

CHAPTER 5 – CONCLUSIONS AND RECCOMENDATIONS

5.1 Conclusions

This final section presents an overview of the primary results from the research on the development of a driver assistance collision warning system. The study successfully achieved its objectives of improving vehicle safety through the implementation of a solution that can be universally applicable. This was achieved through comprehensive design, testing, and validation procedures. The following items summarize the main results of the study:

- The driver assistance collision warning system, built using the Arduino Mega 2560 as the main controller, operates on 5 VDC power supplied via a USB-A connection. This design ensures the system's universal applicability across various vehicle brands and segments, as it can be powered through a vehicle's power outlet, USB charging port, or an external source like a power bank.
- Throughout the calibration phase, the LIDAR sensor provides accurate distance measurements within the tolerance of ± 5 cm, while the Arduino Mega 2560 performs computations without any errors. The combination of the LIDAR sensor's accurate distance readings and the Arduino Mega 2560's ability to compute relative speed, time-to-collision, and fuzzy output will offer precise data for kinematics and perception algorithms.
- The kinematics and perception algorithms were synchronized effectively. This was validated through integration testing, which confirmed that both algorithms work together seamlessly. The successful passing of these tests demonstrates the system's reliability and the robust integration of its core components.
- The collision warning system proves its accuracy through system tests that compare its results with manual calculations, providing reliable decision outputs for both the kinematics and perception algorithms.

- The system's accuracy was thoroughly tested in real-world environment simulations through the acceptance test. The sensors accurately detected objects in various conditions, including fog, rain, dust, varying brightness levels during day and night, and even under direct sunlight. This high level of performance under diverse conditions underscores the system's robustness and reliability.
- One of the primary objectives of the system is to provide users with early warnings to prevent road accidents. By accurately detecting potential collisions well in advance, the system enables drivers to take necessary actions to avoid accidents, thereby enhancing overall road safety.

In conclusion, the driver assistance collision warning system that was developed indicates a significant improvement in the technology that is used to make vehicle safer. As a result of its universal applicability across all vehicle brands and segments, its well-constructed integration of decision-making algorithms, and its excellent accuracy in real-world environments, it is a precious instrument for improving vehicle safety and reducing accidents. This research not only meets its initial objectives but also contributes to the broader goal of improving road safety for all drivers.

5.2 Recommendations

The first recommendations focuses on the current system evaluation. The current driver assistance collision warning system utilizes the Garmin LIDAR Lite v3, which has considerable limits in its performance due to its restricted operational capability at specific angles. In particular, the LIDAR sensor faces difficulties in identifying objects at larger angles, which might result in blind spots and decreased dependability in comprehensive collision avoidance situations. In order to overcome this limitation, it is recommended to improve the system by including advanced sensors that have greater angular resolution and a wider field-of-view. Utilizing sensors with a 360-degree field of view and enhanced precision in detecting objects from different perspectives will greatly enhance the dependability of collision detection systems, resulting in improved performance and enhanced safety for users.

The second point concerns to the potential integration of the driver assistance collision warning system. This system now utilizes a fuzzy logic model to provide braking decision recommendations based on the distance to the next vehicle and the relative speed between vehicles. The possible integration using electric vehicles (EVs) into the target application can be addressed. Integrating this technology has the potential to significantly enhance electric vehicles by optimizing regenerative braking, which is the act of recovering energy during light to moderate braking and storing it in the battery. This would result in improved energy efficiency.

Nevertheless, the successful integration of this technology in electric vehicles requires further study, specifically in comprehending the manner in which to categorize braking situations to optimize the advantages of regenerative braking. In order to enable smooth transitions between regenerative and conventional braking, the system must be able to precisely identify and categorize the severity of braking in real-time. However, it is important that the final choice on brakes is given as a safety warning. This research is essential for improving both safety and energy efficiency.

In its conclusion, this thesis recommends initiating further studies and improvement in the employing advance sensor and integration of collision warning systems with regenerative braking in electric vehicles (EVs). This integration serves as a concurrent solution, both enhancing road safety and improving energy efficiency. This research has the potential to make considerable progress in intelligent electric mobility by improving energy efficiency and ensuring high safety standards. It could lead to important breakthroughs in smart and eco-friendly transportation technology.

