*Electronic Research Archive*

ERA, 31(8): 5083–5103.
DOI: 10.3934/era.2023260
Received: 01 April 2023
Revised: 25 May 2023
Accepted: 19 June 2023
Published: 17 July 2023

http://www.aimspress.com/journal/ERA

---

*Research article*

# Fuzzy logic-based vehicle safety estimation using V2V communications and on-board embedded ROS-based architecture for safe traffic management system in hail city

**Ismail Ben Abdallah[1], Yassine Bouteraa[2,*], Saleh Mobayen[3], Omar Kahouli[4], Ali Aloui[4], Mouldi Ben Amara[4] and Maher JEBALI[5]**

[1] Control and Energy Management Laboratory (CEM Lab.), Ecole Nationale d'Ingenieurs de Sfax (ENIS), University of Sfax, Sfax 3038, Tunisia
[2] Department of Computer Engineering, College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia
[3] Graduate School of Intelligent Data Science, National Yunlin University of Science and Technology, Yunlin 640301, Taiwan
[4] Department of Electronics Engineering, Applied College, University of Ha'il, Ha'il 2440, Saudi Arabia
[5] Department of Computer Science, Applied College, University of Ha'il, Ha'il 2440, Saudi Arabia

* **Correspondence:** Email: yassine.bouteraa@isbs.usf.tn.

**Abstract:** Estimating the state of surrounding vehicles is crucial to either prevent or avoid collisions with other road users. However, due to insufficient historical data and the unpredictability of future driving tactics, estimating the safety status is a difficult undertaking. To address this problem, an intelligent and autonomous traffic management system based on V2V technology is proposed. The main contribution of this work is to design a new system that uses a real-time control system and a fuzzy logic algorithm to estimate safety. The robot operating system (ROS) is the foundation of the control architechture, which connects all the various system nodes and generates the decision in the form of a speech and graphical message. The safe path is determined by a safety evaluation system that combines sensor data with a fuzzy classifier. Moreover, the suitable information processed by each vehicle unit is shared in the group to avoid unexpected problems related to speed, sudden braking, unplanned deviation, street holes, road bumps, and any kind of street issues. The connection is provided through a network based on the ZigBee protocol. The results of vehicle tests show that the proposed method provides a more reliable estimate of safety as compared to other methods.

**Keywords:** V2V; robot operating system; real-time system; safety estimation; fuzzy logic

## 1. Introduction

With the technological revolution in terms of wireless communication, cloud, Internet of Things (IoT), and artificial intelligence, it is obvious that the classic transport system will be abandoned to make way for an intelligent transportation system that meets the requirements of smart cities. These modern communication systems aim to establish communication between vehicle and vehicle on one side and vehicle and infrastructure nodes on the other side. The data acquired by the exteroceptive and proprioceptive sensors of the cars, as well as the data provided by the sensors of the urban infrastructures, will be communicated and shared between vehicles and infrastructure nodes. The goal is to ensure intelligent traffic management, which aims to analyze the issues and propose solutions enabling safer, smarter, and greener transportation. This high level of connectivity based on vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) technologies will be used by vehicle manufacturers, as well as traffic management entities to implement new applications dealing with safety, incident detection, route planning, fleet management, monitoring, etc. Generally, V2V technology allows for the exchange of information between vehicles. This information is data acquired by the vehicles, generally reflecting their states and the state of the road. While V2I technology provides access to the internet and a large database to benefit from several applications and the necessary information on the city, climate, maps, roads, traffic, etc; the combination of the two technologies has given rise to what is known as V2X technology, which harnesses all of this data and applications to provide an intelligent transport system.

### *1.1. Problem statement*

Road crashes kill more than 1.2 million people every year and have a huge impact on health and development [1]. Human shortage and human error are the main causes of accidents. Therefore, it would be very interesting to invest in intelligent driving aids by taking advantage of new processing, detection, and communication technologies to reduce vehicle accidents as much as possible. The conventional traffic management system does not have the ability to monitor traffic remotely, and thus cannot efficiently solve the following traffic problems:

- Many people can die in accidents due to ambulance delays caused by traffic jams.
- The emergency vehicle cannot immediately reach the traffic jam site to control the traffic. The proposed traffic management system will allow the traffic authority to remotely monitor traffic jams.
- Traffic jams waste a lot of time and consume a lot of fuel, which is harmful to the economy, as well as to the environment.
- A major traffic jam or accident may occur due to the traffic lights being damaged; however, by using the proposed traffic management system, we can remotely manage the traffic, even if there are no traffic lights.

### *1.2. Objectives*

The main objective is to set up an intelligent and autonomous traffic management system based on V2V technology, which provides the following functions, among others:

- Safe distance monitoring: As it is known that safe distance is a very important factor to determine the safety status of vehicles on the street, the developed system prevents the driver from approaching neighboring vehicles by overtaking the authorized limit, which is calculated according to the speed of both vehicles and the road situation.
- Sudden braking warning: In general, the driver's view is obscured by the car ahead, and in situations of sudden traffic choking due to one cause or another, the braking reaction made by the driver is only a reaction following the braking made by the car in front of it; if this reaction is done with a little delay, it can cause accidents. Therefore, we propose a procedure to warn all associated vehicles using the ad hoc network by transmitting the warning message between the nodes of this network.

This article focuses on the integration of a real-time V2V communication approach to equip vehicles with a set of services enabling safer, smarter, and more sustainable driving. The document is organized as follows: Section 2 details the tools and methods used in this project; Section 3 shows the experimental results and Section 4 concludes the paper.

## 2. Literature review

Over the past decade, many projects have addressed the subject of V2V communication [2–6], as well as V2I communication [7–12]. The main question is what congestion reduction and traffic safety benefits could be provided by these technological advances. Several techniques are applied to force the vehicle to keep a safe distance between them [13–18]. The time headway policy (CTHP) is one of the famous techniques applied in this context [16–18]. Unlike adaptive cruise control (ACC) systems, which only rely on information from their onboard systems, autonomous vehicles adopt the strategy of V2V and V2I. Note that the objective of reducing the lead time was first addressed by semi-autonomous robots [19–21]. Modern systems, including coordinated adaptive cruise control (CACC) systems, leverage vehicular communication to gather new data in order to decrease uptime [22–24]. They estimate the advantages of V2V communication by accounting for the decrease in the useful time when parasitic lags are present. To model the dynamics of parasitic actuation, they consider a unique disturbance. In the literature [25–29], the advantages of integrating V2V and V2I architecture to enhance traffic management security have been discussed. For instance, [25] discusses the advantages of V2V for preventing emergency lane change collisions in auxiliary vehicles. Inter-vehicle communication makes it easier for vehicles to coordinate during an emergency braking scenario, which lowers the possibility of crashes, as well as the projected number and severity of collisions [26]. In addition, the authors address the problem of decentralized crossings at unsignalized intersections for mixed traffic flows, composed of both connected human-driven vehicles and connected autonomous vehicles [28]. Based on previous research on the 5G new radio (NR) air interface [29], an in-depth tutorial of the third generation partnership project (3GPP) release 16 5G NR V2X standard for V2X communications is discussed. However, future enhancements related to beamforming, side link positioning, and resource allocation should be addressed. Moreover, several recent works focus on safety management based on the new advancement of V2V technologies [30–32].

## 3. Materials and methods

In this section, the suggested navigation system's computer architecture is described. The described hardware and software architectures serve as illustrations for the chosen components and the planned navigation strategy. The design of the architecture, including the choice of each electronic architecture unit, is discussed.

### *3.1. Communication technology*

According to the project's design, the urban infrastructure must be equipped with distributed sensors that effectively communicatewith one another to gather information from various city sites, including traffic data, weather data, energy network data, etc. Several communication technologies can be integrated into these types of applications, such as the following:

- Fiber optic network: this type of technology stands out for its high bandwidth, security and reliability, low attenuation and interference.
- Wi-Fi communication: this type of technology is considered to be the most suitable type for smart cities. Indeed, Wi-Fi networks can be deployed for the intelligent transport system to interconnect vehicles, traffic lights, navigation applications, traffic management services, etc. However, these types of networks are limited by their bandwidth.
- Cell-based communication technology: this type of technology is supported by the advancement of smartphones, especially considering 5G cellular communication technology. In fact, it is distinguished by its wide bandwidth and, most importantly, its distributed infrastructure. Transport-sharing applications, such as Uber and Careem, are currently the most popular applications based on this type of network. In the future, 5G will be the most adopted technology due to its speed and reliability.
- Dedicated short-range communication (DSRC): this type of technology allows intelligent transportation system (ITS) automobiles to communicate with each other and with the infrastructure. DSRC technology operates in the 5.9 GHz band of the radio spectrum over short to medium distances.

