



Driver assistance collision warning system using a LIDAR sensor with kinematics and perception algorithms

Willi Immanuel Susanto¹, Henry Nasution^{2*}, Tanika Dewi Sofianti¹

¹Department of Mechanical Engineering, Faculty of Engineering and Information Technology, Swiss German University, Indonesia

²Renewable Energy Engineering Technology, Faculty of Industrial Technology, Bung Hatta University, Indonesia

Abstract

Road accidents remained a significant global concern, causing loss of life and economic damage. To mitigate this issue, the automotive industry has increasingly invested in Advanced Driver Assistance Systems to enhance vehicle safety. This research presented a Driver Assistance Collision Warning System that incorporated kinematics and perception algorithms to improve collision prevention. The system utilized a LIDAR sensor to capture real-time data regarding the distance to the vehicle in front of it. This data was integrated with an Arduino microcontroller to compute the relative speed and time of collision. Upon detecting a collision risk, the system triggered a warning mechanism, which included an audible alert provided by a buzzer and a visual warning displayed on the head-up display. The system integrated kinematics algorithms, which processed sensor-read values to generate real-time decisions utilizing a specific threshold time to collision, and perception algorithms relied on Fuzzy Logic to handle uncertainty and improve accuracy. Validation was conducted through integration, system, and acceptance testing, demonstrating reliable synchronization of algorithms and accurate operation in real-world environments. The results showed that the system achieved a collision risk detection accuracy of ± 5 cm within five different environmental factors. These findings confirmed the system's potential as a reliable solution for real-world collision prevention.

Keywords:

Collision;
Driver Assistance;
Kinematics Algorithm;
LIDAR;
Perception Algorithm;

Article History:

Received: August 2, 2024
Revised: January 23, 2025
Accepted: February 5, 2025
Published: September 1, 2025

Corresponding Author:

Henry Nasution,
Renewable Energy Engineering
Technology, Faculty of Industrial
Technology, Bung Hatta
University, Indonesia
Email:
henrynasution@bunghatta.ac.id

This is an open-access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



INTRODUCTION

Road accidents are an inevitability. The World Health Organization reports that over 3,000 individuals perish in vehicle accidents daily [1]. The National Transportation Safety Committee (KNKT) of the Republic of Indonesia reports that the total number of road accident cases in Indonesia was 100,028 in 2020. The situation worsened in 2021, with road accidents increasing by approximately 4% [2]. Road traffic injuries cause substantial economic harm to individuals, their families, and the nation. These losses stem from treatment expenses and diminished output

for individuals who are killed or incapacitated due to their injuries, as well as for family members who must forego a job or education to assist the injured. According to reports, material losses in 2021 are likely to reach IDR 247 billion due to monetary impacts [2].

A variety of technologies have been developed in the area of Advanced Driver Assistance Systems (ADAS) that offer safety features designed to enhance driver safety and facilitate a more comfortable driving experience and reduce road accidents. Research by the Traffic Safety Researcher indicates that findings

from police-reported crash analyses are converging, suggesting that Vulnerable Road User ADAS decreases pedestrian crashes by 13% to 27% [3]. Theoretically, the adoption of ADAS in automobiles is supposed to reduce the frequency of road accidents. Notwithstanding, ADAS systems are presently only integrated into contemporary high-end vehicles. The car's price exhibits a linear correlation with the types of technologies used in safety systems. These findings are inconsistent with consumer demand, as the safety system is crucial for all customer segments.

In recent years, vehicle safety systems have been a significant focus of research, with numerous recommendations for methodologies put forth. Numerous studies have been conducted on this subject; the following are several notable studies that address the issue.

H. Kunto D. A. conducted a study on a vehicle anti-collision system utilizing Arduino Uno. This system employs object detection sensors (ultrasonic and laser range finder) as inputs to a warning system output (buzzer and LED) through kinematic logic [4]. In another study, Yuan, Yuwei Lu, and Qi Wang investigate car driving assistance based on the driver's facial positions, utilizing a dataset constructed by the researchers. Technology utilizes machine learning algorithms that require comprehensive pre-trained models to forecast collision risks based on historical driver behavior [5]. While both H. Kunto's kinematic-based system and Yuan, Yuwei Lu, and Qi Wang's perception-based approach provide valuable insights, each has limitations. Kunto's method lacks adaptability to environmental changes, and Yuan's system, though adaptable, is hindered by its reliance on extensive pre-trained models, which compromise real-time responsiveness. To address these issues, the planned research aims to develop a hybrid system that combines the strengths of both kinematic and perception-based approaches. This new system aims to deliver real-time accuracy and adapt to changing conditions, thereby enhancing collision prevention.

This research aims to develop a brand-agnostic Driver Assistance Collision Warning System that ensures universal compatibility across vehicle brands and segments through Arduino, while seamlessly integrating LIDAR-based kinematics algorithms for real-time decision-making with AI-driven perception algorithms utilizing fuzzy logic to enhance collision prediction accuracy and adaptability in diverse driving conditions. The kinematics-based method uses sensor-read values as input to produce a decision output. The perception-based algorithm

employs Artificial Intelligence (AI) logic to analyze rule-based data from the fuzzy system. Rule-based systems are one of the stages of an AI system, where a computer uses rules [6].

The system integrates kinematics algorithms, which process sensor readings to generate real-time decisions, and perception algorithms, which employ fuzzy logic to handle uncertainty and enhance accuracy. This integration ensures intelligent collision reduction, adaptable across vehicle brands and segments.

METHOD

The research paper presents a theoretical framework that establishes the development of the Driver Assistance Collision Warning System. Driver Assistance System reduces exposure to hazardous situations and enhances driving comfort by providing warnings or automating dynamic driving tasks [7]. This system aims to intelligently minimize collisions by integrating kinematic and perception algorithms. The research methodology to be utilized for the Vehicle Collision Warning System is the V-model project milestone approach. The V-model approach to development, well-established in the automotive industry, is subject to high regulations imposed by the requirement for compliance with standards [8]. Compared to the other method, the V-model provides more proper handling for support software integration [9]. The V-model methodology divides the development phase into design, implementation, integration, and system testing.

V-Model Project

The letter "V" symbolizes the development flow, with the left side indicating requirements and specifications, while the right side represents verification processes. The horizontal connection between the left V side and the right V side signifies that verification must adhere to the requirements.