GLOSSARY

Term	Definition
ADAS (Advanced Driver-Assistance Systems)	Intelligent automotive systems that facilitate parking and driving, thereby enhancing safety and convenience.
Artificial Intelligence (AI)	The technology that enables computers to study and think like humans
Head-Up Display (HUD)	A display found in automobiles and aircraft that presents critical data directly to the aviator or driver.
I2C (Inter-Integrated Circuit)	A method by which electronic components within devices can communicate with one another via two connections.
Kemenhub RI (Kementerian Perhubungan Republik Indonesia)	The Indonesian government agency tasked with transportation.
LED (Light Emitting Diode)	A dim, brilliant light found in a variety of electronic devices.
LIDAR (Light Detection and Ranging)	A technology that calculates distance by analyzing the light reflected from an object after striking it with a laser.
OBD-II (On-Board Diagnostics II)	A vehicle system that monitors the vehicle's performance and alerts the driver of any issues.
POLRI (Kepolisian Negara Republik Indonesia)	The Indonesian national security force.
PWM (Pulse Width Modulation)	A method for regulating the power supplied to a device by rapidly cycling it on and off.

SLR (Systematic Literature Review)	A methodical and comprehensive approach to gathering and evaluating all existing studies pertaining to a particular subject matter.
SPI (Serial Peripheral Interface)	A method for electronic devices to rapidly exchange information over small distances.
UART (Universal Asynchronous Receiver/Transmitter)	A component of hardware and software that facilitates communication between devices and computers by converting data between different formats.
UN ECE (United Nations Economic Commission for Europe)	An agency of the United Nations whose mission was to enhance economic relations and standards among European nations.
TTC (Time-to-Collision)	A measurement used by collision warning systems to calculate the duration remained until a vehicle could possibly collide into an object
GUI (Graphical User Interface)	A visual interface that enables users to interact with electronic devices using graphical icons and visual indicators, rather than relying on text-based commands.
FIS (Fuzzy Inference System)	A computer framework that utilizes fuzzy logic to perform tasks such as logic, decision-making, and processing information that is not clear or imprecise.
Centroid (Center of Gravity)	Defuzzification calculation method determined by calculating the weighted average of all potential values, which involves converting fuzzy data into a single, precise output.
BEVs (Battery Electric Vehicles)	Are vehicles that rely completely or partially on electric power stored in batteries.

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] Cun Cahya, "Kakorlantas Sebut dalam Satu Jam Ada 3 Orang Meninggal Dunia Akibat Kecelakaan Lalu Lintas," PT Media Merdeka Network. Accessed: Jul. 08, 2024. [Online]. Available: <https://www.suaramerdeka.com/jawa-tengah/0412034706/kakorlantas-sebut-dalam-satu-jam-ada-3-orang-meninggal-dunia-akibat-kecelakaan-lalu-lintas>
- [3] Komite Nasional Keselamatan Transportasi, *Buku Statistik Investigasi Kecelakaan Transportasi*. 2022.
- [4] Statista Research Department, "Number of road traffic accidents Indonesia 2017-2022," Statista. Accessed: Feb. 03, 2024. [Online]. Available: <https://www.statista.com/statistics/979071/indonesia-number-road-traffic-accidents/>
- [5] Kantor Kepolisian Republik Indonesia, "Jumlah Kecelakaan, Korban Mati, Luka Berat, Luka Ringan, dan Kerugian Materi 2019-2021," Badan Pusat Statistik: Transportasi. Accessed: Feb. 03, 2024. [Online]. Available: <https://www.bps.go.id/id/statistics-table/2/NTEzIzI=/jumlah-kecelakaan--korban-mati--luka-berat--luka-ringan--dan-kerugian-materi.html>
- [6] Badan Pengatur Jalan Tol, "Jumlah Kecelakaan Lalu Lintas Meningkat Jadi 103.645 pada 2021," Data Indonesia. Accessed: Feb. 03, 2024. [Online]. Available: <https://dataindonesia.id/otomotif-transportasi/detail/jumlah-kecelakaan-lalu-lintas-meningkat-jadi-103645-pada-2021>
- [7] 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.
- [8] R. O. Salawu, A.-O. S. Bolatitio, and S. Masibo, "Theoretical and Conceptual Frameworks in Research: Conceptual Clarification," *European Chemical*

