Hardware Simulation of Camera-based Rear-End Collision Avoidance System using Fuzzy Logic

Noor Cholis Basjaruddin¹,² Fazrin Adinugraha² Pebriyanto Pujiputra³ Didin Saefudin⁴ Edi Rakhman⁵

¹,²,³,⁴,⁵Department of Electrical Engineering, Politeknik Negeri Bandung, Bandung, Indonesia.
*Corresponding author. Email: noorcholis@polban.ac.id

ABSTRACT

RCAS (Rear-end Collision Avoidance Systems) is one of the safety features in autonomous cars that can determine what kind of movement that a car installed with the RCAS system should perform to avoid any crashes with another car/object in front of it. This camera-based RCAS uses a Pixy CMUCAM5 camera to measure the distance of a leading car. The measured distance used as the input of a Fuzzy Logic Algorithm will determine the speed and steering angle of the car. This system is applied to a small toy car with an Active Steering Assist to execute the calculated movement that should be performed by the car. The use of this system is proven to improve the level of the safety of the car by successfully avoiding possible collisions.

Keywords: Rear-End Collision-Avoidance System, camera-based RCAS, Fuzzy Logic Algorithm, Active Steering Assist.

1. INTRODUCTION

Collision Avoidance System (CAS) is an automobile safety system designed to prevent or reduce the severity of a collision [1] and in this research, the type of collision that’s focused to be avoided is rear-end collisions, which happens when the leading car stops or suddenly stops while the trailing vehicle is still on some certain speed. There are several similar works in RCAS development. For example, in [2], RCAS system uses two types of sensors that are fused to perform better distance measurements that successfully avoid the car from collisions. In [3], a system was developed to perform automatic braking when the driver is late in avoiding an incoming collision using a radar sensor to estimate the distance and piezoelectric sensor to detect if the driver is stepping on the brake pedal. In [4], a camera and image processing techniques are used to detect cars on the road and calculate the distance to know the Time to Collision as a warning system to avoid a collision. In this research, a Pixy CMUCAM5 is used as an input to measure the distance of the leading car, an Arduino Uno is used as the mainboard, and Fuzzy Logic Algorithm is used to calculate the movement and speed of the car.

RCAS is a subsystem in a smart vehicle that can work alongside with many other subsystems. Several previous works correlated to RCAS are [5] and [6] in which a car is installed with a system that avoids it from getting into collisions using ultrasonic sensors to measure the distance with leading car/object. In [7], a system installed on a car keeps it on the lane using a camera to detect the lane lines, and in [8] a system installed on a car enables it to safely overtake the leading vehicle using an ultrasonic sensor.

This research is a development of [5] and [6]. In those two researches, ultrasonic sensors are used to measure the distance of a leading car. Although ultrasonic sensors have high distance resolution, it doesn’t have high angular resolution [9], in situations which both leading and trailing car is not inline, ultrasonic sensors would face sideways which means they can’t measure the distance of leading car. However, that’s not the case for the camera. As long as an object is within the observation area and is captured by the camera, with some certain image processing techniques, the distance of said object can be measured. This means that the use of a camera to measure the distance in this research as a replacement of an ultrasonic sensor can improve the system’s capability to avoid collisions because it can observe a larger area.

The rest of the paper is organized as follows: Section 2 introduces the research methods, which include How RCAS Works, System’s Block Diagram, Distance
Measuring with Pixy CMUCAM5, Fuzzy Logic
RCAS, and Hardware Simulation, Section 3 presents
the Result and Data Analysis of the system, and
Section 4 shows conclusion and recommendations for
future research.

2. RESEARCH METHODS

2.1. How RCAS Works
The way that RCAS works can be illustrated by Figure
1 where the red, trailing car is a car installed with
RCAS system and the grey, leading car is a car that
simulates other cars that RCAS will crash into. RCAS
starts by measuring the distance between itself and the
leading car, and then it calculates the steering angle
and speed that the car should perform to avoid
collision, if predicted to happens. There are two ways
that RCAS system can avoid itself from running into
collisions, it can either brake right before it hits the
leading car (Figure 1 middle), or if not possible, then
it can perform a maneuver to the sides (Figure 1 bottom).