Finally, this information exchanged must be used to make a decision. Faced with the complicated process of extracting useful data, big data analysis becomes essential; this is because the size and amount of information exchanged are significant as the data is collected by various IoT devices. With cloud computing, users can access shared computing resources through a standard protocol. The ability to scale computing on demand is one of the key benefits of cloud computing. Cloud computing services, such as autonomous decision-making, machine learning, simulation, storage, etc, can be useful for smart cities. A distributed sensor network called a wireless sensor network (WSN) is used to continuously track several physical phenomena, such as temperature, pressure, sound, and others. WSN is utilized in smart cities to track a variety of real-time status and situation resources, including traffic volume, pollution, electricity use, and water availability.

In this paper, the V2I system is composed of two main parts: a car unit and a central station. The car unit is responsible for data collection and sending the processed data to the central station. It is also responsible for receiving the data from the central station and providing it to the driver. On the other side, the central station is responsible for analyzing the data by acquiring data via the Zigbee protocol. The central station is mainly based on a high-performance desktop and an Xbee receiver connected via a serial protocol. Moreover, the central station is connected to the cloud server for data storage and monitoring.

### 3.2. Unit description

The components of the car units are as follows: a Xbee module to transfer security data to the other vehicle units; an IMU sensor to identify road quality, whether it is rough or smooth; a GPS sensor to determine speed, location, etc., which will allow the control system to decide the appropriate (safety) distances based on the GPS reading; and Lidar and Ultrasonic sensors to read the distance between vehicles, so the control system will know whether the vehicle is within safe distance or not.

The goal is to connect the Xbee module, the IMU sensor, and the ultrasonic sensor with a control system; some setup and calibrations are needed for the sensors to work, and the reading value depends on the vehicles and the road situation. One must establish and configure the required configuration for the serial communication port between the control system and the Xbee module to allow two Xbee modules to transfer the required data between them. When the control system receives the safety data through Xbee, in each case, it will display appropriate warnings according to the received data and the vehicle situation. The decision is also transferred via a vocal message via the speech synthesis module. Therefore, a hybrid visual and audio user interface was offered. The input voltage of 12V DC is supported by the power supply. The developed real-time navigation device is depicted in the overview in Figure 1.

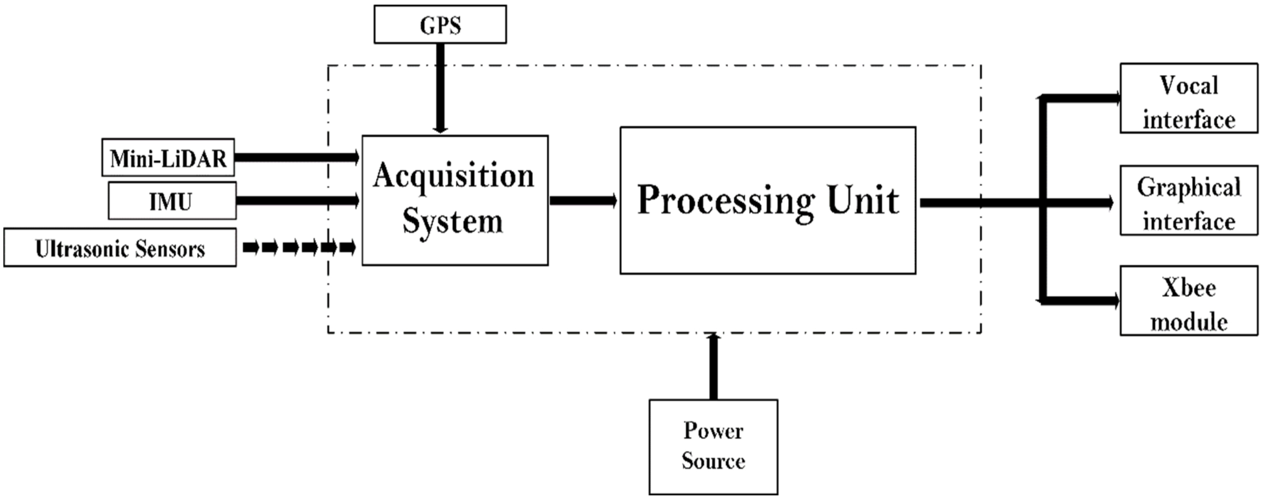


**Figure 1.** Overview of the real-time acquisition and processing unit.

#### 3.2.1. System architecture

The navigation system is constructed on ROS, which is a well-known operating system with resources for building robotic applications. In actuality, ROS is an open-source software development environment for robots. Because of its cutting-edge design, ROS can simultaneously run on multiple machines and link to a wide range of gadgets and programs. The suggested navigation system design is shown in Figure 2. In order to assure sensor data collection and navigation algorithm execution, several nodes were incorporated into the architecture. The hybrid graphical/vocal message containing the choice is sent by the ROS environment, which also sets up communication between nodes. The processing unit manages the obstacle detection and decision-making processes. The Arduino and Raspberry boards are crucial parts of the developed unit and ensure data collection and decision making. Although the above sensors are mostly used for environment recognition, it's important to get a feel of how quickly the vehicle navigation profile is moving. Position, acceleration, and speed are included in the navigation profile. The vehicle's speed can be used to evaluate safety. The XSENS technology was chosen to suit this requirement. Indeed, the MTi-670 is considered a slave node that connects to the processing board using serial communication.

