International Journal of Control, Automation, and Systems (2013) 11(6):1253-1265
DOI 10.1007/s12555-012-0057-6
ISSN:1598-6446 eISSN:2005-4092
http://www.springer.com/12555

Integrated Tracking and Accident Avoidance System for Mobile Robots

Irfan Ullah, Furqan Ullah, Qurban Ullah, and Seoyong Shin*

Abstract: In the intelligent transportation field, various accident avoidance techniques have been applied. One of the most common issues with these is the collision, which remains an unsolved problem. To this end, we developed a Collision Warning and Avoidance System (CWAS), which was implemented in the wheeled mobile robot. Path planning is crucial for a mobile robot to perform a given task correctly. Here, a tracking system for mobile robots that follow an object is presented. Thus, we implemented an integrated tracking system and CWAS in a mobile robot. Both systems can be activated independently. Using the CWAS, the robot is controlled through a remotely controlled device, and collision warning and avoidance functions are performed. Using the tracking system, the robot performs tasks autonomously and maintains a constant distance from the followed object. Information on the surroundings is obtained through range sensors, and the control functions are performed through the microcontroller. The front, left, and right sensors are activated to track the object, and all the sensors are used for the CWAS. The proposed system was tested using the binary logic controller and the Fuzzy Logic Controller (FLC). The efficiency of the robot was improved by increasing the smoothness of motion via the FLC, achieving accuracy in tracking and increasing the safety of the CWAS. Finally, simulations and experimental outcomes have shown the usefulness of the system.

Keywords: Collision avoidance, fuzzy logic, microcontroller, object tracking, range sensor.

1. INTRODUCTION

Recently, the modeling of a secure and safe vehicle control system has attracted many researchers [1]. To reduce vehicle accidents, there have been numerous advances in the development of Collision Warning Systems (CWSs) and Collision Avoidance Systems (CASs). The automatic collision avoidance system is interconnected with the hardware of the vehicle in order to control the acceleration and speed of the vehicle. Via the CWS, alarms are activated only when there is a danger of collision. An approach to human driving behavior focusing on the driver's collision avoidance was studied [2]. In [3], intelligent driver training systems for enhancing road safety were illustrated. Most accidents occur when the driver devotes his or her attention to something else, such as playing with music or using a cell phone. Some precautions include air bags, seat belts, and speed limitations. An intelligent transportation system and CWS were implemented in order to warn drivers of collisions in advance [4,5]. A forward vehicle collision warning system with a seat vibrator was demonstrated in order to reduce traffic accidents [6].

To move in an unknown environment, the mobile robot receives information about the surroundings through sensors, positioning systems, and cameras. Numerous ways of extracting the environmental features from the information gained have been explored. Various navigation techniques have been presented and implemented in robots [7-9]. Recently, many efforts have been devoted to installing multiple sensors in a robot in order to explore the environment more effectively. We use multiple sensors in a mobile robot to obtain information about the surroundings for tracking and the Collision Warning and Avoidance System (CWAS).

The scheme for the theoretical kinematic model of a four-wheeled vehicle using the vehicle's coordinate frame has been described in [10]. In [11], they calculated the error (x_E and y_E) in terms of distance to reach a target point by

$$x_E = x_D - x_B, \tag{1}$$

$$y_E = y_D - y_B, \tag{2}$$

where x_B and y_B are the outputs of the kinematic model. s_D and y_D are the desired target coordinates. Because it was a feedback system, x_B and y_B were compared with x_D and y_D in order to ensure that the vehicle headed toward the target. The distance (d') between the vehicle's current location and the target was determined by

$$d' = \sqrt{x_E^2 + y_E^2}\ . \tag{3}$$

Manuscript received February 3, 2012; Revised October 26, 2012 and January 21, 2013; accepted July 3, 2013. Recommended by Editorial Board member Seul Jung under the direction of Editor Hyouk Ryeol Choi.

Irfan Ullah and Seoyong Shin are with the Department of Information and Communication Engineering, Myongji University, Yongin 449-728, Korea (e-mails: {irfan, sshin}@mju.ac.kr).
Furqan Ullah is with the Department of Mechanical Engineering, Myongji University, Yongin 449-728, Korea (e-mail: furkan @mju.ac.kr).
Qurban Ullah is with the Department of Computer Science and Information Technology, Virtual University of Pakistan, Lahore 54000, Pakistan (e-mail: qurbanullah@hotmail.com).
* Corresponding author.

The proper angle (α) leading to the target was calculated by

$$\alpha = \tan^{-1}\left(\frac{y_E}{x_E}\right). \quad (4)$$

In [12], the Time To Collision (TTC), which is the time would take the vehicles to collide at their current speed, was calculated by

$$TTC = \frac{D}{V_t - V_l}, \quad (5)$$

where D is the distance between the vehicles, obtained via GPS positioning; V_t is the speed of the trailing vehicle; and V_l is the speed of the leading vehicle. Furthermore, the Time Gap (TG), which is the time it would take the trailing vehicle to cover the current distance to the leading vehicle, was determined by

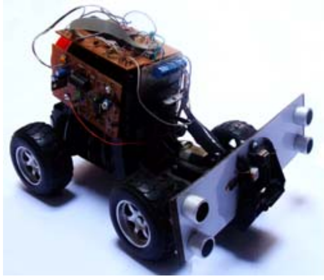
$$TG = \frac{D}{V_t}. \quad (6)$$

In this study, the unique work on the mobile robot is based on the sensor method, in which the position of the robot is updated according to the range information obtained through the range sensors. The range sensor provides information about the position of an object with respect to the robot, and it helps to find the distance between the robot and the object. This information is utilized in order to understand the environment around the robot (e.g., detect an entity, notice the danger of collision, and identify of the object's position and the size of the object). We developed an approach to tracking and CWAS using the binary logic controller and the Fuzzy Logic Controller (FLC). The FLC performs well in order to smoothly drive the servo and Direct Current (DC) motors using the uncertain data. The proposed approach can be used for security purposes in order to prevent automobile accidents and collision between robots.

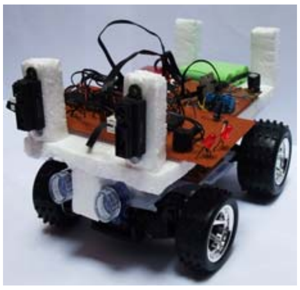
The rest part of this paper is organized as follows. Section 2 gives the background of the CWS and the CAS. In Section 3, object detection via the range sensor is illustrated. Section 4 describes the development of the CWAS. Section 5 presents the development of the object tracking system. In Section 6, we present the implementation of the system using a binary logic controller. In Section 7, the fuzzy logic system is reviewed briefly, and the implementation of the system using FLC is demonstrated. Section 8 illustrates the experiments and results. Finally, some conclusions and future work are described in Section 9.

2. BACKGROUND

The research on communication protocols for a vehicle collision warning system has enabled vehicles to communicate regarding collision warnings. Warning messages are delivered between vehicles in order to address emergency scenarios [13,14]. In these methods,



(a) Distance control robot.



(b) Accident avoidance robot. (c) Tracking robot.

Fig. 1. Physical layout.

the surroundings information about objects is obtained through the already-installed sensors in the vehicle. Therefore, the basic methodology for object detection using sensors must be improved [15]. In [16], Reflexxes^{TM} motion libraries were introduced in order to control the motion of the sensor-based robot. The jerk-limited motion (safe and continuous motion) can be calculated on-line during the motion by using these libraries.