![Figure 1 How RCAS Works](image1)

2.2. Block Diagram
The block diagram of the system can be seen in Figure
2. There are two inputs to the system: a camera sensor
Pixy CMUCAM5 that’s used to measure the distance
of leading car that will be later explained and a push
button to select the two-speed modes of the RCAS car,
this different speed mode is used to test the system in
different situations to test its reliability and response
time. The two inputs will be read by a microcontroller
Arduino Uno which processes the data and will
determine the value of steering angle and speed of the
car to avoid collisions. Those two values are then
executed to the system, the steering angle value is sent
to a front servo motor directly and the speed value is
sent to the rear DC motor that controls the speed of the
car with the help of a motor driver L298N. All of the
data then saved using a data logger for further
analysis.

![Figure 2 RCAS Block Diagram](image2)

2.3. Distance Measurement using Pixy
CMUCAM5
Using a camera sensor to measure distance is not a
novel ideal. Many researchers such as [10] use image
processing techniques, specifically color filtering, to
detect and estimate the distance of an object.
However, due to the low processing power of Arduino
Uno used in this research, we can’t process the image
on the Arduino, thus Pixy CMUCAM5 is used.
Pixy CMUCAM5 is a camera with an on-board
processing unit, meaning the camera captures an
image and perform the image processing techniques
within itself and outputs the data that we wish to get
that will be later be read by Arduino Uno. This makes
way less processing load for the mainboard as it only
needs to read ready-to-use numerical data on the I/O
port.

![Figure 3 Pixy CMUCAM5](image3)

To perform distance measurement, we need to prepare
an object with a distinct color to be detected as a
measurement reference since pixy cameras use hue-
based color filtering to detect the object. In this
research, plain papers with different bright colors are
tested to see which color performs the best, based on
which color that can still be detected at the farthest
distance. We found that orange matches that
description. Orange plain paper is then taped on a small board and put on the back of the leading car.

![Image](image1)

**Figure 4** Color Detection by Pixy CMUCAM5. (a) Sheets of Paper Placed in Front of Camera, (b) Camera View After Detection.

After detecting the colored object, pixy CMUCAM5 sends the width, height, and coordinate of the object through the I/O port. However, what’s used to measure distance is only the width of the detected object (in pixel unit) and linearly convert it into real distance using the formula used in [10]:

\[
F = \frac{P \times D}{W}
\]

Where:
F = Camera Focal Length
P = Object Pixel Width
D = Real Distance
W = Object Real Width

### 2.4. Fuzzy Logic RCAS

Fuzzy logic is the algorithm that’s used to determine the value of steering angle and speed of the car based on the sensor reading. As illustrated in Figure 5, the fuzzy logic has two inputs: **distance**, that is the distance reading coming from the camera, and **inSpeed**, that is the maximum speed of RCAS car that can be changed by pressing the push button, meaning the system will react differently on different speed mode of RCAS car. There are two outputs of the fuzzy logic: **action** and **brake**. **action** is defined as the type of movement that the car should perform. Meanwhile, **brake** is defined as how big is the speed reduction from inSpeed to slow down and stop the car.

![Image](image2)

**Figure 5** Fuzzy Logic RCAS

Figure 6 (a) shows the membership function of **distance**, which is divided into 3 membership functions: “close”, “medium”, and “far”. Figure 6 (b) shows the membership function of **inSpeed**, which is divided into 2 membership functions: “normal” and “fast”. Figure 6 (c) shows the membership function of **brake**, which is divided into 3 membership functions: “none”, “medium”, and “hard”. Figure 6 (d) shows the membership function of **action**, which is divided into 3 membership functions: “forward”, “stop”, and “maneuver”. Table 1 shows the fuzzy rules of the system and Figure 7 shows the fuzzy surface.
Membership Function of (a) distance, (b) inSpeed, (c) brake, and (d) action