The V-model begins with the requirement stage, also known as the pre-development stage, where the system's requirements and architecture are established. Moving down the left side of the "V", the main development stage involves Hardware (H/W) and Software (S/W) development, following three phases: component development, implementation with unit analysis, and final integration. After development, the process shifts to the right side of the "V" for System Verification, ensuring validation and testing align with the defined requirements. [Figure 1](#) illustrates the detailed flow.

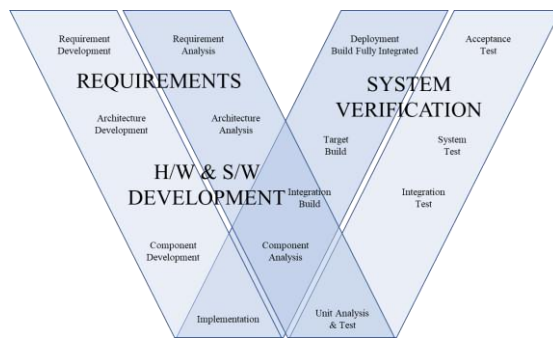


Figure 1. Research Method using V-Model

Pre-development Stage

The pre-development stage defines the essential system requirements in two phases: Requirement and Architecture Development. The driver assistance system selects three primary variables: distance to the front vehicle (d), relative speed (vr), and time-to-collision (TTC). The sensor measures distance in centimetres and is integrated with an Arduino microcontroller, which calculates relative speed to determine the TTC. The TTC notion denotes the time it takes for the front end of the following vehicle to reach the rear end of the leading vehicle, assuming both vehicles proceed at their current speeds and on the same path [10]. The time-to-collision combines the spatial distances with the (relative) velocities to quantify the 'distance' to a collision [11]. Equation (1) and (2) shows the Relative Speed and the TTC formula.

$$vr = \frac{d_B - d_A}{t_B - t_A} \quad (1)$$

$$TTC = \frac{d}{vr} \quad (2)$$

Furthermore, the second requirement is the warning output, categorized into two types: audible warnings, which deliver immediate notifications by sound, and visual warnings. The UN ECE guarantees that these laws are both universally implemented and inclusive, encouraging a global approach to road safety. Standards and conventions developed in UNECE are used worldwide [12].

The audible warning device shall emit a continuous and uniform sound; its acoustic spectrum shall not vary substantially during its operation [13]. At the same time, three visual indication categories are mentioned in the UN ECE Regulation No. 121: Control, Tell-tale, and Indicator [14]. The study incorporates an electric buzzer for audible warnings and LEDs for the visual warning category. The details of the warning symbol are outlined in Table 1.

Table 1. Warning Symbol

Item	Control Symbol	Function	Illumination Color
Emergency Brake		Warning for emergency braking requirement upon detection of an incoming front collision.	Yellow
Brake		Warning for taking braking action: light brake, moderate brake, significant brake	Green, Yellow, Red

Table 2. Kinematics Algorithm Guide

Time-to-Collision (sec)	Actuator	
	Emergency Brake LED	Buzzer
> 3	Off	Off
$2 > TTC \leq 3$	Amber	Off
≤ 2	Amber	On

Table 2 shows the system's decision based on the 3-second rule and speed limit, which serves the final requirement in the pre-development stage. The vehicle collision warning system must adjust its modifications in guidance based on varying conditions. The driver assistance system that has been developed will be implemented as a kinematics and perception algorithm used to make decisions based on facts.

The kinematics algorithm follows a strict true/false approach. It adheres to the 3-second rule as advised by the educational movement from the Toll Road Regulatory Agency of the Ministry of Public Works and Public Housing [2]. The older recommendation is the following 2-second rule. However, based on a study by highway engineers, states and traffic safety organizations have more recently referred to a 3-second rule [15].

Meanwhile, the perception algorithm applies Fuzzy Logic to suggest braking actions based on distance and relative speed. It classifies inputs into four levels: Significant, Moderate, Light, and Safe, determining the appropriate braking response.

The final pre-development stage progresses to architecture development through the components block diagram, as shown in Figure 2.

The system employs the LIDAR sensor to measure the distance to the front vehicle. LIDAR is a distance sensor that is useful for the development of ADAS and autonomous driving [16, 17]. The vehicle's relative speed and Time-to-Collision are derived from calculations performed by the Arduino Mega 2560.

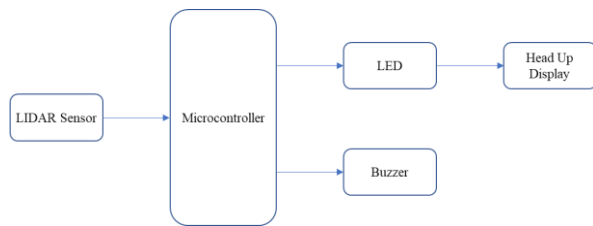


Figure 2. Collision Warning System Block Diagram

It operates at 5 V and is easy to use, not least because several electronic components operate at the same 5 V [18]. To assist the driver with critical safety information, the system provides safety warnings through two actuators: an LED-based head-up display (HUD) for visual alerts and a buzzer for auditory signals to grab the driver's attention.

Calibration, Testing, and Validation

A critical phase of unit analysis and testing is calibration, which involves evaluating and validating each component. Calibration is essential to guarantee that the data collected is not only crisp but also accurately represented. Calibration enables the systems to be adjusted for natural driving habits, hence enhancing customer acceptance of driver assistance systems [19]. The calibration procedure consists of two phases: confirming that the LED and buzzer react appropriately to the microcontroller's inputs and ensuring the LIDAR sensor accurately measures the actual distance.

Additionally, testing and validation ensure that the methodology, data, and results align with the research objectives. According to the V-Model project milestone, the validation phase necessitates verifying the accuracy of the temporal configuration by comprehensive testing of the implementation on the target [4]. The phase involves three testing methods: integration testing, system testing, and acceptance testing.

RESULTS AND DISCUSSION

The current phase has progressed to the primary development stage of the V-model, which provides an in-depth examination of the structured steps involved in development, calibration, and testing. These discussions are based on the design concepts established during the pre-development phase.

Hardware Development

In terms of hardware, this necessitates workable specifications that are feasible. A workable specification is needed to start producing a prototype for the design, ensuring that the design specification is well-drafted [20]. The

wiring diagram serves as a workable specification, detailing all electrical connections, including cable arrangement, components, and connection points. Figure 3 illustrates the system's wiring diagram.

