Int J Adv Manuf Technol (2010) 47:795–807
DOI 10.1007/s00170-009-2017-8

ORIGINAL ARTICLE

Controlling the navigation of automatic guided vehicle (AGV) using integrated fuzzy logic controller with programmable logic controller (IFLPLC)—stage 1

Mehdi Yahyaei · J. E. Jam · R. Hosnavi

Received: 24 September 2008 / Accepted: 11 March 2009 / Published online: 3 April 2009
© Springer-Verlag London Limited 2009

Abstract Material handling in manufacturing systems is becoming easier as the automated machine technology is improved. Nowadays, most of the research aims at increasing the flexibility and improving the performance of the automatic guided vehicle (AGV). In this paper, designed and made AGV in the Industrial Control Laboratory in Royce Lab at the University of Manchester Institute of Science and Technology will be presented. For controlling the navigation of the AGV, a newly developed controller integrated fuzzy logic with programmable logic controller will be used. By using integrated fuzzy logic controller with programmable logic controller (IFLPLC), the flexibility of AGV will be increased and we achieved great advantages, which can be used in future. Since this AGV uses programmable logic controller and fuzzy logic controllers together, then it will be useful for factories which implement flexible manufacturing system (FMS). Online maintenance and sending the commands to other machines from AGV and so on are the advantages that can be used in FMS. The aim of this paper is to propose the practical example of IFLPLC. The designed AGV is able to reach the target. Further researches are required for collision avoidance and selecting the best way in FMS

Keywords Automated guided vehicle (AGV) · Fuzzy logic · Programmable logic controller · Flexible manufacturing system (FMS) · Fuzzy logic controllers (FLC)

M. Yahyaei · J. E. Jam · R. Hosnavi (✉)
Malek-e-Ashtar University,
Isfahan, Iran
e-mail: r_hosnavi@yahoo.com

1 Introduction

Automated guided vehicle (AGV) systems are mainly used for distribution of materials in warehouse environments and movement of material to and from production areas and storage areas in manufacturing facilities. The first AGV application was for transporting groceries in a warehouse [1]. According to statistics in 1989 [2], AGV system installations with respect to their application types were profiled as the following: just-in-time delivery systems (56%), flexible manufacturing (FMS)/flexible assembly transfer system (13%), storage load transfer, non-automated storage and retrieval systems (AS/RS; 12%), AS/RS interface (8%), progressive assembly (7%), mini-load AS/RS interface (1%) and others (3%). Some other applications of AGV systems in non-manufacturing environments include but are not limited to delivering mail, messages and packages in offices and delivering meals and laundry in hospitals

A typical AGV consists of the frame, batteries and electrical system, drive unit, steering, precision stop unit, on-board controller, communication unit, safety system and work platform. So many researches have been done to control the navigation of AGV through the factory.

Basically, a few researchers have worked on controlling the navigation of AGV with a fuzzy logic controller; most researches in this area are in controlling the mobile robot. But mobile robot and AGV are the same in the first perception but by taking a narrower approach one will find that they are different in some properties. Mobile robot is a robot that can explore its surrounding escape from collision with obstacle and be programmed to follow a wall [3]. However, AGV has some limitations. It should be adapted with machines in the factory and it should be able to adjust with pallet in order to pick up and pick down the load and so on. Since all the FMS companies are using programmable logic controller (PLC) for controlling the manufacturing system, controlling the AGV with the PLC gives us so many advantages. In our previous work, integrated fuzzy logic controller with programmable logic controller to increase the flexibility of both fuzzy logic controllers (FLC) and PLC was proposed. Controlling the navigation of AGV using integrated fuzzy logic controller with programmable logic controller (IFLPLC) provides great advantages for factories, which has implemented FMS.

This AGV addresses three objectives. The first one is to remove the guided path for AGV since, in real case, if we employ labours, they would not follow the guided path for handling the material. The second one is to control the navigation of AGV using IFLPLC in order to reach the advantages, which were defined. The third objective is to reduce the cost of the AGV by proposing the new sensing system, which IFLPLC provides.

Using IFLPLC helps us to implement new sensing sensor for AGV. In most of the previous works, in order to detect the object, a ring of ultra-sonic sensor has been used which increases the cost and also it is very difficult to provide inputs for FLC. In this work, an ultra-sonic sensor with an encoder to find the steering angle and distance to the object were implemented. It is worth mentioning that this is the first phase of our AGV. Further research is required, which will be explained in future work in order to meet the real demands of manufacturing environment.