- Bulletin*, vol. 12, no. 12, pp. 2103–2117, 2023, doi: 10.48047/ecb/2023.12.12.139.
- [9] New Car Assessment Program for Southeast Asian Countries (ASEAN NCAP), *ASEAN NCAP ROADMAP 2026-2030*. Selangor: MIROS, 2023. Accessed: Aug. 14, 2024. [Online]. Available: <https://aseancap.org/v4/wp-content/uploads/2023/03/ASEAN-NCAP-Roadmap-2026-2030-V1.0-16-March-2023.pdf>
- [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] Badan Pengatur Jalan Tol, “Mengenal Istilah ‘Rumus Tiga Detik’ Upaya Meminimalisir Kecelakaan Beruntun Di Jalan Tol,” Badan Pengatur Jalan Tol. Accessed: Feb. 03, 2024. [Online]. Available: <https://bpjt.pu.go.id/berita/mengenal-istilah-rumus-tiga-detik-upaya-meminimalisir-kecelakaan-beruntun-di-jalan-tol>
- [12] 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.
- [13] National Highways, “Press Release: National Highways urges drivers to use the two-second rule in new campaign,” UK Government. Accessed: Feb. 03, 2024. [Online]. Available: <https://www.gov.uk/government/news/national-highways-urges-drivers-to-use-the-two-second-rule-in-new-campaign>
- [14] Transport Department Government of West Bengal and Indian Institute of Technology Kharagpur, *Training Manual for Drivers*. 2017.
- [15] R. Sharda, D. Delen, and E. Turban, *Business intelligence and analytics : systems for decision support*, 10th ed. Pearson Education, 2014.
- [16] A. M. Nascimento *et al.*, “A Systematic Literature Review About the Impact of Artificial Intelligence on Autonomous Vehicle Safety,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 12, pp. 4928–4946, Dec. 2020, doi: 10.1109/TITS.2019.2949915.

- [17] 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.
- [18] P. Prakash, P. Kalyan, M. D. Kumar A, P. V Kalyan, J. Diwakar Reddy, and P. G. S, "Vehicle Collision Detection Using IoT," *International Research Journal of Modernization in Engineering Technology and Science*, vol. 2, no. 5, May 2020, [Online]. Available: www.irjmets.com
- [19] D. Katare and M. El-Sharkawy, "Embedded System Enabled Vehicle Collision Detection: An ANN Classifier," in *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, IEEE, Jan. 2019, pp. 0284–0289. doi: 10.1109/CCWC.2019.8666562.
- [20] N. Kumar, D. Acharya, and D. Lohani, "An IoT-Based Vehicle Accident Detection and Classification System Using Sensor Fusion," *IEEE Internet Things J*, vol. 8, no. 2, pp. 869–880, Jan. 2021, doi: 10.1109/JIOT.2020.3008896.
- [21] 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.
- [22] L. J. Moukahal, M. A. Elsayed, and M. Zulkernine, "Vehicle Software Engineering (VSE): Research and Practice," *IEEE Internet Things J*, vol. 7, no. 10, pp. 10137–10149, Oct. 2020, doi: 10.1109/JIOT.2020.3001026.
- [23] 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.
- [24] 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>
- [25] 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>
- [26] Arduino SRL, “Arduino® MEGA 2560 Rev3,” Monza MB, Sep. 2020.
- [27] 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.
- [28] Garmin Ltd., “Lidar Lite v3 Operation Manual and Technical Specifications,” Sep. 2016.
- [29] Pro-signal, “Buzzer Datasheet,” 2016. Accessed: May 14, 2024. [Online]. Available: www.farnell.com
- [30] 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.
- [31] 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.
- [32] 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.
- [33] F. DERNONCOURT, “Introduction to fuzzy logic,” Jan. 2013.
- [34] 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.
- [35] S. Rane, L. K. Ragha, and D. Kurule, “Performance of Mamdani Fuzzy Inference System for Tracking Multiple Targets Using Autopilot System,” *International Journal of Electrical and Electronics Research*, vol. 5, pp. 34–42, 2017, [Online]. Available: www.researchpublish.com