The wiring diagram built with Fritzing software serves as an essential blueprint for the assembly and integration of hardware components. The hardware assembly displayed in Figure 4 is executed by referencing the blueprint specifications in the wiring diagram.

Software Development

Software development requires the creation of a calibration program to synchronize the hardware's output with established standards. The calibration program is organized based on an integrated system of sensors and actuators.

The actuator calibration utilizes a calibration approach that involves a distinctive Arduino program, adhering to the flowchart in Figure 5. The result of the actuator calibration is displayed in Table 3.

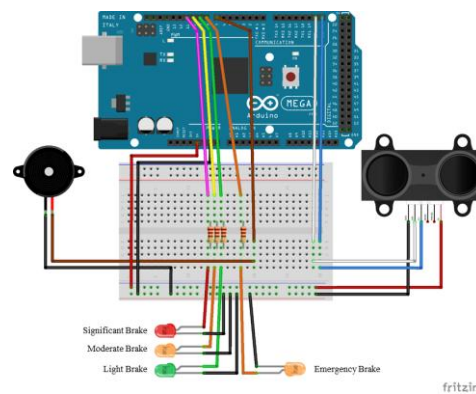


Figure 3. Wiring Diagram of Warning System

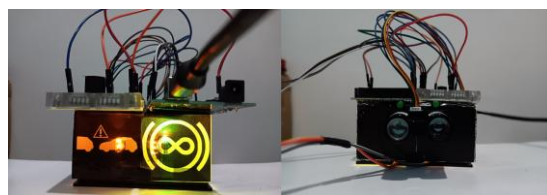


Figure 4. Vehicle Collision Warning System Assy

Table 3. Calibration Result for LIDAR Sensor

Actual Range (cm)	System Read (cm)	Standard (cm)	Result (cm)	Remark
55	56~59	± 5 cm	Max +4 cm	Within Spec
90	93~95	± 5 cm	Max +5 cm	Within Spec
280	284~285	± 5 cm	Max +5 cm	Within Spec
565	566~568	± 5 cm	Max +3 cm	Within Spec
1520	1522~1524	± 5 cm	Max +4 cm	Within Spec

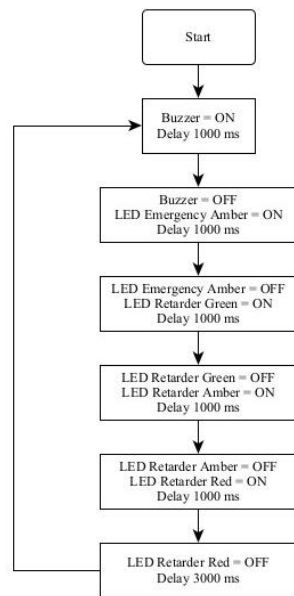


Figure 5. Actuator Calibration Flow Procedure

The results of the sensor calibration are displayed in Table 4. Sensor calibration ensures the accuracy of the Garmin LIDAR Lite v3 by comparing values acquired from a manual distance measurement using a tape measure with the distance output displayed in the Arduino serial monitor.

Integration Build

This phase involves integrating several software and hardware components to create a unified system. The integration of the system entails three steps in the decision-making process: the kinematics algorithm, the perception algorithm, and the comprehensive system that integrates kinematics and perception algorithms.

The Kinematics algorithm approaches the specified target following the 3-second rule guidance. The Arduino program has been set up with three if-conditions, as shown in Table 5, while also considering the computation time required for measuring distance using LIDAR.

Table 4. Actuators Calibration Results

Actuator		Step 1	Step 2	Step 3	Step 4	Step 5
Buzzer		ON	OFF	OFF	OFF	OFF
LED Emergency Brake		OFF	ON	OFF	OFF	OFF
LED Retarder	Green	OFF	OFF	ON	OFF	OFF
LED Retarder	Amber	OFF	OFF	OFF	ON	OFF
LED Retarder	Red	OFF	OFF	OFF	OFF	ON

Table 5. Kinematics Algorithm Decision

Time-to-Collision (sec)	Interval (sec)	Actuator	
		Emergency Brake LED	Buzzer
> 2	1	Off	Off
$1 > TTC \leq 2$	1	Amber	Off
≤ 1	1	Amber	On

The perception algorithm relies on Fuzzy Logic. As more scenarios necessitate decisions that cannot be resolved with a mere yes or no response, the use of fuzzy logic to facilitate decision-making becomes increasingly essential [21]. The development of Fuzzy Logic will be executed utilizing MATLAB. To establish a fuzzy logic system that offers suggestions for braking decisions based on the distance to the front vehicle and the relative speed, the following is a comprehensive and systematic approach to constructing the fuzzy system:

First Step: Address the Inputs and Outputs

The development begins with the identification of the variables. The model's inputs are the distance to the front vehicle and the relative speed. The result displays the Brake Suggestion, specifying the recommended braking action. A crisp value within a specified range must be present in every variable. Table 6 presents comprehensive data regarding the precise crisp values for each variable.

Second Step: Fuzzification

The fuzzification process subsequently transforms crisp value inputs into fuzzy inputs. The condition is accomplished by assigning every variable of input to a collection of linguistic concepts, each indicated by a fuzzy membership function (MF). The MF type is characterized by the application of Trapezoidal and Triangular shapes, attributed to its widespread use and good performance. Nasution's 2011 research indicates that the type is simple, providing good controller performance and being easy to handle [22]. Table 7 illustrates the membership function of each variable.

Subsequently, the crisp value was generated in line with the given requirements and is currently being incorporated into the fuzzy system built with MATLAB.

Table 6. Input-Output of Fuzzy Variables

Variable	Specification	Range
Distance to Front Vehicle	0 – 4 m	0 – 4000
Relative Speed	0 – 4 m/s	0 – 4000
Brake Suggestion	100% brake application max	0 – 100

Table 7. Fuzzy System Membership Function

Variable	Member-ship Function	Type	Parameters
Distance (to Front Vehicle)	Close	Trapezoidal	[0 0 500 1000]
	Fair	Triangular	[500 1500 2500]
	Far	Trapezoidal	[2000 2500 4000 4000]
Relative Speed	Slow	Trapezoidal	[0 0 500 1000]
	Medium	Triangular	[500 1500 2500]
	Fast	Trapezoidal	[2000 2500 4000 4000]
Brake Suggest	Safe	Triangular	[0 15 30]
	Light	Triangular	[20 35 50]
	Moderate	Triangular	[40 55 70]
	Significant	Triangular	[60 80 100]

Figure 6 illustrates the fuzzy system's input and output developed in MATLAB.

