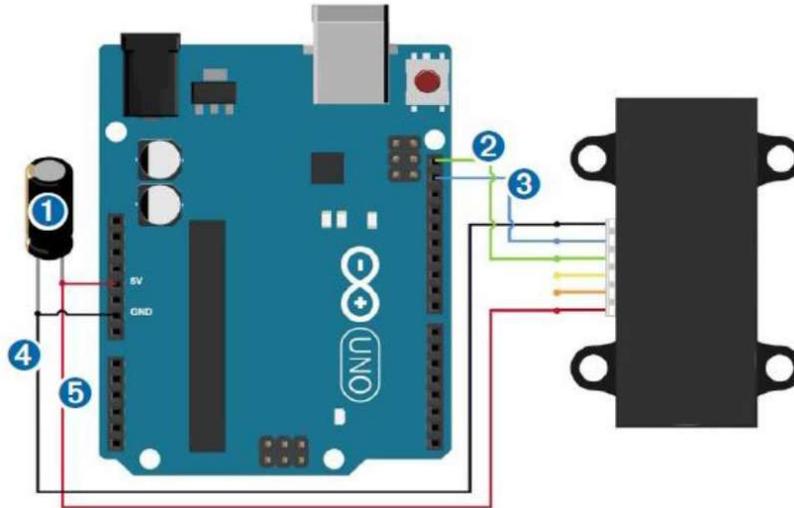


Figure 2.6: Connection scheme for LiDAR-Lite v3 and Elegoo Uno R3

In order to maintain a stable voltage, a $680\mu F$ capacitor is connected between the 5V power supply and GND. This will help mitigate peak current during power up [1].

The LiDAR can be set by choosing between two communication modes:

- I2C: a serial communication bus that allows data to be transmitted between integrated circuits;
- PWM: a bidirectional signal communication mechanism that uses the mode control pin to send and receive distance measurements.

To meet the project's needs, the I2C serial interface was chosen, with the sensor acting as "slave" device and the board as "master". The I2C is a bidirectional synchronous communication protocol that allows data to be transmitted between integrated circuits over two communication lines: the Serial Data (SDA) line and the Serial Clock (SCL).

Devices which use I2C communication are classified as either "master" or "slave." By delivering clock pulses on the SCL line, the master device initiates communication and manages transmission synchronization. The slave device sends the requested data in response to the master's requests. The I2C protocol is widely used in electronic systems to communicate between devices such as sensors, memory, and displays. It allows for various data transmission speeds, including "fast mode" at 400kHz [26].

From a software point of view, the I2C interface must be used to communicate between the LiDAR-Lite v3 sensor and the STM32F4 Nucleo-144 board. By using STM32IDE, it is necessary to create a new project and configuring the microcontroller's pinout according to the specifications of the LiDAR sensor. Pins PB6 (D15) for SCL and PB9 (D14) for SDA were utilized for this scope. STM32IDE is an open-source integrated development environment for programming STM32 microcontrollers, offering a user-friendly interface and comprehensive development tools, including HAL (Hardware Abstraction Layer) libraries for advanced features [22].

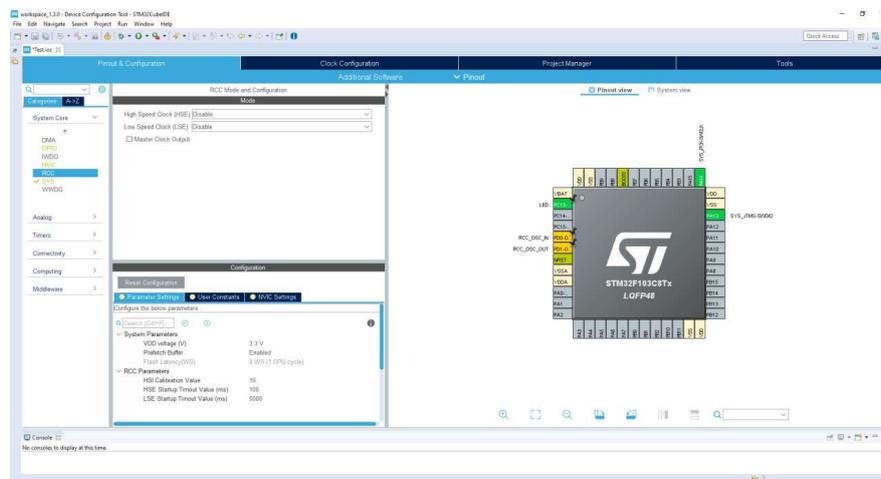


Figure 2.7. STM32IDE visual programming interface example

2.2.3.1.2 Gimbal to PC

The Storm32bgc gimbal board interfaces with the PC via universal serial bus (USB) standard, facilitating parameter configuration, programming and acquisition.

From a software point of view, it is possible to configure the Storm32bgc board and tune the gimbal control parameters through the Windows GUI “o323BGCTool_v096”, which

is freely accessible. After selecting the proper board type, this software allows access to different setup choices and gimbal control parameters. To tune the system's performance, it is possible to configure the sensor type used by the gimbal, the type of motors used, the gimbal's range of motion and PID control settings [25].

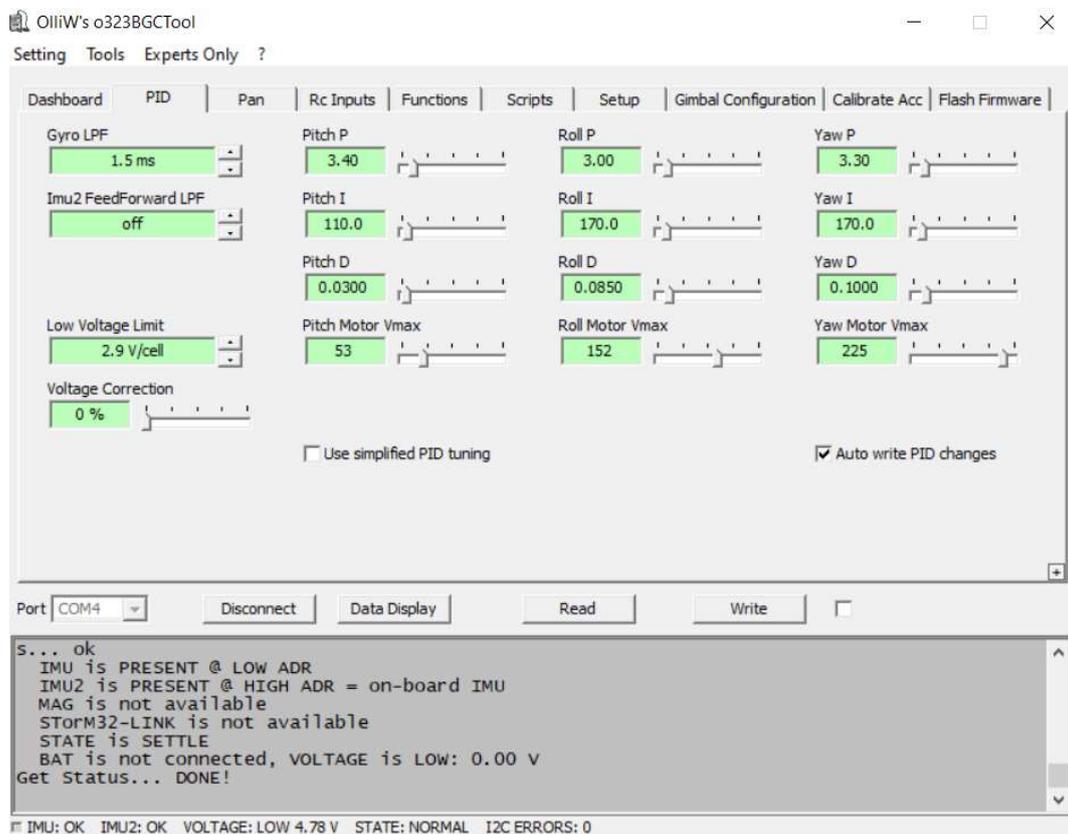


Figure 2.8. Windows GUI example – PID parameters configuration section

During the configuration process, the Z-axis of the gimbal was reversed so that it would be oriented vertically, making the system more effective for the targets of the project.



Figure 2.9.1. Vertical position of the gimbal (reversed Z-axis)



Figure 2.9.2. Vertical position of the gimbal (reversed Z-axis)

MATLAB's visualization tools allow for the production of visually appealing data representations, which improves the interpretation of results. The integration with Simulink enables the design and testing of dynamic systems [28].

2.2.4 Acquisition algorithm

This section explores the data acquisition algorithm, providing an overview and a detailed explanation of the acquisition algorithm.

2.2.4.1 Introduction

The algorithm's purpose is to manipulate the LiDAR within a $5m \times 5m$ area in front of the testing vehicle in order to acquire distance measurements [29, 30, 31, 32]. Each recorded data point D_{ij} ($i, j = 1, 2, \dots, 10$) collected in the *lidar_distance_acquisitions_buffer* matrix will be assigned a separate color based on L_{ij} , the distance between each data point D_{ij} and the LiDAR, after collecting a total of 100 data points equally distributed ($555,6mm$ apart from one to the other) throughout this acquisition square. The reason for collecting this set of data points with varied colors is to develop a representation that can be used to identify possible obstacles in front of the vehicle. The goal is to use the LiDAR's capabilities to build a visual representation of the environment, offering useful insights for object detection and assisting in navigation or obstacle avoidance tasks.

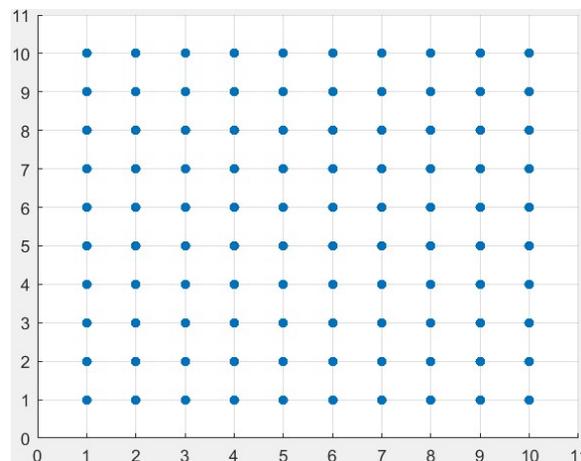


Figure 2.11. Example of a 2D representation of 100 data points equally distributed over a $5m \times 5m$ area

The logic of the algorithm can be summarized as follows ($i, j = 1, 2, \dots, 10$):

- 1) Save and set to gimbal the initial angle configuration $(\rho, \vartheta, \varphi)_{setup}$;
- 2) Perform the following actions until both the matrices *lidar_distance_acquisitions_buffer* and *constrained_distances* are completely filled, meaning they do not contain any values equal to zero:
 - a. Collect the LiDAR measurement samples into the variable *dataAll*, apply filtering techniques and compute the average value, which is the data point pointing distance D_{ij} . This is necessary as the acquired samples may exhibit variations. Save it in the *lidar_distance_acquisitions_buffer* matrix;
 - b. Compute L_{ij} , the distance between each data point D_{ij} and the LiDAR, and save it in the *constrained_distances* matrix;
 - c. Save and set to gimbal the new angle configuration $(\rho, \vartheta, \varphi)_{ij}$ to proceed correctly with the acquisition, taking into account:
 - the arbitrary terrain scanning path chosen at the design level;
 - the technical constraint imposed by the gimbal motors;
 - the spatial constraint imposed by the acquisition area chosen at design level;
- 3) Once both *lidar_distance_acquisitions_buffer* and *constrained_distances* matrices have been filled with the acquired and processed distances, it is possible to visualize the scanned terrain.

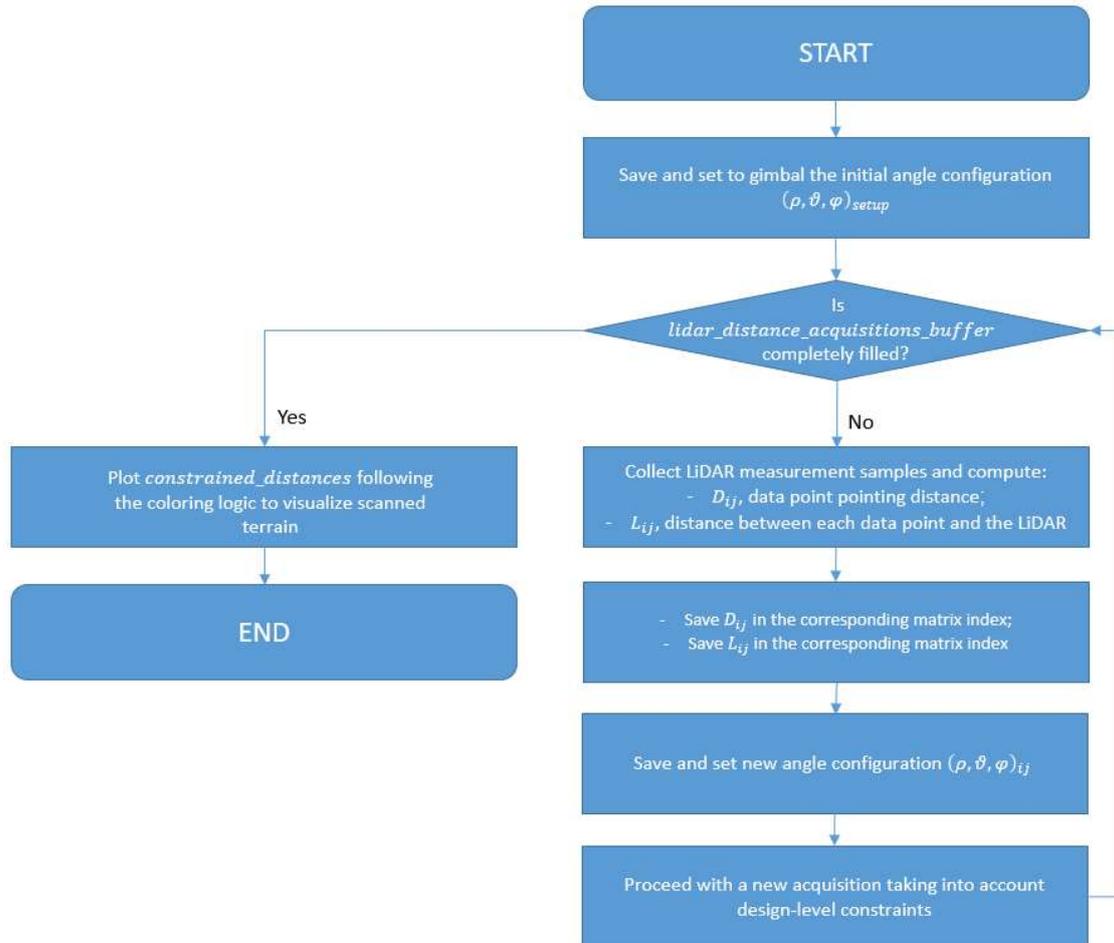


Figure 2.12. Acquisition algorithm flow chart

In summary, the procedure will go in the following order: first, the initial angle configuration is set. Data points will be collected, then the next orientation angles will set, the gimbal will be moved correspondingly, and the process will be repeated. This recurrent cycle of acquire-calculate-move will assure accurate positioning and data collecting. By combining these elements, it is possible to generate a visual representation of the mapped area, providing valuable insights and information for further analysis.

2.2.4.2 Variables definition

To ensure complete coverage of the whole acquisition region and get a knowledge of the expected outcomes, the primary goal of this section is to identify:

- the maximum angles values at which the LiDAR should be tilted;
- acquisition region's constraints.

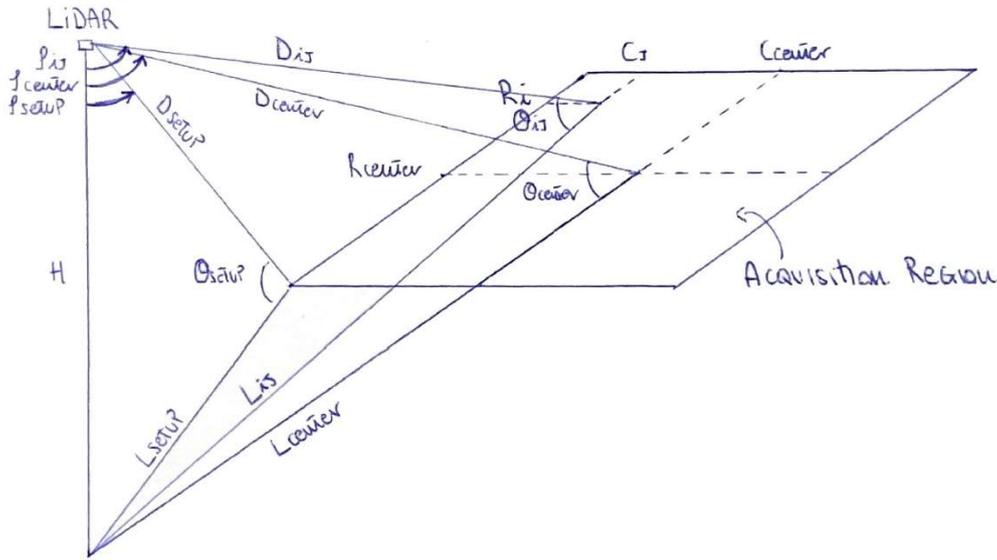


Figure 2.13. The LiDAR (on the left) and the acquisition region (on the right) are illustrated schematically.

The constant and variable values involved in the algorithm have been calculated and reported in the tables below:

Name	Type	Value	Unit of measure	Description
L_{setup}	Constant	$\frac{5\sqrt{(17)}}{2}$	m	Distance between the acquisition starting point and the LiDAR.
lim_inf	Constant	1	m	If anything is found closer than this distance, the measurement must be redone.
lim_sup	Constant	19	m	If anything is found beyond this distance, the measurement must be redone.
L_{ij}	Variable	-	m	The distance between each data point D_{ij} and the LiDAR. This information is required to correctly manage the LiDAR movement algorithm and the 2D plot data visualization.
H	Constant	1	m	The fixed height at which the LiDAR is mounted is an arbitrary decision aimed at optimizing the pitch angle θ and offering greater flexibility in the mapping design phase.
D_{setup}	Constant	$\frac{\sqrt{(429)}}{2}$	m	Theoretical unobstructed pointing distance refers to the distance measured by the LiDAR without any obstacles or obstructions present at the initial acquisition point.
D_{ij}	Variable	-	m	The pointing distance is determined by the measurements samples obtained from the LiDAR.
$[\rho, \theta, \varphi]_{setup}$	Constant	$[-14.04, 5.54, 0]$	°	Angle triplet which represents the initial gimbal configuration, serving as the starting point for data acquisition.
d_p	Constant	3.3	°	Depending on the algorithm's status, this value is utilized to increase or decrease the yaw value.
d_θ	Constant	0.6	°	Depending on the algorithm's status, this value is utilized to increase or decrease the pitch value.

Figure 2.14. Table of constant and variables

In particular:

$$L_{setup} = \sqrt{10^2 + (5/2)^2} = \frac{5\sqrt{17}}{2} m$$

$$D_{setup} = \sqrt{L_{setup}^2 + 1^2} = \frac{\sqrt{429}}{2} m$$

$$\rho_{setup} = -\sin^{-1}\left(\frac{1 * 2.5}{L_{setup}}\right) = -14.04^\circ$$

$$\vartheta_{setup} = \sin^{-1}\left(\frac{1 * \sin(90)}{D_{setup}}\right) = 5.54^\circ$$

$$\varphi_{setup} = 0$$

Acquisition region's points of interest	ρ	ϑ	L	D
Center	0	4,574	12.5	$\frac{\sqrt{(629)}}{2}$
Bottom left vertex (setup)	-14.04	5.541	$\frac{5\sqrt{(17)}}{2}$	$\frac{\sqrt{(429)}}{2}$
Top left vertex	-9.463	3.763	$\frac{\sqrt{(37)}}{2}$	$\frac{\sqrt{(929)}}{2}$
Top right vertex	9.463	3.763	$\frac{\sqrt{(37)}}{2}$	$\frac{\sqrt{(929)}}{2}$
Bottom right vertex	14.04	5.541	$\frac{\sqrt{(17)}}{2}$	$\frac{\sqrt{(429)}}{2}$

Figure 2.15. Table of the initial setup parameters

The calculations were carried out taking into account the following factors:

- The bottom-left vertex of the acquisition region is the initial setup point from which the LiDAR will begin the measurement process;
- Positive yaw (ρ_{ij}) values indicate a right turn, while negative values indicate a left turn relative to the horizon;

- Positive pitch (ϑ_{ij}) values indicate a downward movement, while negative values indicate an upward movement relative to the horizon;
- The angular values provided were derived with the LiDAR pointing at the horizon as the zero reference;
- The initial acquisition point corresponds to the index (1,1) of the *lidar_distance_acquisitions_buffer* and *constrained_distances* matrices;
- Because the LiDAR is not impacted by roll, the phi angle (ϕ_{ij}) will be excluded from the subsequent calculations. In fact, the gimbal's function is to keep the LiDAR aligned with the horizon line ($\phi_{ij} = \phi = 0$);
- The values $lim_inf = 1m$ and $lim_sup = 19m$ are used as discriminators to eliminate values that are too low or too high for the project's purposes.

The choice of the *lim_inf* value was guided by the “bumper principle”, a concept associated with vehicle bumpers. While in Italy it is not specify a fixed minimum distance in meters to maintain between vehicles when they are stopped at a traffic light or in congested traffic, the Italian Highway Code (article 141) emphasizes the importance of keeping a safe following distance. This distance is crucial to let drivers to come to a safe stop in the event of sudden or unforeseen braking. The “bumper principle” reflects the general common-sense notion of keeping a sufficient distance, allowing a driver to see the rear bumper of the vehicle in front, thereby minimizing the risk of rear-end collisions and ensuring an appropriate safety margin in both moving traffic and congested queues. The actual average following distance may vary depending on road conditions and the size of the vehicles involved. However, for many traffic scenarios and most vehicles, maintaining an approximate distance of 1-2 meters between your vehicle's front bumper and the preceding vehicle's rear bumper can be considered a typical estimate [33]. In contrast, the *lim_sup* value was chosen taking into consideration exceptional cases, such as sensor system malfunctions or instances where the vehicle is not in normal operating conditions (e.g. the electric scooter has fallen over).