- [36] D. R. Solanke, K. D. Chinchkhede, and A. B. Manwar, "Design & Implementation of Fuzzy Inference System For Automatic Braking System," *International Journal of Advance Research in Science and Engineering*, vol. 6, no. 9, 2017.
- [37] 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>
- [38] M. S. bin Mohd Aras, S. M. S. b Syed Abdul Hamid, F. binti Abdul Azis, F. A. binti Ali, and S. S. b Abdullah, "Study of the effect in the output membership function when tuning a Fuzzy Logic Controller," in *2011 IEEE International Conference on Control System, Computing and Engineering*, 2011, pp. 1–6. doi: 10.1109/ICCSCE.2011.6190485.
- [39] 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.

CURRICULUM VITAE

Name : Willi Immanuel Susanto

Phone Number : (+62) 85641446000

Email : willisusanto08@gmail.com

Address : Perumahan Kota Baru No. 52, Kel.
Blotongan, Kec. Sidorejo, Kota Salatiga



Working Experience:

2023 – Now : R&D Regulation and Certification Engineer – PT. Hartono Istana Teknologi (Polytron)

2022 – 2023 : Part Development Specialist – PT. Hyundai Motor Manufacturing Indonesia

2019 – 2022 : Purchasing Strategy Staff – PT. Honda Prospect Motor

2017 : IT Support Intern – PT. Formulatrix Indonesia

Education:

2018 : International Intern Research Program – National Chung Cheng University, Taiwan

2014 – 2018 : Bachelor of Electrical Engineering – Jenderal Soedirman University

6. Appendix 1: Actuator Calibration Arduino Code

```
/*-----  
Connections:  
pin 7 as a Buzzer  
pin 8 as an Emergency Brake (Amber) - Wire White then Purple  
pin 9 as an Retarder (Green) - Wire Blue then Brown  
pin 10 as an Retarder (Amber) - Wire Amber then Black  
pin 11 as an Retarder (Red) - Wire Red then White  
GND - Wire Gray then Gray  
-----*/  
  
void setup() {  
    pinMode(7, OUTPUT);  
    pinMode(8, OUTPUT);  
    pinMode(9, OUTPUT);  
    pinMode(10, OUTPUT);  
    pinMode(11, OUTPUT);  
}  
  
void loop() {  
    digitalWrite(7, HIGH);  
    delay(1000);  
    digitalWrite(7, LOW);  
    digitalWrite(8, HIGH);  
    delay(1000);  
    digitalWrite(8, LOW);  
    digitalWrite(9, HIGH);  
    delay(1000);  
    digitalWrite(9, LOW);  
    digitalWrite(10, HIGH);  
    delay(1000);  
    digitalWrite(10, LOW);  
    digitalWrite(11, HIGH);  
    delay(1000);  
    digitalWrite(11, LOW);  
    delay(3000);  
}
```

7. Appendix 2: LIDAR Sensor Calibration Arduino Code

```
/*-----  
Connections:  
LIDAR-Lite 5 Vdc (red) to Arduino 5v  
LIDAR-Lite I2C SCL (green) to Arduino SCL  
LIDAR-Lite I2C SDA (blue) to Arduino SDA  
LIDAR-Lite Ground (black) to Arduino GND  
-----*/  
  
#include <Wire.h>  
#include <LIDARLite.h>  
  
LIDARLite myLidarLite;  
  
void setup()  
{  
  Serial.begin(9600);  
  myLidarLite.begin(0, true);  
  myLidarLite.configure(0);  
}  
  
void loop()  
{  
  Serial.println(myLidarLite.distance());  
  for(int i = 0; i < 50; i++)  
  {  
    Serial.print(myLidarLite.distance(false));  
    Serial.println(" cm");  
    delay(1000);  
  }  
}
```