Third Step: Define Rule

The rules consist of a series of IF-THEN statements that establish a logical relationship between the inputs and the outputs of the fuzzy system. With two variables, each containing three membership functions, the total number of viable rules amounts to nine. The decision is determined based on the initial forecast after evaluating all possible rules. The system's Fuzzy Rules are illustrated in Figure 7.

Fourth Step: Fuzzy Inference Method (Engine)

The MATLAB Fuzzy Logic Designer offers two types of inference engines: the Sugeno and Mamdani types. The Mamdani model will be employed in the collision warning system due to its enhanced reliability in producing accurate outcomes. A previous study by Mateichyk et al. on the energy efficiency of vehicles has shown that the Mamdani-type fuzzy system yields better results compared to the Sugeno-type [23].

Fifth Step: Defuzzification

Defuzzification is the process of transforming fuzzy outputs into exact results. The collision warning system will utilize the centroid approach for defuzzification. As previously stated in the research, the centroid method is one of the most widely used methods in engineering applications, where membership values are treated as weights to produce a balanced and representative crisp output [24].

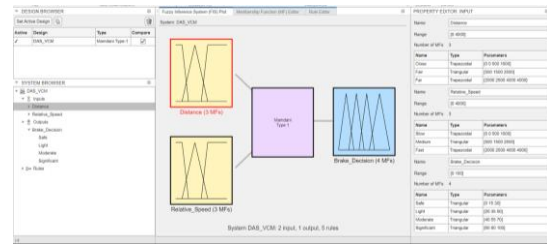


Figure 6. Fuzzy System Membership Function

Rule	Weight	Name
1 If Distance is Close and Relative_Speed is Slow then Brake is Moderate	1	rule1
2 If Distance is Fair and Relative_Speed is Slow then Brake is Light	1	rule2
3 If Distance is Far and Relative_Speed is Slow then Brake is Safe	1	rule3
4 If Distance is Close and Relative_Speed is Medium then Brake is Significant	1	rule4
5 If Distance is Fair and Relative_Speed is Medium then Brake is Moderate	1	rule5
6 If Distance is Far and Relative_Speed is Medium then Brake is Light	1	rule6
7 If Distance is Close and Relative_Speed is Fast then Brake is Significant	1	rule7
8 If Distance is Fair and Relative_Speed is Fast then Brake is Significant	1	rule8
9 If Distance is Far and Relative_Speed is Fast then Brake is Moderate	1	rule9

Figure 7. Fuzzy System Rules Mapping

In the centroid method, the fuzzified value, $d_{CA}(C)$, is defined as the value within the range of variable z for which the area under the graph of the membership function C is divided into two equal subareas. For the discrete case, in which C is defined as a finite universal set $\{z_1, z_2, \dots, z_n\}$, the formula is presented in (3) [25].

$$d_{CA}(C) = \frac{\sum_{k=1}^n c_{(x_k)} z_k}{\sum_{k=1}^n c_{(z_k)}} \quad (3)$$

Upon finalizing the construction of the fuzzy system from the first to the fifth step, attention now turns to the final phase of testing and optimization.

First Final Phase: Implementation and Testing

The first final step of implementation and testing demonstrates the system's operational use, which includes the development of a graphical user interface (GUI) using MATLAB's rule inference capabilities.

In the MATLAB GUI, the crisp input for distance is defined as 2100, corresponding to 2100 cm, and the relative speed is defined as 700, equating to 700 cm/s. The output from the fuzzy system is 35, indicating a requirement of 35% brake application. To determine the system's functionality, a manual calculation of the fuzzy decision must be performed, as in Figure 8.

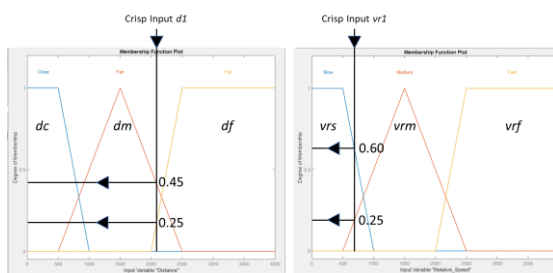


Figure 8. Manual Fuzzification

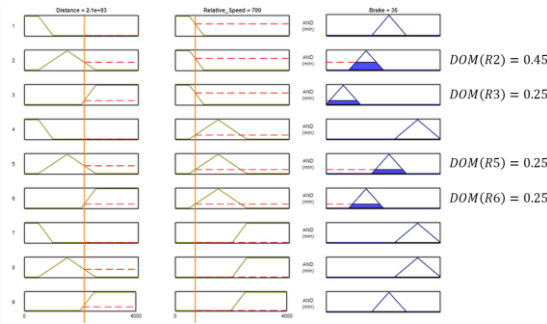


Figure 9. Rule Evaluation

a. Fuzzification

Utilize the crisp inputs $d1$ and $vr1$ to calculate their respective degrees of membership (DOM) within the relevant fuzzy sets.

Crisp input: $d1 = 2100$; $vr1 = 700$

Membership function: Close (dc); Fair/Moderate (dm); Far (df); Slow (vrs); Medium (vrn); Fast (vrf)

Figure 9 shows the estimate of the Degree of Membership (DOM), with the results outlined below:

$$\begin{aligned} \text{DOM}_{x=\text{dm}} &= 0.45 & \text{DOM}_{x=\text{df}} &= 0.25 \\ \text{DOM}_{y=\text{vrs}} &= 0.60 & \text{DOM}_{y=\text{vrn}} &= 0.25 \end{aligned}$$

b. Rule Evaluation

This phase aims to validate the intersection of the inputs within the corresponding rules. The fuzzy operator employed to obtain the single DOM representation of the rule is the AND fuzzy operation, signifying the intersection of fuzzy sets. The procedure is delineated by the formula shown in (4).

$$\text{DOM}_{\text{AUB}}(x) = \min[\text{DOM}_A(x), \text{DOM}_B(x)] \quad (4)$$