2.2.4.3 Data acquisition

The outcome of this section is a dataset that serves as the starting point for the later stages of the research, allowing to get useful insights and draw meaningful conclusions about the target area.

Since configuring the STM32F4 board to establish communication with the LiDAR was not in the scope of this thesis, the priority targets to be reached are:

- 1) to tune the interaction between PC and STM32F4;
- 2) to develop a MATLAB script which collects and stores all the measurement samples acquired by PC.

Regarding the first target, the LiDAR operates in standard mode at 1000Hz , taking one measurement $sample_k$ every $t = 1/1000 = 0.001\text{s}$. Each measurement sample, representing a distance measured in centimeters, is stored in a dedicated data buffer, ready to be read by the PC through the STM32F4. This means that within a minute, up to 60000 samples can be obtained. Once the buffer is filled, the PC reads the data at a baud rate of 115200, which corresponds to 115200 symbols per second. Each symbol is encoded with 8bits , resulting in a bit rate of 921600bits/s . Thus, the PC can read the 60000 samples in 0.521s , assuming the buffer is already filled. However, if the buffer is not yet filled by the LiDAR (which takes 60s to obtain 60000 samples), one must wait for the process to complete [1].

Since the new Proof of Concept (POC) is a real-time system, the system's output must be provided within a certain response time in accordance with the functionality of the system itself. As the system under consideration is used for obstacle detection, the following reasoning needs to be taken into account:

- the total time required for the LiDAR movement without any acquisition to cover the acquisition area was measured at 1.3s . Considering the acquisition area divided into 100 steps, each step takes 0.0013s . In order to collect some acquisitions during each step maintain the same movement lead time, a reduced number of samples was needed to be selected to match exactly the same time that one step requires to be completed. For this reason, the number of acquired

samples was decreased to 1300, 13 for each step. Since a single sample is acquired by the LiDAR every 0.001s, the process of collecting 13 samples takes exactly 0.0013s, which is in fact the same lead time for a single step. Since the PC takes 0.521s to read 60000 samples, it takes only 0.0113s to read 1300 samples. This results in a total acquisition time of $t_{acquisition} = 1.3 + 0.0113 = 1.3113s$;

- considering an average speed of 25km/h, the system would obtain a full measurement every $sd_{25} = 7 * 1.3113 = 9.18m$, which is obviously not feasible for the targets of this work. This represent a system limitation due to the insufficient power computation of the hardware installed;
- considering an average speed of 10km/h, the system would obtain a full measurement every $sd_{10} = 2.8 * 1.3113 = 3.67m$, meeting the objectives of this work. With 6.33m available for braking and considering the stopping distance on an asphalt road at 10 km/h is 3.3m, the setup aligns with the thesis targets.

Regarding the second target, the main challenge lies in the asynchronous nature of the LiDAR, STM32F4 board and PC. To read the measurement message correctly from start to finish and avoid starting mid-message, synchronization is crucial.

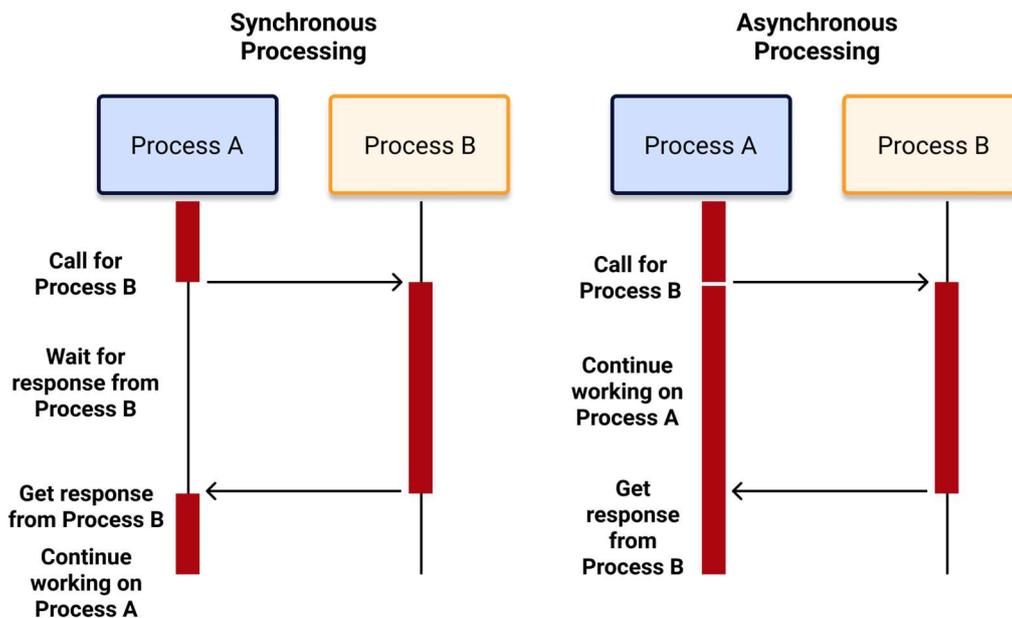


Figure 2.16. Synchronous and asynchronous processing methods

The LiDAR communicates with the STM32F4 board through the I2C protocol. The transmission message is encoded with 2 initial bits indicating the start and 3 final bits indicating the end of the message [1]. It is essential to read the entire message, including the start and end bits, to store the relevant data in the variable called *dataAll*. The following code snippet effectively identifies the initial position for reading the message coming from the STM32F4 board to the PC, which corresponds to one of the specified data values (0, 205, 10). This is achieved by sequentially analyzing each bit of the incoming message. Once the starting point for data storage is determined, the relevant data is then stored. In particular:

- The variable *data* is used to temporary store the incoming stream of data from the board;
- The variable *startFrame* is used as a flag variable;
- The function *read()* is used to collect the data coming from the COM interface.

```
data = read(stm32, 1, "UINT8");
if (data == 0)
    startFrame = 1;
end

data = read(stm32, 1, "UINT8");
if (data == 205)
    startFrame = 1;
end

data = read(stm32, 1, "UINT8");
if (data == 0)
    startFrame = 1;
end

data = read(stm32, 1, "UINT8");
if (data == 10)
    startFrame = 1;
end
```

Figure 2.18. Code snippet identifies the initial position for reading the message coming from the COM

2.2.4.4 Data processing and LiDAR spatial manipulation

The variable *dataAll* contains all the measurement samples $sample_k$ collected by the LiDAR. This dataset will be processed to calculate:

- the pointing distance D_{ij} , which will then be stored in the matrix *lidar_distance_acquisitions_buffer*;
- the distance L_{ij} between each data point D_{ij} and the LiDAR. This value, which will be stored in the matrix *constrained_distances*, is crucial for implementing the logic to reposition the LiDAR according to the path chosen at design level.

To complete this stage, it is important to take advantage of the techniques provided by the data processing field. Data processing is necessary to clean, transform, and extract relevant features from the raw data. This step helps to eliminate any inconsistencies or errors in the data, ensuring its accuracy and reliability.

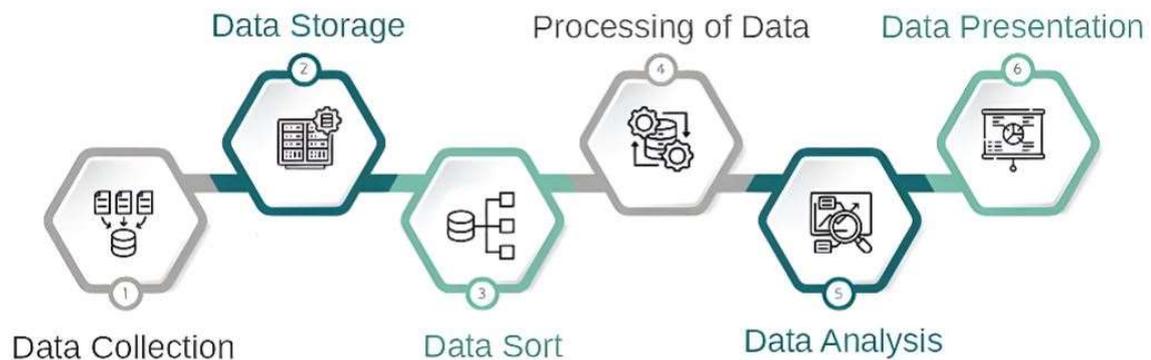


Figure 2.19. Data processing fundamental steps

To proceed with the computation of the variables D_{ij} and L_{ij} , the resulting dataset *dataAll* is subjected to signal frequency analysis. The significance of this analysis lies in its ability to assess whether the system under examination is affected by any form of noise or interference. Within the scope of this study, three distinct digital filtering methods will be considered and compared to understand the most suitable filtering approach taking into account trade-offs between performance, quality, and efficiency.

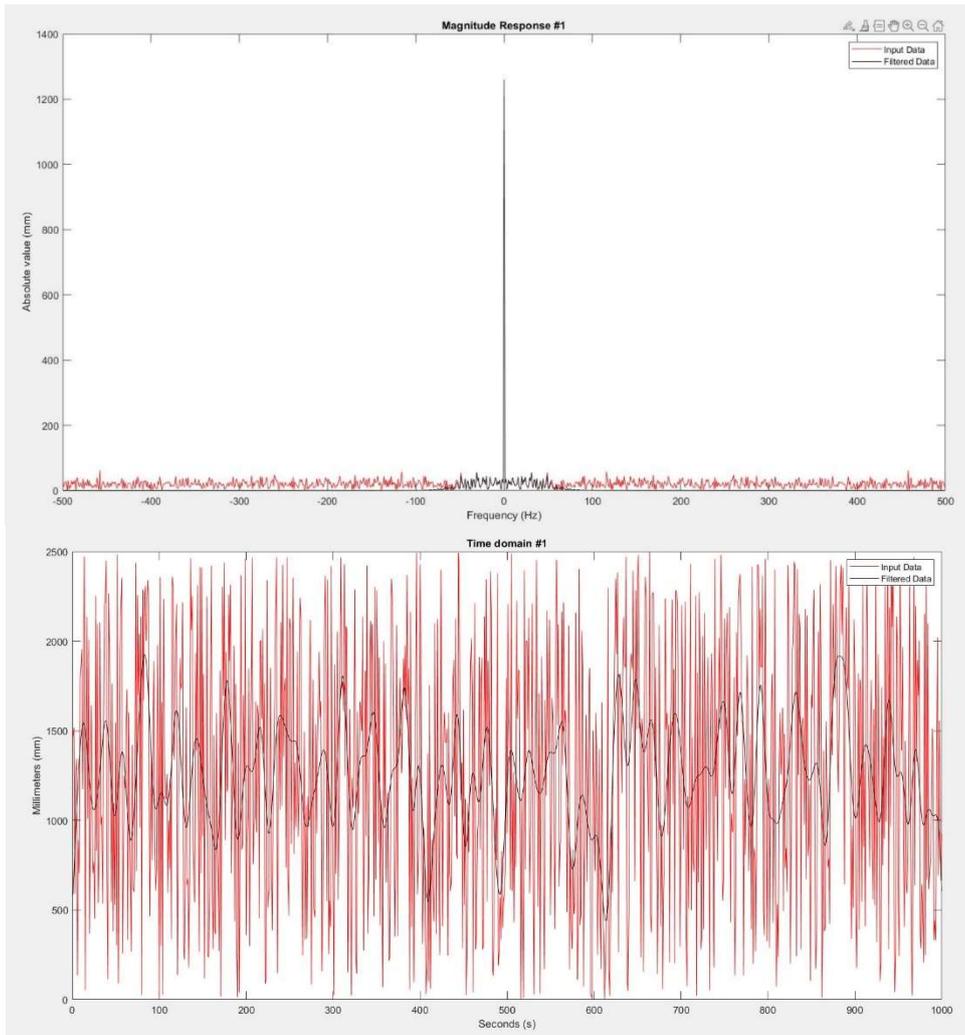


Figure 2.20: Digital filtering method A, frequency and time domains outputs

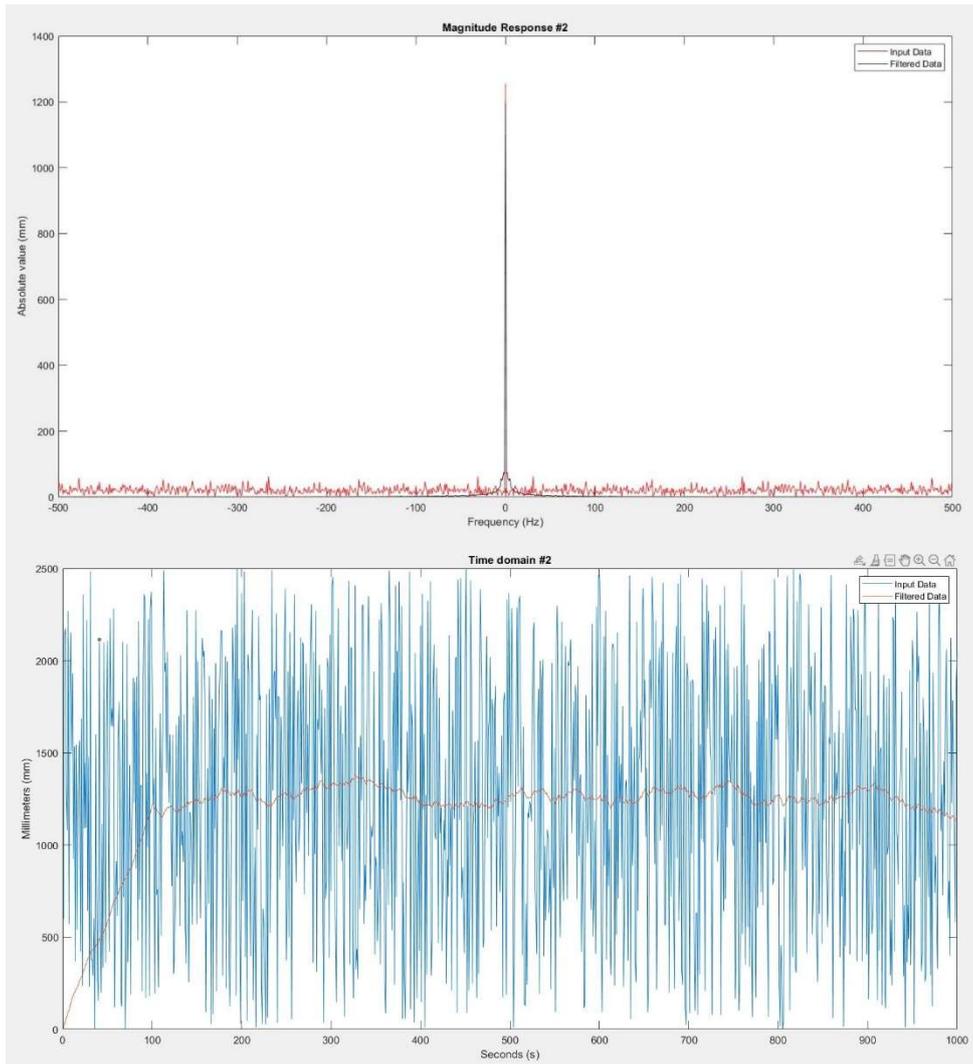


Figure 2.21: Digital filtering method B, frequency and time domains outputs

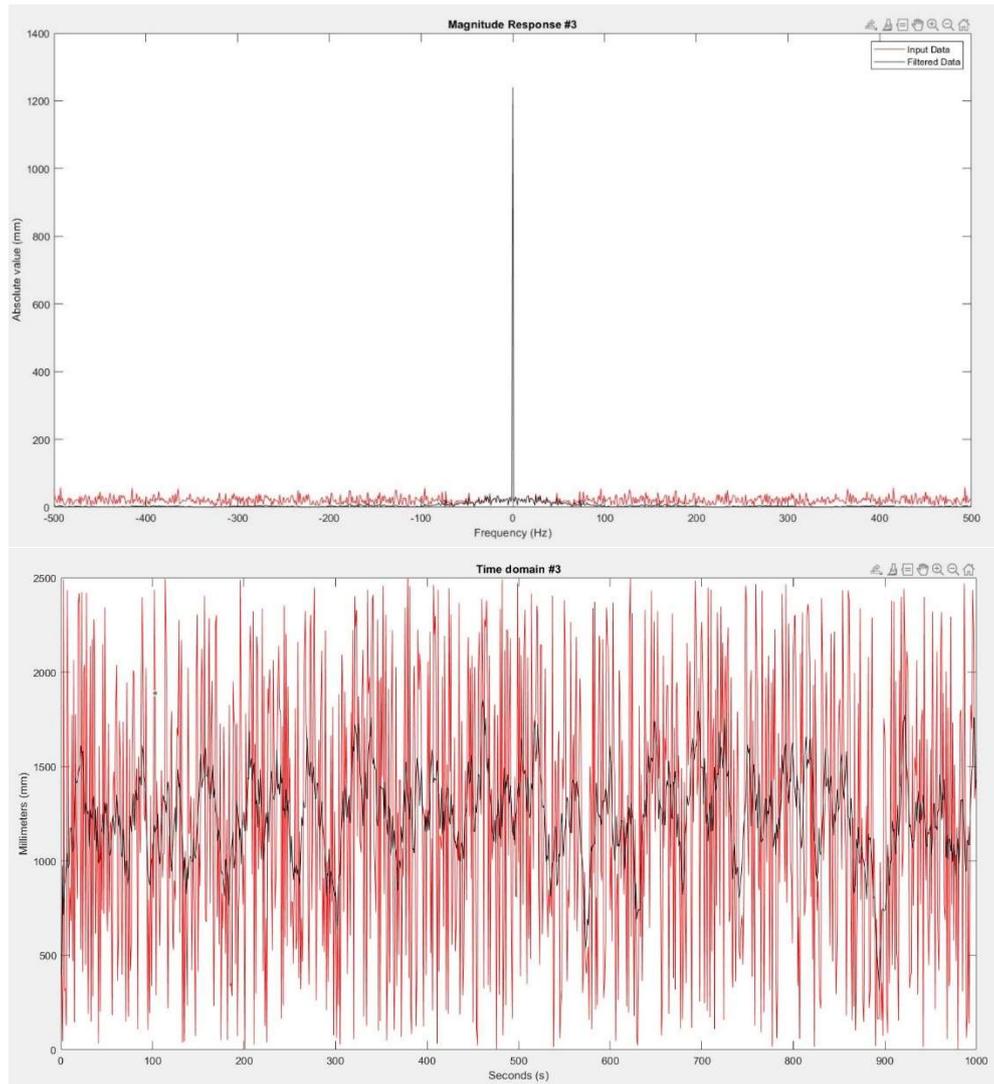


Figure 2.22: Digital filtering method C, frequency and time domains outputs

- Method A: the standard non-ideal low pass filter presents challenges in both the frequency and time domains. It fails to achieve acceptable results, as some non-relevant harmonics remain unfiltered in the frequency domain, leading to errors in the time domain measurements. The root cause of this issue lies in the non-ideality of the applied filter;

- Method B: the parameterized moving-average filter produces highly favorable outcomes in both the frequency and time domains. It effectively filters out non-relevant harmonics in the frequency domain, resulting in significant improvements in the accuracy of time domain measurements;

- Method C: the smooth moving-average filter. This method is similar to the B one but still presents challenges in both the frequency and time domains. It fails to achieve acceptable results, as some non-relevant harmonics remain unfiltered in the frequency domain, leading to errors in the time domain measurements.

After conducting a complete comparison of the results, method B was chosen as the candidate for the testing phase due to its superior performance in terms of the accuracy of time domain measurements. In particular, the initial measurement samples will be excluded from the dataset, as the filter's effectiveness becomes evident only after 100s from the beginning of the initial data acquisition loop. In general, a standard 1-D digital filter, described by the relation $y = filter(b, a, x)$, is utilized to filter the input data x using a rational transfer function defined by the numerator and denominator coefficients b and a . Among various filtering methods, the moving-average filter is commonly employed for smoothing noisy data. This filter slides a window of length $windowSize$ along the data, calculating averages of the data within each window. The following difference equation characterizes a moving-average filter for a vector x :

$$y(n) = \frac{1}{windowSize} (x(n) + x(n-1) + \dots + x(n - (windowSize - 1)))$$

The assigned values for the parameters b and a are the ones below:

```
windowSize = 100;
b = (1/windowSize)*ones(1,windowSize);
a = 1;
dataAll_filtered_2 = filter(b,a, dataAll);
```

Figure 2.23: Code snippet which implements the parameterized moving-average filter (method B)

Once the filtering process is applied to the data, the subsequent step involves computing both the variables D_{ij} and L_{ij} ,

$$D_{ij} = \frac{1}{n} \sum_{k=1}^n sample_k$$

$$L_{ij} = \sqrt{D_{ij} \cos(\vartheta_{ij}) \cos(\rho_{ij})}$$

which will be stored in the corresponding *lidar_distance_acquisitions_buffer* and *constrained_distances* matrices. L_{ij} is one of the most important information needed for the algorithm to work, since it serves the purpose of:

- managing the LiDAR movement algorithm;
- managing the 2D plot data visualization.

Once L_{ij} has been computed, the LiDAR spatial manipulation algorithm can be utilized. The primary objective of this section of the code is to save and set the new angular configuration $(\rho, \vartheta, \varphi)_{ij}$ to the gimbal, enabling the correct acquisition process to proceed. This involves considering the arbitrary terrain scanning path chosen during the design phase, the technical constraints imposed by the gimbal motors and the spatial constraints imposed by the pre-defined acquisition area. By effectively managing these aspects, the algorithm ensures accurate data collection and mapping throughout the scanning process.