8. Appendix 3: Driver Assistance Collision Warning System Arduino Code

`fis_header.h`

```
#define FIS_TYPE float
#define FIS_RESOLUTION 101
#define FIS_MIN -3.4028235E+38
#define FIS_MAX 3.4028235E+38
typedef FIS_TYPE(* FIS_MF)(FIS_TYPE, FIS_TYPE*);
typedef FIS_TYPE(* FIS_ARR_OP)(FIS_TYPE, FIS_TYPE);
typedef FIS_TYPE(* FIS_ARR)(FIS_TYPE*, int, FIS_ARR_OP);
```

`Driver_Assistance_Collision_Warning_System.ino`

```
#include <Wire.h>
#include <math.h>
#include <LIDARLite.h>
#include "fis_header.h"

LIDARLite myLidarLite;
const int buzzer = 7;
const int ledEmerg = 8;
const int ledRetardGreen = 9; // Green LED for FIS Light Brake Decision
const int ledRetardAmber = 10; // Amber LED for FIS Moderate Brake Decision
const int ledRetardRed = 11; // Red LED for FIS Significant Brake Decision

const int fis_gcI = 2; // Number of inputs to the fuzzy inference system
const int fis_gcO = 1; // Number of outputs to the fuzzy inference system
const int fis_gcR = 9; // Number of rules to the fuzzy inference system

FIS_TYPE g_fisInput[fis_gcI];
FIS_TYPE g_fisOutput[fis_gcO];

void setup()
{
  Serial.begin(9600);
  pinMode(buzzer, OUTPUT);
  pinMode(ledEmerg, OUTPUT);
  pinMode(ledRetardGreen, OUTPUT);
  pinMode(ledRetardAmber, OUTPUT);
  pinMode(ledRetardRed, OUTPUT);
  myLidarLite.begin(0, true); // initialize LiDAR Lite v3
  myLidarLite.configure(0); // default configuration
```

```
}

void loop()
{
    static unsigned long lastTime = 0;
    unsigned long currentTime;
    int currentDistance;
    float vr, ttc;
    static int lastDistance = readLiDAR(); //read current distance
    currentTime = millis(); // get current time in milliseconds

    if (currentTime - lastTime >= 1000)
    {
        // Calculate Vr (relative speed) in meters/second
        currentDistance = readLiDAR();
        int toleranceDistance = fabs(currentDistance - lastDistance);
        if (toleranceDistance > 5) // Check both distances are greater than zero
        {
            vr = (float)(currentDistance - lastDistance) / ((currentTime - lastTime) / 1000.0);
        }
        else
        {
            vr = 0.0f; // Set Vr to 0 if one of the distances is not detected
        }

        // Calculate TTC (time to collision) in seconds
        if (vr != 0)
        {
            ttc = fabs((float)currentDistance / vr);
        }
        else
        {
            ttc = 0; // Indefinite time (no movement)
        }

        // Resetting all outputs
        digitalWrite(ledEmerg, LOW);
        digitalWrite(buzzer, LOW);
        digitalWrite(ledRetardGreen, LOW);
        digitalWrite(ledRetardAmber, LOW);
        digitalWrite(ledRetardRed, LOW);

        // Display Distance, Vr, and TTC on Serial Monitor
        Serial.print("\nDistance: ");
        Serial.print(currentDistance);
        Serial.print(" cm, Vr: ");
        Serial.print(vr);
        Serial.print(" cm/s, TTC: ");
```

```
Serial.print(ttc);
Serial.print(" s,");

// Kinematics Decision-making
if (ttc <= 0)
{
    // For TTC <= 0 or error, indicate safe state with green LED
    digitalWrite(ledRetardGreen, HIGH);
}
else if (ttc > 1 && ttc <= 2)
{
    digitalWrite(ledEmerg, HIGH); // Turn on LED
    digitalWrite(buzzer, LOW); // Turn off buzzer
}
else if (ttc > 0 && ttc <= 1)
{
    digitalWrite(ledEmerg, HIGH); // Turn on LED
    digitalWrite(buzzer, HIGH); // Turn on buzzer
}

// Perception Decision-making
else if (ttc > 2)
{
    // Fuzzy Logic Input
    g_fisInput[0] = (FIS_TYPE)currentDistance;
    g_fisInput[1] = (FIS_TYPE)vr;
    fuzzy_process();

    Serial.print(" Output: ");
    Serial.print(g_fisOutput[0]);

    // Decision Output for FIS
    digitalWrite(ledRetardRed, LOW);
    digitalWrite(ledRetardAmber, LOW);
    digitalWrite(ledRetardGreen, LOW);

    if (g_fisOutput[0] > 75) // Significant
    {
        digitalWrite(ledRetardRed, HIGH);
        digitalWrite(buzzer, HIGH);