Figure 10 shows the specifics captured by the Rule Inference tab in MATLAB's Fuzzy Logic Designer. The result of the DOM is determined in the following calculation.

$$\begin{aligned} \text{DOM}_{\text{dmUvrs}}(\text{Rules2}) &= \min[0.45, 0.60] = 0.45 \\ \text{DOM}_{\text{dfUvrs}}(\text{Rules3}) &= \min[0.25, 0.60] = 0.25 \\ \text{DOM}_{\text{dmUvrn}}(\text{Rules5}) &= \min[0.45, 0.25] = 0.25 \\ \text{DOM}_{\text{dfUvrn}}(\text{Rules6}) &= \min[0.25, 0.25] = 0.25 \end{aligned}$$

c. Rule Outputs Aggregation

Throughout the aggregation process, the results of all rules will be combined. The method will combine the membership functions of any rule consequents that have been previously modified, generating the unified fuzzy set $\sum \text{DOM}$. Figure 10 illustrates the outcome of the fuzzy set.

According to the aggregate outcome, only three Membership Functions (MF) cross within the fuzzy set. These entities are categorized as Safe, Light, and Moderate. Calculating the parameter's

mean determines the range of each variable for the three membership functions.

Brake Safe: $z_A = 15$

Brake Light: $z_B = 35$

Brake Moderate: $z_C = 55$

d. Defuzzification

The defuzzification process utilizes the aggregated fuzzy output set as input to generate a specific numerical output. The defuzzification phase of this calculation applies the centroid method, as outlined in Equation 3. $C(z_k)$ represents the degree contributions of each membership z_k ; hence, the crisp output is computed as outlined in the subsequent calculations.

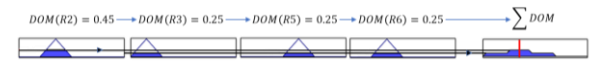


Figure 10. Aggregation of the Outputs

$$\text{DOM}(z_A) = \text{DOM}(\text{Rules3}) = 0.25$$

$$\text{DOM}(z_B) = \text{DOM}(\text{Rules2}) = 0.45$$

$$\text{DOM}(z_C) = \text{DOM}(\text{Rules5}) = 0.25$$

$$d_{CA}(C) = \frac{\text{DOM}(z_A)z_A + \text{DOM}(z_B)z_B + \text{DOM}(z_C)z_C}{\text{DOM}(z_1) + \text{DOM}(z_2) + \text{DOM}(z_3)}$$

$$d_{CA}(C) = 35$$

The result of the manually calculated fuzzy decisions has become $d_{CA}(C) = 35$. The manual calculation result remains consistent once the system has been finalized. This result indicates that the MATLAB system operates efficiently in alignment with basic concepts.

Second Final Step: Optimization and Fine-Tuning

The second final step in the fuzzy development process is to optimize and refine the fuzzy system. The technique offers extensive sampling evaluations across multiple scenarios to ensure the system operates as intended.

A prior study published in the IEEE journal suggests that the tuning process for Fuzzy Logic Controllers lacks a systematic methodology, necessitating a trial-and-error approach that involves modifying fuzzy rules and mapping membership functions until satisfactory outcomes are achieved.

This technique will be incorporated into the tuning method of this research. The fine-tuning process employs a trial-and-error approach to enhance the fuzzy logic rules and membership functions until acceptable outcomes are achieved.

a. Rules Optimization

This phase involves assessing a set of existing rules that specify the system's decision-making architecture. This stage primarily aims to

improve the "THEN" statement part of the rules, which define the brake decision output. The optimization requires recalibrating the rules to guarantee that outputs are adjusted to extreme values. Rules Mapping Optimization is shown in Figure 11. Figure 12 illustrates the results of rule optimization.

Three rules, specifically rule2, rule5, and rule9, had modifications as a result of fine-tuning. The rules' output has been adjusted to accommodate for extreme values. The aim of producing these extreme values in the rules' output is to assist with centroid method of defuzzification, which calculates the mean of the aggregated outputs, consequently creating an appropriate brake recommendation.

b. Membership Function Optimization

Following an adjustment of rules, work is directed towards the optimization of membership functions. Throughout the tuning implementation attempt, modifications should prioritize output adjustment. The accuracy of the system has limitations, as the centroid method for defuzzification limits the extraction of the extreme values (minimum and maximum) required for specific driving conditions.

To address the matter, the membership function for the braking decision has been adjusted by extending the range of these attributes further the original 0 to 100% range. The adjustments of the membership function for the braking choice are illustrated in Figure 12.

System: DAS_VCM_2

Add All Possible Rules Clear All Rules

Rule	Weight	Name
1 If Distance is Close and Relative_Speed is Slow then Brake is Moderate	1	rule1
2 If Distance is Fair and Relative_Speed is Slow then Brake is Safe	1	rule2
3 If Distance is Far and Relative_Speed is Slow then Brake is Safe	1	rule3
4 If Distance is Close and Relative_Speed is Medium then Brake is Significant	1	rule4
5 If Distance is Fair and Relative_Speed is Medium then Brake is Significant	1	rule5
6 If Distance is Far and Relative_Speed is Medium then Brake is Light	1	rule6
7 If Distance is Close and Relative_Speed is Fast then Brake is Significant	1	rule7
8 If Distance is Fair and Relative_Speed is Fast then Brake is Significant	1	rule8
9 If Distance is Far and Relative_Speed is Fast then Brake is Significant	1	rule9

Figure 11. Rules Mapping Optimization

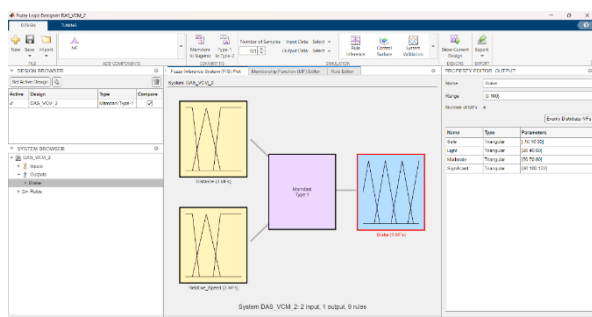


Figure 12. Membership Function Optimization

The enhanced adjustment simply modifies the minimum and maximum values, enabling the system to effectively gather data and respond to more extreme circumstances with a broader range of braking applications. This improvement ensures that the output can be converted into a more accurate braking action, thereby enhancing the overall efficiency and precision of the system in real-world driving situations.