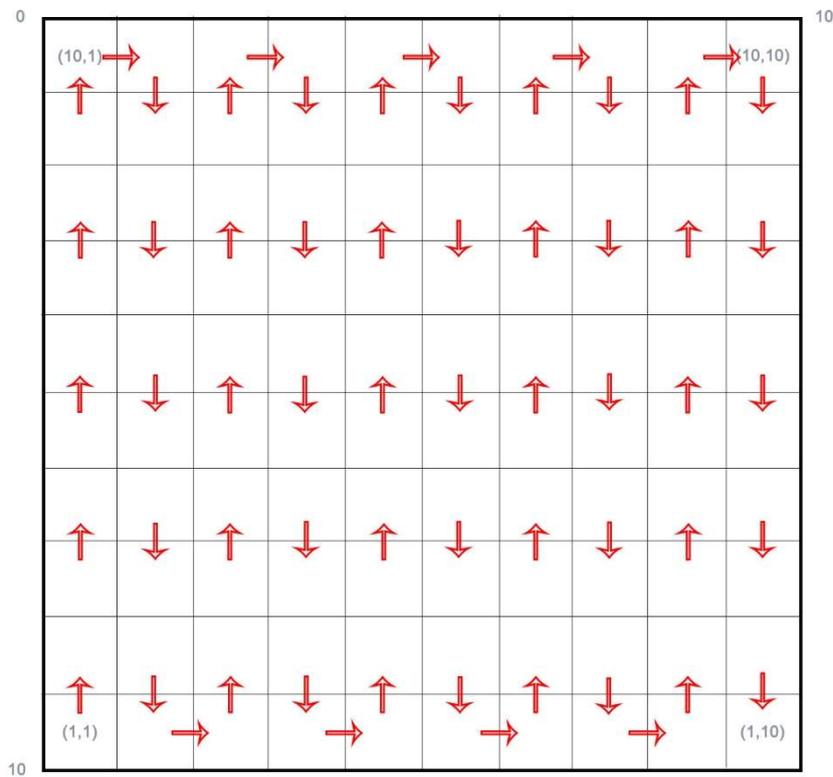


Figure 2.24. Arbitrary terrain scanning path chosen at the design level: the numbers shown inside the square (e.g., (10,1)) indicate the position of the data within the indices of the matrix.

In particular, the LiDAR spatial manipulation algorithm can be visualized as shown in the following flow chart:

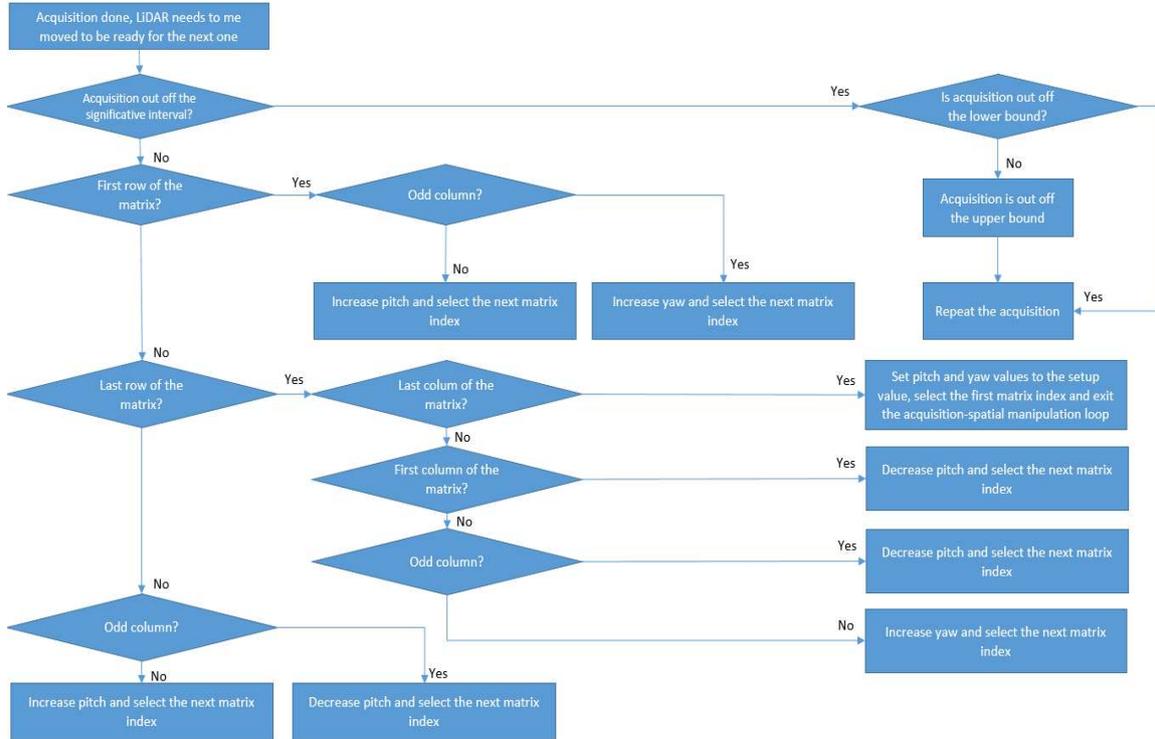


Figure 2.25. LiDAR spatial manipulation algorithm flow chart

The described processes of data points acquisition and LiDAR position manipulation continues to be executed until both the matrices *lidar_distance_acquisitions_buffer* and *constrained_distances* are fully populated, ensuring that there are no remaining zero values within it.

2.2.4.5 Data visualization

Once both the matrices *lidar_distance_acquisitions_buffer* and *constrained_distances* are completely filled, it is possible to proceed with the data visualization step. The idea behind the visualization logic is that for every data point D_{ij} a distinct color is selected, according to the value of L_{ij} . This assortment of colored data points aims to construct a visual representation of potential obstacles in front of the vehicle.

The logic behind the coloration process is straightforward: in a 2D plane, comprised of a grid with 10 columns and 10 rows, 100 equidistant data points g_{ij} are visualized. The coloration of these points is determined by their respective distance values L_{ji} compared to different distances thresholds, already decided at design level, so that the color intensity is inversely proportional to this distance: in other words, closer points are depicted with darker shades, while farther points are portrayed with lighter shades. A white color is provided accounting for corner cases where the measurement might be outside the inferior and superior thresholds lim_inf and lim_sup . This coloring scheme provides a visual representation that vividly conveys distance information across the 2D grid. In summary, if an obstacle exists within the acquisition area, the coloring will indicate a darker shade in the region where the obstacle is situated, contrasting with the coloration that would have been observed without the obstacle. There are a total of 22 distance thresholds, ranging from 1m (lim_inf) up to 19m (lim_sup). Each threshold in the interval [1025,1475]cm corresponds to a 25cm increment. Mathematically, the inequalities are as follows ($constrained_distances(j, i) = L_{ji}$ where $j, i = 1, 2, \dots, 10$):

$$L_{ji} < lim_inf$$

$$lim_inf \leq L_{ji} \ \&\& \ L_{ji} \leq 1025$$

.

.

$$1475 < L_{ji} \ \&\& \ L_{ji} \leq lim_sup$$

$$L_{ji} > lim_sup$$

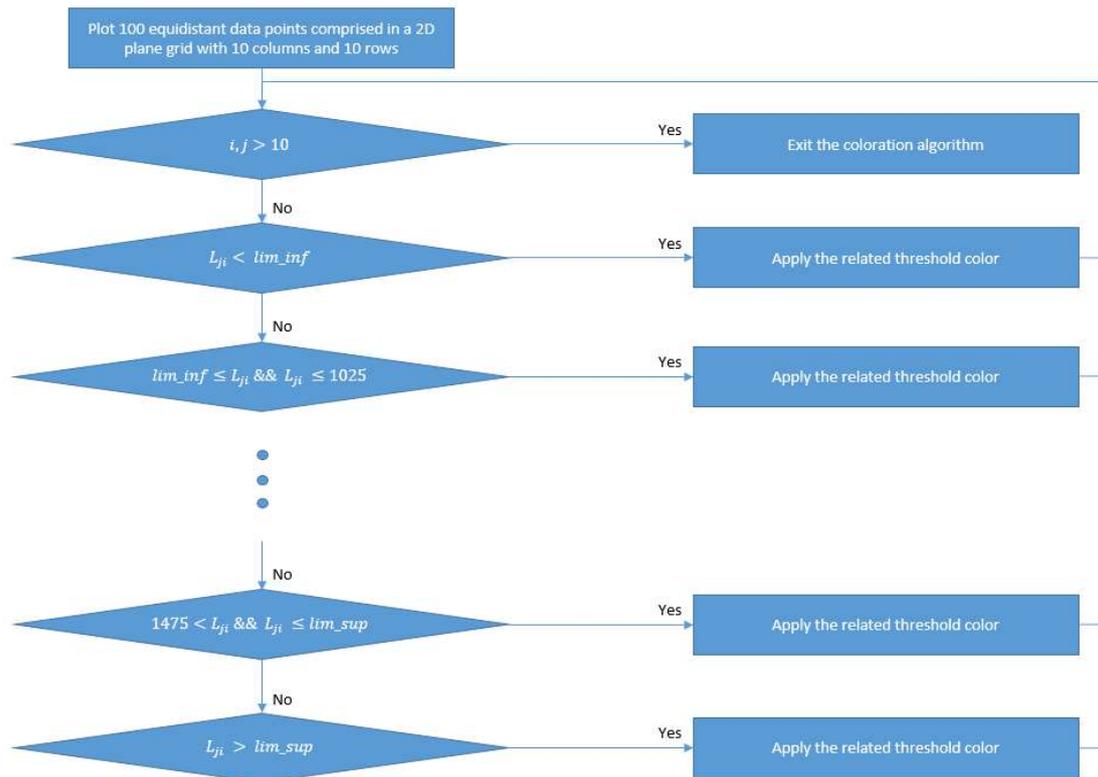


Figure 2.26. Coloration algorithm flow chart

Below is shown an illustration depicting the coloration logic applied to a dataset comprising 100 randomly generated data points, with distances ranging from 0cm to 2000cm. This representation aligns with the predetermined coloring scheme established during the design phase. It is important to note that the generated image contains no scientifically meaningful information due to the random nature of the data.

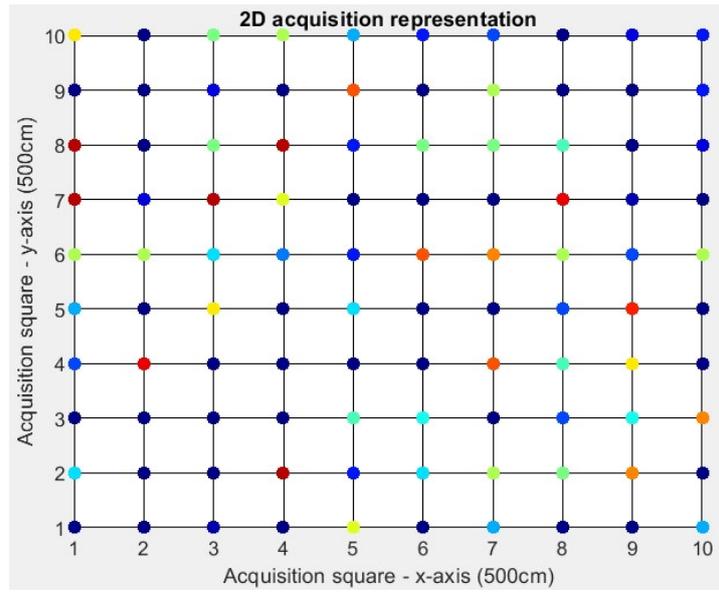


Figure 2.27. Coloration logic applied to 100 randomly generated data points representation, with distances ranging from 0cm to 2000cm, in a 2D plane

3 Finalizing and testing

This part will go into the final steps of the project. The testing methodologies used to evaluate the project's effectiveness and reliability will be described in detail. Furthermore, the final test results will be examined and compared to the project's requirements to guarantee compliance.

3.1 System's refinement

To prepare the system for testing, it must first be installed on the testing vehicle, which is an electric scooter. The electric scooter under consideration for the test is the KUGOO S1 Pro. This micromobility vehicle was chosen because it adapts well to the testing dynamics and because it was already available, i.e. no purchase was required. Moreover, the handlebar's design perfectly fit the sensing system footprint, allowing for simple installation.



Figure 3.0. KUGOO S1 Pro electric scooter

The installation requirements are the following:

- the sensing system was securely affixed at the center of the handlebar using some strong adhesive tape. This approach ensured a secure fixture;
- the system requires two independent power sources to function properly: one for the STM32F4 board and another for the gimbal. The first one was directly linked to the PC and drew a 5V supply, whereas the second one was connected to an electrical outlet through a specific power adapter with a voltage range of 14V to 16V.



Figure 3.1. Complete sensing system installed on the electric scooter

In alignment with the principles established in the earlier chapters, the sensor was positioned at a height of 1m. The sensor's orientation was carefully adjusted to achieve a pointing distance as close as possible to the theoretical stopping distance of 10m, which in fact coincide with the lower bound of the acquisition region area.



Figure 3.2. Visual representation of the sensor's slight inclination angle

3.2 Calibration

The resulting complete system can be considered ready to be tested as it was calibrated accurately. In particular, the LiDAR features an auto-calibration procedure called “Bias correction”. This process initializes upon powering on, gathering baseline data. It then computes corrections based on this data to counter measurement errors influenced by environmental factors like temperature and atmospheric pressure. These corrections are subsequently applied to all future measurements to maintain accuracy across diverse environments.

The gimbal, like the LiDAR, has its own calibration procedure. Calibration of the accelerometers on the IMU modules, in particular, is critical for optimal gimbal function. The STorM32 supports two calibration methods, "1-point" and "6-point," and the first was chosen for simplicity. The Windows GUI “o323BGCTool_v096” completely automates the procedure.

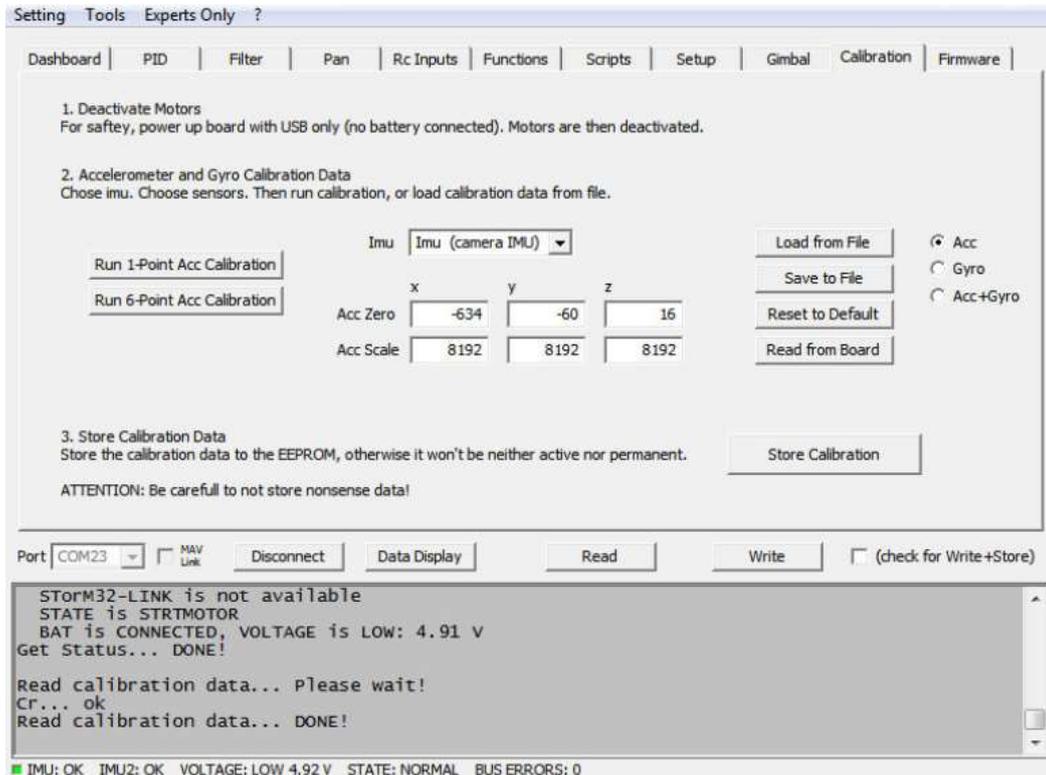


Figure 3.3. Windows GUI example – Calibration parameters configuration section

Calibrating the gimbal is quite important, as it corrects device-specific accelerometer inaccuracies, which are critical for accurate data representation and camera leveling. Calibration has a big impact on horizon tilt since it aligns the accelerometer of the Z-axis with the one of the gimbal for better precision. In terms of gyro calibration, the offsets of the gyros are generally calculated during gimbal startup, a few seconds after power up. As a result, the gimbal must be kept at rest throughout this time. This is by far the greatest choice and should be utilized whenever possible.

3.3 Testing and evaluation

The goal of the testing activity is to determine whether the actual system has been improved by the new adjustments. The primary goal is to verify that the 2D acquisition operates correctly. In other words, the acquisition system must be capable of capturing a dataset reflecting the acquisition area, and the coloring logic must be applied correctly as

determined at the design stage. This thesis does not provide the software logic for distinguishing obstacles in the acquisition dataset.

The following test configuration was taken into account:

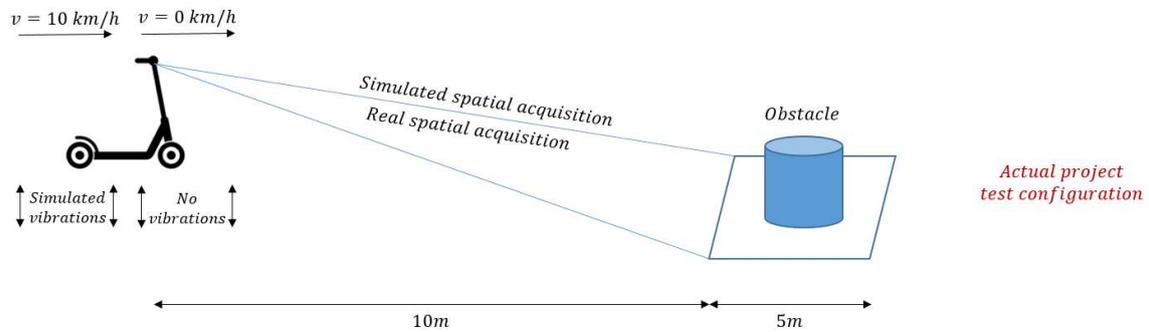


Figure 3.4. Test configuration

To evaluate this scenario, two specific metrics were considered:

- area of acquisition measurements precision: the ability of the sensing system to capture a dataset which actually reflects the acquisition area;
- robustness: how resilient the sensing system is to vibrations.

Three specific instances were considered in order to acquire the necessary information:

- 1) Scenario #1, simulated case:
 - a. $v = \text{const} = 10 \text{ km/h}$;
 - b. simulated road vibrations;
 - c. simulated suspension system and tires;
 - d. no gimbal dampening action;
 - e. simulated acquisitions.

- 2) Scenario #2, simulated case:
 - a. $v = \text{const} = 10 \text{ km/h}$;
 - b. simulated road vibrations;
 - c. simulated suspension system and tires;
 - d. simulated gimbal dampening action;
 - e. simulated acquisitions.

- 3) Scenario #3, real case:
- a. $v = 0 \text{ km/h}$;
 - b. no road vibrations;
 - c. no suspension system and tires;
 - d. real gimbal dampening action;
 - e. real acquisitions.

All the three instances were applied to each testing phases detailed below. The testing procedure begins with a basic distinction between free road and obstacle detection sensing. Following that, new features were gradually implemented, with each stage systematically testing the sensor's performance.

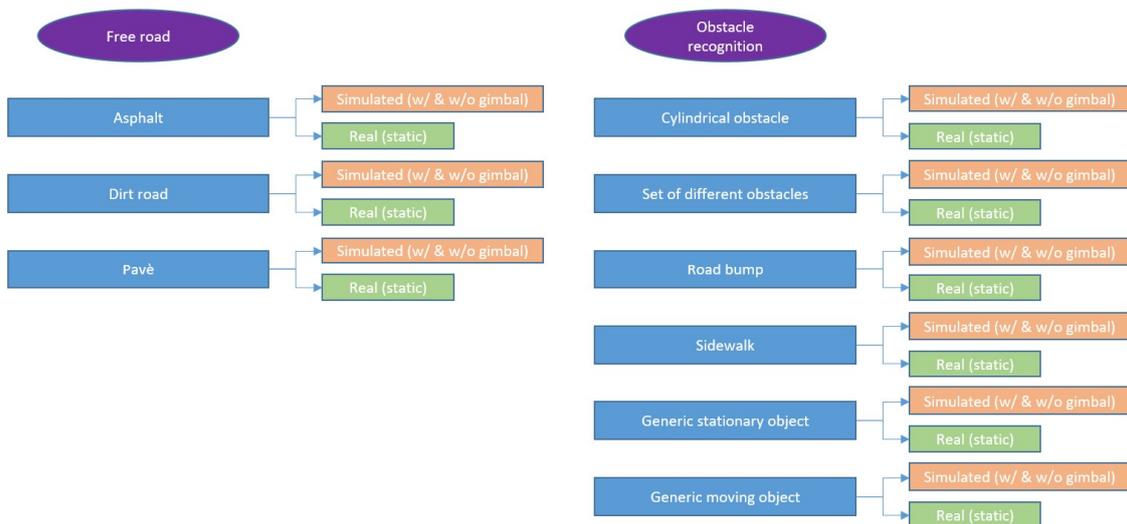


Figure 3.5. Testing activity phases

The testing phase aims to generate simulated datasets for scenarios #1 and #2 based on the real dataset acquired through scenario #3.

3.3.1 Simulation models

In the real-world environment of land vehicles, road vibrations are dynamic movements and oscillations felt as the vehicles drives over the road surface. These vibrations are

caused by the interaction of several elements, including road irregularities, vehicle attributes and external environmental conditions. When a vehicle travels over uneven terrain, its tires and suspension system interact with the textures of the road dampening the effect of those vibrations.



Figure 3.6. Road surface degraded by potholes

Understanding and correctly modeling these road vibrations, as well as the dampening effects of the suspension system, tires and gimbal, is crucial for progressing through the testing process.