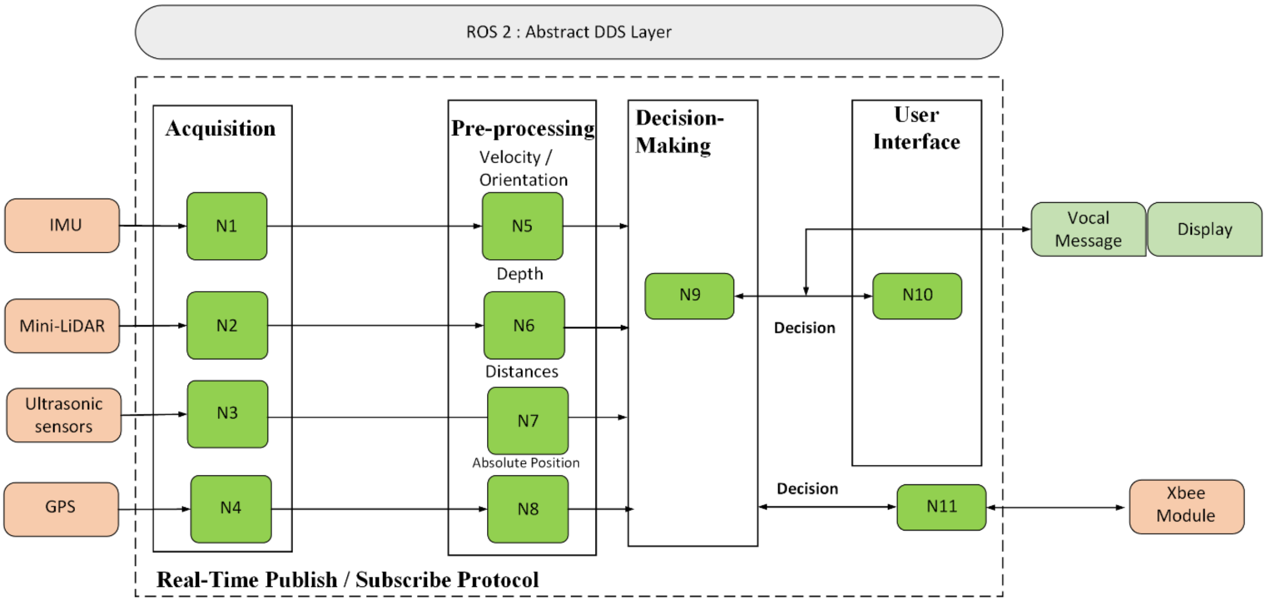


**Figure 2.** High level description of the proposed navigation system.

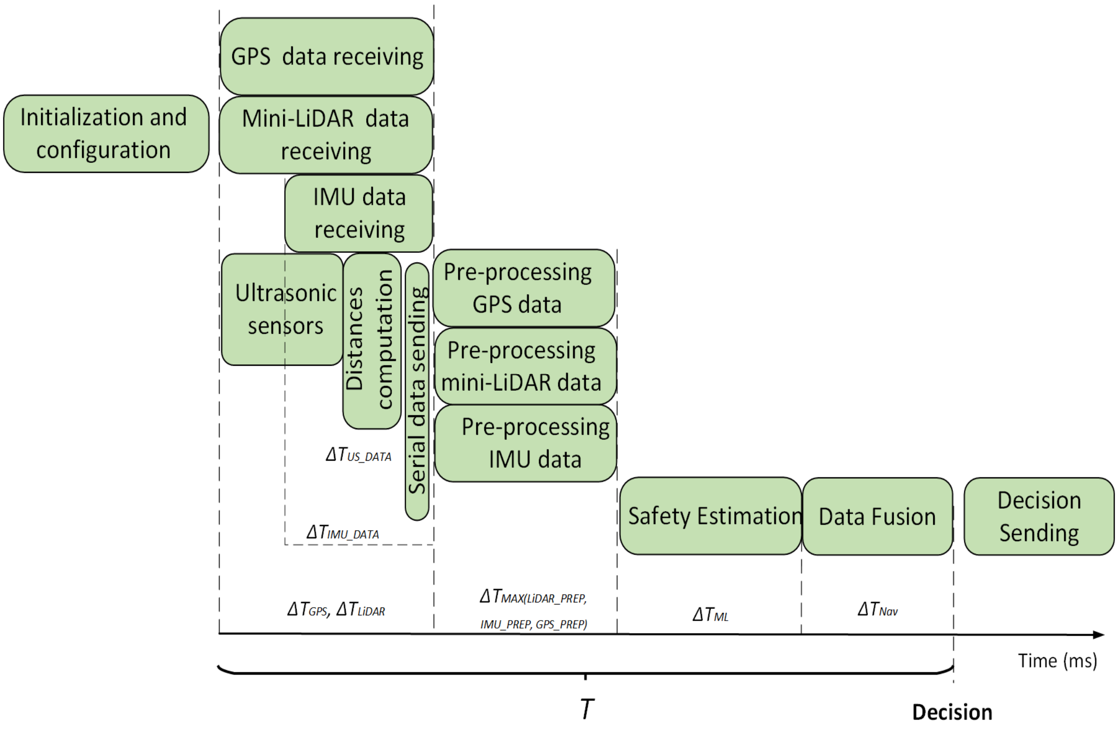


**Figure 3.** Task scheduling in parallel processing.

Parallel processing is the main focus of the low-level design (Figure 3). Task scheduling is a method for splitting an operation into multiple processes and running them all simultaneously on various CPUs or processors for parallel processing. This design offers concurrency, which removes timing constraints and enables the solution of larger problems. Additional devices may also be integrated using it. To meet real-time requirements, the system operations must be finished within the sample time T, depending on the principal components (including the acquisition, the computing, and the decision-making). The ROS-based nodes architecture is shown in Figure 4.

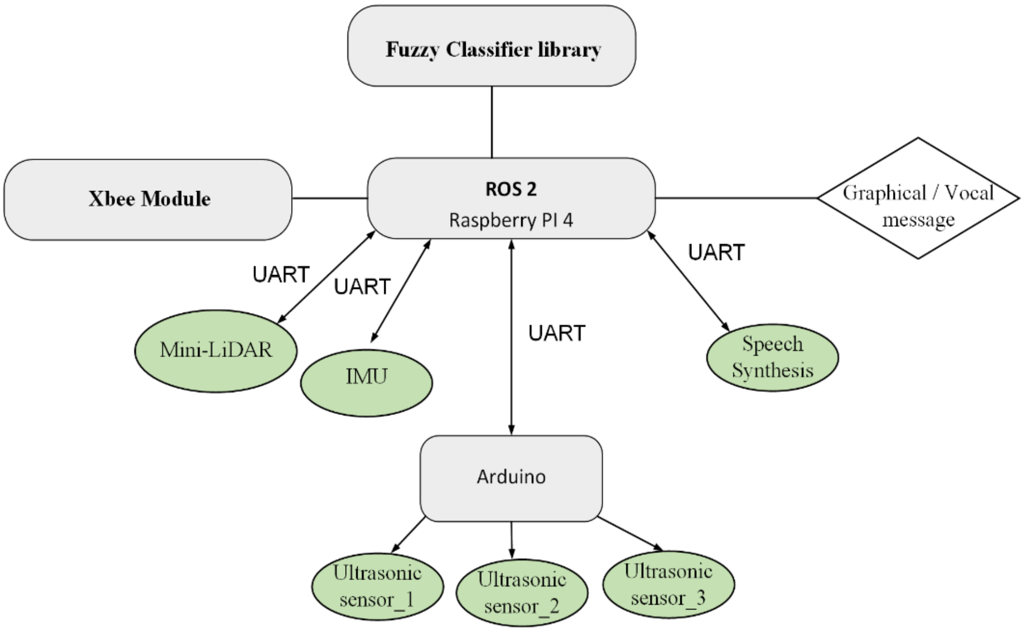


**Figure 4.** ROS-based hardware architecture.

### 3.2.2. Operating system selection

Significant advancements in software engineering lifecycle dependability are required due to the growing complexity of autonomous navigation systems. One of the most challenging research areas for overall systems is middleware. Runtime application adaptation, communication across heterogeneous systems at various middleware layers, and runtime safety assurance in the case of failure are all instances of middleware issues.

Since ROS1 does not provide real-time performance, we switched to the robot operating system (ROS2) architecture. In actuality, ROS1 shares many of the same drawbacks, including the lack of a standardized approach for creating a multi-robot system. Furthermore, ROS1 lacks a real-time design, which forces us to stretch our design to match the high real-time performance requirements and tight real-time performance indicators of our navigation system.

In order to ensure data integrity, ROS1's distributed strategy also needs a stable network environment; however, the network is unsecured and unencrypted. ROS2 improves network performance for multi-robot communication over ROS1 and adds functionality for multi-robot systems. Real-time control is also supported by ROS2, which can enhance the intended system's performance, as well as the timeliness of control. The architecture of ROS2, which is organized into multiple levels, improves fault tolerance capabilities because communication is based on the data distribution service (DDS) standard (Figure 5). Additionally, ROS2's intra-process communication technique is more efficient.

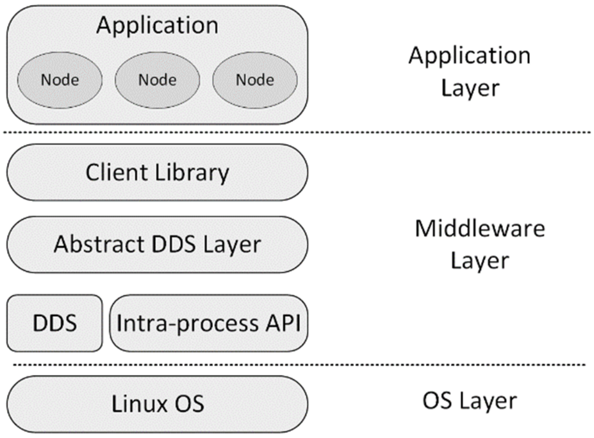


**Figure 5.** ROS2 architecture overview.

### *3.3. Navigation approach and safety assessment algorithm*

An Arduino Due and a Raspberry PI 4 formed the basis of the control system's architecture. In particular, the Raspberry Pi board was designed for data merging and decision-making. As a data collection and alerting node, the Arduino board was added to the opposite side. The ROS-Serial capability enabled a UART interface connection between the Raspberry and the Arduino. Phase 1 of the navigation approach involved gathering geometric data from ultrasonic sensors. As a result, the designated scanning region detected the presence of impediments. If one or more of the three implanted sensors detected an obstruction, the system alerted the user verbally and instructed them to switch to the freest side direction. In fact, the processing unit's most liberated side will verbally tell the user. To determine the car's safe path, a fuzzy logic classifier was built using the Lidar sensor and the IMU. Indeed, the mini-LiDAR module was built specifically to measure depth and detect the curve of the ground. To determine the vehicle's velocity, an IMU was used. The fuzzy controller processed and fused the vehicle velocity and depth data as inputs to produce a route risk estimate. By way of a voice message, the user was notified of the controller's choice. Figure 6 illustrates the method for evaluating safety, and Figure 7 shows the proposed fuzzy logic system. The fuzzy logic system's inputs were the computed speed and depth, and its output was the calculated safety level. The suggested fuzzy approach used the vehicle's speed and ground depth to determine the safety level while navigating. Input and output membership functions are displayed in Figures 8 and 9, respectively. A Gaussian membership function was an optimum type to employ in the. Defuzzification is performed using the center of area (CoA) technique. The LabVIEW software was used to implement a rules database, as shown in Figure 10. Indeed, the database shown in Table 1 is used to build fifteen if/then rules. The rules of the fuzzy classifier are defined by an expert by analyzing inputs and then outputs the estimated safety level. Indeed, different scenarios were simulated to generate the related rules suitable for safety estimation. In this phase, simulated scenarios show that the velocity of the vehicle and the geometric shape of the road are the most important factors, which can affect the safety of the car. The relationship between these factors is also important to improve safety. Therefore, the interconnection between the vehicle's speed and the depth data is considered in the fuzzy rules design. Several other factors can affect safety, such as weather constraints and the driver's health conditions. However, the weight of these criteria is relatively low compared to other factors. Hence, only the two mentioned criteria are taken into account for safety estimation. The selected inputs of the fuzzy classifier can be easily interpreted. Indeed, the depth data provides the state of the road in the front side, including the ground shape and any other existing obstacle. A low value of the depth data means that a very close obstacle is detected against the related vehicle, while a high value means that the obstacle is far away. Regarding the second input, the speed level is proportional to the acquired value of the car's velocity. The combination of these inputs can provide a reliable idea of the risk level, which can improve the safety of the driver. The testing outcomes of the developed fuzzy system are shown in Figure 11.