Table 1 Fuzzy Rules

<table>
<thead>
<tr>
<th>Antecedent</th>
<th>Consequent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distance</td>
<td>Speed</td>
</tr>
<tr>
<td>Far</td>
<td>&amp; Fast</td>
</tr>
<tr>
<td>Medium</td>
<td>&amp; Fast</td>
</tr>
<tr>
<td>Close</td>
<td>&amp; Fast</td>
</tr>
<tr>
<td>Far</td>
<td>&amp; Medium</td>
</tr>
<tr>
<td>Medium</td>
<td>&amp; Medium</td>
</tr>
<tr>
<td>Close</td>
<td>&amp; Medium</td>
</tr>
</tbody>
</table>

Figure 7 Fuzzy Surface

2.5 Hardware Simulation

The system is installed on an RC car to simulate a real car as seen in Figure 8. A Pixy CMUCAM5 is put at the front part of the car while other components are put in the middle of the car. The result of the hardware simulation can be seen in Figure 8.

Figure 8 Implementation of The System: (a) Top View, (b) Side View, (c) Front View

3. RESULT AND DATA ANALYSIS

The system is tested on two different scenarios: still and moving leading car. The result of the experiment with still leading car can be seen in Figure 9 and Figure 10. In this experiment, in the beginning, both cars are placed in a far distance (5 and 8 meters), then RCAS moves forward and closer to a still leading car. In both experiments, the maximum speed for RCAS car is 120 PWM.

As shown in Figure 9 (a) and Figure 10 (a), in both experiments with a still car, we can see that the system starts with no brake because the distance between the car installed with the system and the car in front of it is still part of “far” membership
function of distance, as defined in the fuzzy logic, so the car is moving on its full speed but as the distance gets shorter (65th data on Figure 9 (a) and 31st data on Figure 10 (a)) and the distance is below 300cm, which is part of “medium” membership function of distance. At this point, the brake starts to increase, making the car to move slower, and finally when the distance is very short (103rd data on Figure 9 (a) and 73rd data on Figure 10 (a)), the distance is below 120cm, which is part of “close” in the membership function of distance. At this point, there is a probability for a collision to happen, that is why the brake increases significantly to finally stop the car and the collision is then avoided. In Figure 9 (b) and Figure 10 (b), we can see that the car still has a remaining safe distance after stopping. This means that collision is successfully avoided by the system.

The result of the experiment with moving the leading car can be seen in Figure 11. In this experiment, in the beginning, both cars are placed in a close distance, then the leading car moves forward and after a few moments, RCAS moves forward and then the leading car is simulated to suddenly stop. In this experiment, the maximum speed for RCAS car is 160 PWM.

In the graph we can see that in beginning, because the distance is very close, less than 50cm, which is part of the “close” membership function of distance, the brake value is big so that the car doesn’t move but as soon as the leading car moves forward, and the distance between the two cars increases until it is more than 100cm (121st data on Figure 11 (a)), which is part of the “medium” membership function of distance, the brake value decreases, making it to move slowly, until the leading car is more than 150cm (181st data on Figure 11 (a)), which is part of the “far” membership function of distance, the brake is fully lifted, making the car to go on its full speed. When the leading car is simulated to suddenly stops, and the distance starts to decrease again to the point that it is part of the “medium” and then “close”
membership function of distance, the brake value increases again and the car stops behind it at a safe distance and collision is completely avoided. Same with the experiment with still leading car, in Figure 11 (b) we can see that after stopping, the car is still at a safe distance from the leading car, meaning that collision is successfully avoided.

4. CONCLUSION AND RECOMMENDATIONS

The result of this experiment shows a promising future for a safer and better Rear-end Collision Avoidance System. Although the system works as designed, there’s still room for improvement. There are two recommendations for the future development of the system: one, this system needs to be implemented on a bigger car that can simulate the system better to get closer to the real-size car. Two, a better measurement methodology needs to be used. Use a better camera with higher sensor quality that’s more robust to different lighting situations and use other types of sensors to help the camera to do better distance reading, that’s not affected by the lighting situation of the environment that the car operates in like an ultrasonic sensor. Then we can perform sensor fusion for better and more robust sensor reading to improve the system’s safety in avoiding collisions.

ACKNOWLEDGMENT

This paper is funded by DRPM Grant 2020 (Contract Number: 135/SP2H/AMD/LT/DRPM/2020) from Indonesian Ministry of Research, Technology, and Higher Education.

REFERENCES