    }
    else if (g_fisOutput[0] > 50) // Moderate
    {
        digitalWrite(ledRetardRed, HIGH);
    }
    else if (g_fisOutput[0] > 25) // Light
    {
        digitalWrite(ledRetardAmber, HIGH);
    }
}
```

```
    }
    else if (g_fisOutput[0] > 25) // Light
    {
        digitalWrite(ledRetardGreen, HIGH);
    }
}

lastTime = currentTime; // Update last time
lastDistance = currentDistance; // Update last distance
}
}

int readLiDAR()
{
    return myLidarLite.distance(false);
}

void fuzzy_process()
{
    fis_evaluate();
}

/*-----
Support functions for Fuzzy Inference System
-----*/

// Trapezoidal Member Function
FIS_TYPE fis_trapmf(FIS_TYPE x, FIS_TYPE* p)
{
    FIS_TYPE a = p[0], b = p[1], c = p[2], d = p[3];
    FIS_TYPE t1 = ((x <= c) ? 1 : ((d < x) ? 0 : ((c != d) ? ((d - x) / (d - c)) : 0)));
    FIS_TYPE t2 = ((b <= x) ? 1 : ((x < a) ? 0 : ((a != b) ? ((x - a) / (b - a)) : 0)));
    return (FIS_TYPE) min(t1, t2);
}

// Triangular Member Function
FIS_TYPE fis_trimf(FIS_TYPE x, FIS_TYPE* p)
{
    FIS_TYPE a = p[0], b = p[1], c = p[2];
    FIS_TYPE t1 = (x - a) / (b - a);
    FIS_TYPE t2 = (c - x) / (c - b);
    if ((a == b) && (b == c)) return (FIS_TYPE) (x == a);
    if (a == b) return (FIS_TYPE) (t2*(b <= x)*(x <= c));
    if (b == c) return (FIS_TYPE) (t1*(a <= x)*(x <= b));
    t1 = min(t1, t2);
    return (FIS_TYPE) max(t1, 0);
}
```

```
FIS_TYPE fis_min(FIS_TYPE a, FIS_TYPE b)
{
    return min(a, b);
}

FIS_TYPE fis_max(FIS_TYPE a, FIS_TYPE b)
{
    return max(a, b);
}

FIS_TYPE fis_array_operation(FIS_TYPE *array, int size, _FIS_ARR_OP pfnOp)
{
    int i;
    FIS_TYPE ret = 0;

    if (size == 0) return ret;
    if (size == 1) return array[0];

    ret = array[0];
    for (i = 1; i < size; i++)
    {
        ret = (*pfnOp)(ret, array[i]);
    }

    return ret;
}

/*-----
Data for Fuzzy Inference System
-----*/

_FIS_MF fis_gMF[] =
{
    fis_trapmf, fis_trimf
};

// Count of member function for each Input
int fis_gIMFCount[] = { 3, 3 };

// Count of member function for each Output
int fis_gOMFCount[] = { 4 };

// Coefficients for the Input Member Functions
FIS_TYPE fis_gMFI0Coeff1[] = { 0, 0, 500, 1000 };
FIS_TYPE fis_gMFI0Coeff2[] = { 500, 1500, 2500 };
FIS_TYPE fis_gMFI0Coeff3[] = { 2000, 2500, 4000, 4000 };
```

```

FIS_TYPE* fis_gMFI0Coeff[] = { fis_gMFI0Coeff1, fis_gMFI0Coeff2,
fis_gMFI0Coeff3 };
FIS_TYPE fis_gMFI1Coeff1[] = { 0, 0, 500, 1000 };
FIS_TYPE fis_gMFI1Coeff2[] = { 500, 1500, 2500 };
FIS_TYPE fis_gMFI1Coeff3[] = { 2000, 2500, 4000, 4000 };
FIS_TYPE* fis_gMFI1Coeff[] = { fis_gMFI1Coeff1, fis_gMFI1Coeff2,
fis_gMFI1Coeff3 };
FIS_TYPE** fis_gMFICoeff[] = { fis_gMFI0Coeff, fis_gMFI1Coeff };

// Coefficients for the Output Member Functions
FIS_TYPE fis_gMFO0Coeff1[] = { -10, 10, 30 };
FIS_TYPE fis_gMFO0Coeff2[] = { 20, 40, 60 };
FIS_TYPE fis_gMFO0Coeff3[] = { 50, 70, 90 };
FIS_TYPE fis_gMFO0Coeff4[] = { 80, 100, 120 };
FIS_TYPE* fis_gMFO0Coeff[] = { fis_gMFO0Coeff1, fis_gMFO0Coeff2,
fis_gMFO0Coeff3, fis_gMFO0Coeff4 };
FIS_TYPE** fis_gMFOCoeff[] = { fis_gMFO0Coeff };

// Input membership function set
int fis_gMFI0[] = { 0, 1, 0 };
int fis_gMFI1[] = { 0, 1, 0 };
int* fis_gMFI[] = { fis_gMFI0, fis_gMFI1 };

// Output membership function set
int fis_gMFO0[] = { 1, 1, 1, 1 };
int* fis_gMFO[] = { fis_gMFO0 };

// Rule Weights
FIS_TYPE fis_gRWeight[] = { 1, 1, 1, 1, 1, 1, 1, 1, 1 };

// Rule Type
int fis_gRType[] = { 1, 1, 1, 1, 1, 1, 1, 1, 1 };

// Rule Inputs
int fis_gRI0[] = { 1, 1 };
int fis_gRI1[] = { 2, 1 };
int fis_gRI2[] = { 3, 1 };
int fis_gRI3[] = { 1, 2 };
int fis_gRI4[] = { 2, 2 };
int fis_gRI5[] = { 3, 2 };
int fis_gRI6[] = { 1, 3 };
int fis_gRI7[] = { 2, 3 };
int fis_gRI8[] = { 3, 3 };
int* fis_gRI[] = { fis_gRI0, fis_gRI1, fis_gRI2, fis_gRI3, fis_gRI4, fis_gRI5, fis_gRI6,
fis_gRI7, fis_gRI8 };

// Rule Outputs
int fis_gRO0[] = { 3 };