Integrated System:

An integrated system of sensors and actuators with kinematics and perception algorithms has been constructed. The final phase of the integration build signifies the comprehensive integration of the system. The integration begins with translating the Fuzzy System developed in MATLAB into Arduino language, followed by integrating the code to make it work together.

Figure 13 illustrates the operational logic of a Driver Assistance Collision Warning System built in Arduino, which integrates kinematics and perception algorithms. The primary assessment is conducted by evaluating the time-to-collision (TTC). If the TTC is less than 2 seconds, the system will apply a kinematic algorithm to determine the result. When the TTC exceeds 2 seconds, the system applies fuzzy-perception-based logic to determine the proper output.

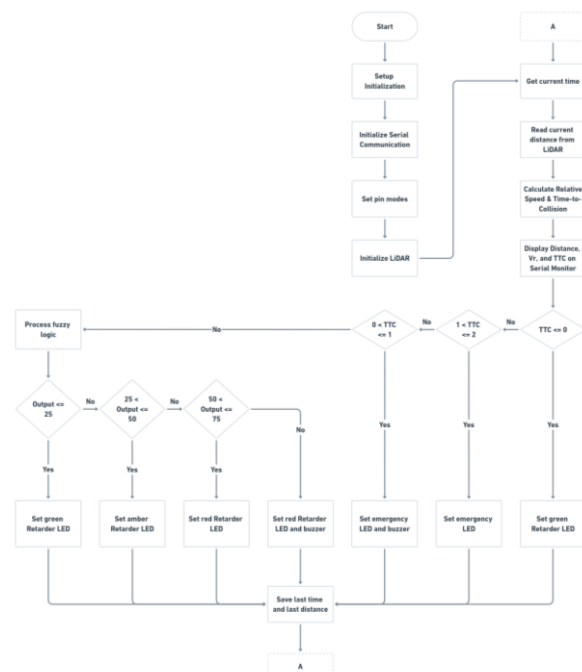


Figure 13. Kinematics and Perception Algorithm Integration System Flowchart

Integration Test

Subsequent to the integration build. The integration testing shall be performed to validate the seamless connection between the kinematics algorithm (time-to-collision threshold) and the perception algorithm (fuzzy logic). This phase ensures the absence of interference between the two algorithms.

Test Condition: The collision warning system is positioned atop the model remote-controlled car and oriented towards the solid barrier. The remote-controlled car nears the solid barrier, resulting in a change in the distance measurement.

Acceptance Criteria: The system's warning shall precisely represent the system's algorithm. The system's output warning must correspond with its logic while executing kinematics or perception decisions. The criteria table for the system is presented in Table 8.

The system demonstrated its capability for accurate decision-making in many scenarios, as indicated by the outcomes shown in Table 9. The system's integration test passed under the specified conditions. The system proved to make correct and precise choices during the testing scenarios, thereby confirming its adherence to the established acceptance requirements.

System Test

The objective of the system test phase is to validate the precision of distance information passing from the LIDAR to the Arduino, assess the accuracy of relative speed and time-to-collision computations, and confirm that the fuzzy logic system provides ideal warning responses.

Table 8. Integration Test Criteria

TTC (sec)	Algorithm Decision	Warning Output
0	Kinematics	Brake – Green
$0 < \text{TTC} < 1$	Kinematics	Retarder – Amber, Buzzer
$1 < \text{TTC} < 2$	Kinematics	Retarder – Amber
$\text{TTC} > 2$	Perception	Brake – Green/Amber/Red

Table 9. Integration Test Result

Test No.	TTC (cm/s)	Fuzzy Output	Decision Algorithm	Indicator	Judge
1	2.81	55.25	Perception	Brake – Red	Pass
2	0.44	-	Kinematics	Retarder, Buzzer	Pass
3	2.45	70.00	Perception	Brake – Red	Pass
4	2.53	62.51	Perception	Brake – Red	Pass

Acceptance Criteria: The system must accurately calculate relative speed and time-to-collision. Furthermore, the system's output must adhere to the system's logic, whether the decision is kinematics or perception algorithms.

Data Source: The data was obtained from the integration test, including time-to-collision results and other essential information.

Fuzzy Output Calculation: signifies utilizing the rule inference feature of the MATLAB Fuzzy Logic Designer by inputting the crisp values of distance and relative speed to obtain the brake decision output value, as shown in Figure 14.

Calculation: utilizing basic equations to confirm the system's computation result. The TTC can be calculated by applying (2).

a. Test-1

$$d_A = 455, d_B = 707$$

$$t_A = 06.649 \approx 7, t_B = 07.660 \approx 8$$

$$vr = (707-455)/(8-7) = 252 \text{ cm/s}$$

$$\text{TTC} = 707/252 = 2.81 \text{ s}$$

b. Test-2

$$d_A = 707, d_B = 215$$

$$t_A = 07.660 \approx 8, t_B = 08.689 \approx 9$$

$$vr = (215-707)/(9-8) = -492 \text{ cm/s}$$

$$\text{TTC} = |215/-492| = 0.44 \text{ s}$$

c. Test-3

$$d_A = 215, d_B = 363$$

$$t_A = 18.744 \approx 19, t_B = 19.757 \approx 20$$

$$vr = (363-215)/(20-19) = 148 \text{ cm/s}$$

$$\text{TTC} = 363/148 = 2.45 \text{ s}$$

d. Test-4

$$d_A = 363, d_B = 601$$

$$t_A = 19.757 \approx 20, t_B = 20.760 \approx 21$$

$$vr = (601-363)/(21-20) = 238 \text{ cm/s}$$

$$\text{TTC} = 601/238 = 2.53 \text{ s}$$

The accuracy of the decision-making algorithms in the system is validated by comparing the manual calculations, which employ basic equations, with the system's computation result.

Table 10 shows that there are no deviations in Time-to-Collision values between the system output and the manual calculation.



Figure 14. MATLAB Fuzzy Result Test No. 1, 3, 4

Table 10. System Test Result

Test No.	System Result		Validation		Judge
	TTC (cm/s)	Fuzzy Result	TTC Manual (cm/s)	MATLAB Output	
1	2.81	55.25	2.81	55.3	Pass
2	0.44	-	0.44	-	Pass
3	2.45	70.00	2.45	70	Pass
4	2.53	62.51	2.53	62.5	Pass

Furthermore, the fuzzy output yields the same results as the system output and MATLAB calculations, with the exception of test number 2, where the Time-to-Collision is less than 2 seconds, utilizing a kinematics approach and not requiring fuzzy output.