Numerous automatic distance control methods have been proposed and implemented on vehicles and robots. A low-cost distance control algorithm was proposed for the mobile robots, which can be implemented in the real vehicles [17,18]. The schematic diagram of the distance control robot is shown in Fig. 1(a). The distance from the object was controlled by the microcontroller via the infrared (IR) range sensor. This approach was used to track an object; however, the robot did not provide smooth tracking, because the distance control algorithm was not used for the ultrasonic sensors, which were used for the left and right rotation. An accident avoidance mobile robot was demonstrated, as shown in Fig. 1(b), which had four IR range sensors to detect objects from the front and back sides [19,20]. It was not able to detect objects from the left and right sides. Therefore, it was not able to avoid collisions from the left and right sides. Furthermore, an object-tracking robot was presented, as shown in Fig. 1(c), in which the ultrasonic sensors were used for the range measurements only from the front side [21,22]. When the object took a turn to the left or right, which was beyond the limits of the robot, it did not track the object in some scenarios.

In this study, we describe the development of the robots that are described above. We increased the number of sensors and updated the control algorithm. For the CWAS, the security was increased by inserting more sensors on all sides. For the tracking system, the left and right sensors were utilized in order to track the object efficiently. As a result, we implemented both systems in the robot. The integrated system is particularly helpful because it can be controlled autonomously and by humans.

3. OBJECT DETECTION

We chose IR range finders because of their relatively low cost as compared to cameras, automotive radars, and laser rangefinders. Sharp™ GP2D12 range sensors with a detection rage of 10-80 cm were utilized. A block diagram for measuring the distance from the range sensor is shown in Fig. 2 [23]. The sensors were mounted vertically on the robot, with the transmitter is above the receiver, in order to detect the object efficiently, as shown in Fig. 3 [23]. The output of the sensor is in analog form. We collected analog voltage values from the sensor by moving five objects in front of the sensor one by one. In Fig. 4, we show a graph created from the collected data, in which the divergence between the curves is very small.

In our case, the robot should detect an object in the range of 0–80 cm. When the distance is less than 10 cm, it may fail to detect an obstacle, which may lead to the possibility of collision. The effective detection range of the IR range sensor is 10–80 cm; however, the sensor can also detect an object in the range of less than 10 cm [23]. However, the sensor was not able to give the actual value for distance measured in the range of less than 10 cm. Thus, we used an effective distance measuring range of 10–80 cm. To solve this problem, a short-range sensor can be used (e.g., the Sharp™ GP2D120 range sensor, which has an effective detection range of 4–30 cm).

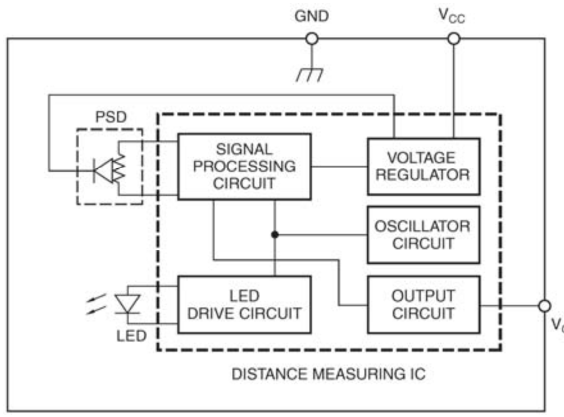


Fig. 2. Block diagram for measuring distance via the range sensor.

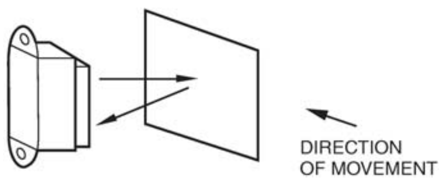


Fig. 3. Schematic showing the detection of horizontal movement via the range sensor.

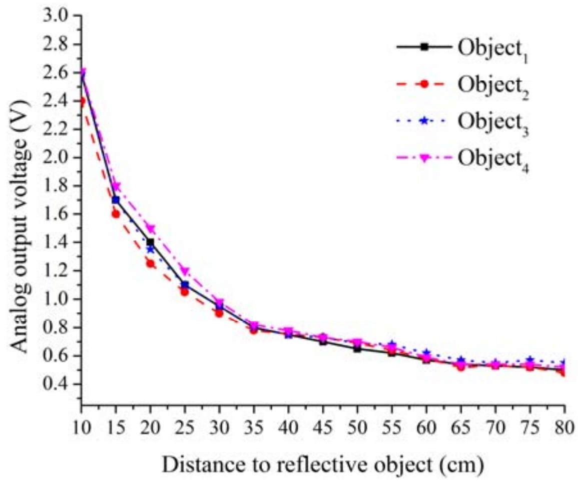
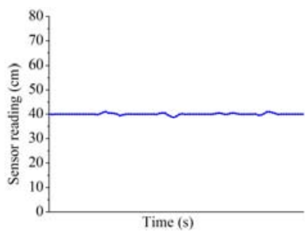
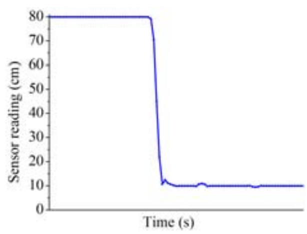


Fig. 4. Comparison of the output of the range sensor with five different objects.

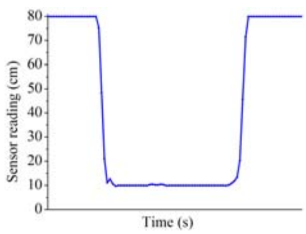
Here, we shall discuss some future work. Each sensor on the robot provides a single 10–80 cm range reading every cycle. If the distance reading is the maximum (80 cm), this means no object is detected. We can translate the distance reading into a count of object size and type [24]. However, this study does not attempt to determine the object size and type. The distance reading provides a one-dimensional view of the distance to the nearest object. In Fig. 5(a), we show an example of the trace produced by the sensor. The object is detected at a distance of 40 cm, and the surface of the object is flat because the dip is not produced. The dip is representative of the length and surface (flat or curved) of the object. In Figs. 5(b) and 5(c), two scenarios are shown: out of range and detected, respectively. Thus, we can determine the length of the object from the dips by detecting the horizontal movement. If two objects are very closely spaced, a small dip appears, as shown in Fig. 5(d).



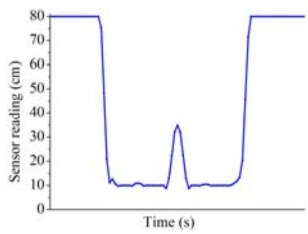
(a) Object at a constant distance with no dip.



(b) Object at nearest position.



(c) Complete object detected.



(d) Two very closely spaced objects.

Fig. 5. One-dimensional view of the distance to the object.

4. CWAS

The idea behind the system is that the control unit receives information from various sources (sensors and remotely controlled device) and performs specific tasks. The control unit contains a microcontroller that sends commands to all modules (motor driver and alarms). To avoid collision, information on the surroundings should be known from all sides (left, right, front, and back). For this purpose, we used eight range sensors that provide range information. We placed three sensors on the front side of the robot. Similarly, three sensors were placed on the back side, and the other sensors were mounted on the left and right sides. The arrangement of sensors on the robot is shown in Fig. 6. A remotely controlled device was employed to move the robot manually. We mounted all modules in a vehicle that had four wheels. To rotate the wheels, DC and servo motors were used. The DC motor was used for the forward and backward motions. The servo motor was used for the left and right motions.

The overview of the CWAS is illustrated in Fig. 7. Initially, the object was detected through the range sensors and then control functions were performed. The robot was controlled in two modes: manual mode and CWAS mode. In the manual mode, the robot was controlled via a remotely controlled device. In the CWAS mode, the motion of the robot was controlled via instructions from the range sensors. In this mode, the forward, backward, left, right, and stop functions were performed. When the object was detected in the range of 40–80 cm on any side, sound and light alarms began automatically. When the object was detected in the range of 20–40 cm from the front or back sides, the stop function was carried out. When the object was detected at a distance of less than 20 cm from the front side, the backward function was performed. Similarly, when the object was detected at a distance of less than 20 cm from the back side, the forward function was activated. If the S_L or S_R sensor indicated the presence of the object, the right or left mode was activated in the range of 10–40 cm, respectively.