```

```

int fis_gRO1[] = { 1 };
int fis_gRO2[] = { 1 };
int fis_gRO3[] = { 4 };
int fis_gRO4[] = { 4 };
int fis_gRO5[] = { 2 };
int fis_gRO6[] = { 4 };
int fis_gRO7[] = { 4 };
int fis_gRO8[] = { 4 };
int* fis_gRO[] = { fis_gRO0, fis_gRO1, fis_gRO2, fis_gRO3, fis_gRO4, fis_gRO5,
fis_gRO6, fis_gRO7, fis_gRO8 };

FIS_TYPE fis_gIMin[] = { 0, 0 }; //Input range Min

FIS_TYPE fis_gIMax[] = { 4000, 4000 }; //Input range Max

FIS_TYPE fis_gOMin[] = { 0 }; //Output range Min

FIS_TYPE fis_gOMax[] = { 100 }; //Output range Max

FIS_TYPE fis_MF_out(FIS_TYPE** fuzzyRuleSet, FIS_TYPE x, int o)
{
    FIS_TYPE mfOut;
    int r;

    for (r = 0; r < fis_gcR; ++r)
    {
        int index = fis_gRO[r][o];
        if (index > 0)
        {
            index = index - 1;
            mfOut = (fis_gMF[fis_gMFO[o][index]])(x, fis_gMFOCoeff[o][index]);
        }
        else if (index < 0)
        {
            index = -index - 1;
            mfOut = 1 - (fis_gMF[fis_gMFO[o][index]])(x, fis_gMFOCoeff[o][index]);
        }
        else
        {
            mfOut = 0;
        }

        fuzzyRuleSet[0][r] = fis_min(mfOut, fuzzyRuleSet[1][r]);
    }
    return fis_array_operation(fuzzyRuleSet[0], fis_gcR, fis_max);
}