The successful test results serve as concrete proof of the system's capability to operate with accuracy as well as reliability in practical conditions, ensuring both safety and performance. The system test has been assessed as Pass.

Acceptance Test

An acceptance test verifies that the system meets end-user requirements and operates reliably across various environmental conditions. The collision warning system employs a LIDAR sensor as its primary component; hence, the acceptance test will concentrate on LIDAR performance. According to the research by Park J. et al. from Hyundai Motor Company, it contains eight distinct environmental factors that could be used to evaluate the performance of LIDAR: cover contamination, strong sunlight, high temperature, low temperature, vibration, interference, reflectivity of a target, and transitions between day and night [26].

To highlight the driver experience, the acceptance tests for the system will incorporate real-world driving scenarios with three tests performance evaluations. The initial assessment examines cover contamination, with fog, rain, and dust. The second test features strong sunlight, and the third test incorporates the transition from day to night.

In the preliminary phase, conduct tests 1 through 3 to simulate cover contamination by fog, rain, and dust. The initial test approximated cover contamination by generating foggy situations. The container was filled with artificial smoke to generate a dense fog, significantly reducing visibility and simulating actual fog conditions. The second test examines contamination caused by rain.

Table 11. Acceptance Test Result

Test No.	Req. & Test	Spec. Std. (cm)	Deviation Result (cm)	Remark	Judge
1	Fog	± 5	± 2	No intervention	Pass
2	Rain	± 5	± 1	No intervention	Pass
3	Dust	± 5	± 5	No intervention	Pass
4	Strong Sunlight	± 5	± 2	No intervention	Pass
5	Day-night transition	± 5	± 3	No intervention	Pass

A manual droplet generation equipment was constructed to simulate the process of rainfall by producing artificial raindrops. The third test concentrated on dust pollution. In this scenario, the sensor was obscured by an acrylic cover that had been previously coated with powder to simulate a dusty environment. The fourth test simulated prompt day-night transitions. The lighting in the chamber executed two prompt transitions, alternating between brightness and darkness. This simulation aims to replicate the illumination change during the transition from day to night or vice versa. At last, the fifth test subjected the sensor to strong sunlight conditions. A flashlight was employed in the same chamber to illuminate the sensor from multiple angles, simulating the effects of direct sunlight and glare.

The test results in Table 11 indicate that the sensor-read value deviation is within ± 5 cm, signifying that the deviation remains within the LIDAR sensor standard. In conclusion, the sensor successfully passed the acceptance test performed under simulated real-world conditions, indicating excellent reliability and performance across all assessed scenarios, hence meeting the acceptance criteria.

Discussion

This research effectively met its goals of enhancing vehicle safety through the establishment of a universally applicable solution. The main results of the research into the development of a driver assistance collision warning system are summarized in the subsequent items:

The driver assistance collision warning mechanism, utilising the Arduino Mega 2560 as its microcontroller, operates on a 5 VDC power supply, ensuring system interoperability and adaptability across various vehicle brands and categories. Its power can be sourced through a

vehicle's electrical port, a USB charging interface, or an auxiliary power source such as a portable battery pack. During the calibration phase, the LIDAR sensor gives highly accurate distance measurements with a margin of error of ± 5 cm, while the Arduino Microcontroller precisely executes computational processes.

The integration of the LIDAR sensor's precise distance measurement with the Arduino Mega 2560's calculations for time-to-collision estimation, relative speed, and fuzzy logic output facilitates reliable data for kinematics and perception algorithms. The kinematics and perception algorithms were effectively harmonized, as verified through comprehensive integration testing, which confirmed their seamless collaboration. The successful completion of these evaluations underscores the system's dependability and the strong cohesion of its fundamental components.

The collision warning mechanism demonstrates its precision through systematic validation, wherein its outputs are benchmarked against manual computations, ensuring consistent and reliable decision-making for both kinematic modeling and perceptual algorithms. The system's accuracy was rigorously evaluated in simulated actual environments using the acceptance test. The sensors precisely identified objects under various circumstances, including fog, rain, dust, and contrasting brightness levels during both day and night, as well as bright sunlight. This exceptional performance across varied settings highlights the system's resistance and reliability. The system's primary objective is to provide users with early warnings to avoid collisions between vehicles. By precisely identifying potential collisions in advance, the system allows drivers to implement crucial precautions to prevent accidents, hence improving overall road safety.

CONCLUSION

In conclusion, the system demonstrates a significant enhancement in the technology used to improve vehicle safety. With its broad compatibility across different vehicle brands and segments, its seamless integration of decision-making algorithms, and its high-performance accuracy in real-world applications, it serves as an essential instrument for enhancing vehicle safety. By integrating kinematics and perception, this study overcomes the limitations of previous research, achieving real-time accuracy while reducing reliance on pre-trained models. This research achieves its primary objectives and significantly

advances the broader goal of enhancing road safety for all vehicle drivers.

ACKNOWLEDGMENT

I am deeply thankful to the Swiss German University for the curriculum and instruction program, which has offered me a stimulating and enriching academic experience. My gratitude extends to Polytron for providing me with the practical insights and professional environment that have enriched my research experience.