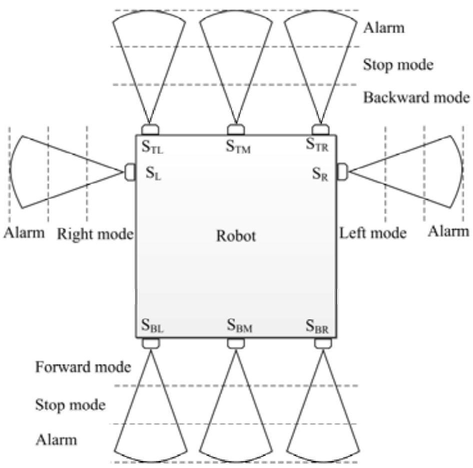


Fig. 6. Collision warning and avoidance from various directions.

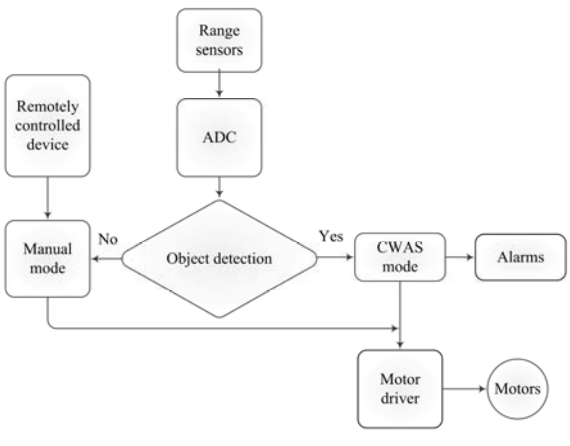


Fig. 7. Flow diagram of the CWAS design.

5. OBJECT TRACKING SYSTEM

To follow an object, the robot requires range measurements from the front, left, and right directions. Therefore, we used five range sensors for the tracking system. The S_{TM} sensor was used for the forward and backward motions, and the remaining sensors were used for the left and right rotations. The S_{TL}, S_{TM}, and S_{TR} sensors gave the range measurements from the front side. The S_L sensor gave the range information from the left side; likewise, the S_R sensor gave the range measurements from the right side. The S_{TM} sensor was reserved for the forward and backward motions, and the other four sensors were used for the left and right rotations. Fifteen different obstacle environments used for object tracking are shown in Fig. 8.

We assumed that the distance for near was below 20 cm, medium was between 20 and 40 cm, and far was between 40 and 80 cm. If the S_{TL}, S_{TM}, and S_{TR} sensors detected the object in the near region, the robot moved in the forward direction. If the S_{TL} sensor detected the object in the near region, the S_{TM} sensor detected the object in the constant region, and the S_{TR} sensor detected

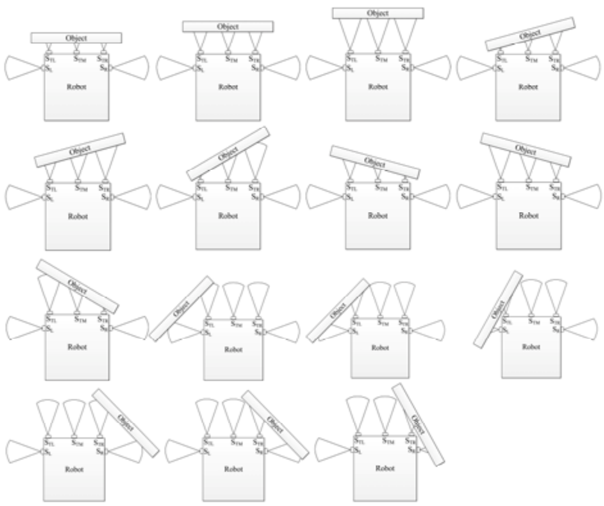


Fig. 8. Schemetic illustrating various scenarios for object tracking.

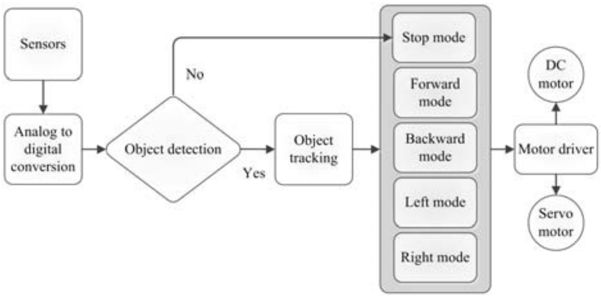


Fig. 9. Flow diagram of the object tracking system design.

the object in the far region, the robot went to forward in the left direction. If no object was detected through the sensors, the robot was stopped.

Here, we faced some complex scenarios. If the object is not straight (e.g., trumpet-shaped), the front sensors (S_{TL}, S_{TM}, and S_{TR}) can simultaneously detect the object, even if it should be detected by only one sensor. Then, the action is performed according to the algorithm, and this action may be different from the action required. We tried to solve this problem by mounting the sensors vertically (Fig. 3) and by installing the sensors at a certain distance from each other in order to avoid interference. We placed the S_{TM} sensor at the center for the forward and backward motions, and the S_{TL} and S_{TR} sensors were placed on sides for the left and right motions. This placement also helped to avoid interference.

The flowchart of the tracking system for the mobile robot is shown in Fig. 9. Initially, the sensors determined the position of the object, and then, different operations were performed on the range data to obtain the final decision. Finally, commands were sent to the DC and servo motors. In the tracking system, we used hardware modules that were used for the CWAS.

6. BINARY LOGIC CONTROLLER

The hardware circuits were designed in Proteus™. The circuit diagram used to test the range sensor is shown in Fig. 10. A changeable analog voltage source was used to perform the same tasks as the IR range sensor. Since the output of the range sensor was in the form of analog voltages, an Analog to Digital Converter (ADC), the ADC0808, was used to convert the analog signals into a digital form. All the components were interfaced with the microcontroller.

In the CWAS, the control unit was developed by connecting the remotely controlled unit, the ADC, and the motor module with the microcontroller. To send instructions from the remotely controlled device, a transmitter and a receiver were used. The receiver was directly connected to the microcontroller. The transmitter module contained an encoder (PT2262), and a decoder (PT2272), which was used on the receiver side. From the given signals, the microcontroller sent commands to the motors. To accept logic levels and drive the motors, a dual full-bridge motor driver, the L298, was used.

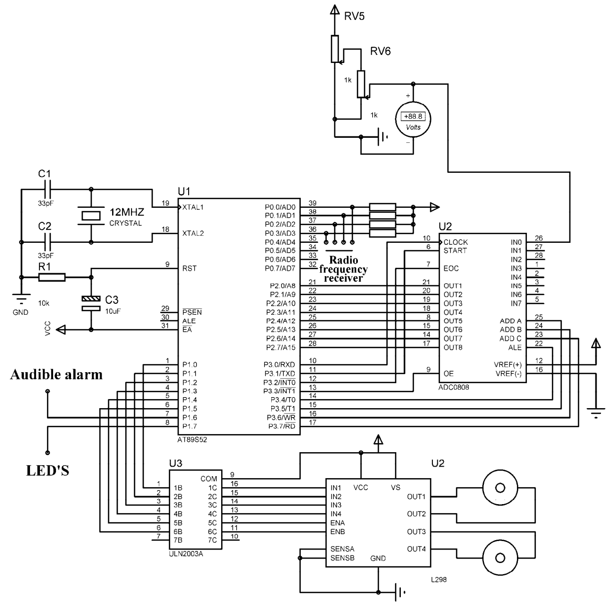


Fig. 10. Integrated hardware circuit to test the range sensor and other hardware modules.

```
a[] = {1...b...c...d};   // array of hexadecimal numbers
for e = 1...8; do        // reading data from eight sensors
        sdata[] = readsensors;
if sdata[] = 0; then     // object is out of range
        manual();
else {
        for j = a[];     // for forward and backward motion from S_TL, S_TM,
                         // S_TR, S_BL, S_BM, and S_BR sensors
        do {
            if sdata[0,1,2] = 1...b; then   // range between 0-20 cm
                    Backward();
            else if sdata[0,1,2,3,4,5] = b...c; then   // range between 20-40 cm
                    Stop();
            else if sdata[3,4,5] = 1...b; then
                    Forward();
        }
        for j = a[];     // for left and right motion from S_L and S_R sensors
        do {
            if sdata[6] = 1...b; then
                    Right-half();
            else if sdata[6] = b...c; then
                    Right();
            else if sdata[7] = 1...b; then
                    Left-half();
            else if sdata[7] = b...c; then
                    Left();
        }
        for j = a[];     // for alarm from all sensors
        do {
            if sdata[0,1,2,3,4,5] = c...d; then   // range between 40-80 cm
                    Alarm-high();
            else if sdata[6,7] = c...d; then
                    Alarm-slow();
        }
        manual();
}
go to 2;
```

Fig. 11. CWAS algorithm for binary logic controller.