**Table 1.** Rules database.

| | | Velocity | | |
|---|---|---|---|---|
| | | High | Medium | Low |
| Depth | Very Low | very high | very high | high |
| | Low | very high | high | medium |
| | Medium | high | medium | medium |
| | High | medium | medium | low |
| | Very high | medium | low | very low |

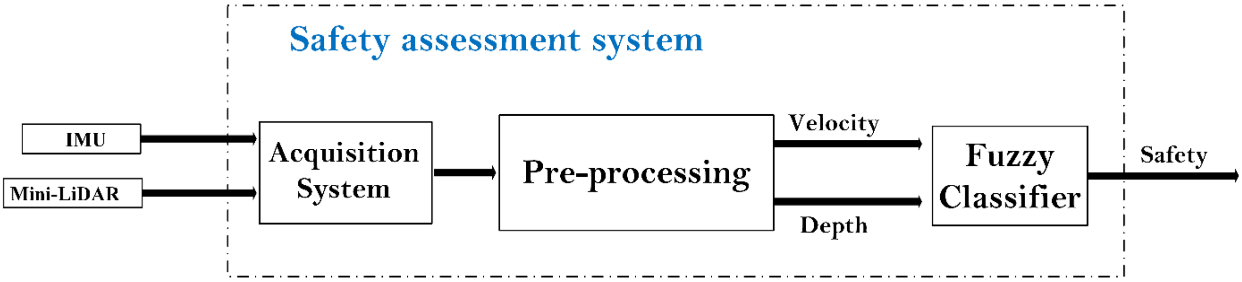


**Figure 6.** System for safety evaluation that combines sensor data fusion with fuzzy classification.

**Figure 7.** Fuzzy logic system.

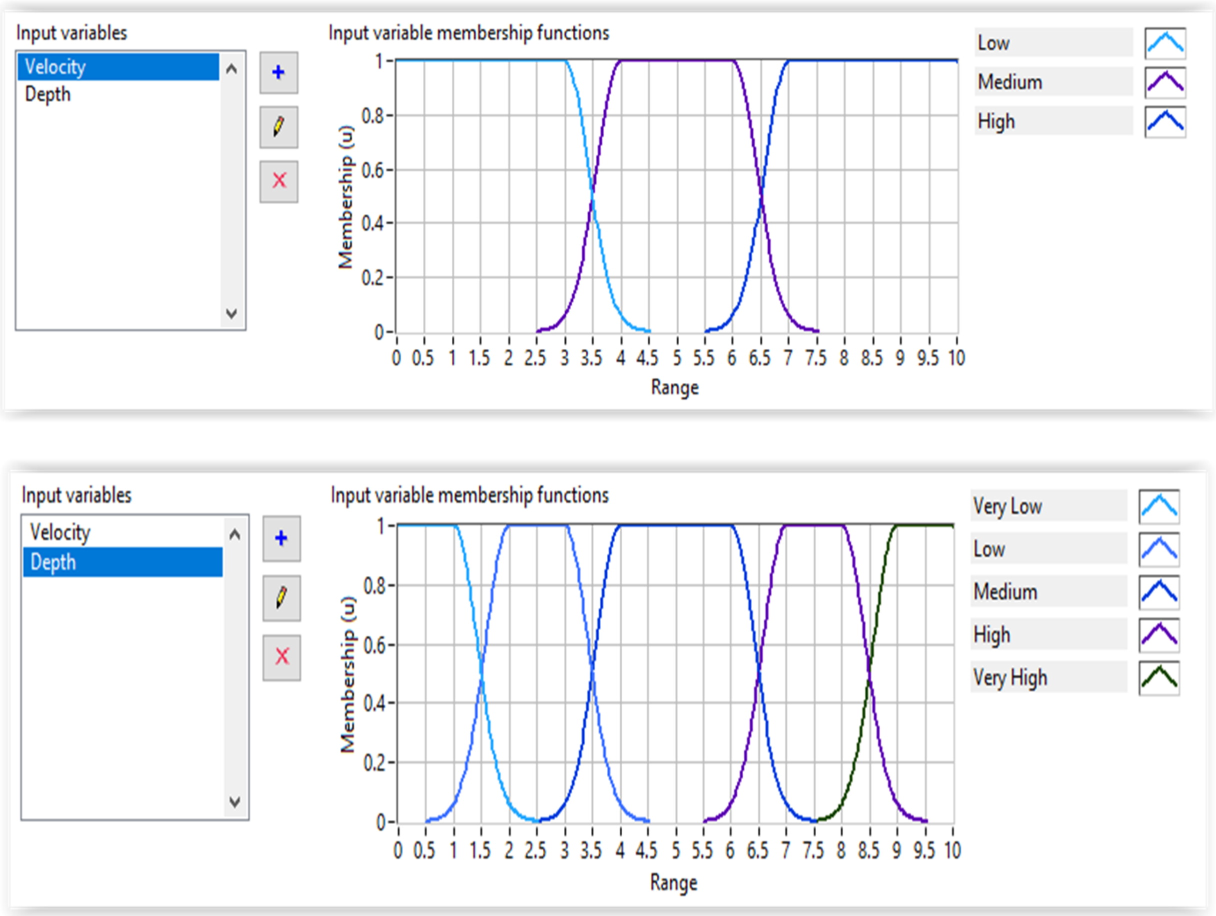


**Figure 8.** Inputs of the fuzzy classifier.

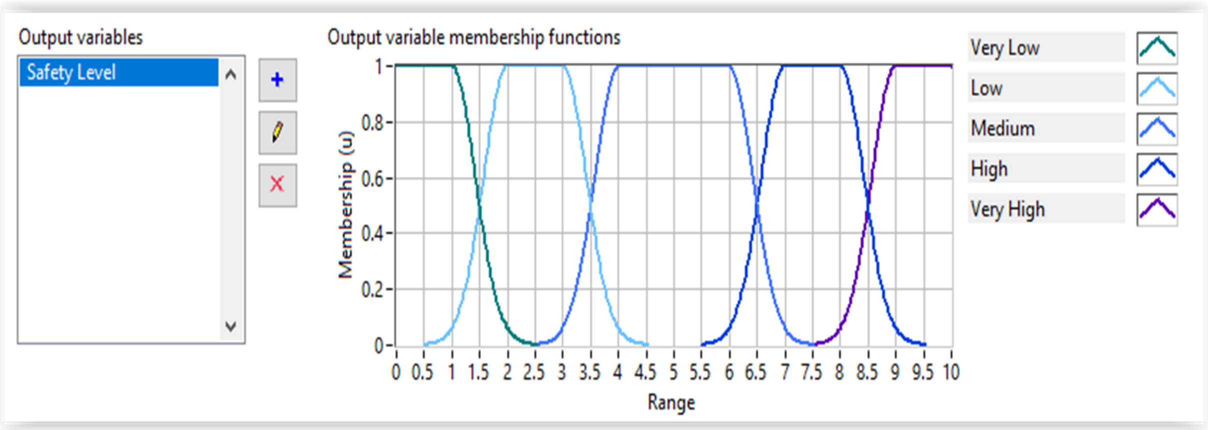
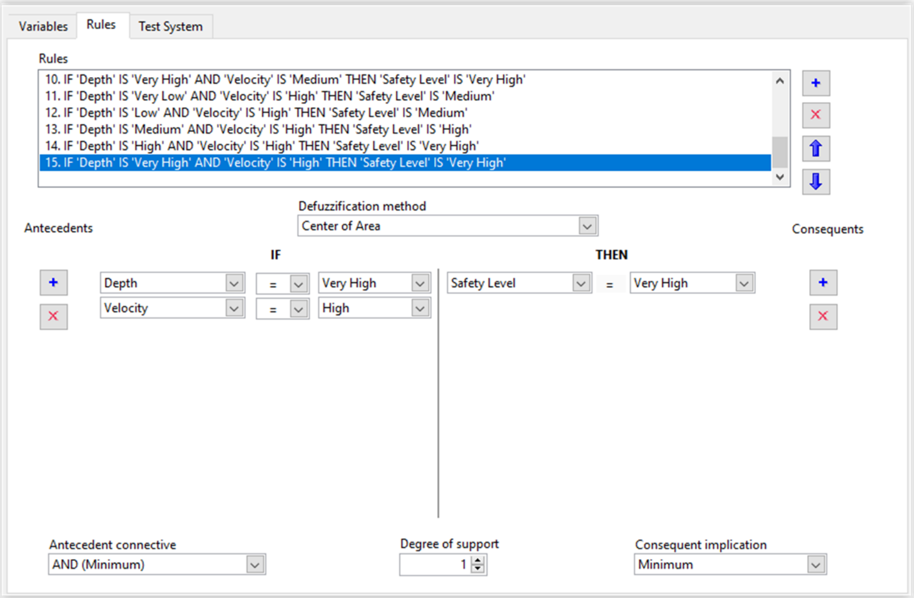