3.3.1.1 Road vibrations model

In the real world, road vibrations are influenced by various factors resulting in essentially random vibrations. Gaussian noise, a form of random noise with a normal distribution, is well-suited for simulating these unpredictable vibrations as it can account for road imperfections, texture changes and external disturbances that contribute to sensor vibrations. Normal distributions, commonly known as Gaussian distributions, play a significant role in statistics and find wide application in the natural, social and engineering sciences for modeling real-valued random variables with unknown distributions. This is particularly relevant because of the central limit theorem, which asserts that, given certain conditions, the average of multiple samples from a random variable with finite mean and variance becomes a random variable. As the sample size increases, this random variable converges toward a normal distribution. As a result,

physical quantities that result from the summation of numerous independent processes, like measurement errors, tend to exhibit distributions that closely resemble the normal distribution [36, 37, 38, 39, 40, 41].

The probability density function φ of a Gaussian random variable x is given by

$$\varphi_{\mu,\sigma^2}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

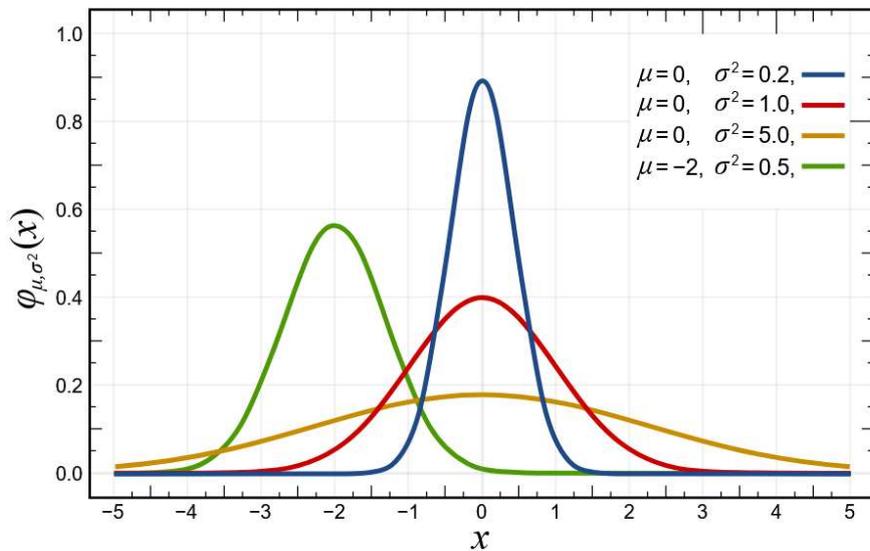


Figure 3.7. Families of probability density functions of a random variable x . The red line is the standard normal distribution.

where $x \in \mathbb{R}$ represents the random noise, $\mu \in \mathbb{R}$ is the mean of the distribution, $\sigma^2 \in \mathbb{R}_{>0}$ is the variance and so $\sigma \in \mathbb{R}_{\geq 0}$ the standard deviation. Considering the standard normal distribution (red line), approximately 70% of the x values fall within the range $[\mu - \sigma, \mu + \sigma]$, and about 95% within the range $[\mu - 2\sigma, \mu + 2\sigma]$. The use of Gaussian noise adds realism to the simulated data, replicating the variability of real-world conditions. Its flexibility in adjusting parameters, like variance and mean, allows for precise control over the intensity and nature of the simulated vibrations [42].

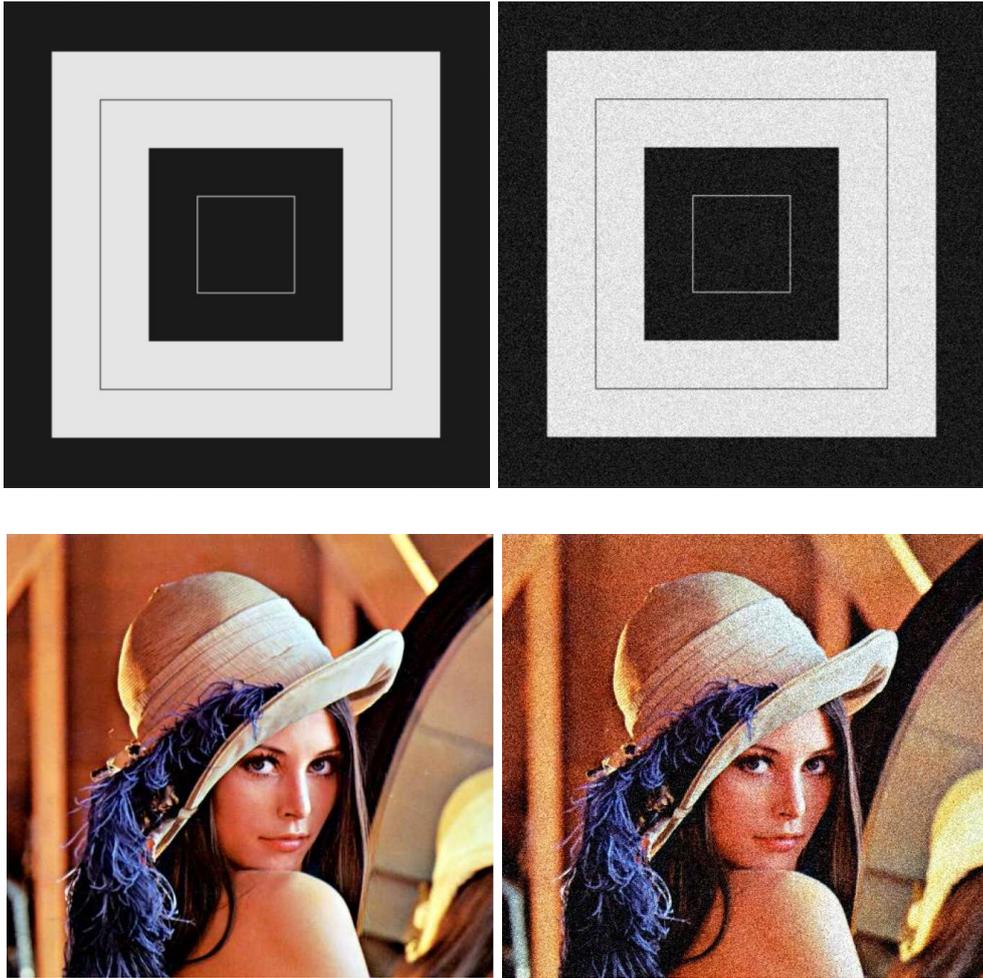


Figure 3.8. A visualization of Gaussian noise in grayscale (above) and RGB (below) digital pictures. The images on the left are the original; the ones on the right are affected by Gaussian noise ($\sigma=25$, $\mu=0$).

Road class	σ^2	μ
Very good	1 ... 4	1
Good	4 ... 16	1
Average	16 ... 64	1
Poor	64 ... 256	1
Very poor	256 ... 1024	1

Figure 3.9. Road profile parameters (σ^2 , μ) for different road classes according to ISO8608

In the context of this thesis, the following variance and mean values are used to characterize and simulate the three types of road surfaces under study [34, 35]:

- $\sigma^2 = 64$, $\mu = 1$ (asphalt);
- $\sigma^2 = 256$, $\mu = 1$ (pavè);
- $\sigma^2 = 1024$, $\mu = 1$ (dirt road).

3.3.1.2 Suspensions and tires models

Vehicle suspensions are sophisticated systems designed to reduce the transmission of vibrations from the road to the vehicle's frame. The primary purpose is to improve ride quality by reducing discomfort and shocks felt by the driver. These systems include a variety of mechanical components, such as springs and dampers (shock absorbers), and they frequently include extra control systems for fine-tuning. It is critical to remember that suspensions are largely dampening systems rather than generators of vibrations when simulating road vibrations.



Figure 3.10. An illustration of a pair of scooter rear shock absorber suspensions

In general, the suspensions of an electric scooter cannot be regarded as a second-order, linear time-invariant system (see appendix A, section A, for a detailed description): they

are made up of complex mechanical components such as springs, shock absorbers, and, in some cases, electronic control systems. These components have the potential to introduce nonlinearity into the system's response:

- shock absorbers' nonlinear effect: shock absorbers, which are crucial components of suspensions, frequently display nonlinear behavior. For example, in the presence of strong forces, shock absorber resistance may grow excessively;
- spring characteristics: suspension springs frequently exhibit nonlinear behavior, particularly when subjected to large deformations;
- electronic control action: some electric scooters include electronic suspension control systems, which might result in nonlinearity.



Figure 3.11. Examples of typical front and rear electric scooter suspension systems

Certain assumptions must be met in order to view the scooter's shock absorbers as a second-order, linear time-invariant system and so to utilize an exponential damping model without compromising the validity of the analysis and simulation:

- 1) system linearity: the shock absorbers have to exhibit linear behavior in terms of excitations. In other words, the damping force (the resistance given by the shock absorbers) must be proportional to the relative velocity (e.g., compression or extension velocity) between the shock absorber components;

- 2) stationarity: the system must be stationary, which means that its behavior must be constant over time. During the period of interest for the analysis, the shock absorbers on the scooter should retain a constant damping behavior;
- 3) damping effectiveness: the shock absorbers must have significant effects on the vibrations to be represented. In other words, their dampening should be important in terms of road surface vibrations;
- 4) damping ratio satisfaction: the damping ratio must be chosen in accordance with the actual behavior of the shock absorbers since it controls the rate at which the damping mechanism minimizes vibrations.

Because all of the previous premises are met in the case of the thesis, the suspension can be identified as a second-order, linear time-invariant system. In particular, with respect to the KUGOO S1 Pro:

- 1) the system's linearity is legitimate since the suspensions tend to exhibit linear behavior in response to excitations under normal conditions. The damping forces should be proportionate to the relative velocities of the suspension components;
- 2) stationarity is a legitimate assumption because, during regular driving on an asphalt road, the suspensions should exhibit consistent damping behavior during the analysis time;
- 3) the damping effect of the suspensions is significant with respect to the vibrations induced by the road surface;
- 4) the damping factor will be chosen to be suitable with the real behavior of the suspensions interacting with an asphalt road, since the testing activity will be conducted primarily on asphalt.

In addition, because tires are studied in conjunction with suspensions and must comply to the same linearity parameters as suspensions, which have been considered verified, the

damping effect of the tires can be taken into account during the damping ratio value selection [43, 44]. Since the linearity of both the suspension system and the tires has been verified, it is possible to assume that the combination of the two systems can be modelled as a second-order, linear time-invariant system with equation of motion

$$m \frac{d^2 x}{dt^2} + c \frac{dx}{dt} + kx = 0$$

where:

- $m = m_s + m_t$ is the mass of the system;
- $c = c_s + c_t$ is the damping coefficient;
- $k = k_s + k_t$ is the spring constant (*stiffness*);
- $F(t) = 0$ (*steady state*) represents the case where the system is not subjected to external forces or inputs, and its behavior is only determined by its mass, spring constant and damping coefficient.

With this premises, it is possible to use a decreasing exponential function, such as an exponentially damped sinusoid, to represent this damping action. With respect to the vibrations induced by the road, this approximation is a simpler, more effective compromise between realism and computing simplicity. Exponential decay models are frequently used to represent systems in which an effect decreases in proportion to its current magnitude [43, 44]. The general equation may be represented as

$$y(t) = Ae^{-\lambda t} \cos(\omega t - \varphi)$$

where:

- $y(t)$ is the instantaneous amplitude at time t ;
- A is the initial amplitude at $t = 0$;
- λ is the decay rate (t^{-1});
- φ is the phase angle at $t = 0$;
- ω is the angular frequency;
- f is the number of cycles per time unit ($= \frac{\omega}{2\pi}$);

- ζ is the damping ratio, non-dimensional characterization of the decay rate relative to the frequency of the system. It is a measure of how rapidly the oscillations or vibrations are reduced or dampened over time.

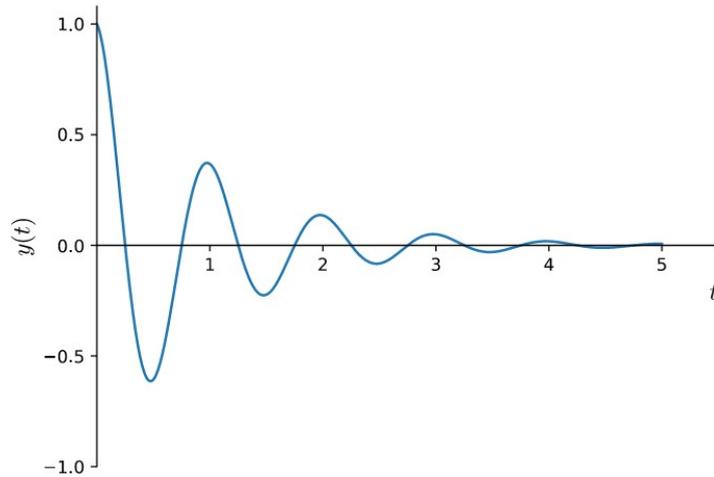


Figure 3.12. Example of a damped sinusoidal wave represented as the function $y(t) = e^{-t}\cos(2\pi t)$

The second order system under study can be identified as underdamped, characterized by the mathematical inequality $0 \leq \zeta < 1$. This is supported by the fact that, in presence of vibrations, it reaches a stable state without continuous oscillations after a certain amount of time. Knowing that the following mathematical relationship exists

$$\zeta = \frac{c}{2\sqrt{km}} = \frac{\lambda}{\sqrt{\lambda^2 + \omega^2}}$$

it is possible to choose a specific value of ζ to simulate the damping behavior of the system under analysis with respect to vibrations [43, 44].

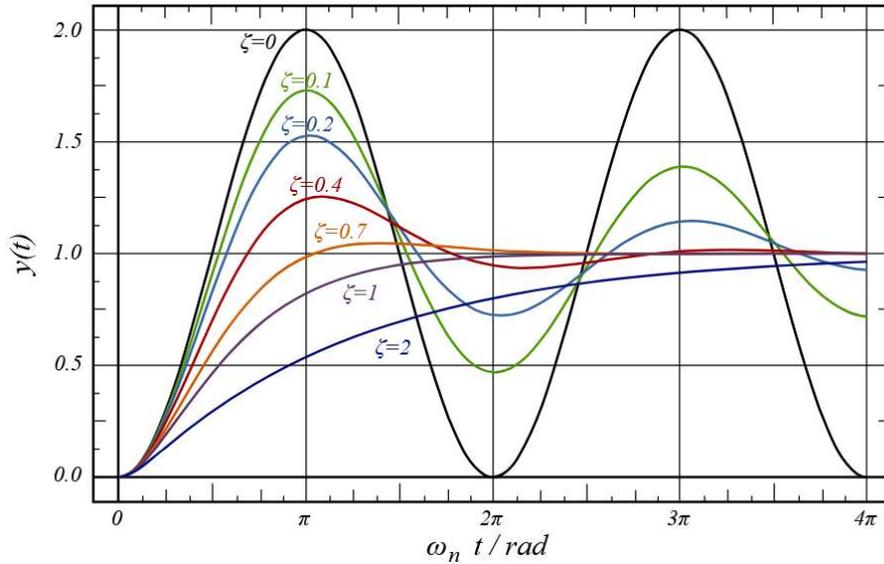


Figure 3.13. The effect of varying damping ratio on a second-order system

In the context of this thesis, the damping ratio of $\zeta = 0.35$ will be used for simulation, since the range of values that reflects real damping behavior when considering an electric scooter is between $[0.2, 0.7]$. This figure implies that the strength of the vibrations is decreasing in a balanced way due to the damping action of both the suspension system and tires. Adjusting the damping factor allows to fine-tune the damping amount for certain simulation circumstances [43, 44].

3.3.1.3 Gimbal dumping effect model

A gimbal is a mechanical device consisting of interconnected rings or ball joints that enable movement along multiple axes. Its primary purpose is to maintain the orientation of an object, such as a camera or sensor, regardless of the movement of the supporting object. This is achieved by allowing the gimbal to freely spin along at least three orthogonal axes. Gimbals leverage gyroscopic principles to preserve object alignment while reducing the impact of vibrations and undesired movements.



Figure 3.14. An illustration of a generic 2-dof gimbal

Gimbals, on the other hand, are complex systems characterized by non-linearity due to their structures and the involved physical forces. In particular:

- 1) complex mechanical structure: gimbals are made up of a complicated mechanical framework. Because interactions between distinct pieces might respond differently under different situations, this complexity invariably creates non-linearity in the overall response;
- 2) electric motors and control: non-linear properties are common in gimbal motors, especially during rapid accelerations or decelerations;
- 3) structural flexibility: the structural flexibility of the gimbal can influence its reaction in real-world settings. Elastic deformation or nonlinear movement of structural components might contribute to the system's overall nonlinearity;
- 4) friction and resistances: mechanical frictions and resistances can exhibit non-linear behavior, particularly when subjected to repeated use or changes in environmental conditions;

- 5) electromagnetic interactions: because of the complexity of magnetic forces and field changes, gimbals that use magnets or involve electromagnetic interactions can produce non-linearity behaviors.

To simulate the damping effect given by the gimbal, it is possible to follow a similar approach used for suspensions and tires in order to demonstrate the assumption of second-order, linear time-invariant system [43, 44]. However, gimbals are complicated systems with distinct properties, and the analysis would necessitate a large number of resources that were not available at the time the thesis was completed. Furthermore, numerous scientific sources suggest that it is possible to assume the gimbal's linear behavior without the analysis losing its scientific purpose [45, 46]. In the context of the thesis, considering the gimbal as a second-order, linear time-invariant system, it becomes feasible to compute and incorporate the damping contribution introduced by this system. This can be achieved by adjusting the damping ratio, which is currently set to $\xi = 0.8$. This figure is supported by the fact that, in theory, a properly tuned gimbal should prevent any form of vibration from affecting the acquisition system.

Once all the simulation models have been configured and fine-tuned, the next step is the transition to the testing phase. The testing phase will proceed as outlined below:

- generating the dataset for scenario #3 through real acquisition;
- simulating the dataset for scenario #1 starting from the dataset of scenario #3;
- simulating the dataset for scenario #2 starting from the dataset of scenario #3.

Before to start, an ideal acquisition matrix is presented below for reference. In particular, the illustration depicts a sorted and simulated dataset comprising 100 data points, with distances ranging from 1000cm (last matrix row) to 1500cm (first matrix row). This representation aligns with the predetermined coloring scheme established during the design phase. The generated image represents a scenario in which there are no disturbances, such as vibrations, and no obstacles recognized by the sensing device (free road condition).

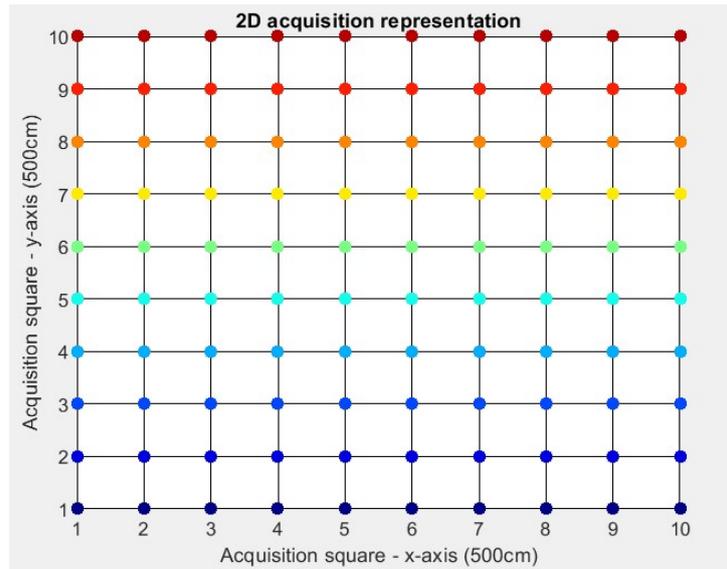


Figure 3.15. 100 sorted and simulated data points representation, with distances ranging from 1000cm to 1500cm, in a 2D plane

3.3.2 Free road

In the following paragraphs, the sensor's response across three different road types will be tested using an internal courtyard within the Teoresi company: asphalt, dirt road and pavè (cobblestone). No obstacles were introduced during this phase as the goal was to evaluate the accuracy of the coloring logic alongside both simulated and real system acquisitions.

3.3.2.1 Asphalt

Asphalt, widely found in urban areas, stands as the primary road surface for transportation networks. This scenario represents the most common and practical road texture in terms of real-world applications for the sensing system under test.