In the tracking system, the control unit was developed by connecting the ADC and the motor module with the microcontroller. As the robot autonomously navigates in the tracking mode, we did not need to control the robot motion manually. Thus, a remotely controlled device was not required for the tracking system.

In Fig. 11, we summarize the algorithm for the CWAS. The S_{TL}, S_{TM}, S_{TR}, S_{BL}, S_{BM}, S_{BR}, S_L, and S_R sensors are represented by sdata[0]–sdata[7] in the algorithm, respectively. An array of hexadecimal numbers was obtained by interfacing the analog voltage source, ADC, and microcontroller. The voltages were varied from the analog voltage source within the range of 0.01–3.0 V by a step size of 0.01 V. The output of the ADC was interfaced on a parallel port of the microcontroller. It read the

8-bit data in hexadecimal form. After the object detection via the sensor, the digital values were compared with the predefined array. If the 8-bit was found from the predefined array, the corresponding function was activated. For example, if the value from the S_{TM}, S_{TL}, or S_{TR} sensors was between 0 and 20 cm, the backward function was activated. Likewise, if the obstacle was detected via the S_L sensor in the range of 20 to 40 cm the right function was activated. If the obstacle was detected in the range of 40 to 80 cm, the alarm function was activated.

We summarize the algorithm for the tracking system in Fig. 12. Like the CWAS algorithm, this algorithm performed task of comparing the current 8-bit data with the predefined array. If the obstacle was not detected via any sensor, the stop function was executed. If the 8-bit data was found from the array the corresponding function was activated according to the tracking system. For example, if the object was detected through the S_{TM} sensor, the corresponding backward, stop, and forward functions were activated. The corresponding straight, left, and right functions were activated after object detection via the S_{TL}, S_{TR}, S_L, and S_R sensors. For example, if the object was detected through the S_{TL} and S_{TR} sensors in the range of 0 to 20 cm, the straight function was executed. If the S_{TL} and S_{TR} sensors detected the object in the range of 0 to 20 cm and 40 to 80 cm, respectively, the left function was performed. Likewise, if the object was detected through the S_{TR} and S_R sensors in the range of 20 to 40 cm and 40 to 80 cm, respectively, the right function was executed.

Both algorithms were successfully implemented and verified in simulation by using C-language and µVision™ software. The source code was compiled, and a HEX file was generated. It was used in Proteus™ for the integrated simulation of the microcontroller with other hardware modules.

```
a[] = {1...b...c...d};              // array of hexadecimal numbers
for e = 1...8; do                   // reading data from eight sensors
        sdata[] = readsensors;
if sdata[] = 0; then                // object is out of range
        stop();
        for j = a[];   // for forward and backward motion
                       // from S_TM sensor
                if sdata[1] = 1...b; then        // range between 0-20 cm
                        backward();
                        delay = 1...sdata[1];
                else if sdata[1] = b...c; then  // range between 20-40 cm
           do           stop();
                else if sdata[1] = c...d; then  // range between 40-80 cm
                        forward();
                        delay = 1...sdata[1];
        for j = a[];   // for left and right rotation from S_TL, S_TR, S_L,
                       //and S_R sensors
                if sdata[0,2,6,7] = a[];
                        if sdata[0] = 1...b & sdata[2] = 1...b; then
                                straight();
                        else if sdata[0] = b...c & sdata[2] = b...c; then
                                straight();
                        else if sdata[0] = c...d & sdata[2] = c...d; then
                                straight();
                        else if sdata[0] = 1...b & sdata[2] = b...c; then
                                left-half();
else                    else if sdata[0] = b...c & sdata[2] = c...d; then
                                left-half();
                        else if sdata[0] = 1...b & sdata[2] = c...d; then
                                left();
                        else if sdata[0] = b...c & sdata[2] = 1...b; then
                                right-half();
           do   then    else if sdata[0] = c...d & sdata[2] = b...c; then
                                right-half();
                        else if sdata[0] = c...d & sdata[2] = 1...b; then
                                right();
                        else if sdata[6] = c...d & sdata[0] = c...d; then
                                left();
                        else if sdata[6] = b...c & sdata[0] = b...c; then
                                left-full();
                        else if sdata[6] = 1...b & sdata[0] = 1...b; then
                                left-full();
                        else if sdata[2] = c...d & sdata[7] = c...d; then
                                right();
                        else if sdata[2] = b...c & sdata[7] = b...c; then
                                right-full();
                        else if sdata[2] = 1...b & sdata[7] = 1...b; then
                                right-full();
go to 2;
```

Fig. 12. Object tracking algorithm for binary logic controller.

7. PROPOSED FUZZY LOGIC CONTROLLER

Over the last decades, fuzzy logic has experienced a rapid growth in control nonlinear systems [25]. Fuzzy logic has been applied to the development of mobile robot navigation and in various industrial applications [26]. The main advantage of the Fuzzy Logic System (FLS) is that it can extract heuristic rules that contain if-then statements from human experience. Thus, it is easy to implement FLC. To control the uncertain data from the range sensors, FLC is used because it captures the uncertainties in an efficient way. In robot navigation, FLS is preferred in producing the outputs that are used to control various actions via the inputs from the range sensors. FLC performs fuzzification, rule base, fuzzy inference, and defuzzification, as shown in Fig. 13. FLC received the surrounding information from the sensors. This information was the input, which was used for the fuzzification in order to create membership functions. Fuzzy rules were established from the input and output data. FIS produced the result for each rule. After the defuzzification, the actual outputs for the vehicle drive system were obtained. Finally, control actions were performed for motion of the robot. The activity diagram of the system is shown in Fig. 14.

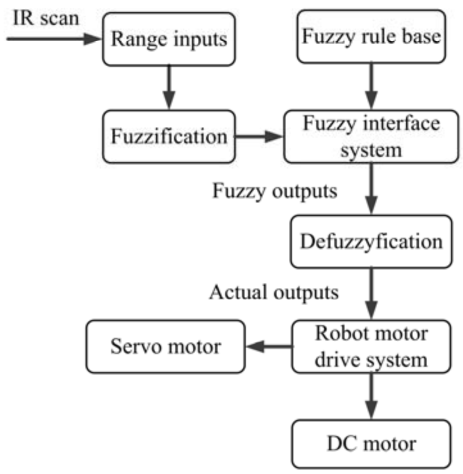


Fig. 13. Illustration of the FLC.

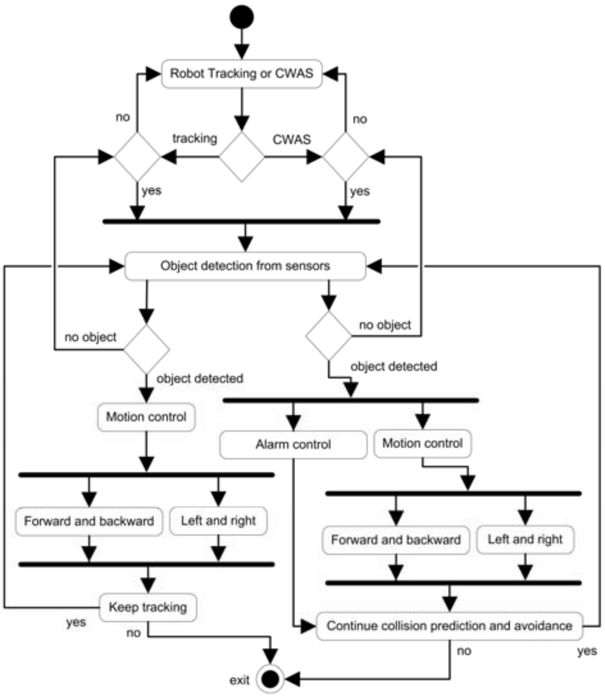


Fig. 14. Activity diagram of the tracking and CWAS.