**Figure 9.** Output of the fuzzy classifier.



Rules

10. IF 'Depth' IS 'Very High' AND 'Velocity' IS 'Medium' THEN 'Safety Level' IS 'Very High'
11. IF 'Depth' IS 'Very Low' AND 'Velocity' IS 'High' THEN 'Safety Level' IS 'Medium'
12. IF 'Depth' IS 'Low' AND 'Velocity' IS 'High' THEN 'Safety Level' IS 'Medium'
13. IF 'Depth' IS 'Medium' AND 'Velocity' IS 'High' THEN 'Safety Level' IS 'High'
14. IF 'Depth' IS 'High' AND 'Velocity' IS 'High' THEN 'Safety Level' IS 'Very High'
15. IF 'Depth' IS 'Very High' AND 'Velocity' IS 'High' THEN 'Safety Level' IS 'Very High'

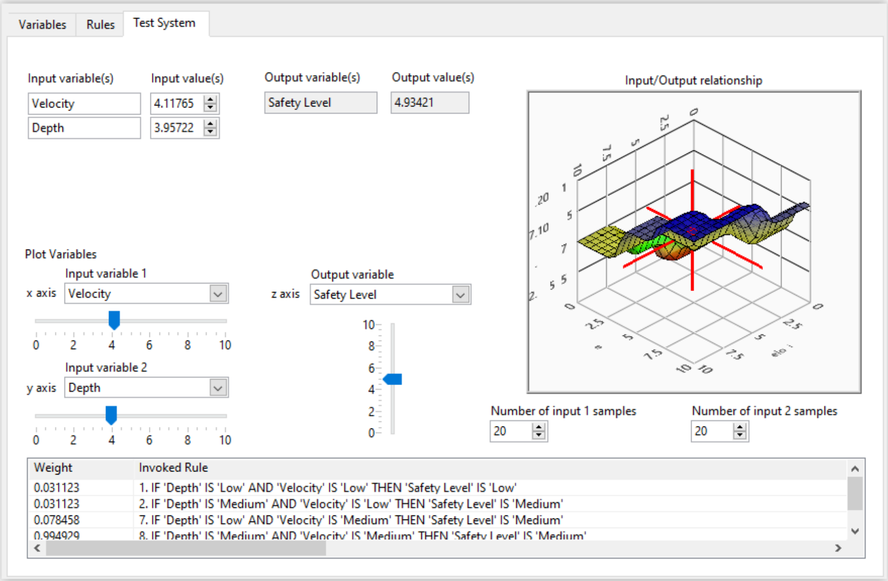
Defuzzification method: Center of Area

Antecedents
Depth = Very High
Velocity = High

Consequents
Safety Level = Very High

Antecedent connective: AND (Minimum)
Degree of support: 1
Consequent implication: Minimum

**Figure 10.** Implementation of the rules database.



| Input variable(s) | Input value(s) |
|---|---|
| Velocity | 4.11765 |
| Depth | 3.95722 |

| Output variable(s) | Output value(s) |
|---|---|
| Safety Level | 4.93421 |

Input variable 1
x axis: Velocity

Input variable 2
y axis: Depth

Output variable
z axis: Safety Level

Number of input 1 samples: 20
Number of input 2 samples: 20

| Weight | Invoked Rule |
|---|---|
| 0.031123 | 1. IF 'Depth' IS 'Low' AND 'Velocity' IS 'Low' THEN 'Safety Level' IS 'Low' |
| 0.031123 | 2. IF 'Depth' IS 'Medium' AND 'Velocity' IS 'Low' THEN 'Safety Level' IS 'Medium' |
| 0.078458 | 7. IF 'Depth' IS 'Low' AND 'Velocity' IS 'Medium' THEN 'Safety Level' IS 'Medium' |
| 0.994929 | 8. IF 'Depth' IS 'Medium' AND 'Velocity' IS 'Medium' THEN 'Safety Level' IS 'Medium' |

**Figure 11.** Testing system.

Figure 12 represents the navigation system's operational flowchart. This flowchart shows how the primary controller combines information and selects the safe course. Even if the three ultrasonic sensors—dedicated to identifying high obstacles—are acquired and processed directly by the Arduino board, the master board will still receive the measured values for data fusion and decision-making. The appropriate verbal message is generated by the master board and sent to the user. Figure 13, which depicts the main inputs and outputs of the suggested method, may be used to explain the core algorithm of the master board. Indeed, the safe path generation algorithm process inputs and generates the safe orientation and sends it to the driver. The inputs concerning the safety level estimated by the fuzzy classifier, the ultrasonic sensors data processed by the Arduino slave board, the current yaw angle provided by the IMU, and the GPS data are all useful information received via the Xbee protocol of other vehicles located in the surrounding area. The GPS information (latitude/longitude) should be used to determine the relative coordinates of the primary vehicle and the closest vehicle. The earth-centered, earth-fixed geodetic coordinate system is known as the GPS. It is necessary to convert this coordinate to the earth centered/earth fixed rectangular reference frame and to a local tangent plane in order to convert it to a local geographic coordinate system with the main vehicle as the origin. The earth centered/earth fixed rectangular coordinates can be calculated from the earth centered/earth fixed geodetic coordinate system, as shown in the following equation:

$$\begin{gathered} X = (h + n)cos\gamma \;\; cos\varphi \\ X = (h + n)cos\gamma \;\; sin\varphi \\ Z = (h + (1 + E^2) \;\; n)sin\gamma \end{gathered} \tag{1}$$

where $\gamma$ and $\varphi$ are the latitude and longitude, respectively, E deals with the eccentricity, and n represents the distance from the surface to the z axis along the ellipse normal as illustrated by Eq (2):

$$n = \frac{a}{\sqrt{1-E^2 \;\; sin^2\gamma}} \tag{2}$$

where $a$ is the WGS-84 earth semimajor axis. The following equation demonstrates how to change the earth centered/earth fixed rectangular reference frame to a local tangent plane coordinate system:

$$\begin{bmatrix} -sin\varphi & cos\varphi & 0 \\ -\; cos\varphi \;\; sin\gamma & -\; sin\gamma \;\; sin\varphi & cos\gamma \\ cos\gamma \;\; cos\varphi & cos\gamma \;\; sin\varphi & sin\gamma \end{bmatrix} = \begin{bmatrix} X - X_0 \\ Y - Y_0 \\ Z - Z_0 \end{bmatrix} \tag{3}$$

The local coordinate system is concluded by rotating the heading angle of the closest vehicle if X, Y, and Z are the earth centered/earth fixed rectangular coordinates of the primary vehicle and $X_0$, $Y_0$, and $Z_0$ are the coordinates of the nearest vehicle. Nevertheless, there is a flaw in that an inaccuracy occurs in the estimated relative coordinates if the heading angle of the closest vehicle is calculated with a poor accuracy. To get around this limitation, we approximated where the closest car was using the coordinates retrieved from the IMU. Accurate coordinate measurements are an advantage; however, additional labor is needed to filter and optimize the acquired data, which is a drawback. To address the issues mentioned above, we put forward a technique for calculating the precise location by combining the relative coordinates obtained from the IMU with the location data derived through V2V communication. The main vehicle's position can be accurately calculated, even though the relative position acquired through V2V communication is inaccurate.

A continuous value represents the vehicle trajectory. To give the driver a heads-up and avoid a collision, it is essential to estimate the location of the main vehicle in real-time. Given the width of each vehicle, the lateral position accuracy in the safe orientation prediction system is crucial for determining the width of the lane on the road. As a result, we divided the areas into nine grid cells, as shown in Figure 14, where the main vehicle is anticipated to be located after a specific length of time. We also propose two systems: one that anticipates the trajectory to determine the grid's horizontal lines and another that anticipates the desire to change lanes to determine the grid's vertical lines.