Figure 3.15. Asphalt road

Displayed below are representative images of the three instances that were considered for the testing activity:

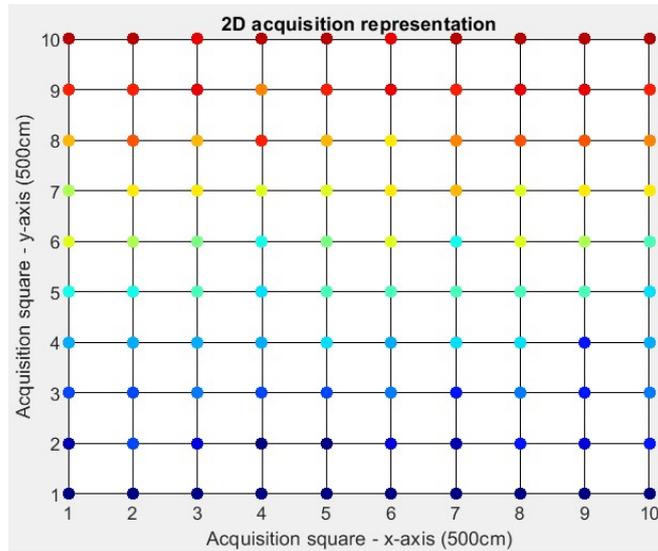


Figure 3.16. Asphalt phase: scenario #1

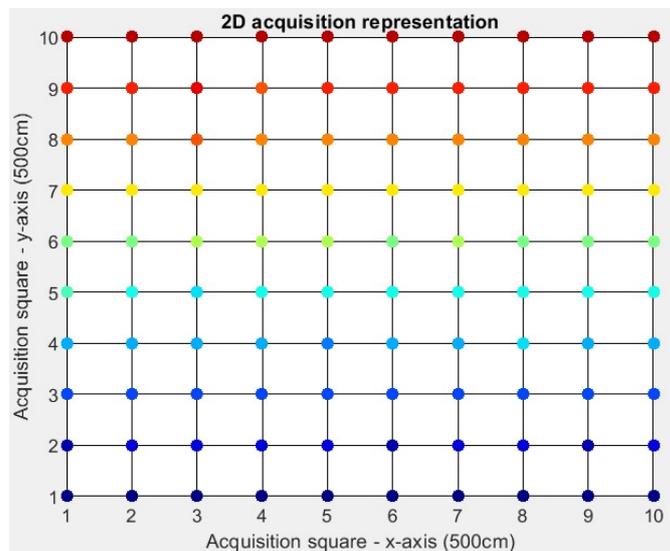


Figure 3.17. Asphalt phase: scenario #2

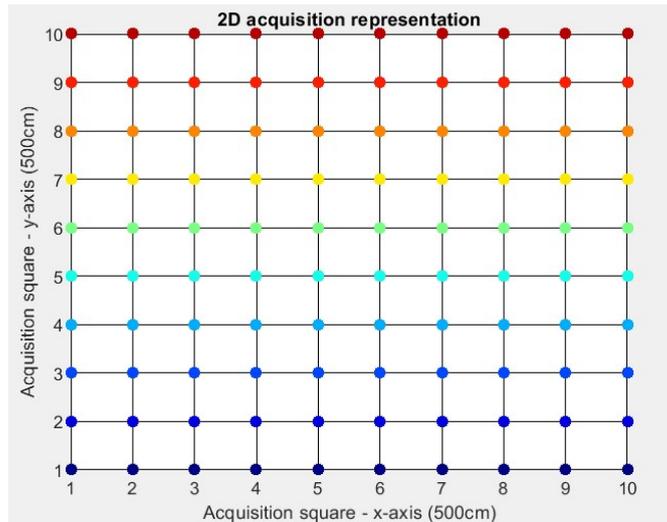


Figure 3.18. Asphalt phase: scenario #3

Scenario #1:

- the acquisition appears significantly distorted compared to the ideal one (figure 3.15) due to the effect of vibrations on the acquisition system. The coloration logic fails to be applied, potentially leading to a false recognition of an obstacle;
- the system proves to be highly susceptible to vibrations.

Scenario #2:

- the acquisition appears clean compared to the ideal one. The coloring logic is sufficiently respected, avoiding a case of false recognition of an obstacle;
- the system appears resistant to vibrations.

Scenario #3:

- the acquisition is identical to the ideal one. The coloring logic is perfectly respected. There is no possibility of encountering a case of false recognition of an obstacle;

- the resilience to vibrations cannot be evaluated as this scenario is of a static nature.

3.3.2.2 Dirt road

Dirt roads, commonly observed in rural or less urbanized regions, serve as fundamental pathways for various transport routes. Although this type of road surface is less common in urban areas, it is crucial to consider it for testing purposes as it may give rise to significant vehicle oscillations that could considerably impact the measurement values.



Figure 3.19. Dirt road

Displayed below are representative images of the three instances that were considered for the testing activity:

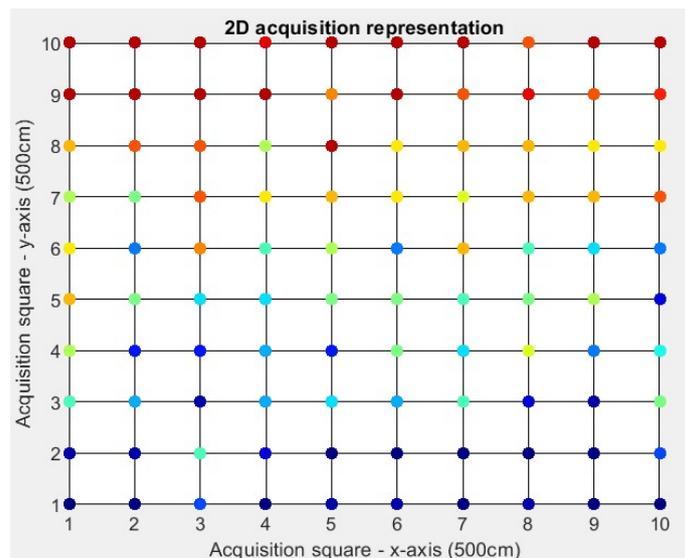


Figure 3.20. Dirt road phase: scenario #1

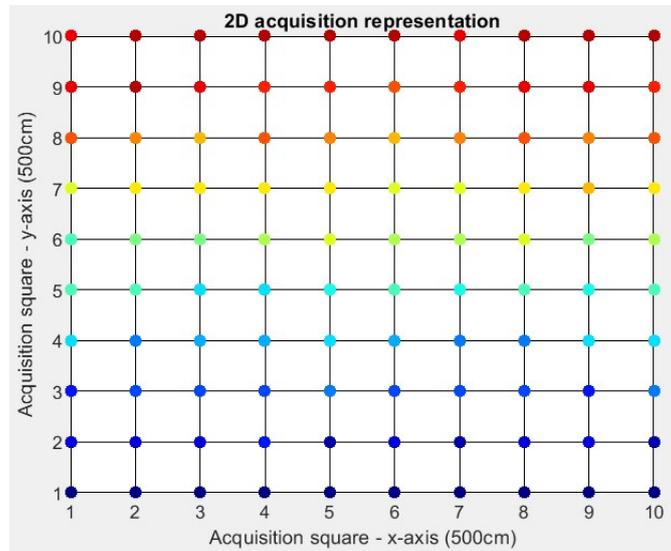


Figure 3.21. Dirt road phase: scenario #2

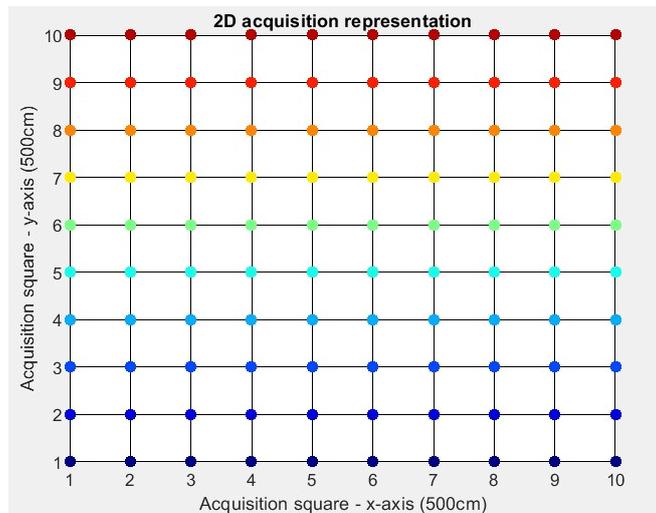


Figure 3.22. Dirt road phase: scenario #3

Scenario #1:

- the acquisition appears totally distorted compared to the ideal one (figure 3.15) due to the effect of vibrations on the acquisition system. The coloration logic fails to be applied, potentially leading to a false recognition of an obstacle;

- the system proves to be highly susceptible to vibrations.

Scenario #2:

- the acquisition appears clean compared to the ideal one. The coloring logic is sufficiently respected, avoiding a case of false recognition of an obstacle;
- the system appears resistant to vibrations.

Scenario #3:

- the acquisition is identical to the ideal one. The coloring logic is perfectly respected. There is no possibility of encountering a case of false recognition of an obstacle;
- the resilience to vibrations cannot be evaluated as this scenario is of a static nature.

3.3.2.3 Pavè

Pavè refers to a traditional outdoor road surface constructed from stone or porphyry cubes. It's a common feature in pedestrian centers and city hubs, potentially encountered in real-world scenarios by the sensing system. These cubes typically measure around ten centimeters. Despite its irregular surface, this paving style may induce considerable vehicle oscillations, mainly because the wheels of electric scooters are smaller in comparison.



Figure 3.23. Pavè

Displayed below are representative images of the three instances that were considered for the testing activity:

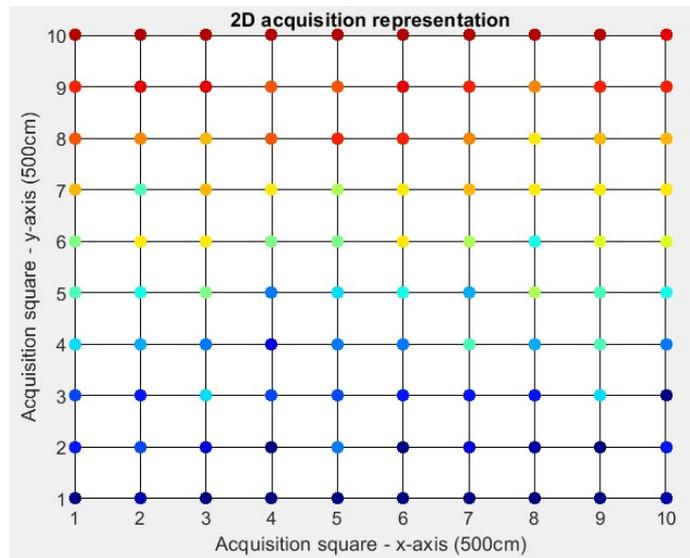


Figure 3.24. Pavè phase: scenario #1

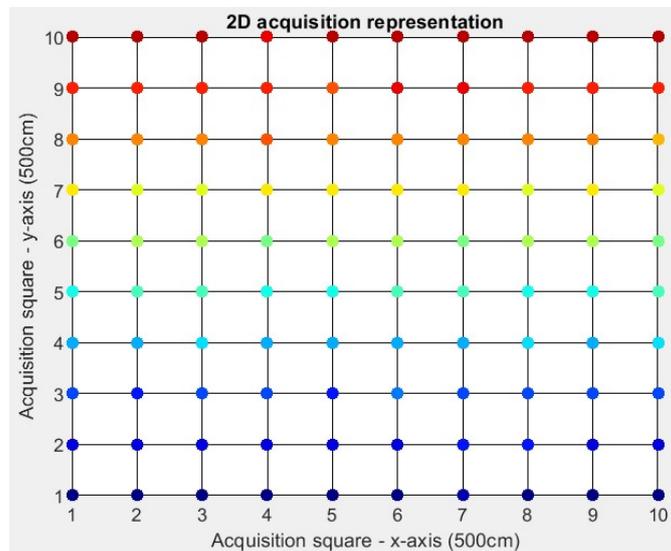


Figure 3.25. Pavè phase: scenario #2

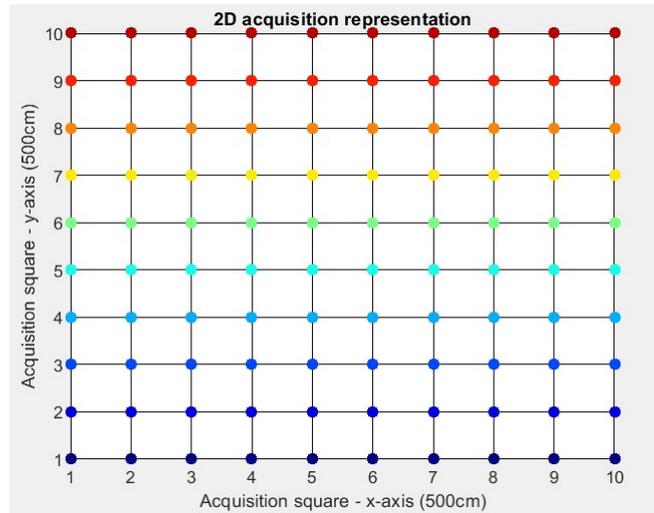


Figure 3.26. Pavè phase: scenario #3

Scenario #1:

- the acquisition appears totally distorted compared to the ideal one (figure 3.15) due to the effect of vibrations on the acquisition system. The coloration logic fails to be applied, potentially leading to a false recognition of an obstacle;
- the system proves to be highly susceptible to vibrations.

Scenario #2:

- the acquisition appears clean compared to the ideal one. The coloring logic is sufficiently respected, avoiding a case of false recognition of an obstacle;
- the system appears resistant to vibrations.

Scenario #3:

- the acquisition is identical to the ideal one. The coloring logic is perfectly respected. There is no possibility of encountering a case of false recognition of an obstacle;

- the resilience to vibrations cannot be evaluated as this scenario is of a static nature.

3.3.3 Obstacle recognition

After analyzing the sensor's performance across different types of pavement, the focus shifts to obstacle recognition. The setup considered throughout the six consecutive analysis phases will be the asphalt pavement since it is the most common and practical road texture. The considered obstacles include:

- one cardboard box 26cm high;
- one cardboard box 34cm high;
- one cemented polystyrene cylinder 30cm high;
- two roll-ups 10cm high.



Figure 3.27. Different types of obstacles deployed during the testing activity

3.3.3.1 Cylindrical obstacle

The cylindrical object was intentionally placed approximately at the center of the acquisition area to test the sensor's accuracy and functionality. Displayed below are representative images of the three instances that were considered for the testing activity:

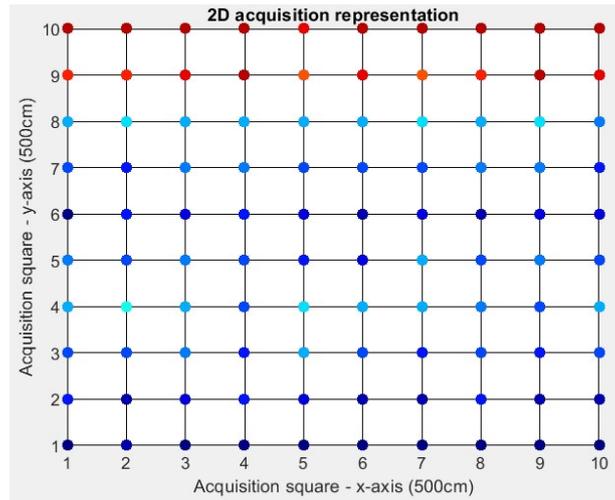


Figure 3.28. Obstacle recognition: Cylindrical obstacle phase: scenario #1

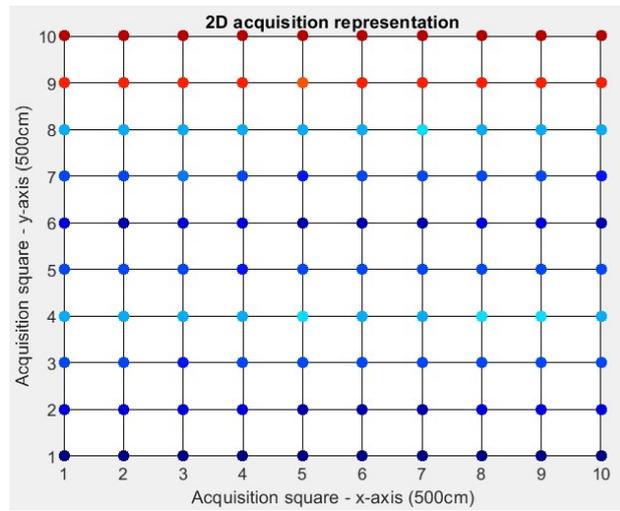


Figure 3.29. Obstacle recognition: Cylindrical obstacle phase: scenario #2

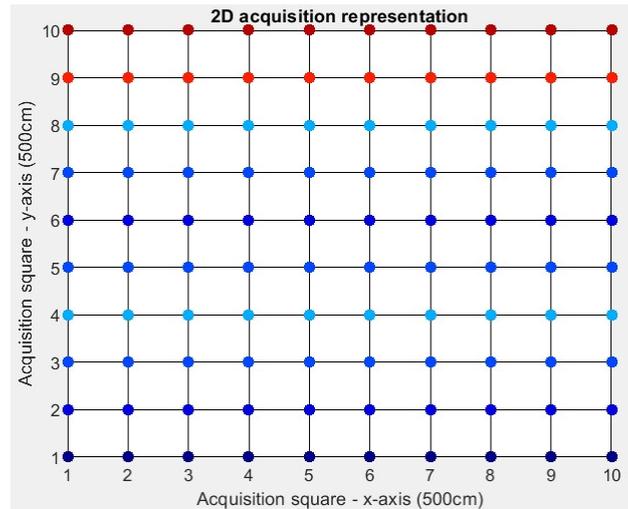


Figure 3.30. Obstacle recognition: Cylindrical obstacle phase: scenario #3

Scenario #1:

- the acquisition appears significantly distorted due to the effect of vibrations on the acquisition system. The coloration logic fails to be applied, potentially leading to a false recognition of an obstacle. In this scenario, the cylindrical obstacle cannot be detected by the system;
- the system proves to be highly susceptible to vibrations.

Scenario #2:

- the acquisition appears clean. The coloring logic is sufficiently respected, avoiding a case of false recognition of an obstacle. In this scenario, the cylindrical obstacle can be identified by the system (matrix rows 5-8);
- the system appears resistant to vibrations.

Scenario #3:

- the acquisition is error-free. The coloring logic is perfectly respected. There is no possibility of encountering a case of false recognition of an obstacle. In this

scenario, the cylindrical obstacle can be identified by the system (matrix rows 5-8);

- the resilience to vibrations cannot be evaluated as this scenario is of a static nature.

3.3.3.2 Set of different obstacles



Figure 3.31. Obstacle recognition: set of different obstacles

The two objects were intentionally placed at the beginning and at the end of the acquisition area to test the sensor's accuracy and functionality. Displayed below are representative images of the three instances that were considered for the testing activity:

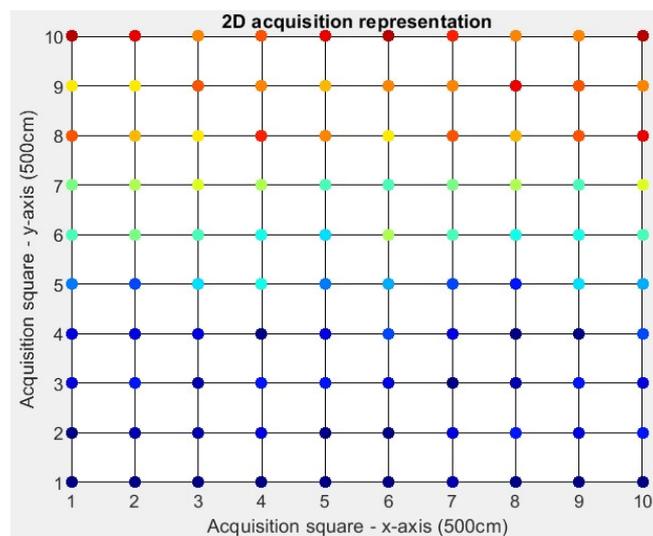


Figure 3.32. Obstacle recognition: Set of different obstacles phase: scenario #1

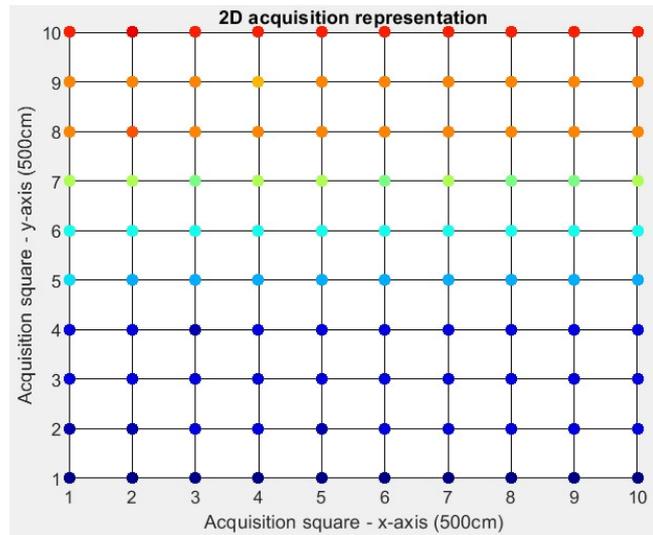


Figure 3.33. Obstacle recognition: Set of different obstacles phase: scenario #2

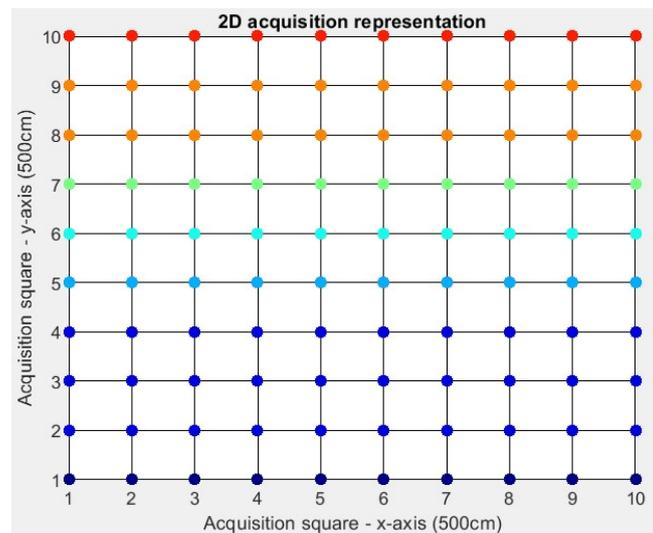


Figure 3.34. Obstacle recognition: Set of different obstacles phase: scenario #3

Scenario #1:

- the acquisition appears significantly distorted due to the effect of vibrations on the acquisition system. The coloration logic fails to be applied, potentially leading to

a false recognition of an obstacle. In this scenario, the two obstacles cannot be detected by the system;

- the system proves to be highly susceptible to vibrations.

Scenario #2:

- the acquisition appears clean. The coloring logic is sufficiently respected, avoiding a case of false recognition of an obstacle. In this scenario, the two obstacles can be identified by the system (matrix rows 1-4 and 8-10);
- the system appears resistant to vibrations.

Scenario #3:

- the acquisition is error-free. The coloring logic is perfectly respected. There is no possibility of encountering a case of false recognition of an obstacle. In this scenario, the two obstacles can be identified by the system (matrix rows 1-4 and 8-10);
- the resilience to vibrations cannot be evaluated as this scenario is of a static nature.

3.3.3.3 Road bump

Since a road bump was not available in the Teoresi courtyard, the two roll-ups were used next to each other to simulate this case.