7.1. Object tracking system

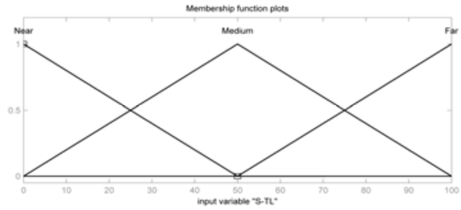
The MATLAB® fuzzy logic toolbox was used to implement the FLC. It had five input fuzzy variables and two output fuzzy variables. We represented the membership functions for each fuzzy input, where the linguistic variables were labeled as S-TL, S-TM, S-TR, S-L, and S-R, as shown in Fig. 15. Values were plotted vs. degrees of membership for each input fuzzy variable. The input from each sensor to FLC was defined by three fuzzy sets labeled as Near, Medium, and Far. The input is fuzzified using a triangular membership function to reduce the computational load on the microcontroller, and the degree of membership of a particular fuzzy set is scaled between 0 and 100 instead of from 0 to 1 to prevent floating point storage and to reduce calculations. The membership functions for the sensor can be defined by [21,22]

$$\mu_{\text{Sensor, Near}} = \begin{cases} -2l_S + 100 & 0 \le l_S \le 50 \\ 0 & l_S \ge 50, \end{cases} \tag{7}$$

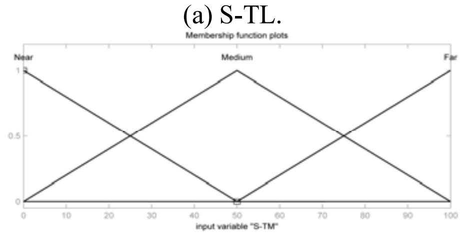
$$\mu_{\text{Sensor, Med}} = \begin{cases} 2l_S & 0 \le l_S \le 50 \\ -2l_S + 100 & 50 \le l_S \le 100, \end{cases} \tag{8}$$

$$\mu_{\text{Sensor, Far}} = \begin{cases} 2l_S - 100 & 50 \le l_S \le 100 \\ 0 & l_S \ge 100. \end{cases} \tag{9}$$

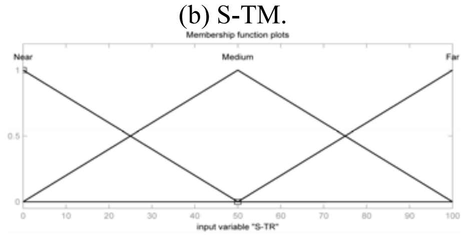
The amounts of membership functions were determined by using the inputs and outputs. Large membership functions give a high computational load; however, the performance of the system is improved by increasing the membership functions. The fuzzy output variable DC motor was decomposed into three fuzzy sets labeled Back, Normal, and Fast, as shown in Fig. 16(a). The fuzzy output variable servo motor was decomposed into seven fuzzy sets labeled Left-full, Left, Left-half,



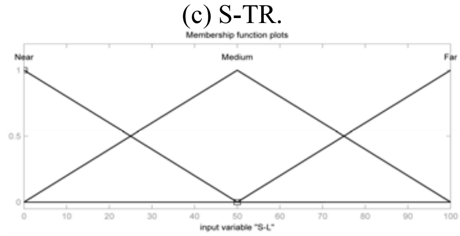
(a) S-TL.



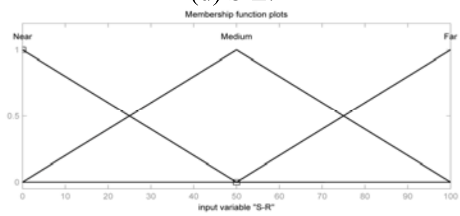
(b) S-TM.



(c) S-TR.

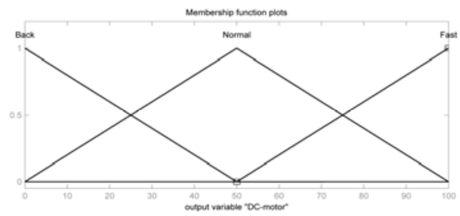


(d) S-L.

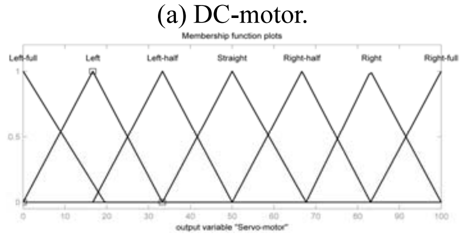


(e) S-R.

Fig. 15. Defininitions of the tracking system membership functions for the fuzzy input variables.



(a) DC-motor.



(b) Servo-motor.

Fig. 16. Defininitions of the tracking system membership functions for the fuzzy ouput variables.

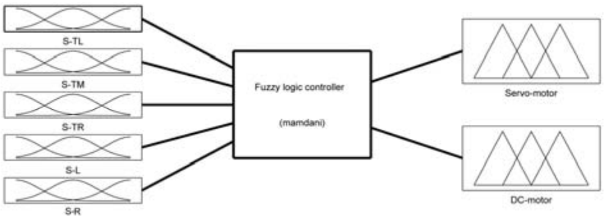


Fig. 17. FIS for the tracking system.

Table 1. Rule base with fuzzy output variable servo motor and fuzzy input variables S-TL and S-TR.

S-TL	S-TR		
	Near	**Medium**	**Far**
Near	Straight	Left-half	Left
Medium	Right-half	Straight	Left-half
Far	Right	Right-half	Straight

Table 2. Rule base with fuzzy output servo motor and fuzzy input variables S-TL and S-L.

S-TL	S-L	Servo-motor
Near	Near	Left-full
Medium	Medium	Left-full
Far	Far	Left

Table 3. Rule base with fuzzy output servo motor and fuzzy input variables S-TR and S-R.

S-TR	S-R	Servo-motor
Near	Near	Right-full
Medium	Medium	Right-full
Far	Far	Right

Table 4. Rule base with fuzzy output DC motor and fuzzy input variable S-TM.

S-TM	DC-motor
Near	Back
Medium	Normal
Far	Fast

Straight, Right-half, Right, and Right-full, as shown in Fig. 16(b). FIS for the tracking system is depicted in Fig. 17.

To calculate the consequent part of each rule, Mamdani's fuzzy reasoning method, which is comprised of a simple min-operation and a max-operation, was preferred. To obtain better results, fuzzy rules were determined through different experiments and improved via a number of tests. Fuzzy rules were accumulated that contained the AND operator for the tracking system, as well as a collection of IF-THEN statements. Table 1 shows the rule base in which nine fuzzy rules were defined with output fuzzy variable servo motor and input fuzzy variables S-TL and S-TR. Table 2 shows the rule base for which three fuzzy rules were declared with fuzzy output variable servo motor and input fuzzy variables S-TL and S-L. Similarly, three fuzzy rules were defined with fuzzy output variable servo motor and fuzzy input variables S-TR and S-R, as shown in Table 3. In Table 4, we represent the rule base with fuzzy output variable DC motor and fuzzy input variable S-TM.