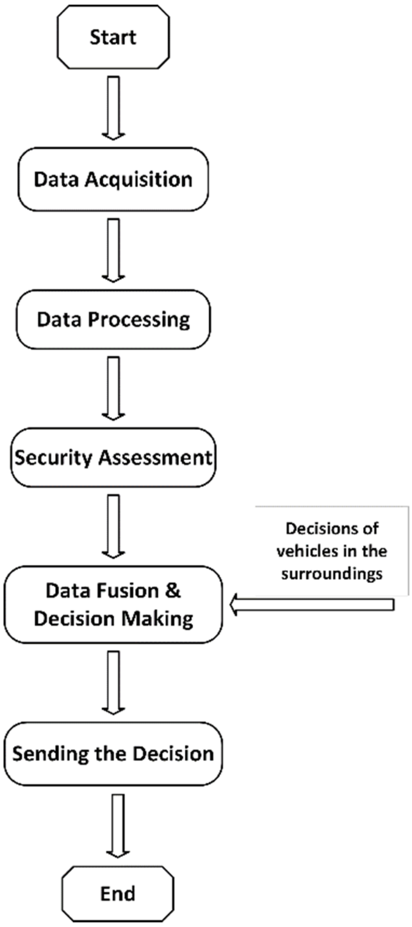


**Figure 12.** Operating flowchart.

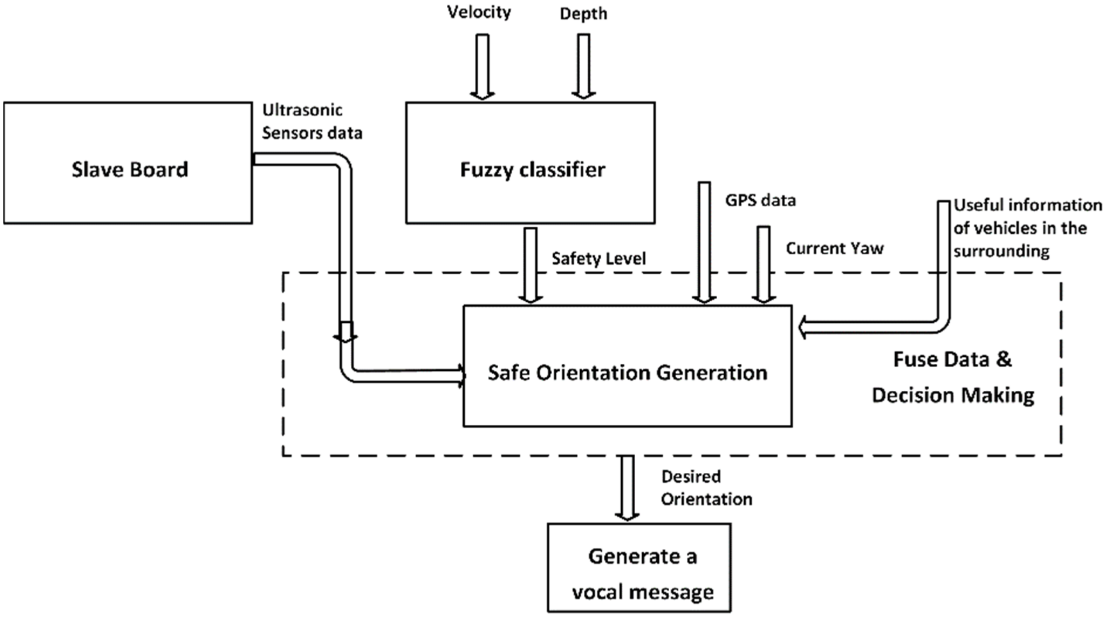


**Figure 13.** Block diagram of the main controller.

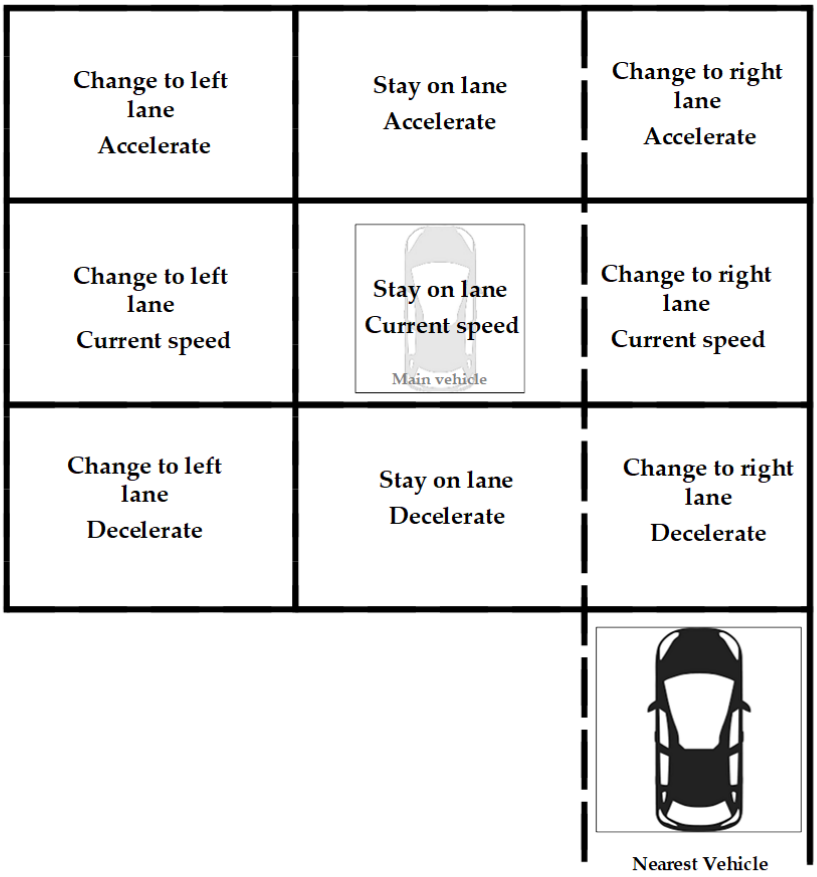


**Figure 14.** Grid with potential sites of the main vehicle.

## 4. Experimental results and discussion

### *4.1. Experimental setup*

The V2I system consists of two parts: a car unit and a central station. The car unit is responsible for collecting data and sending it to the central station. It is also responsible for receiving the data from the central station and providing it to the driver. The central station is responsible for analyzing the data. The car unit consists of nine pieces: 1) Raspberry PI, 2) Arduino Due, 3) GPS sensor, 4) IMU, 5) display, 6) Lidar sensor, 7) Xbee module, 8) Ultrasonic sensors and 9) Alcohol sensor. The microcontroller is the heart of the unit. It is responsible for collecting data from the sensors, analyzing it and forwarding it to the Xbee module. The GPS sensor will provide the location and speed. The location is used to locate drunk drivers, accidents, and traffic jams. The unit will only send location data in these situations. Regarding the speed, the system will warn the driver if he exceeds the speed limit, then issue a penalty. The use of the GPS to find the speed instead of the IMU ensures that the data is as accurate as possible. An alcohol sensor is used to detect the driver's mental state, since they are considered as a great threat to people's safety. The IMU is used to detect accidents which can be triggered by abnormal acceleration caused by a crash. The display provides the driver with the required information and warnings. The speech module is activated in case of severe warnings. The Xbee module is used to communicate with other vehicles and the central station via the roadside infrastructure.

The first step consists of installing the designed system. Ultrasonic sensors were plugged into the three sides of the front of the vehicle in order to detect the obstacles on the front side. The Lidar sensor is mounted into the lower area of the front of the vehicle detecting the ground shape. The remaining components were placed inside the vehicle in order to communicate with driver. The component plugging is illustrated by Figure 15. Figure 16(a),(b) shows the installation of the control system inside the vehicle and the placement of the GPS antenna on the vehicle's center, respectively. The main control system detects the vehicle's speed and orientation and alerts the driver about the safety level. Distances from other vehicles were acquired and processed to alert the driver about the desired speed and maintaining a safe distance. After the experimental setup, the designed system is implemented in three vehicles. All used vehicles are relatively new and in a good state. On the other hand, drivers are healthy and show a good experience in driving. Participants are trained to use the designed system, how to interpret the different outputs, and how to collaborate with it in different situations. They return good feedback that the designed system is high-friendly and easy to use. The testing phase is performed in good weather and road conditions.

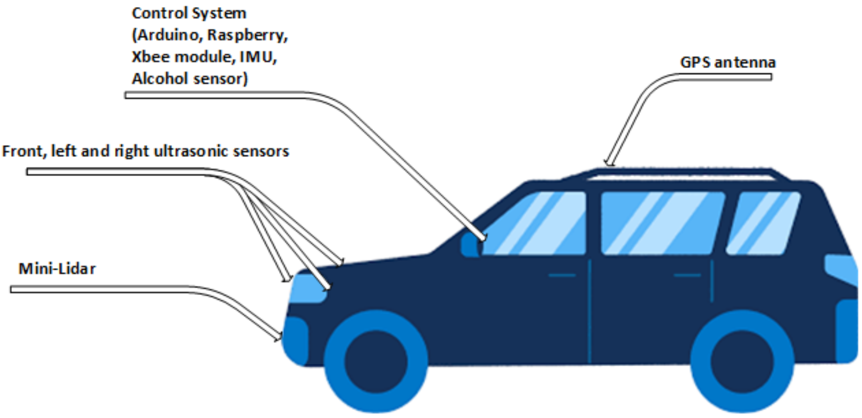


**Figure 15.** System plugging: (a) GPS placed on vehicle's center, (b) on board control system.

(a) (b)

**Figure 16.** System plugging: (a) GPS placed on vehicle's center, (b) on board control system.

### 4.2. Results and discussion

The designed system is implemented in three vehicles to test the performance in different situations. The number of trials is calculated using the full-factorial design approach. Indeed, a full factorial design of experiment (DOE) is one of several approaches to design an experiment to determine the effect that various levels of system inputs will have on the outputs. The purpose of the DOE is to determine at what levels of the inputs will the outputs be optimized. It can be calculated based on the following formula:

$$\#Runs = X * K \tag{4}$$

where X is the number of settings and K is the number of variables for factors.

In our experiment, we have three number of settings (X = 3), which are the vehicle speed, the safe distance and the road state. The number of variables is reduced in only one variable (K = 1), resulting in three scenarios. Each scenario combines three factors emulating the most real cases. The following selected scenarios are tested:

Scenario 1: One vehicle moves at a high speed and the remaining vehicles move at regular speed on the front side of the first vehicle, keeping a safe distance between them.