Figure 3.35: Generic road bump

The “road bump” was intentionally placed approximately at the center of the acquisition area to test the sensor's accuracy and functionality. Displayed below are representative images of the three instances that were considered for the testing activity:

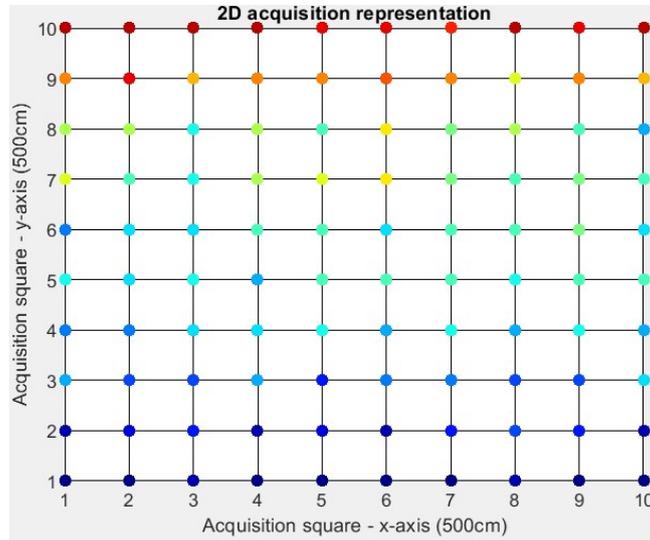


Figure 3.36. Obstacle recognition: Road bump of 10cm phase: scenario #1

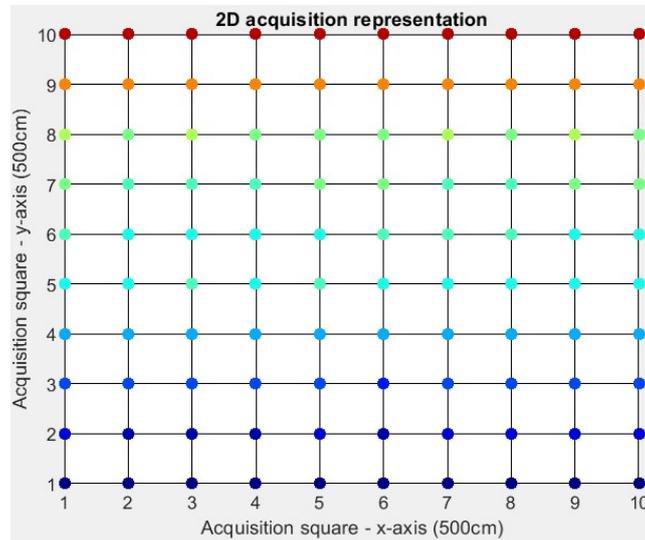


Figure 3.37. Obstacle recognition: Road bump of 10cm phase: scenario #2

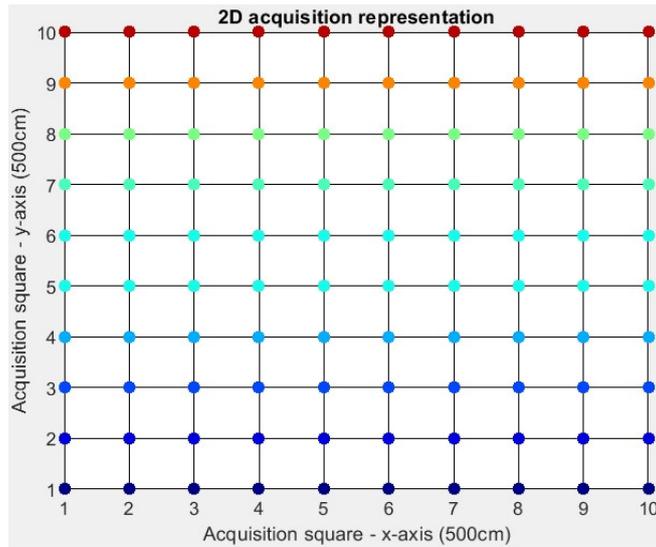


Figure 3.38. Obstacle recognition: Road bump of 10cm phase: scenario #3

Scenario #1:

- the acquisition appears significantly distorted due to the effect of vibrations on the acquisition system. The coloration logic fails to be applied, potentially leading to a false recognition of an obstacle. In this scenario, the “road bump” cannot be detected by the system;
- the system proves to be highly susceptible to vibrations.

Scenario #2:

- the acquisition appears clean. The coloring logic is sufficiently respected, avoiding a case of false recognition of an obstacle. In this scenario, the “road bump” can be identified by the system (matrix rows 6-8). However, based on the pictures, it seems that the system's sensitivity may not be enough or the coloring logic might not be finely calibrated to detect obstacles below a specific height threshold;
- the system appears resistant to vibrations.

Scenario #3:

- the acquisition is error-free. The coloring logic is perfectly respected. There is no possibility of encountering a case of false recognition of an obstacle. In this scenario, the “road bump” can be identified by the system (matrix rows 6-8). However, based on the pictures, similar considerations to those made for scenario #2 could be taken into account;
- the resilience to vibrations cannot be evaluated as this scenario is of a static nature.

3.3.3.4 Sidewalk

In the testing area, there was a sidewalk adjacent to the asphalt road. Consequently, the available sidewalk 15cm high was utilized for conducting the tests.



Figure 3.39. Sidewalk

The sensing system was intentionally positioned so that the sidewalk was approximately at the center of the acquisition area to test the sensor's accuracy and functionality. Displayed below are representative images of the three instances that were considered for the testing activity:

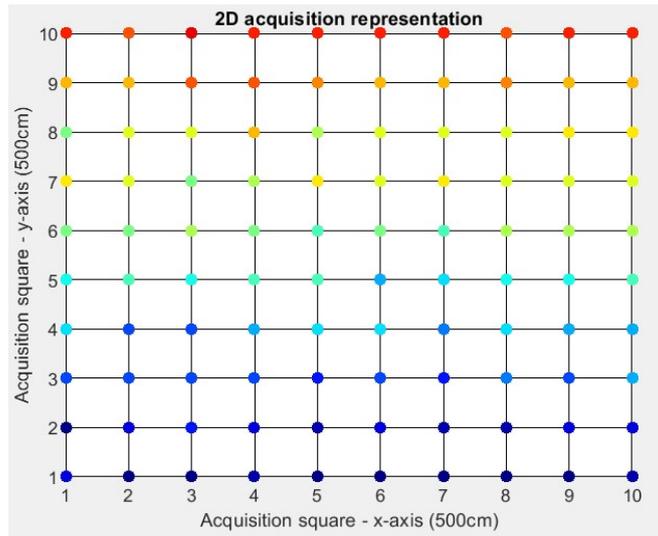


Figure 3.40. Obstacle recognition: Sidewalk phase: scenario #1

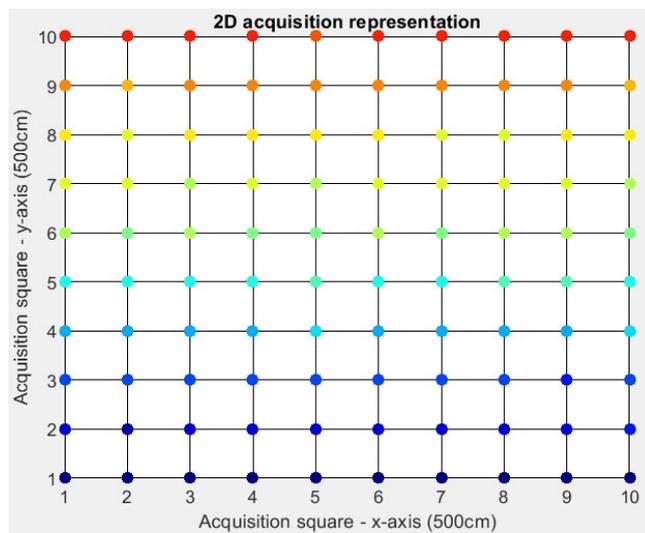


Figure 3.41. Obstacle recognition: Sidewalk phase: scenario #2

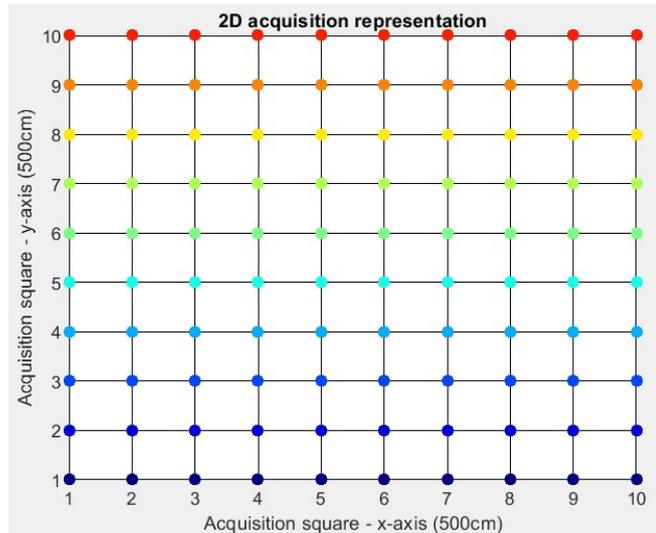


Figure 3.42. Obstacle recognition: Sidewalk phase: scenario #3

Scenario #1:

- the acquisition appears significantly distorted due to the effect of vibrations on the acquisition system. The coloration logic fails to be applied, potentially leading to a false recognition of an obstacle. In this scenario, the sidewalk cannot be detected by the system;
- the system proves to be highly susceptible to vibrations.

Scenario #2:

- the acquisition appears clean. The coloring logic is sufficiently respected, avoiding a case of false recognition of an obstacle. In this scenario, the sidewalk can be identified by the system (matrix rows 6-10). However, based on the pictures, it seems that the system's sensitivity may not be enough or the coloring logic might not be finely calibrated to detect obstacles below a specific height threshold;
- the system appears resistant to vibrations.

Scenario #3:

- the acquisition is error-free. The coloring logic is perfectly respected. There is no possibility of encountering a case of false recognition of an obstacle. In this scenario, the sidewalk can be identified by the system (matrix rows 6-10). However, based on the pictures, similar considerations to those made for scenario #2 could be taken into account;
- the resilience to vibrations cannot be evaluated as this scenario is of a static nature.

3.3.3.5 Generic stationary object

To simulate the recognition of a stationary obstacle, one cardboard box was positioned upright on its longer side, exceeding the sensor's height. The specific height of the obstacle is not relevant for the scope of the tests.



Figure 3.43. Generic stationary object

The generic stationary object was intentionally placed approximately at the center of the acquisition area to test the sensor's accuracy and functionality. Displayed below are representative images of the three instances that were considered for the testing activity:

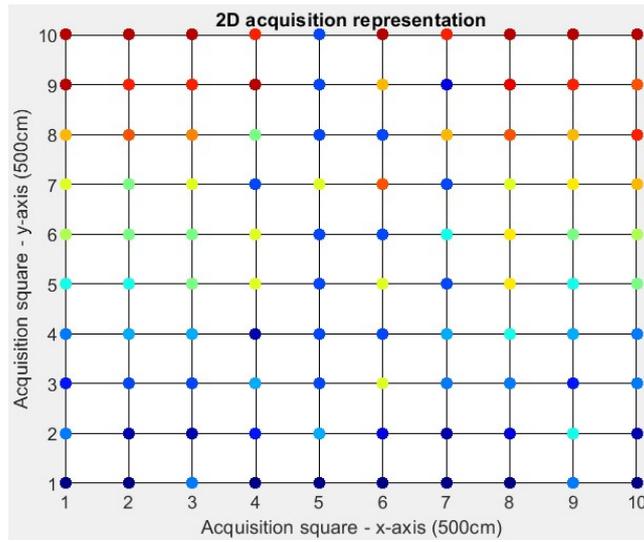


Figure 3.44. Obstacle recognition: Generic stationary object phase: scenario #1

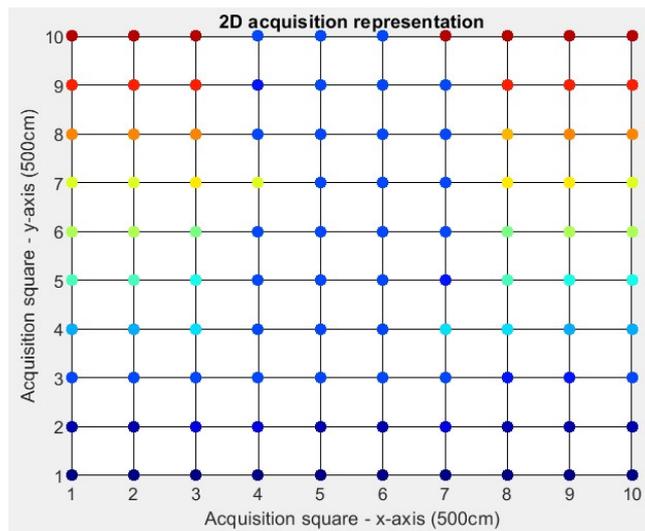


Figure 3.45. Obstacle recognition: Generic stationary object phase: scenario #2

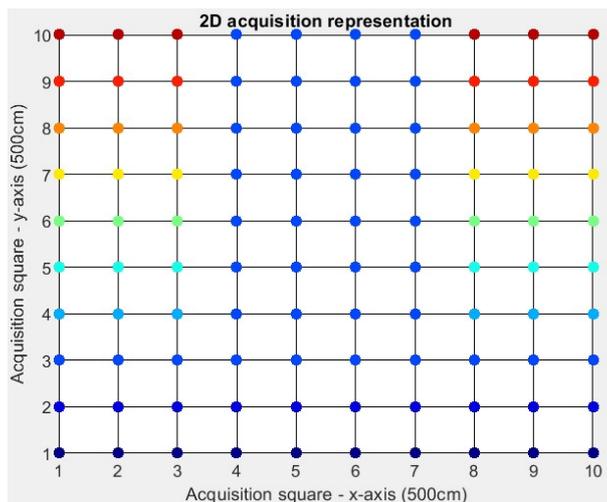


Figure 3.46. Obstacle recognition: Generic stationary object phase: scenario #3

Scenario #1:

- the acquisition appears significantly distorted due to the effect of vibrations on the acquisition system. The coloration logic fails to be applied, potentially leading to a false recognition of an obstacle. In this scenario, the generic stationary object cannot be detected by the system;
- the system proves to be highly susceptible to vibrations.

Scenario #2:

- the acquisition appears clean. The coloring logic is sufficiently respected, avoiding a case of false recognition of an obstacle. In this scenario, the generic stationary object can be identified by the system (matrix rows 4-10);
- the system appears resistant to vibrations.

Scenario #3:

- the acquisition is error-free. The coloring logic is perfectly respected. There is no possibility of encountering a case of false recognition of an obstacle. In this

scenario, the generic stationary object can be identified by the system (matrix rows 4-10);

- the resilience to vibrations cannot be evaluated as this scenario is of a static nature.

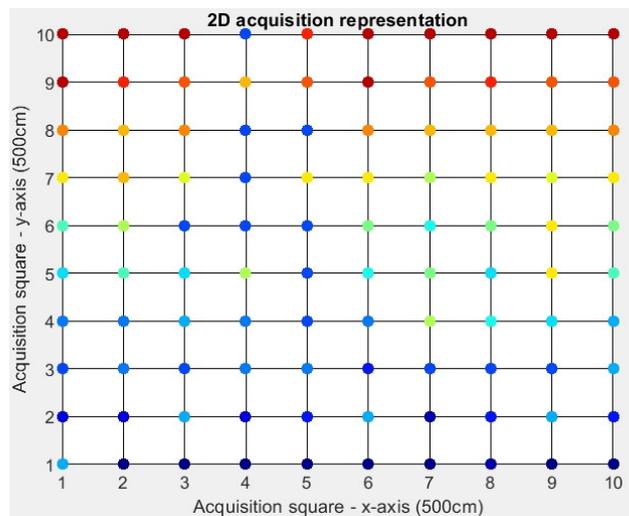
3.3.3.6 Generic moving object

To conduct this testing phase, an additional person was engaged. Specifically, the objective of this phase is to simulate a real person holding a cellphone in their right hand, walking perpendicular to the sensor's acquisition direction, thus evaluating the sensor's response in the presence of a moving obstacle.

Each scenario involved three different cases:

- Case #1: a person moving from right to left with respect to the acquisition system;
- Case #2: a person moving from left to right with respect to the acquisition system;
- Case #3: a person moving at the same speed as the measurement sensor's movement, from left to right within the acquisition area.

Displayed below are representative images of the three instances that were considered for the testing activity:



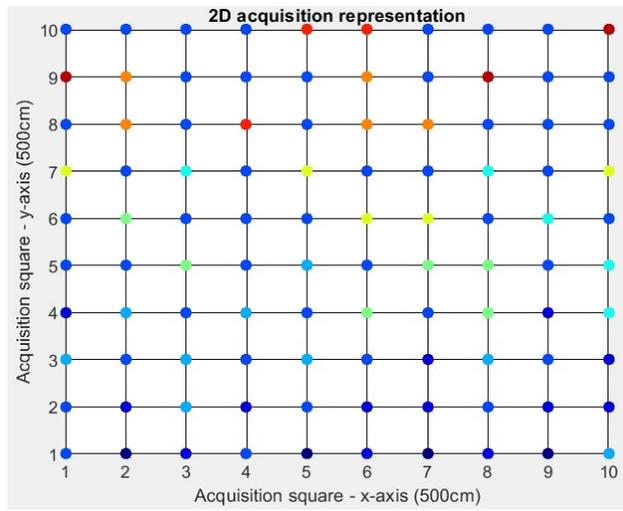
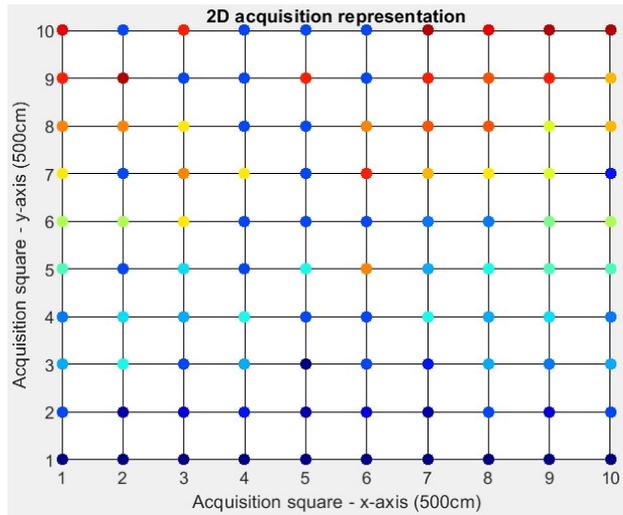
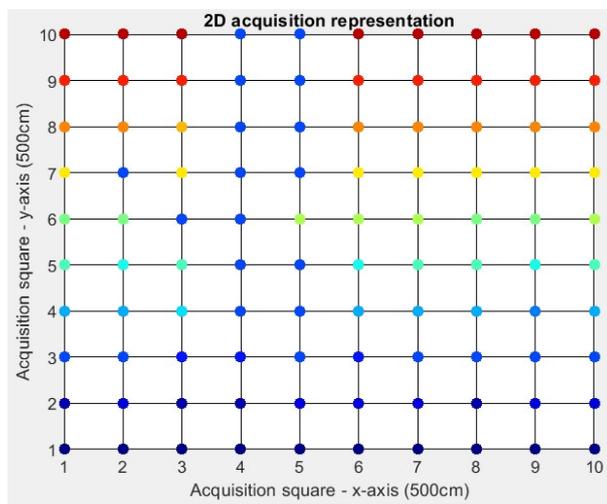


Figure 3.47. Obstacle recognition: Generic moving object phase: scenario #1



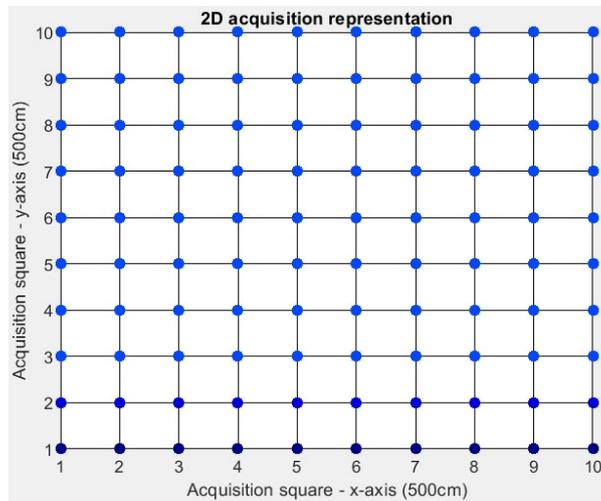
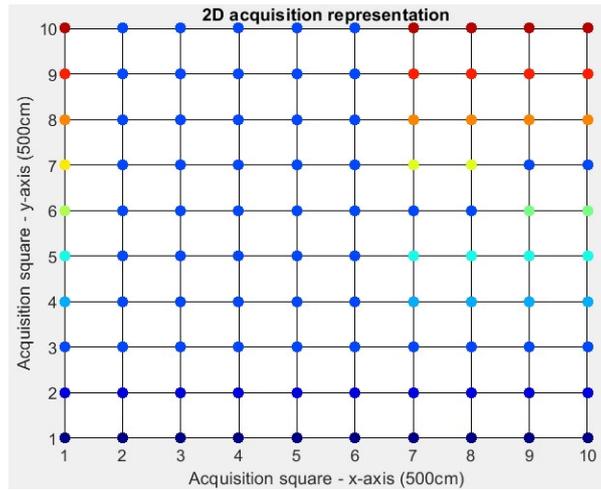
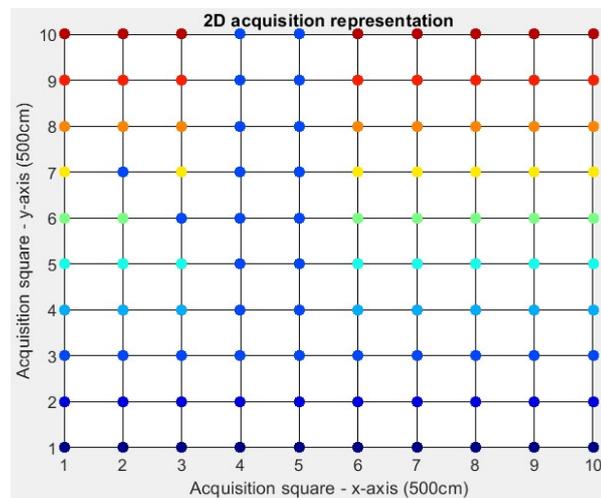


Figure 3.48. Obstacle recognition: Generic moving object phase: scenario #2



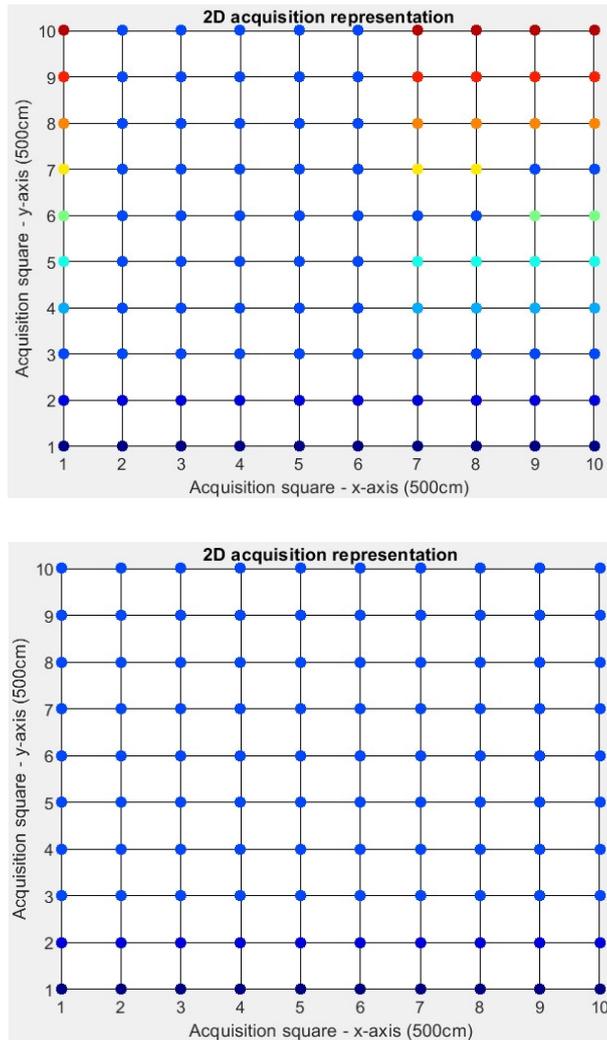


Figure 3.49. Obstacle recognition: Generic moving object phase: scenario #3

Scenario #1:

- the acquisitions appear significantly distorted due to the effect of vibrations on the acquisition system. The coloration logic fails to be applied, potentially leading to a false recognition of an obstacle. In this scenario, a generic moving object cannot be detected by the system in any of the three cases considered;
- the system proves to be highly susceptible to vibrations.