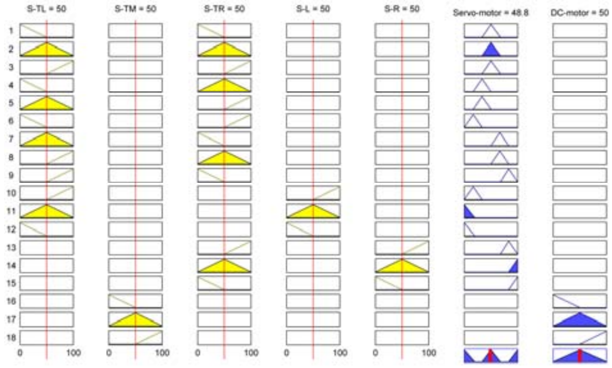
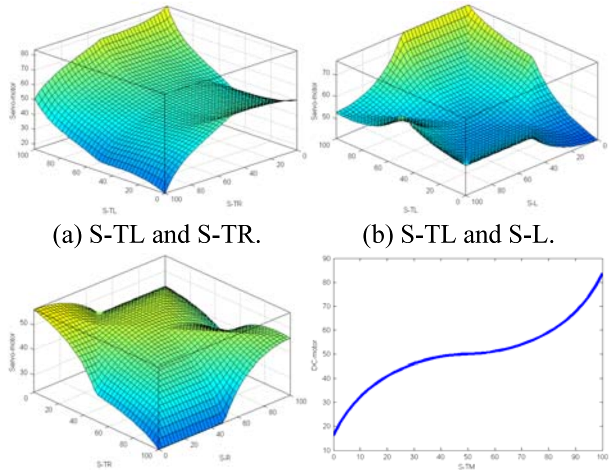


Fig. 18. Rule base for the tracking system.



(a) S-TL and S-TR. (b) S-TL and S-L.

(c) S-TR and S-R. (d) Fuzzy output variable DC-motor with fuzzy input variable S-TM

Fig. 19. Control surface for the tracking system. Fuzzy output variable (servo motor) with fuzzy input variables.

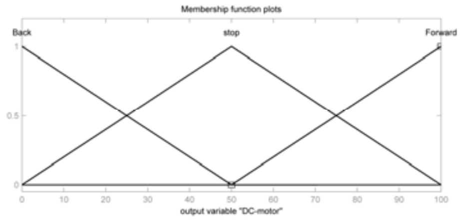
The center of gravity method has been confirmed as giving accurate and efficient results [27,28]. In the proposed controller, it was used for the defuzzification. It can be expressed by [29]

$$Z_a = \frac{\int \mu_C(Z) \cdot Z \, d_Z}{\int \mu_C(Z) \, d_Z}, \tag{10}$$

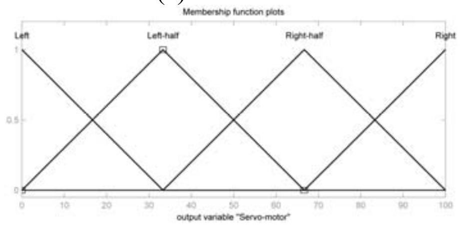
where Z_a is the defuzzified output, $\mu_c(Z)$ is the degree of membership, and Z is the output variable. The rule base from five fuzzy input and two fuzzy output variables for the tracking system is depicted in Fig. 18. The control surface for the tracking system is shown in Fig. 19.

7.2. CWAS

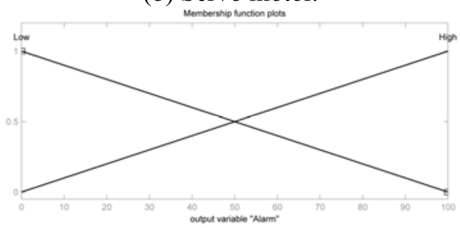
For the CWAS, there were eight input variables and three output variables. We represented membership functions for each fuzzy input, where the linguistic variables were labeled S-TL, S-TM, S-TR, S-BL, S-BM, S-BR, S-L, and S-R. The input to the fuzzy controller from each fuzzy input variable was defined by three fuzzy sets labeled Near, Medium, and Far. The fuzzy output DC motor variable was decomposed into three



(a) DC motor.



(b) Servo motor.



(c) Alarm.

Fig. 20. Defininitions of the CWAS membership functions for the fuzzy output variables.

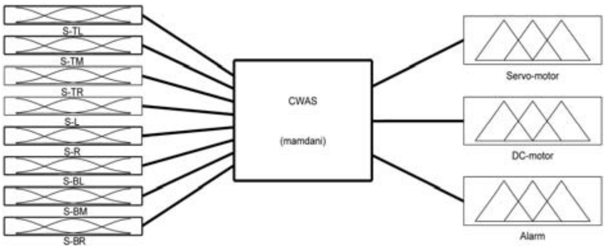


Fig. 21. FIS for the CWAS.

Table 5. Rule base for the CWAS with fuzzy output variables DC-motor and alarm.

S-TL	Near		Medium	Far
S-TM	Near		Medium	Far
S-TR	Near		Medium	Far
S-BL		Near	Medium	Far
S-BM		Near	Medium	Far
S-BR		Near	Medium	Far
DC-motor	Back	Forward	Stop	
Alarm				High

fuzzy sets labeled Back, Stop, and Forward, as shown in Fig. 20(a); similarly, the servo motor was decomposed into four fuzzy sets labeled Left, Left-half, Right-half, and Right, as shown in Fig. 20(b). The fuzzy output variable alarm was decomposed into two fuzzy sets labeled High and Low, as shown in Fig. 20(c). For the CWAS, the FIS is depicted in Fig. 21.

Fuzzy rules containing the OR operator were accumulated for the CWAS, as well as a collection of IF-THEN statements. For the fuzzy output variables DC motor, alarm, and servo motor, nine fuzzy rules were conducted as shown in Tables 5 and 6. The rule base from the eight fuzzy input and three fuzzy output variables for the CWAS is shown in Fig. 22. The control surface for the CWAS is shown in Fig. 23.

Table 6. Rule base for the CWAS with the fuzzy ouput variables servo-motor and alarm.

S-L	Near		Medium		Far
S-R		Near		Medium	Far
Servo-motor	Right-half	Left-half	Right	Left	
Alarm					Slow

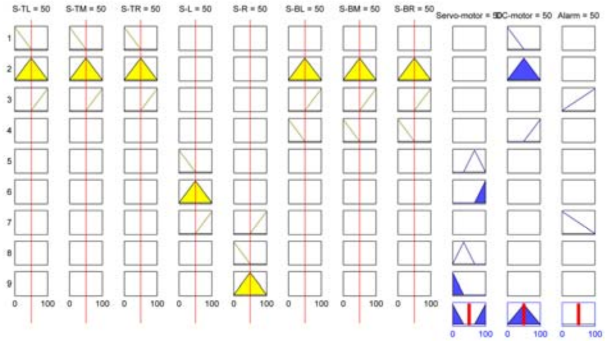
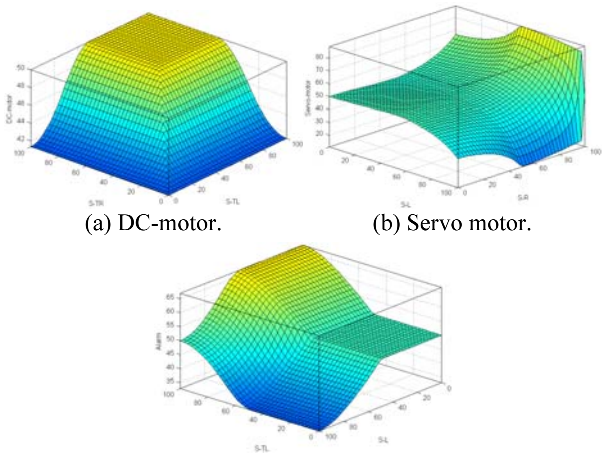


Fig. 22. Rule base for CWAS.



(a) DC-motor. (b) Servo motor.