Scenario 2: The first vehicle moves at a regular speed to face a speed bump. The remaining two vehicles move at regular speed, keeping a safe distance between them.

Scenario 3: All vehicles move at high speed and the safe distance is not kept by two vehicles.

**Table 2.** Results of the tested scenarios.

| Scenario | Vehicle | Displayed message before driver intervening | Estimated risk level |  | Alarm |
|---|---|---|---|---|---|
|  |  |  | Before driver intervening | After driver intervening |  |
| Scenario-1 | Vehicle-1 | Stay on lane<br>Current speed | Very low | Very low | No alarm |
|  | Vehicle-2 | Stay on lane<br>Current speed | Medium | Very low | Smooth beep |
|  | Vehicle-3 | keep safe distance<br>Decelerate | High | Very low | Continuous beep |
| Scenario-2 | Vehicle-1 | Sudden Brake | High | Low | Continuous beep |
|  | Vehicle-2 | Stay on lane<br>Current speed | Very low | Low | No alarm |
|  | Vehicle-3 | Stay on lane<br>Current speed | Very low | Very low | No alarm |
| Scenario-3 | Vehicle-1 | Stay on lane<br>Decelerate | Medium | Very low | Smooth beep |
|  | Vehicle-2 | Stay on lane<br>Decelerate | Medium | Very low | Smooth beep |
|  | Vehicle-3 | keep safe distance<br>Decelerate | Very high | Low | Continuous beep |

Table 2 shows the results of the three tested scenarios that show the displayed messages and the estimated level of security. The driver was notified of the decision and took the relevant action to avoid the accident. Hence, the estimated level of danger is reported before and after the driver's intervention.

Figure 17 shows the warning messages "sudden brake" and "keep safe distance". At the same time, the GPS sensor collects locations of the traveled area (See Figure 18). GPS tracking is useful for the localization of all connected cars, and therefore, any accident can be detected and located rapidly. In addition, the vehicle subject to an accident can be easily identified.

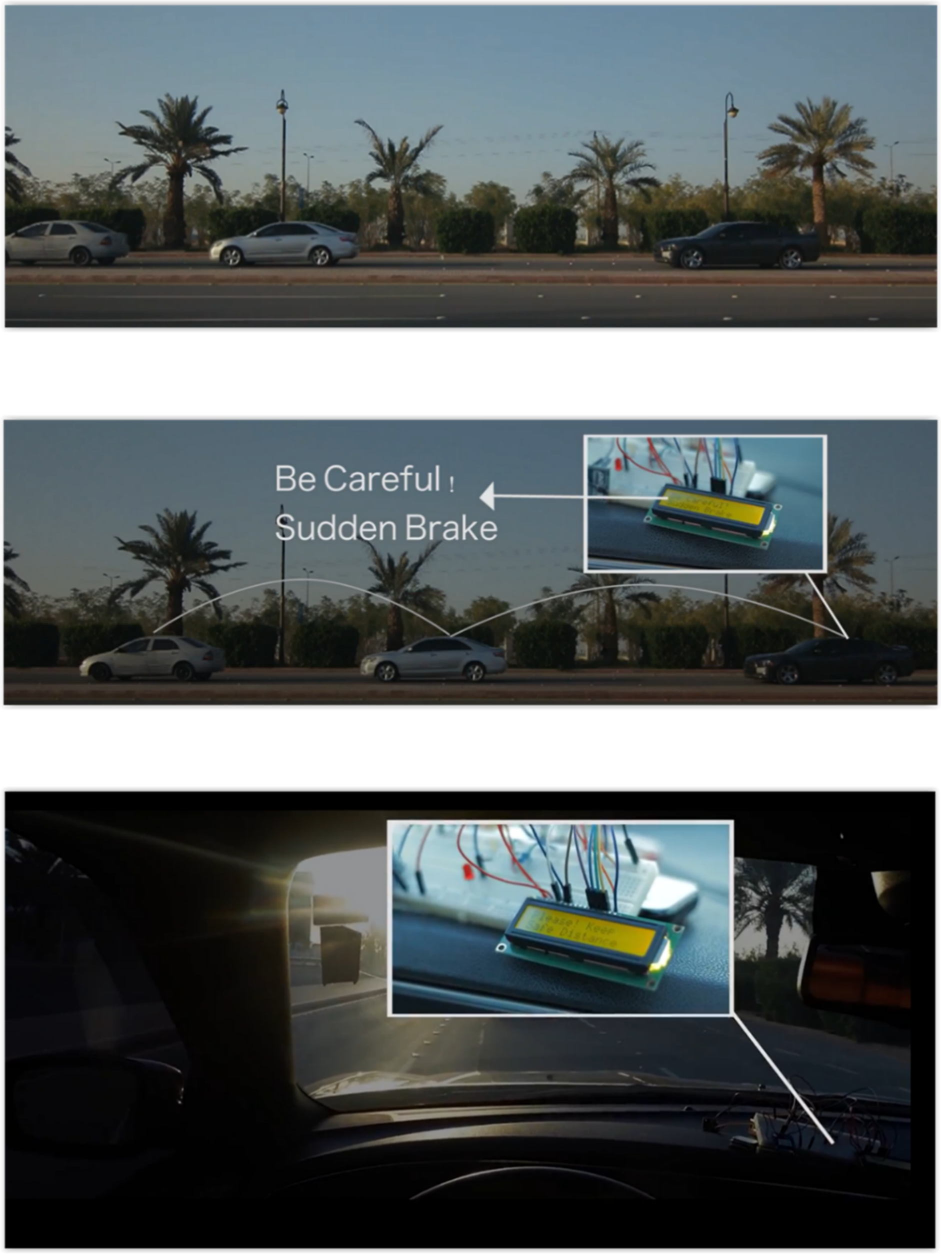


**Figure 17.** Warning messages: "Sudden Brake" and "keep safe distance".

Results show that the developed safety estimation system can immediately detect the risk level due to the real-time specifications of the ROS environment. The decision-making system provides a reliable risk level before the driver intervenes. The beep alarm is very important to alert the driver of the risk detection, and therefore, the driver can show the graphical message to apply the related actions. The driver is limited by 1 second to apply the desired action. Otherwise, they will be in a complicated situation if the decision is not applied in the allowable time. This special scenario is not simulated because the accident can occur.

**Figure 18.** Locations collected from GPS sensor.

The accuracy of the classifier and the response time are reported in Table 3. The accuracy is computed by comparing the system decision with the real situation. The designed system shows a 90% accuracy as an average value. This result is sensitive to the speed value since the speed can be acquired in real-time via the IMU. On the other hand, the accuracy can decrease if the main risk source is the depth data due to the latency of the acquisition time of the depth data. This can be accepted if we assume that the car's velocity is a more important effect if the accident occurred. Moreover, the average value of the response time to take the decision can be optimized to be more suitable for non-expert drivers. After applying the decision, the stabilization time is relatively acceptable since the risk is fully avoided.

**Table 3.** Performance evaluation.

| Scenario | Accuracy | Response time (to take decision) | Stabilization time (after applying decision) |
|---|---|---|---|
| Scenario-1 | 85% | 230 ms | 513 ms |
| Scenario-2 | 90% | 255 ms | 644 ms |
| Scenario-3 | 95% | 195 ms | 550 ms |
| Average | 90% | 226 ms | 569 ms |

Although the testing results show a good performance for risk avoidance, the current system is limited by the back side screening. Indeed, only the front side is considered for data acquisition. However, some special scenarios require the scanning data of the back side to take a more reliable decision. Moreover, the experiment environment can be considered as an ideal environment with no external disturbance. Indeed, it's important to run the selected scenarios in more complicated traffic by integrating more external vehicles. The next step consists of applying the designed system in the real-word application.

## 5. Conclusions

A real environment was used to create and test a real-time V2V architecture. High-performance sensors and an integrated computer running ROS comprised the suggested architecture. Advanced sensors that provide information about hazards in front of the driver, as well as navigational metrics (position, speed, acceleration, and orientation), are built into the designed architecture. To determine the vehicle's safe route, a safety management system that combines sensor data fusing and fuzzy classifier is implemented. Drivers in various environments tested the proposed V2V architecture in real-time, and the results are encouraging. Real driver testing revealed that the designed navigation aid system improved obstacle identification, ramp and cavity recognition, and interactions with other vehicles. Future works consist of integrating the designed system into the control system of the intelligent car. Since the ROS-based architecture is open source and can be easily extended, the designed system can collaborate with the internal embedded control system of the intelligent car. On the other hand, weather conditions should be acquired by integrating more sensors in the acquisition system improving the safety of the related vehicle. Moreover, the decision-making system can be optimized by integrating more reliable machine learning algorithms such as deep learning to improve safety. However, this may require more computing resources to meet real-time requirements.

## Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

## Acknowledgments

This research has been funded by Scientific Research Deanship at University of Ha'il-Saudi Arabia through project number RD-21 112.

## Conflict of interest

The authors declare there is no conflict of interest.