Scenario #2:

- the acquisition appears clean. The coloring logic is sufficiently respected, avoiding a case of false recognition of an obstacle. In this scenario, a generic moving object can be identified by the system. In particular:
 - o Case #1: the profile of the person holding the phone is identified in columns 2-5 of the matrix;
 - o Case #2: the profile of the person holding the mobile phone is identified in columns 2-10 of the matrix;
 - o Case #3: it is not possible to identify the profile of the person holding the cell phone since they move at exactly the same speed as the sensor;
- the system appears resistant to vibrations.

Scenario #3:

- the acquisition is error-free. The coloring logic is perfectly respected. There is no possibility of encountering a case of false recognition of an obstacle. In this scenario, a generic moving object can be identified by the system. In particular:
 - o Case #1: the profile of the person holding the phone is identified in columns 2-5 of the matrix;
 - o Case #2: the profile of the person holding the mobile phone is identified in columns 2-10 of the matrix;
 - o Case #3: it is not possible to identify the profile of the person holding the cell phone since they move at exactly the same speed as the sensor;
- the resilience to vibrations cannot be evaluated as this scenario is of a static nature.

3.4 Tests results and optimizations

The focus of this chapter is on reviewing the results of the tests and identifying potential optimizations. The chapter includes a complete examination of the data collected from the testing scenarios, with the goal of validating the system's performance under diverse conditions. Furthermore, this part explores potential optimizations, with the goal of improving the system's functionality and efficiency.

3.4.1 Validating the results

Before proceeding with the analysis of the obtained results it is necessary to verify their validity, especially considering a potential implementation of the system on a dedicated embedded system. For this purpose, the Model-in-the-loop technique was employed.

Model-in-the-loop (MIL) is a development and testing procedure used in the context of system design and validation, which is especially common in engineering, software development and control systems. The MIL technique involves running simulations or mathematical models of specific components or subsystems prior to actual hardware implementation. This enables early-stage evaluation, design verification and performance evaluation of the system's components in a safe and cost-effective virtual environment [47].

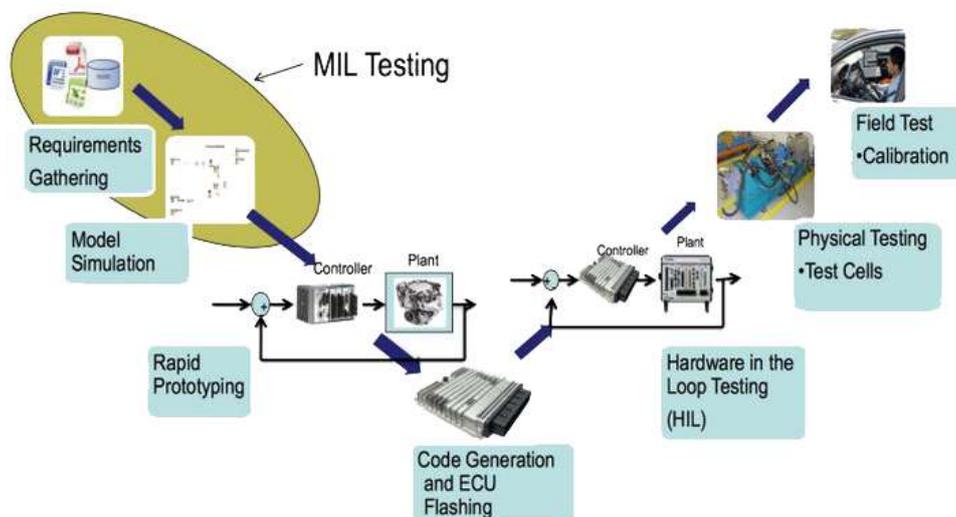


Figure 3.50. Embedded software development process (V-shape)

In this project's context, the outputs of the following software modules will be simulated and compared against those obtained through the script, with respect to the scenario #3 case:

- data processing;
- LiDAR spatial manipulation.

To begin, the procedure involves translating the segment of the script intended for testing into Simulink block language. The aim here is to derive a block description and verify its functionality. If the simulation of the model generates identical outcomes as obtained from executing the script, it confirms the success of the test [48].

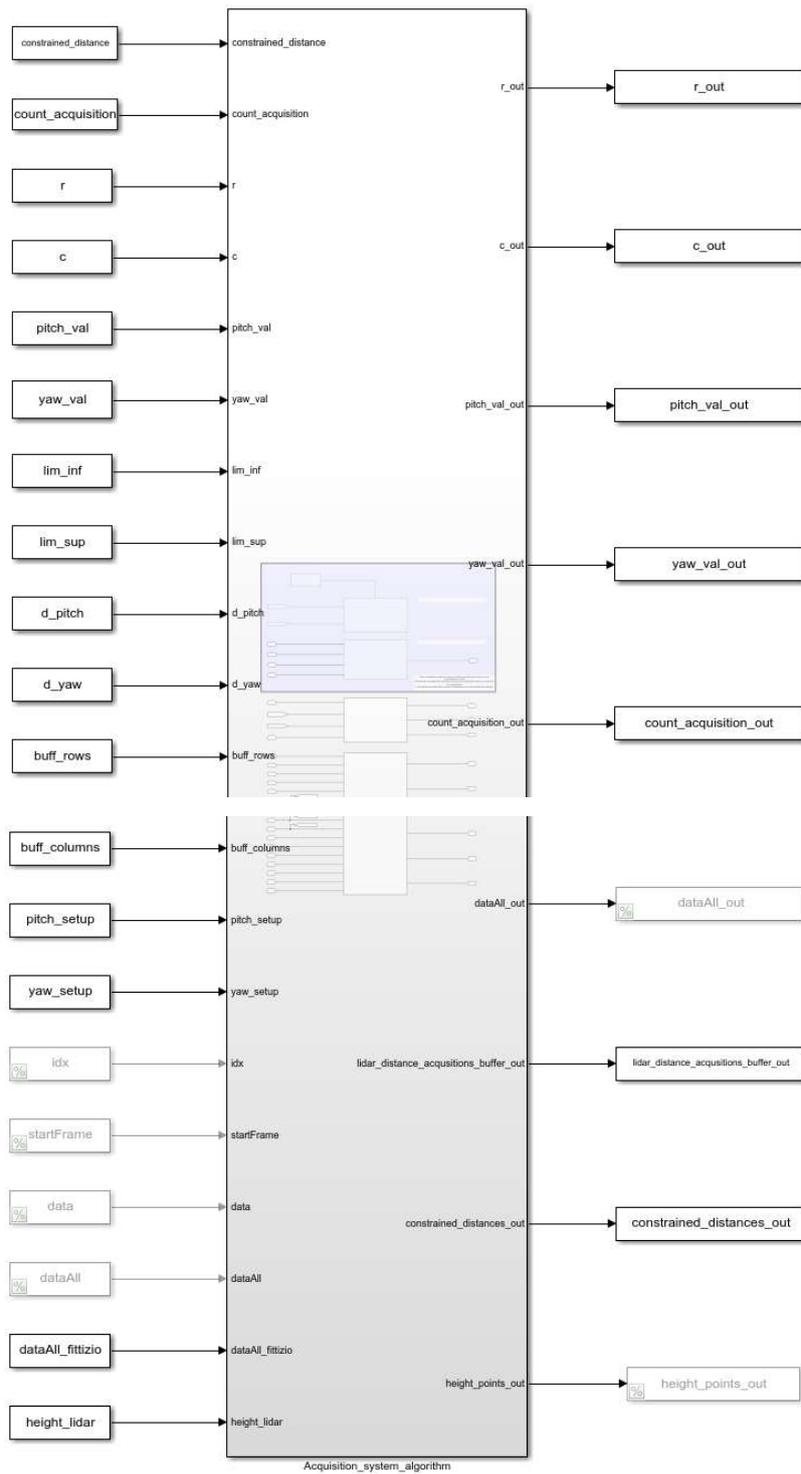


Figure 3.51. Acquisition system model, Simulink block language and MAAB guidelines

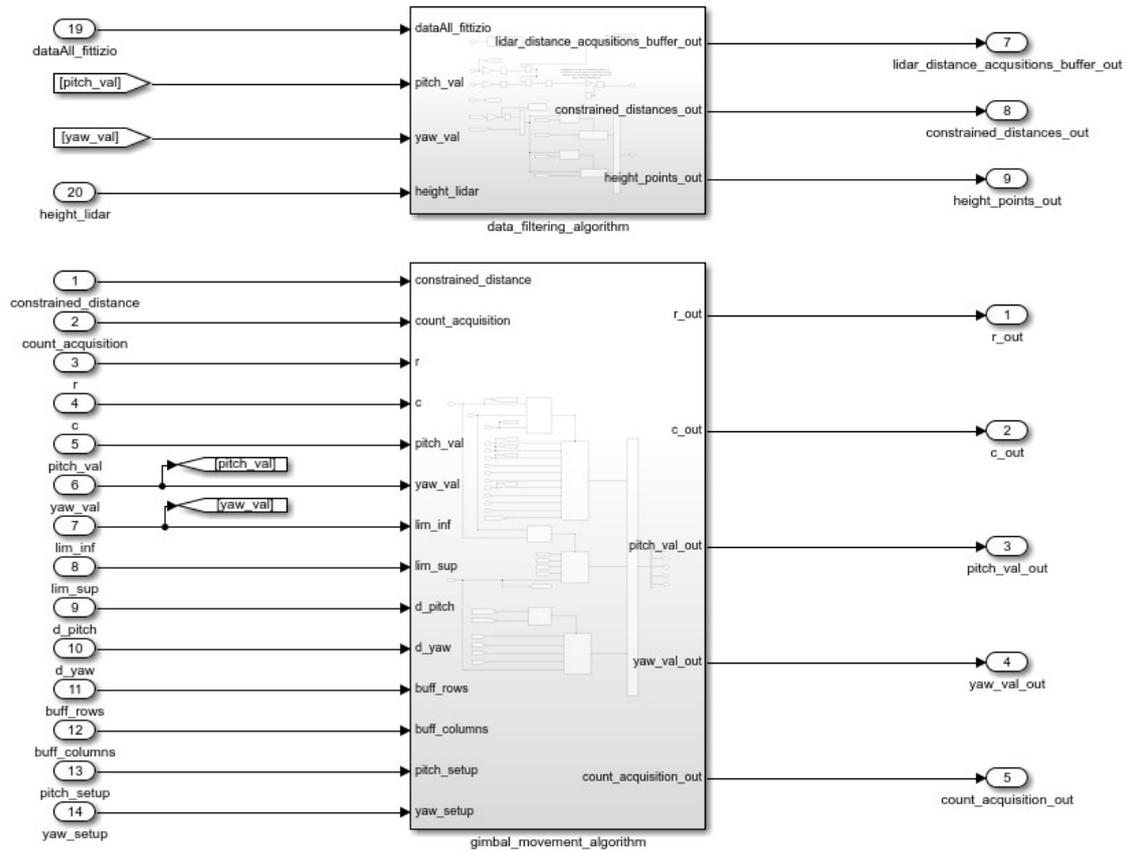


Figure 3.52. Data processing and LiDAR spatial manipulation models, Simulink block language and MAAB guidelines

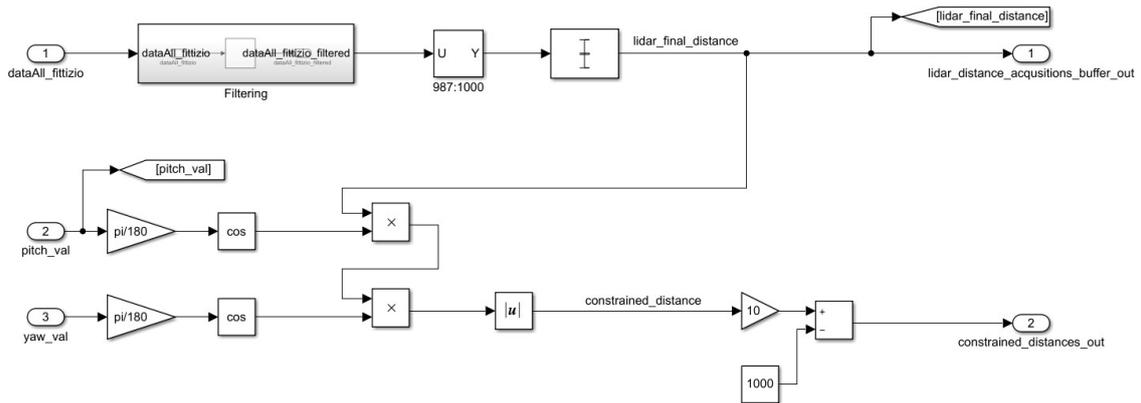


Figure 3.53. Data processing model, Simulink block language and MAAB guidelines

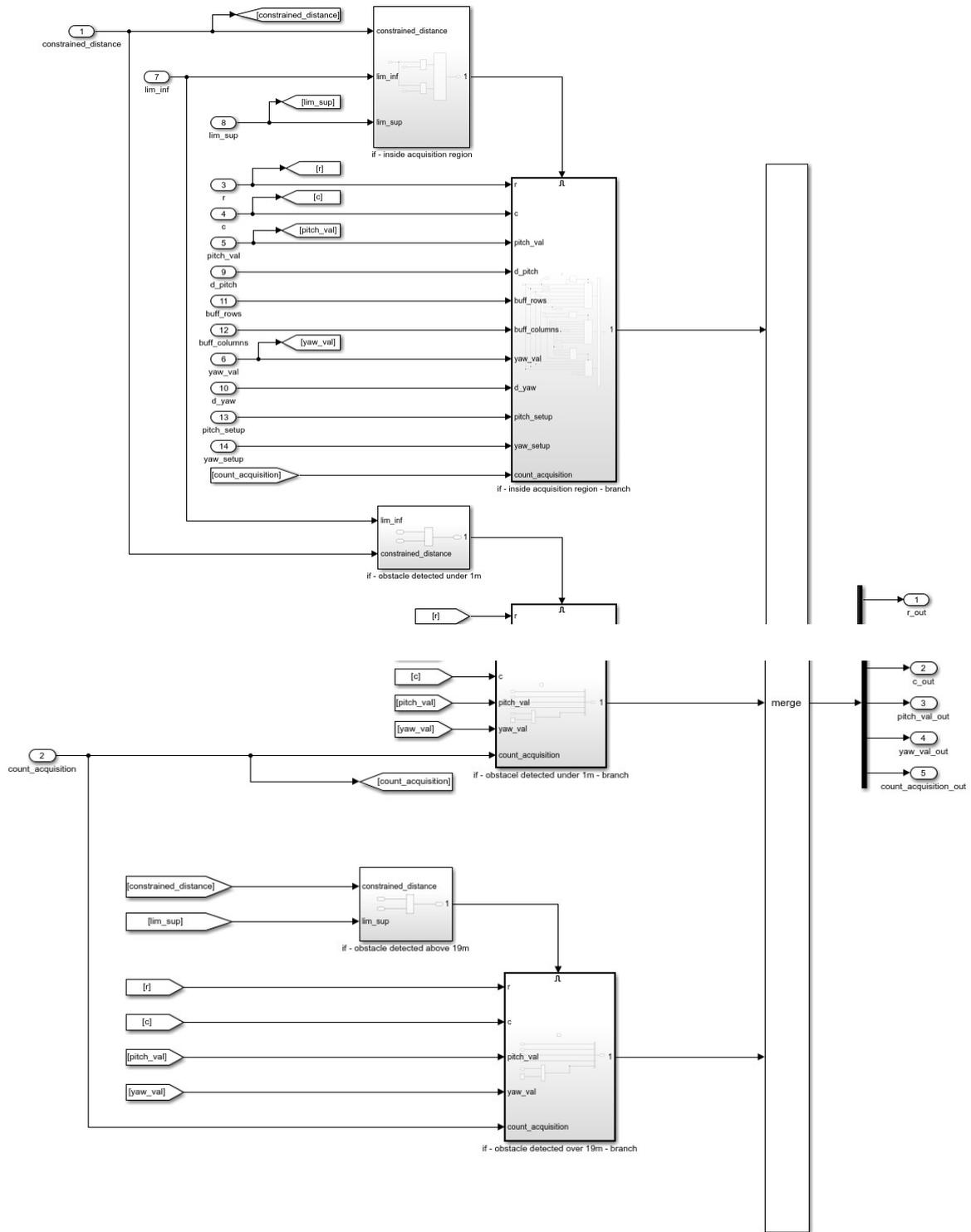


Figure 3.54. LiDAR spatial manipulation model, Simulink block language and MAAB guidelines

It is important to highlight that each variable within the model should be regarded as a "signal" obtained from an external device. In fact, considering each variable as simulated signals from external devices, the model should employ "From/To Workspace" blocks rather than "Const" blocks. This is because the input signals are perceived as dynamic signals and not stationary values. The same principle applies to the outputs, where they will be considered as an output signals. The simulation parameters are as follows:

- *Start time* = 0.0s;
- *Stop time* = 1.4s;
- *Type* = *Fixed – step*;
- *Solver* = *discrete (no continuous states)*;
- *Fixed step size* = 0.001s.

The model was simulated and the results, in terms of graphical output, were identical to those obtained by the script, thus verifying the script's validity. This could be considered as a first step towards scalability. Moreover, this is a fundamental stage for code generation. Code generation is the process of automatically developing source code or executable software from a higher-level representation, such as models developed in tools like Simulink. In the context of Simulink, this involves converting a Simulink model into code that may be deployed on embedded devices or other platforms. Generated code provides optimal performance adapted to specific platforms or hardware, resulting in faster execution. It offers portability across varied systems with common structures, improving reliability and maintainability. Furthermore, its executable form enables rapid prototype, testing, and scalability for complex systems while assuring consistency throughout development [49].

3.4.2 Test results analysis and optimizations

Extensive investigation and rigorous testing have led to significant advancements and refinements in the system's functionalities. Specifically, the upgraded system has successfully achieved the objectives outlined in the thesis, and now it is capable of a:

- larger coverage area, allowing for a broader range of data collecting and processing. This increased coverage improves its overall effectiveness in capturing and interpreting data within its operational environment;
- greater resistance to vibrations. It has become significantly less vulnerable to external vibrations, by integrating peculiar dampening techniques, ensuring more accurate and steady measurements even in dynamic conditions;
- more responsive output as a result of the enhanced microprocessor;
- higher level of reliability. The technology now performs with greater precision and consistency;
- good capacity to distinguish increasingly elaborate and sophisticated things. It currently displays improved skill in identifying varied objects as a result of algorithm optimizations and advancements.

During the testing phase, certain areas that require improvements were identified. In particular, there are:

- challenges in detecting obstacles resembling road bumps and sidewalks: to achieve enhanced system output and a more accurate representation, the following steps are recommended:
 - increase the quantity of acquired samples (data points) from 100 to a higher count, necessitating an improved hardware computational capacity;
 - enhance the color scale in the coloring logic algorithm to generate a sharper image. This involves augmenting the number of thresholds in the coloring logic and employing more suitable color schemes for this purpose.
- challenges encountered in identifying obstacles during adverse weather conditions: since the sensor's measured distance is based on the detection of reflected energy received by the sensor, the reflected laser beam may be disrupted

in this cases. This results in inaccurate signal reception by the sensor. Additionally, liquids on the road surface can cause dispersion of the laser beam.