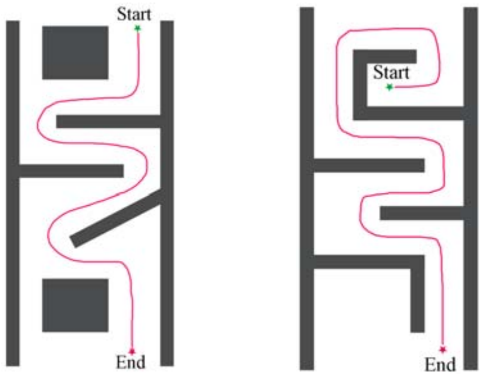
(c) Alarm.

Fig. 23. Control surface for the CWAS. Output fuzzy variables.

8. EXPERIMENTS AND RESULTS

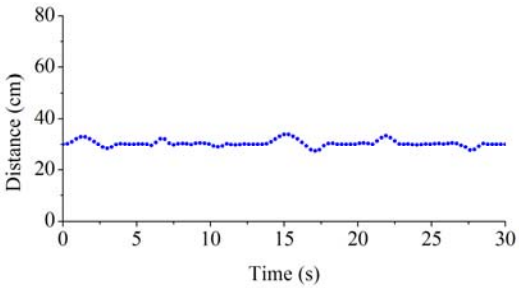
The use of multiple sensors on the mobile robot increases its safety level. Therefore, we used multiple sensors to increase the sensitivity of the mobile robot. All the sensors and other modules were tested, installed, and managed well. Various trials were conducted to validate the proposed system.

Because the robot performed object tracking autonomously, a vehicle was moved in front of the robot. The robot followed the object by maintaining a constant distance from it. The speed of the followed vehicle was also varied, and the robot precisely matched its speed with that of the followed vehicle. The robot was also monitored by changing the motion of the followed vehicle in the forward, backward, left, and right direc-

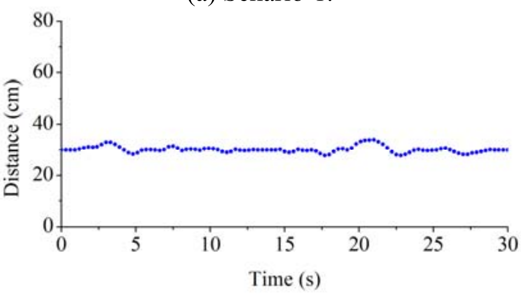


(a) Scenario-1. (b) Scenario-2.

Fig. 24. Object tracking experiments with convex and concave obstacles.



(a) Senario-1.



(b) Senario-2.

Fig. 25. Deviation in terms of maintaining a constant distance from the object during tracking.

tions. After that, the robot was tested via experiments in various obstacle environments (Fig. 8). The robot followed the object by achieving the desired turning angle properly and attained the correct speed (forward or backward) very smoothly. Furthermore, two tracks were developed to test the tracking system, as shown in Fig. 24. In scenario 1, a vehicle was moved on the track and was controlled through a remotely controlled device. The robot followed the vehicle by maintaining a distance of 30cm from it. The deviation in terms of maintaining a constant distance during tracking is shown in Fig. 25(a). It can be seen that the deviation was high at the starting point. In scenario 2, an object was moved independently on the track. The robot followed the object by maintaining a distance of 30cm from it, and the deviation in terms of maintaining a constant distance during tracking is shown in Fig. 25(b). It can be seen that the deviation was high when the object suddenly stopped and started its motion. Thus, the robot can track an object

Fig. 26. CWAS experiment in the dynamic environment with convex and concave obstacles.

that is controlled manually or moved independently. Because the robot was designed to maintain a distance of 30 cm from the object, it successfully achieved the desired constant distance during tracking. Furthermore, we achieved better smoothness during tracking than previous approaches did [17,18,21,22].

We faced some complex scenarios during object tracking. For example, if the detected object had a very high speed, which was beyond the limit, it was considered out of range after a certain interval. Likewise, if the detected object had very high speed in the backward direction, there might be an accident. If multiple objects were detected through various sensors at the same time, the robot was not able to recognize the followed object. The robot followed the object that was in front of it.

In the CWAS, tests were conducted by moving the robot among objects and hurdles via the remotely controlled device. The robot indicated warning alarms, and the respected motion mode was activated in order to avoid collision. In another experiment, objects were moved on various sides of the robot, and the robot met its goal without collision. The robot was also tested via different experiments under different scenarios in order to verify each function properly (Fig. 6). Furthermore, a track was developed to verify the CWAS. Since the robot was manually controlled in the CWAS, it was moved on the track via a remotely controlled device, as shown in Fig. 26. The robot performed the collision warning and avoidance functions automatically. Because the robot did not have prior knowledge of the environment, it turned to the right when the object was detected in the left direction. Similarly, it turned to the left when the object was detected in the right direction. The possibility of collision for the robot during navigation is shown in Fig. 27. When a hurdle was detected in the left or right direction at a near distance, the collision possibility became high in that direction. If multiple objects were detected via various sensors, the robot performed operations one by one sequentially, depending on the algorithm. In some situations, it was difficult for the robot to make decision correctly. Overall, the robot gave acceptable results. These experimental results indicate that the proposed approach is particularly helpful for

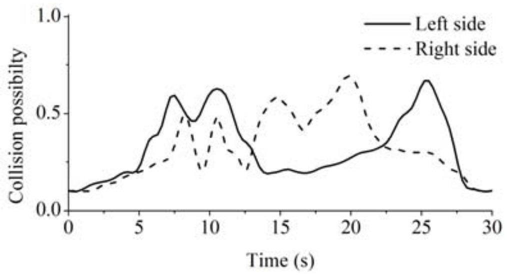


Fig. 27. Collision possibility during navigation with the CWAS.

collision prediction and collision avoidance. Additionally, this approach gives better collision prediction and avoidance than the previous approach did [19,20].

The experiments were conducted by using a binary logic controller and FLC. It was concluded that FLC can handle uncertain data better than a simple binary logic controller can and that it can make decisions in a very natural way. Therefore, the motion of the robot was smoother when using FLC than when using the binary logic controller. Finally, both approaches to the mobile robot were developed, tested, and verified through experiments.

9. CONCLUSIONS

In this paper, we presented an integrated object tracking system and CWAS for a mobile robot. The proposed approach relied on techniques in which the positions of static and moving objects were determined. The system was tested with a mobile robot. It was controlled through instructions by human and through the range sensors. The robot can be used in two ways. First, for tracking an object autonomously by maintaining a constant distance from it. It can also be used for collision prediction, which has a critical importance in terms of alerting the driver before an accident, and collision avoidance in which a safe distance is predefined for each function; thus, both are combined in order to increase the safety level.

In the object tracking system, the robot was autonomously instructed regarding the navigation. We conducted experiments for various scenarios. The robot successfully tracked the object by maintaining a constant distance of 30 cm from the tracked object in various situations.

In the CWAS, the robot performed security functions precisely. It is concluded that collision prediction, achieved via the range measurements, helps to reduce vehicle accidents and that the driver can be notified before an accident via light and sound alarms.

The conventional binary logic controller and the FLC were utilized in order to handle the uncertain data from the sensors. Both controllers performed well for both approaches. Since the FLC can handle uncertainties well, smoothness was achieved in terms of object tracking. It demonstrated precise results in terms of object tracking. However, computational complexity was high in the fuzzy approach.

Future work will focus on creating a well-defined sensor architecture by increasing the sensor precision. In object tracking, the speed variation of the robot will be improved by estimating the speed of the followed vehicle precisely, and a separate control unit will be installed in order to identify the tracked object.

REFERENCES

[1] K. Morita, J. Mashiko, and T. Okada, "A study on delay in braking operation when drivers looking aside from road ahead," *Transactions of Society of Automotive Engineers of Japan*, vol. 33, no. 3, pp. 161-166, July 2002.

[2] J. Kim, S. Hayakawa, T. Suzuki, K. Hayashi, S. Okuma, N. Tsuchida, M. Shimizu, and S. Kido, "Modeling of driver's collision avoidance maneuver based on controller switching model," *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 35, no. 6, pp. 1131-1143, 2005.