```

```

FIS_TYPE fis_defuzz_centroid(FIS_TYPE** fuzzyRuleSet, int o)
{
    FIS_TYPE step = (fis_gOMax[o] - fis_gOMin[o]) / (FIS_RESOLUTION - 1);
    FIS_TYPE area = 0;
    FIS_TYPE momentum = 0;
    FIS_TYPE dist, slice;
    int i;

    // calculate the area under the curve formed by the MF outputs
    for (i = 0; i < FIS_RESOLUTION; ++i){
        dist = fis_gOMin[o] + (step * i);
        slice = step * fis_MF_out(fuzzyRuleSet, dist, o);
        area += slice;
        momentum += slice*dist;
    }

    return ((area == 0) ? ((fis_gOMax[o] + fis_gOMin[o]) / 2) : (momentum / area));
}

/*-----
Fuzzy Inference System
-----*/
void fis_evaluate()
{
    FIS_TYPE fuzzyInput0[] = { 0, 0, 0 };
    FIS_TYPE fuzzyInput1[] = { 0, 0, 0 };
    FIS_TYPE* fuzzyInput[fis_gcI] = { fuzzyInput0, fuzzyInput1, };
    FIS_TYPE fuzzyOutput0[] = { 0, 0, 0, 0 };
    FIS_TYPE* fuzzyOutput[fis_gcO] = { fuzzyOutput0, };
    FIS_TYPE fuzzyRules[fis_gcR] = { 0 };
    FIS_TYPE fuzzyFires[fis_gcR] = { 0 };
    FIS_TYPE* fuzzyRuleSet[] = { fuzzyRules, fuzzyFires };
    FIS_TYPE sW = 0;

    // Transforming input to fuzzy Input
    int i, j, r, o;
    for (i = 0; i < fis_gcI; ++i)
    {
        for (j = 0; j < fis_gIMFCount[i]; ++j)
        {
            fuzzyInput[i][j] =
                (fis_gMF[fis_gMFI[i][j]])(g_fisInput[i], fis_gMFCoeff[i][j]);
        }
    }

    int index = 0;
    for (r = 0; r < fis_gcR; ++r)
    {

```

```
    if (fis_gRType[r] == 1)
    {
        fuzzyFires[r] = FIS_MAX;
        for (i = 0; i < fis_gcI; ++i)
        {
            index = fis_gRI[r][i];
            if (index > 0)
                fuzzyFires[r] = fis_min(fuzzyFires[r], fuzzyInput[i][index - 1]);
            else if (index < 0)
                fuzzyFires[r] = fis_min(fuzzyFires[r], 1 - fuzzyInput[i][-index - 1]);
            else
                fuzzyFires[r] = fis_min(fuzzyFires[r], 1);
        }
    }
    else
    {
        fuzzyFires[r] = FIS_MIN;
        for (i = 0; i < fis_gcI; ++i)
        {
            index = fis_gRI[r][i];
            if (index > 0)
                fuzzyFires[r] = fis_max(fuzzyFires[r], fuzzyInput[i][index - 1]);
            else if (index < 0)
                fuzzyFires[r] = fis_max(fuzzyFires[r], 1 - fuzzyInput[i][-index - 1]);
            else
                fuzzyFires[r] = fis_max(fuzzyFires[r], 0);
        }
    }

    fuzzyFires[r] = fis_gRWeight[r] * fuzzyFires[r];
    sW += fuzzyFires[r];
}

if (sW == 0)
{
    for (o = 0; o < fis_gcO; ++o)
    {
        g_fisOutput[o] = ((fis_gOMax[o] + fis_gOMin[o]) / 2);
    }
}
else
{
    for (o = 0; o < fis_gcO; ++o)
    {
        g_fisOutput[o] = fis_defuzz_centroid(fuzzyRuleSet, o);
    }
}
}
```