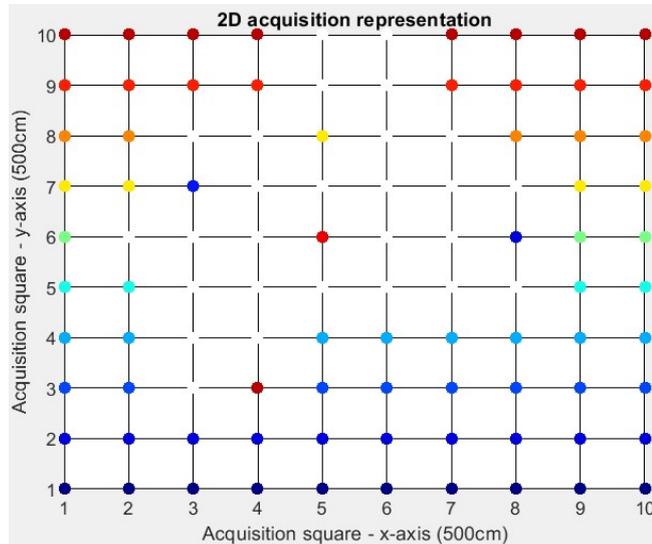


Figure 3.55. Climatic condition influence: presence of a water puddle within the acquisition area (asphalt, scenario #3)

Potential solutions to address this issue could include:

- disabling the system during unfavorable weather conditions;
 - utilize a hardware that is not susceptible to this problem.
- difficulties in acquiring data at high speeds: the study reveals that at a speed of 25km/h , with the current state of the proof of concept, a complete acquisition occurs every 9.18m . This is not acceptable as it implies that the system can only detect an obstacle when it's already in close proximity to the acquisition area, making the system entirely ineffective and posing safety concerns. One possible solution is a more high-performing hardware.

4 Conclusions and future works

All set targets have been met, verifying the initial premises. The system has shown a greatly expanded coverage area of acquisition, good ability to discern increasingly complicated and sophisticated objects, greater resistance to vibrations, improved output responsiveness and a higher level of overall dependability.

Despite that, it is possible to outline some possible future developments, setting the framework for additional research and innovation in this sector. Some of them can be, for example:

- automated control of the system's inclination angle: the current acquisition algorithm works optimally only within a specific distance range of 10m to 15m. Any alteration in the LiDAR's inclination disrupts the system's functionality. A new project would involve refining the algorithm to enable the acquisition system to automatically adjust its inclination angle upon activation, ensuring consistent and accurate operation at all times;
- driver warnings and object recognition: actually the sensing system does not provide warnings to the driver upon detecting an obstacle, nor does it differentiate between various types of obstacles. Future tasks could include adding an audio warning system to alert the driver when an obstacle is detected and implementing software-level obstacle differentiation to create the framework for possible artificial intelligence integration;
- the system was designed initially for 2D data collecting, but its capabilities may be expanded to include 3D acquisitions. Unlocking and increasing this functionality would be an exciting prospect.

Appendix A

[A] The following differential equation

$$m \frac{d^2 x}{dt^2} + c \frac{dx}{dt} + kx = 0$$

represents a simple damped harmonic oscillator, commonly used in physics and engineering. The variables involved are:

- m : the mass of the object involved in the oscillation;
- $x(t)$: the position of the object at time t ;
- c : the damping coefficient, which indicates the damping force or resistance of the object against motion;
- k : the elasticity or stiffness constant, which represents the force of restoring the object towards its equilibrium position.

The equation represents the sum of three contributions:

- the first part $m \frac{d^2 x}{dt^2}$ is the inertial force, proportional to the acceleration $x''(t)$ multiplied by the mass m ;
- the second part $c \frac{dx}{dt}$ represents the damping force, proportional to the speed $x'(t)$ multiplied by the damping coefficient c ;
- the third part kx represents the elastic force, proportional to the position $x(t)$ multiplied by the elasticity constant k .

This differential equation describes the behavior of a system subjected to a force that brings the object towards an equilibrium position, but with a damping force that progressively reduces the amplitude of the oscillations over time. Moreover, it represents a second-order, linear time-invariant system. To rewrite this equation in standard terms of a linear, time-invariant second-order system, it is possible to normalize it by dividing by the mass m . This allows to write the equation in terms of a dimensionless variable:

$$\frac{d^2 x}{dt^2} + \frac{c}{m} \frac{dx}{dt} + \frac{k}{m} x = 0$$

where:

- $\omega_n^2 = \frac{k}{m}$ is the natural frequency of the system;
- $2\zeta\omega_n = \frac{c}{m}$ is the damping factor;
- ω_n is the natural frequency of the undamped system;
- ζ is the damping ratio.

The normalized differential equation then becomes:

$$\frac{d^2x}{dt^2} + 2\zeta\omega_n \frac{dx}{dt} + \omega_n^2 x = 0$$

This standard representation of a linear, time-invariant second-order system shows the relationship between the natural frequency ω_n and the damping factor ζ with the derivatives of the position $x(t)$ with respect to time.

A system is considered linear if it satisfies two fundamental properties:

- Superposition Property: if $x_1(t)$ produces a response $y_1(t)$ and $x_2(t)$ produces $y_2(t)$, then $Ax_1(t) + Bx_2(t)$ must produce $Ay_1(t) + By_2(t)$, where A and B are constants;
- Homogeneity Property: if $x(t)$ produces $y(t)$, then $Cx(t)$ must produce $Cy(t)$, where C is a constant.

The differential equation mentioned satisfies both properties. When the forces $x(t)$ are added or multiplied by constants, the associated responses $y(t)$ follow the same addition or multiplication operations, respecting the linearity of the system.

A system is considered of a second-order when the differential equation describing it involves second order derivatives with respect to time ($\frac{d^2x}{dt^2}$). In this case, the equation contains an acceleration term, which characterizes a second-order system.

A system is considered time invariant if its response does not depend on absolute time but only on the time difference ($t - t_0$). If the input changes, same output at a later time t_0 will be obtained, as long as the input is delayed by t_0 . The linear, second-order differential equation mentioned is time invariant because the relationship between input

and output does not change over time, as long as the system parameters (m, c, k) remain constant over time [43, 44, 50].

References

Bibliography:

[1] M. d'Addato, D. Trincherio and A. Bertone (2020). "Design of a smart sensor for obstacles detection to be installed on electric scooters or bicycles". Teoresi Group, Torino.

[2] A. Li, P. Zhao, H. Haitao, A. Mansourian and K. W. Axhausena (2021). "How did micro-mobility change in response to COVID-19 pandemic? A case study based on spatial-temporal-semantic analytics". Computers, Environment and Urban Systems, Volume 90, 101703.

[3] S. Carrese, E. Cipriani, C. Colombaroni, U. Crisalli, G. Fusco, A. Gemma, N. Isaenko, L. Mannini, M. Petrelli, V. Busillo and S. Saracchi (2021). "Analysis and monitoring of post-COVID mobility demand in Rome resulting from the adoption of sustainable mobility measures". Transport Policy, Volume 111, pp. 197-215.

[4] T. K. Trivedi, C. Liu, A. L. M. Antonio, N. Wheaton, V. Kreger, A. Yap, D. Schriger, J. G. Elmore (2019). "Injuries associated with standing electric scooter use". JAMA Network Open, Los Angeles.

[5] P. Feiffer (2019). "Better Safety with Emergency Brake Assist for Motorcycles". ATZworldwide, Volume 121, pp. 60-65.

[6] J. Woongsun and R. Rajesh (2018). "Rear Vehicle Tracking on a Bicycle Using Active Sensor Orientation Control". IEEE Transaction on intelligent transportation systems, Volume 19, No. 8.

[7] P. Wei, L. Cagle, T. Reza, j. Ball and J. Gafford (2018). "LiDAR and Camera Detection Fusion in a Real-Time Industrial Multi-Sensor Collision Avoidance System". MDPI journal Electronics, 7(6), 84.

Sitography:

[8] Wikipedia (2022). “Micromobility”. Wikipedia.com. View online:

<https://en.wikipedia.org/wiki/Micromobility>

[9] M. Gucci, Nikada and South Agency (2020). “Emission-based driving restrictions: how do they work?”. RenaultGroup.com. View online:

<https://www.renaultgroup.com/en/news-on-air/news/learning-your-way-around-alternate-traffic-restrictions-emissions-based-driving-restrictions-and-low-emission-zones/>

[10] C. Rudinschi (2020). “The impact of COVID-19 on public transportation and future mobility solutions”. Ingenuity.siemens.com. View online:

<https://ingenuity.siemens.com/2020/09/the-impact-of-covid-19-on-public-transportation-and-future-mobility-solutions/>

[11] Apple (2020). “COVID-19 – Mobility Trends Reports”. Apple.com. View online:

<https://covid19.apple.com/mobility>

[12] R. Clewlow (2020). “The Micro-Mobility Revolution”. Medium.com. View online:

<https://medium.com/populus-ai/the-micro-mobility-revolution-95e396db3754>

[13] Elettricosmart (2020).” Qual è il Monopattino Elettrico più Venduto al Mondo e in Italia?”. Elettricosmart.it. View online: <https://www.elettricosmart.it/qual-e-il-monopattino-elettrico-piu-venduto-al-mondo-e-in-italia/>

<https://www.elettricosmart.it/qual-e-il-monopattino-elettrico-piu-venduto-al-mondo-e-in-italia/>

[14] Wikipedia (2022). “Motorized scooter”. Wikipedia.com. View online:

https://en.wikipedia.org/wiki/Motorized_scooter

[15] A. Vai (2020). “Monopattini elettrici, con la diffusione aumentano gli incidenti”.

LaStampa.com. View online:

<https://www.lastampa.it/motori/attualita/2020/08/12/news/monopattini-elettrici-con-la-diffusione-aumentano-gli-incidenti-1.39186873/>

[16] N. McCarthy (2019). “How Dangerous Are Electric Scooters?”. Statista.com. View online: <https://www.statista.com/chart/16895/electric-scooter-injuries-and-accidents-in-southern-california/>

<https://www.statista.com/chart/16895/electric-scooter-injuries-and-accidents-in-southern-california/>

- [17] Ministero dell'Interno (2020). “Le regole per i monopattini elettrici”. Interno.gov. View online: <https://www.interno.gov.it/it/notizie/regole-i-monopattini-elettrici>
- [18] Youmath (2023). “Calcolo dello spazio di frenata in base alle condizioni”. Youmath.it. View online: <https://www.youmath.it/domande-a-risposte/view/6684-spazio-di-frenatura.html>
- [19] Garmin (2018). “Varia RTL510”. Garmin.com. View online: <https://www.garmin.com/en-US/p/601468/pn/010-01980-00>
- [20] Wikipedia (2022). “Sensor fusion”. Wikipedia.com. View online: https://en.wikipedia.org/wiki/Sensor_fusion
- [21] Garmin (2018). “LIDAR-Lite v3”. Garmin.com. View online: <https://www.garmin.com/it-IT/p/557294/pn/010-01722-00>
- [22] Garmin (2018). “LIDAR-Lite v3 datasheet”. Garmin.com. View online: https://static.garmin.com/pumac/LIDAR_Lite_v3_Operation_Manual_and_Technical_Specifications.pdf
- [23] Sparkfun (2018). “LIDAR-Lite v3”. Sparkfun.com. View online: <https://www.sparkfun.com/products/14032>
- [24] STMicroelectronics (2018). “Nucleo-F429ZI”. St.com. View online: <https://www.st.com/en/evaluation-tools/nucleo-f429zi.html>
- [25] Yang/wdaehn, OlliW and alcurtis93. “Getting Started”. OlliW.eu. View online: http://www.olliw.eu/storm32bgc-wiki/Getting_Started
- [26] Wikipedia (2023). “I2C”. Wikipedia.com. View online: <https://en.wikipedia.org/wiki/I%C2%B2C>
- [27] MathWorks (2023). “MATLAB”. MathWorks.com. View online: <https://it.mathworks.com/products/matlab.html>
- [28] Wikipedia (2023). “MATLAB”. Wikipedia.com. View online: <https://it.wikipedia.org/wiki/MATLAB>

- [29] Usman Ali Butt (2018). “How to use a LiDAR sensor with Arduino”. Engineersgarage.com. View online: <https://www.engineersgarage.com/how-to-use-a-lidar-sensor-with-arduino/>
- [30] Abhinav (2018). “Arduino LIDAR”. Hackster.com. View online: https://www.hackster.io/Abhinav_Abhi/arduino-lidar-917404
- [31] Rchow012 (2018). “Arduino based LIDAR 3d mapping”. Arduino.cc. View online: <https://forum.arduino.cc/t/arduino-based-lidar-3d-mapping/529820>
- [32] David Cambridge (2018). “Hobby Lidar Build”. EEVblog.com. View online: <https://www.eevblog.com/forum/projects/hobby-lidar-build/msg1567591/>
- [33] Automobile Club d’Italia (2023). “Codice della strada”. Aci.it. View online: <https://www.aci.it/i-servizi/normative/codice-della-strada.html>
- [34] Roland Hostettler (2012). “Vehicle Parameter Estimation using Road Surface Vibrations”. Hostettler.com. View online: <http://hostettler.co/>
- [35] ISO 8608: Mechanical Vibration – Road Surface Profiles – Reporting of Measured Data (1995).
- [36] H. E. M. Hunt (1991). “Stochastic modelling of traffic-induced ground vibration”. Journal of Sound and Vibration, vol. 144, no. 1, pp. 53 – 70.
- [37] Wikipedia (2023). “Probability density function”. Wikipedia.com. View online: https://en.wikipedia.org/wiki/Probability_density_function
- [38] Wikipedia (2023). “Normal distribution”. Wikipedia.com. View online: https://en.wikipedia.org/wiki/Normal_distribution
- [39] International Encyclopedia of the Social Sciences (2023). “Normal distribution”. Encyclopedia.com. View online: https://en.wikipedia.org/wiki/Standard_deviation
- [40] G. Casella; R. L. Berger (2001). Statistical Inference (2nd ed.). Duxbury.
- [41] A. Lyon (2014). “Why are Normal Distributions Normal?”. The British Journal for the Philosophy of Science. View online: https://aidanlyon.com/normal_distributions.pdf

- [42] S. Battiato (2020). “Restauro e Ricostruzione da Immagini”. Dmi.unict.it. View online: www.dmi.unict.it/battiato/Parte_8_-_Image_Restoration.pdf
- [43] Wikipedia (2023). “Damping”. Wikipedia.com. View online: <https://en.wikipedia.org/wiki/Damping>
- [44] J.Rahman, M. Mushtaq, A. Ali, Y.N. Anjam, S. Nazir (2014). "Modelling damped mass spring system in MATHLAB Simulink". Journal of Faculty of Engineering & Technology. 2. View online: <https://www.researchgate.net/publication/338158709>
- [45] C. Monti, A. Giorgetti, A. Girgenti (2015). “An Axiomatic Design approach for a motorcycle steering damper”. 9th International Conference on Axiomatic Design – ICAD 2015. View online: <https://www.researchgate.net/publication/283194907>
- [46] E. Akagündüz, O. Cifdaloz (2021). “Dynamical system parameter identification using deep recurrent cell networks”. Neural Computing and Applications. View online: <https://www.researchgate.net/publication/353154924>
- [47] Wikipedia (2019). “Model-in-the-loop”. Wikipedia.com. View online: <https://it.wikipedia.org/wiki/Model-in-the-loop>
- [48] MathWorks (2023). “What are MIL, SIL, PIL, and HIL, and how do they integrate with the Model-Based Design approach?”. Mathworks.com. View online: <https://it.mathworks.com/matlabcentral/answers/440277-what-are-mil-sil-pil-and-hil-and-how-do-they-integrate-with-the-model-based-design-approach>
- [49] Wikipedia (2023). “Model-driven engineering”. Wikipedia.com. View online: https://en.wikipedia.org/wiki/Model-driven_engineering
- [50] Wikipedia (2023). “Vibration”. Wikipedia.com. View online: <https://en.wikipedia.org/wiki/Vibration>

Images' link:

Figure 1.0. View online: https://auto.hwupgrade.it/news/urban-mobility/la-micromobilita-di-helbiz-ora-e-integrata-in-google-maps-bici-monopattini-e-scooter-elettrici-per-l-ultimo-miglio_104079.html

Figure 1.1. View online: <https://todaysenvironmentalist.com/2018/12/18/air-pollution-reduces-global-life-expectancy-by-nearly-2-years/>

Figure 1.2. View online: <https://covid19.apple.com/mobility>

Figure 1.3. View online: <https://www.elettricosmart.it/qual-e-il-monopattino-elettrico-piu-venduto-al-mondo-e-in-italia/>

Figure 1.4. View online: <https://www.profsharemarketresearch.com/electric-scooters-market-report/>

Figure 1.5. View online: <https://cdn.statcdn.com/Infographic/images/normal/16895.jpeg>

Figure 1.6. View online: <https://jamanetwork.com/journals/jamanetworkopen/fullarticle/2722574>

Figure 1.7. View online: <https://www.garmin.com/it-IT/p/601468>

Figure 1.8. View online: <https://link.springer.com/article/10.1007/s38311-019-0097-0>

Figure 1.9. View online: <https://ieeexplore.ieee.org/document/8101535>

Figure 1.10. View online: <https://arxiv.org/abs/1807.10573>

Figure 1.11. View online: <https://www.shutterstock.com/it/search/rough-road>

Figure 1.12, 1.13, 1.14, 1.15, 1.16, 1.17, 1.18, 1.19, 1.20, 1.21, 3.15, 3.19, 3.23, 3.27, 3.31, 3.39, 3.43. M. d'Addato, D. Trincherò and A. Bertone (2020). "Design of a smart sensor for obstacles detection to be installed on electric scooters or bicycles". Teoresi Group, Torino, p. 84, p. 16, p. 22, p. 20, p. 24, p. 41, p. 48, p. 50, p. 56, p. 58, p. 71. p. 86, p. 87, p. 89, p. 90, p. 92, p. 94, p. 97 respectively.

Figure 1.22. View online: <https://www.elegoo.com/products/elegoo-uno-r3-board>

Figure 1.23. View online: https://ca.wikipedia.org/wiki/Suspensi%C3%B3_de_Cardan

Figure 1.24. View online:

<https://www.coppeliarobotics.com/helpFiles/en/pointClouds.htm>

Figure 1.25. View online:

https://it.mathworks.com/help/simulink/mdl_gd/maab/jc_0630usageofmultiportswitchblocks.html

Figure 2.0, 2.1, 2.4, 2.8, 2.9.1, 2.9.2, 2.11, 2.12, 2.13, 2.14, 2.15, 2.17, 2.19, 2.20, 2.21, 2.22, 2.23, 2.24, 2.25, 2.26, 2.27, 3.1, 3.2, 3.3, 3.4, 3.5, 3.16, 3.17, 3.18, 3.20, 3.21, 3.22, 3.24, 3.25, 3.26, 3.28, 3.29, 3.30, 3.32, 3.33, 3.34, 3.36, 3.37, 3.38, 3.40, 3.41, 3.42, 3.44, 3.45, 3.46, 3.47, 3.48, 3.49, 3.51, 3.52, 3.53, 3.54, 3.55. View online: personal image.

Figure 2.2. View online: <https://www.st.com/en/evaluation-tools/nucleo-f429zi.html>

Figure 2.3. View online: <http://www.olliw.eu/2013/storm32bgc/>

Figure 2.5 – 2.6. View online:

https://static.garmin.com/pumac/LIDAR_Lite_v3_Operation_Manual_and_Technical_Specifications.pdf

Figure 2.7. View online: <https://predictabledesigns.com/wp-content/uploads/2020/05/HeroImage-1.jpg>

Figure 2.10. View online: <https://it.wikipedia.org/wiki/MATLAB>

Figure 2.16. View online: <https://www.koyeb.com/static/images/blog/sync-vs-async-schema.png>

Figure 2.18. View online: <https://www.educba.com/what-is-data-processing/>

Figure 3.0. View online: <https://www.kugooitaly.com/collections/kugoo-scooter-elettrici/products/kugoo-s1-pro-electric-scooter>

Figure 3.6. View online: <https://www.vicenzatoday.it/foto/cronaca/strada-dissestata-pericolo-per-i-ciclisti/>

Figure 3.7. View online: https://en.wikipedia.org/wiki/Normal_distribution

Figure 3.8. View online: https://en.wikipedia.org/wiki/Gaussian_noise

Figure 3.9. View online: <http://hostettler.co/publications/>

Figure 3.10. View online: <https://m.media-amazon.com/images/I/61qdYyRHQDL.jpg>

Figure 3.11. View online:

https://ulipstore.com/cdn/shop/products/01_400x@2x.jpg?v=1670659531 ,

<https://m.media-amazon.com/images/I/71rp1dLR7uL.jpg>

Figure 3.12. View online: <https://en.wikipedia.org/wiki/Damping>

Figure 3.13. View online: <https://en.wikipedia.org/wiki/Damping>

Figure 3.14. View online: <https://www.researchgate.net/publication/353154924>

Figure 3.35. View online: <https://pacificcoastpaving.com/difference-between-speed-bumps-and-speed-humps/>

Figure 3.50: View online: <https://sysorganic.wordpress.com/2016/12/03/model-in-the-loop-mil/>

Acknowledgements

I am very grateful to Professor Massimo Violante for accepting to be my academic supervisor as well as to my business supervisor Alberto Bertone for allowing me to finish my academic career at the Teoresi Spa company. Next, I want to express my gratitude to Antonio Arena and Jacopo Filippi, who supported me as my tutors throughout the entire thesis process.

Second of all, I would like to thank Giorgia, my significant other, for her constant support and unconditional love. I also want to thank my wonderful family, without whom I could not have accomplished what I have.

Then, I would like to thank all the people who have been close to me over this last academic year, giving me the chance to enjoy their support, guidance, and encouragement throughout my journey.

In particular, I would like to thank my roommate Antonio, my personal feeder as well as professional binge watcher, and my second family, Adriano, Lorenzo, Leonardo and Arash, with whom I had a lot of fun hanging out around Turin.

Last but not least, I would like to mention all my close friends from Thiene. Their continuous support and closeness have improved my life and made every moment more special.

Finally, I want to thank me,
for believing in me,
for doing all this hard work,
for having no days off,
for never quitting,
for always being a giver and try to give more than I receive,
for trying to do more right than wrong,
for being me at all time.