## References

1. M. Peden, Global collaboration on road traffic injury prevention, *Int. J. Inj. Control Saf. Promot.*, **12** (2005), 85–91. https://doi.org/10.1080/15660970500086130
2. T. J. C. Sousa, V. Monteiro, J. C. A. Fernandes, C. Couto, A. A. N. Meléndez, J. L. Afonso, New perspectives for vehicle-to-vehicle (V2V) power transfer, in *IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society*, IEEE, (2018), 5183–5188. https://doi.org/10.1109/IECON.2018.8591209
3. T. Petrov, M. Dado, K. E. Ambrosch, P. Holečko, Experimental topology for V2V communication based on Internet of Things, in *2016 ELEKTRO*, IEEE, (2016), 72–76.
4. B. M. Masini, A. Bazzi, A. Zanella, A survey on the roadmap to mandate on board connectivity and enable V2V-based vehicular sensor networks, *Sensors*, **18** (2018), 2207. https://doi.org/10.3390/s18072207

5. M. Shurrab, S. Singh, H. Otrok, R. Mizouni, V. Khadkikar, H. Zeineldin, An efficient vehicle-to-vehicle (V2V) energy sharing framework, *IEEE Internet Things J.*, **9** (2021), 5315–5328. https://doi.org/10.1109/JIOT.2021.3109010
6. B. S. Umesh, V. Khadkikar, H. H. Zeineldin, S. Singh, H. Otrok, R. Mizouni, Direct electric vehicle to vehicle (V2V) power transfer using on-board drivetrain and motor windings, *IEEE Trans. Ind. Electron.*, **69** (2021), 10765–10775. https://doi.org/10.1109/TIE.2021.3121707
7. J. Santa, F. Pereñíguez, J. C. Cano, A. F. Skarmeta, C. T. Calafate, P. Manzoni, Comprehensive vehicular networking platform for V2I and V2V communications within the walkie-talkie project, *Int. J. Distrib. Sens. Netw.*, **9** (2013), 676850. https://doi.org/10.1155/2013/676850
8. A. R. Khan, M. F. Jamlos, N. Osman, M. I. Ishak, F. Dzaharudin, Y. K. Yeow, et al., DSRC technology in vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) IoT system for intelligent transportation system (ITS): a review, *Recent Trends Mech. Ind. 4.0*, **730** (2022), 97–106. https://doi.org/10.1007/978-981-33-4597-3_10
9. C. Dong, H. Wang, Y. Li, Y. Liu, Q. Chen, Economic comparison between vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) at freeway on-ramps based on microscopic simulations, *IET Intel. Transport Syst.*, **13** (2019), 1726–1735. https://doi.org/10.1049/iet-its.2018.5537
10. R. Q. Malik, H. A. Alsattar, K. N. Ramli, B. B. Zaidan, A. A. Zaidan, Z. H. Kareem, et al., Mapping and deep analysis of vehicle-to-infrastructure communication systems: coherent taxonomy, datasets, evaluation and performance measurements, motivations, open challenges, recommendations, and methodological aspects, *IEEE Access*, **7** (2019), 126753–126772. https://doi.org/10.1109/ACCESS.2019.2927611
11. P. Liu, W. D. Fan, Exploring the impact of connected and autonomous vehicles on mobility and environment at signalized intersections through vehicle-to-infrastructure (V2I) and infrastructure-to-vehicle (I2V) communications, *Transp. Plann. Technol.*, **44** (2021), 129–138. https://doi.org/10.1080/03081060.2020.1868088
12. A. Kovalenko, R. F. Hussain, O. Semiari, M. A. Salehi, Robust resource allocation using edge computing for vehicle to infrastructure (v2i) networks, in 2019 *IEEE 3rd International Conference on Fog and Edge Computing (ICFEC)*, IEEE, 2019. https://doi.org/10.1109/CFEC.2019.8733151
13. A. Vinel, N. Lyamin, P. Isachenkov, Modeling of V2V communications for C-ITS safety applications: A CPS perspective, *IEEE Commun. Lett.*, **22** (2018), 1600–1603. https://doi.org/10.1109/LCOMM.2018.2835484
14. J. Thota, N. F. Abdullah, A. Doufexi, S. Armour, V2V for vehicular safety applications, *IEEE Trans. Intell. Transp. Syst.*, **21** (2019), 2571–2585. https://doi.org/10.1109/TITS.2019.2920738
15. H. S. Basheer, C. Bassil, A review of broadcasting safety data in V2V: Weaknesses and requirements, *Ad Hoc Networks*, **65** (2017), 13–25. https://doi.org/10.1016/j.adhoc.2017.07.004
16. H. Chehardoli, A. Ghasemi, Adaptive centralized/decentralized control and identification of 1-D heterogeneous vehicular platoons based on constant time headway policy, *IEEE Trans. Intell. Transp. Syst.*, **19** (2018), 3376–3386. https://doi.org/10.1109/TITS.2017.2781152
17. E. Abolfazli, B. Besselink, T. Charalambous, On time headway selection in platoons under the MPF topology in the presence of communication delays, *IEEE Trans. Intell. Transp. Syst.*, **23** (2021), 8881–8894. https://doi.org/10.1109/TITS.2021.3087484
18. E. Abolfazli, B. Besselink, T. Charalambous, Reducing time headway in platoons under the MPF topology in the presence of communication delays, *IFAC-PapersOnLine*, **53** (2020), 15267–15274. https://doi.org/10.1016/j.ifacol.2020.12.2319

19. T. C. Au, S. Zhang, P. Stone, Autonomous intersection management for semi-autonomous vehicles, *Handb. Transp.*, (2015), 88–104.
20. J. Wang, J. Wang, R. Wang, Trajectory replanning in V2V lane exchanging with consideration of driver preferences, in *2016 American Control Conference (ACC)*, *IEEE*, (2016), 3492-3497. https://doi.org/10.1109/ACC.2016.7525454
21. M. Repoux, N. Geroliminis, M. Kaspi, Operational analysis of an innovative semi-autonomous on-demand transportation system, *Transp. Res. Part C Emerging Technol.*, **132** (2021), 103373. https://doi.org/10.1016/j.trc.2021.103373
22. E. V. Nunen, J. Verhaegh, E. Silvas, E. Semsar-Kazerooni, N. Wouw, Robust model predictive cooperative adaptive cruise control subject to V2V impairments, in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2017. https://doi.org/10.1109/ITSC.2017.8317758
23. V. Milanés, S. E. Shladover, Handling cut-in vehicles in strings of cooperative adaptive cruise control vehicles, *J. Intell. Transp. Syst.*, **20** (2016), 178–191. https://doi.org/10.1080/15472450.2015.1016023
24. S. E. Shladover, C. Nowakowski, X. Y. Lu, R. Ferlis, Cooperative adaptive cruise control: Definitions and operating concepts, *Transp. Res. Rec.*, **2489** (2015), 145–152. https://doi.org/10.3141/2489-17
25. M. Gupta, J. Benson, F. Patwa, R. Sandhu, Secure V2V and V2I communication in intelligent transportation using cloudlets, *IEEE Trans. Serv. Comput.*,**15** (2020), 1912–1925. https://doi.org/10.1109/TSC.2020.3025993
26. J. A. Sanguesa, J. Barrachina, M. Fogue, P. Garrido, F. J. Martinez, J. Cano, et al., Sensing traffic density combining V2V and V2I wireless communications, *Sensors*, **15** (2015), 31794–31810. https://doi.org/10.3390/s151229889
27. M. S. Sheikh, J. Liang, A comprehensive survey on VANET security services in traffic management system, *Wireless Commun. Mobile Comput.*, **2019** (2019), 2423915. https://doi.org/10.1155/2019/2423915.
28. G. N. Bifulco, A. Coppola, A. Petrillo, S. Santini, Decentralized cooperative crossing at unsignalized intersections via vehicle-to-vehicle communication in mixed traffic flows, *J. Intell. Transp. Sys.*, (2022), 1–26. https://doi.org/10.1080/15472450.2022.2124868
29. M. H. C. Garcia, A. Molina-Galan, M. Boban, J. Gozalvez, B. Coll-Perales, T. Şahin, et al., A tutorial on 5G NR V2X communications, *IEEE Commun. Surv. Tutorials*, **23** (2021), 1972–2026. https://doi.org/10.1109/COMST.2021.3057017
30. Y. Jo, J. Jang, J. Ko, C. Oh, An in-vehicle warning information provision strategy for V2V-based proactive traffic safety management, *IEEE Trans. Intell. Transp. Syst.*, **23** (2022), 19387–19398. https://doi.org/10.1109/TITS.2022.3156923
31. S. Bahramnejad, N. Movahhedinia, A reliability estimation framework for cognitive radio V2V communications and an ANN-based model for automating estimations, *Computing*, **104** (2022), 1923–1947. https://doi.org/10.1007/s00607-022-01072-7
32. A. E. Cox, A. S. Mueller, J. B. Cicchino, Safety potential of crash avoidance features, improved headlights, and V2V-enhanced technologies for older drivers, *J. Saf. Res.*, **85** (2023), 200–209. https://doi.org/10.1016/j.jsr.2023.02.004