REFERENCES

- [1] Y. Lobanova and S. Evtiukov, "Role and methods of accident ability diagnosis in ensuring traffic safety," *Transportation Research Procedia*, vol. 50, pp. 363–372, 2020, doi: 10.1016/j.trpro.2020.10.043.
- [2] W. I. Susanto, H. Nasution, and T. D. Sofianti, "Driver Assistance Collision Warning System Using LIDAR Sensor with Kinematic and Perception Algorithms," *Thesis*, Swiss German University, 2024.
- [3] I. Isaksson Hellman and M. Lindman, "Estimating the crash reducing effect of Advanced Driver Assistance Systems (ADAS) for vulnerable road users," *Traffic Safety Research*, vol. 4, p. 000036, Nov. 2023, doi: 10.55329/blzz2682.
- [4] H. Kunto, H. Nasution, and D. Hendriana, "Design and Develop Vehicle Anti-Collision Warning System," *Conference on Management and Engineering in Industry (CMEI)*, vol. 4, no. 1, pp. 19–24, Aug. 2022.
- [5] Y. Yuan, Y. Lu, and Q. Wang, "Adaptive forward vehicle collision warning based on driving behavior," *Neurocomputing*, vol. 408, pp. 64–71, Sep. 2020, doi: 10.1016/j.neucom.2019.11.024.
- [6] G. Šimić, "Can AI be conscious," *Abstract book of A4L Symposium in Zagreb 2024*, 2024.
- [7] L. Masello, G. Castignani, B. Sheehan, F. Murphy, and K. McDonnell, "On the road safety benefits of advanced driver assistance systems in different driving contexts," *Transp Res Interdiscip Perspect*, vol. 15, p. 100670, Sep. 2022, doi: 10.1016/j.trip.2022.100670.
- [8] C. Salazar Dorn and C. L. Knüvener, "Building the Bridge Between Automotive SPICE® and Agile Development," *Systems, Software and Services Process Improvement*, 2021, pp. 391–405. doi: 10.1007/978-3-030-85521-5_26.
- [9] L. J. Moukahal, M. A. Elsayed, and M. Zulkernine, "Vehicle Software Engineering

- (VSE): Research and Practice," *IEEE Internet Things Journal*, vol. 7, no. 10, pp. 10137–10149, Oct. 2020, doi: 10.1109/JIOT.2020.3001026.
- [10] S. Das and A. K. Maurya, "Defining Time-to-Collision Thresholds by the Type of Lead Vehicle in Non-Lane-Based Traffic Environments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 12, pp. 4972–4982, Dec. 2020, doi: 10.1109/TITS.2019.2946001.
- [11] J. Cordes, M. Chraïbi, A. Tordeux, and A. Schadschneider, "Time-To-Collision Models for Single-File Pedestrian Motion," *Collect Dyn*, vol. 6, p. A133, Jan. 2022, doi: 10.17815/CD.2021.133.
- [12] United Nations. Economic Commission for Europe, World Forum for Harmonization of Vehicle Regulations (WP. 29) : how it works, how to join it., 4th ed. 2022.
- [13] United Nations. Economic Commission for Europe, Uniform provisions concerning the approval of audible warning devices and of motor vehicles with regard to their audible signals. 2020. Accessed: Jul. 17, 2024. [Online]. Available: <https://unece.org/transport/documents/2021/05/standards/un-regulation-no-28-amend6>
- [14] United Nations. Economic Commission for Europe, Uniform provisions concerning the approval of vehicles with regard to the location and identification of hand controls, tell-tales and indicators. 2023. Accessed: Jul. 17, 2024. [Online]. Available: <https://unece.org/transport/documents/2023/12/standards/addendum-120-un-regulation-no-121-revision-2-amendment-6>
- [15] J. F. McDonald, "Following Rules and Traffic Flow: What If Everyone Used the Three-Second Rule?," *SSRN Electronic Journal*, 2023, doi: 10.2139/ssrn.4540900.
- [16] K. YOSHIOKA, "A Tutorial and Review of Automobile Direct ToF LiDAR SoCs: Evolution of Next-Generation LiDARs," *IEICE Transactions on Electronics*, vol. E105.C, no. 10, p. 2021CTI0002, Oct. 2022, doi: 10.1587/transele.2021CTI0002.
- [17] Z. Dai, A. Wolf, P.-P. Ley, T. Glück, M. C. Sundermeier, and R. Lachmayer, "Requirements for Automotive LiDAR Systems," *Sensors*, vol. 22, no. 19, p. 7532, Oct. 2022, doi: 10.3390/s22197532.
- [18] H. K. Kondaveeti, N. K. Kumaravelu, S. D. Vanambathina, S. E. Mathe, and S. Vappangi, "A systematic literature review on prototyping with Arduino: Applications, challenges, advantages, and limitations," *Comput Sci Rev*, vol. 40, p. 100364, May 2021, doi: 10.1016/j.cosrev.2021.100364.
- [19] M. Markofsky, M. Schäfer, and D. Schramm, "Use Cases and Methods of Virtual ADAS/ADS Calibration in Simulation," *Vehicles*, vol. 5, no. 3, pp. 802–829, Jul. 2023, doi: 10.3390/vehicles5030044.
- [20] A. Goeritno and I. Setyawibawa, "An Electronic Device Reviewed by Diagnosing on the Modules Embodiment," *International Journal of Electronics and Communications Systems*, vol. 1, no. 2, pp. 41–55, Dec. 2021, doi: 10.24042/ijecs.v1i2.10383.
- [21] D. Romahadi, D. Feriyanto, F. Anggara, F. P. Wijaya, and W. Dong, "Intelligent system design for identification of unbalance and misalignment using Fuzzy Logic methods," *SINERGI*, vol. 28, no. 2, p. 241, Apr. 2024, doi: 10.22441/sinergi.2024.2.004.
- [22] H. Nasution, H. Jamaluddin, and J. Mohd. Syeriff, "Energy Analysis for Air Conditioning System Using Fuzzy Logic Controller," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 9, no. 1, p. 139, Apr. 2011, doi: 10.12928/telkomnika.v9i1.680.
- [23] V. Mateichyk, N. Kostian, M. Smieszek, J. Mosciszewski, and L. Tarandushka, "Evaluating Vehicle Energy Efficiency in Urban Transport Systems Based on Fuzzy Logic Models," *Energies (Basel)*, vol. 16, no. 2, p. 734, Jan. 2023, doi: 10.3390/en16020734.
- [24] J. J. Cardiel-Ortega and R. Baeza-Serrato, "Failure Mode and Effect Analysis with a Fuzzy Logic Approach," *Systems*, vol. 11, no. 7, p. 348, Jul. 2023, doi: 10.3390/systems11070348.
- [25] G. J. Klir and B. Yuan, "Fuzzy sets and fuzzy logic - theory and applications," Prentice Hall, 1995. [Online]. Available: <https://api.semanticscholar.org/CorpusID:46622061>
- [26] J. Park, J. Cho, S. Lee, S. Bak, and Y. Kim, "An Automotive LiDAR Performance Test Method in Dynamic Driving Conditions," *Sensors*, vol. 23, no. 8, p. 3892, Apr. 2023, doi: 10.3390/s23083892.