[3] H. Malik and A. Rakotonirainy, "The need of intelligent driver training systems for road safety," *Proc. of the 19th International Conference on Systems Engineering*, pp. 183-188, 2008.

[4] Y. Pan, T. Acarman, and U. Ozguner, "Nonstandard safety technology," *Proc. of IEEE Intelligent Transportation Systems*, pp. 1126-1131, 2001.

[5] J. Wang, F. Gao, S. Shi, and J. Wang, "A new method for distance and relative velocity measurement in vehicle collision warning system," *Proc. of the Sixth International Conference on Intelligent Systems Design and Applications*, vol. 1, pp. 1142-1146, 2006.

[6] M. Higuchi and P. Raksincharoensak, "Seat vibrotactile warning interface for forward vehicle collision avoidance," *Proc. of SICE Annual Conference*, pp. 1370-1373, 18-21 August 2010.

[7] Z. S. Lim, S. T. Kwan, and M. G. Joo, "Multi-object identification for mobile robot using ultrasonic sensors," *International Journal of Control, Automation, and Systems*, vol. 10, no. 3, pp. 589-593, 2012.

[8] J. H. Lee, C. Lin, H. Lim, and J. M. Lee, "Sliding mode control for trajectory tracking of mobile robot in the RFID sensor space," *International Journal of Control, Automation, and Systems*, vol. 7, no. 3, pp. 429-435, 2009.

[9] I. Ullah, F. Ullah, and K. Park, "Sensor based robotic navigation and distance control," *Proc. of International Conference on Intelligence and Information Technology*, vol. 2, pp. 59-63, October 2010.

[10] P. Corke, *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*, Springer, Berlin, 2011.

[11] I. Kecskes, Z. Balogh, and P. Odry, "Modeling and fuzzy control of a four-wheeled mobile robot," *Proc. of IEEE 10th Jubilee International Symposium on Intelligent Systems and Informatics*, pp. 205-210, September 2012.

[12] V. Milanés, J. Pérez, J. Godoy, and E. Onieva, "A

fuzzy aid rear-end collision warning/avoidance system," *Expert Systems with Applications*, vol. 39, pp. 9097-9107, 2012.
[13] Y. Wang, E. Wenjuan, D. Tian, G. Lu, G. Yu, and Y. Wang, "Vehicle collision warning system and collision detection algorithm based on vehicle infrastructure integration," *Proc. of the 7th Advanced Forum on Transportation of China*, pp. 216-220, October 2011.
[14] H. Zhu and G. Singh, "A communication protocol for a vehicle collision warning system," *Proc. of IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing*, pp. 636-644, December 2010.
[15] I. Ullah, Q. Ullah, F. Ullah, and S. Shin, "Integrated collision avoidance and tracking system for mobile robot," *Proc. of International Conference on Robotics and Artificial Intelligence*, pp. 68-74, October 2012.
[16] T. Kroger, "Opening the door to new sensor-based robot applications—the reflexes motion libraries," *Proc. of IEEE International Conference on Robotics and Automation*, pp. 1-4, May 2011.
[17] I. Ullah, Q. Ullah, F. Ullah, and S. Shin, "Sensor-based autonomous robot navigation with distance control," *J. Comput. Intell. Electron. Syst.*, vol. 1, no. 2, pp. 161-168, 2012.
[18] I. Ullah, Q. Ullah, F. Ullah, and S. Shin, "Mobile robot navigation with distance control," *Proc. of International Conference on Robotics and Artificial Intelligence*, pp. 61-67, October 2012.
[19] I. Ullah, F. Ullah, Q. Ullah, and S. Shin, "Sensor-Based Robotic Model for Vehicle Accident Avoidance," *J. Comput. Intell. Electron. Syst.*, vol. 1, no. 1, pp. 57-62, 2012.
[20] I. Ullah, F. Ullah, and Q. Ullah, "A sensor based robotic model for vehicle collision reduction," *Proc. of International Conference on Computer Networks and Information Technology*, pp. 251-255, July 2011.
[21] I. Ullah, F. Ullah, Q. Ullah, and S. Shin, "Object following fuzzy controller for a mobile robot," *J. Comput. Intell. Electron. Syst.*, vol. 1, no. 1, pp. 86-90, 2012.
[22] I. Ullah, F. Ullah, and Q. Ullah, "Real-time object following fuzzy controller for a mobile robot," *Proc. of International Conference on Computer Networks and Information Technology*, pp. 251-255, July 2011.
[23] SHARP^{TM}, GP2D12 Datasheet. Available: http://www.sharpsma.com/webfm_send/1203
[24] S. Mathur, T. Jin, N. Kasturirangan, J. Chandrasekaran, W. Xue, M. Gruteser, and W. Trappe, "ParkNet: drive-by sensing of road-side parking statistics," *Proc. of the 8th International Conference on Mobile Systems, Applications, and Services*, pp. 123-136, 2010.
[25] K. P. Valavanis, L. Doitsidis, M. Long, and R. R. Murphy, "A case study of fuzzy-logic-based robot navigation," *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 93-107, September 2006.
[26] N. C. Tsourveloudis, K. P. Valavanis, and T. Hebert, "Autonomous vehicle navigation utilizing electrostatic potential fields and fuzzy logic," *IEEE Trans. on Robotics and Automation*, vol. 17, no. 4, pp. 490-497, August 2001.
[27] S. X. Yang, L. Hao, M. Q.-H. Meng, and P. X. Liu, "An embedded fuzzy controller for a behavior-based mobile robot with guaranteed performance," *IEEE Trans. on Fuzzy Systems*, vol. 12, no. 4, pp. 436-446, 2004.
[28] S. S. Lancaster and M. J. Wierman, "Empirical study of defuzzification," *Proc. of International Conference of the North American Fuzzy Information Processing Society*, pp. 121-126, 2003.
[29] E. Van Broekhoven and B. De Baets, "Only smooth rule bases can generate monotone mamdani–assilian models under center-of-gravity defuzzification," *IEEE Trans. on Fuzzy Systems*, vol. 17, no. 5, pp. 1157-1174, 2009.

Irfan Ullah received his B.S. degree in Electrical (Telecommunication) Engineering from COMSATS Institute of Information Technology, Lahore, Pakistan, in 2011. He is currently working toward an integrated M.S. and Ph.D. degree in Information and Communication Engineering at Myongji University, Yongin, Korea. His research interests include robot navigation, robot control, fuzzy logic, solar concentrated energy, solar concentrators, daylighting systems, solid-state lighting.

Furqan Ullah received the M.S. degree in Mechanical and Automobile Engineering from Myongji University, Yongin, Korea, in 2009. He is currently working toward a Ph.D. degree at Graduate School of Mechanical Engineering, Myongji University, Yongin, Korea. His research interests include computer graphics, computational geometry, real-time 3D measurement systems, 3D scanners, computer vision, image processing, CNC simulations, physics-based modeling, simulation of complex systems, virtual reality, haptic rendering, and human computer interaction.

Qurban Ullah received his B.S. degree in Mathematics and Statistics from University of Punjab, Lahore, Pakistan, in 2003. He received the Master in Information Technology degree from Virtual University of Pakistan, Lahore, Pakistan, in 2009. He is currently working toward an M.S. degree in Computer Science and Information Technology from Virtual University of Pakistan, Lahore, Pakistan. His research interests include mobile robotics, automation system, software development, web development, and operating systems.

Seoyong Shin received his B.S. degree in Control and Instrumentation Engineering from Seoul National University, Seoul, Korea in 1987. He received his M.S. degree in Electrical Engineering from Florida Institute of Technology, Florida, U.S.A. in 1989. He received his Ph.D. degree in Electrical Engineering from Texas A&M University, Texas, U.S.A. in 1992. In 1994 he joined the faculty of the Department of Information and Communication Engineering, Myongji University, where he became a full professor in 2003. His research areas include control and instrumentation, solar daylighting system, and optical communication functional modules.