In the first part of this paper, previous work on controlling the AGV using fuzzy logic controller will be reviewed and also different sensing system, which have been used for mobile robot or AGV, will be presented.

2 Previous work

2.1 Literature review on controlling the navigation of AGV using fuzzy logic controller

Senoo et al. [4] proposed the steering control of AGV using fuzzy control. They tested and discussed this controller for its fast response and energy saving in the case of the step change of guide tape. Their designed AGV is tricycle, which followed the guide tape, and the front wheel is controlled by potential metre steering motor. The variables in the membership function in the antecedent part were the position deflection, the displacement of position deflection and the displacement of steering deflection. The variable in the membership function in the consequent part was steering manipulation volume. They found that the fuzzy control is superior to the proportional integral control. By adding the displacement of steering deflection to the fuzzy reasoning, it is possible to save the steering energy for the step response regardless of the travelling speeds.

One of the good researches in navigation of AGV has been performed by Wuwei et al. [5]. They presented the new navigation method for AGV with fuzzy neuro network controller when in the presence of obstacles. Their AGV can avoid the dynamic and static obstacle and reach the target safely and reliably. They utilise a navigation scheme based on searching points on an arc. Their experimental AGV prototype performed successfully. Real-time control experiments verify that the AGV is capable of safe navigation in an experimental environment and follows the reference paths smoothly.

Wu et al. [6] used fuzzy logic control and artificial potential field (APF) for AGV navigation. The APF method is used to calculate the repulsive force between the vehicle and the closest obstacle and the attractive force generated by the goal. Then, the resultant force guides the AGV to its destination. When trap situations occur and the AGV may not reach the goal or may even collide with the obstacles, a fuzzy logic controller is used to modify the direction of the AGV. In their paper based on the angle between the obstacle and the goal and the status of the AGV, a correction angle is generated by the simple fuzzy rules.

Lin and Wang [7] proposed a fuzzy logic controller for collision avoidance for AGV. They combined fuzzy logic with crisp reasoning to guide an AGV to get out of trap since memories of path and crisp sequence flows are handled by non-fuzzy processing. That is, fuzzy logic and crisp reasoning are combined to simulate "human-like" behaviour–decision abilities in the AGV. Their designed AGV was able to avoid collision with unknown obstacle. Their designed AGV can turn its orientation to avoid collision with moving obstacle. They proposed a prediction method, which is first used to check whether a collision between the AGV and a moving obstacle will happen in the future. Then, they used fuzzy rule to guide the AGV without colliding with moving obstacle.

2.2 Literature review on different sensing system for mobile robot or AGV

Macleod et al. [8] developed a position measurement system which consists of a fast rotating laser on board of the AGV and three beacons that are placed along the border of the area to be covered. Every time the laser hits a beacon, a pulse is registered. In this way, the three beacons give three pulses per revolution of the laser. The delay times between these pulses are then used to calculate the X, Y position of the AGV. The additional heading pulse, broadcasted from the AGV to the roadside, allows for measurement of the yaw angle ψ.

Also, Alves and Junior [9] used a step motor to turn the direction of the ultra-sonic sensors, so that each sensor can substitute two or more sensors in mobile robot navigation.

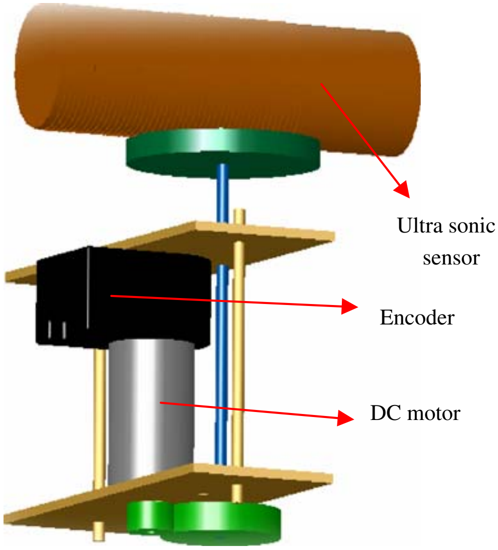


Fig. 1 The sketch of the mechanism for calculating the steering angle and distance

Since a general automatic guided vehicle follows the guided path, its sensing system is not complex because it is limited in recognising the guided path but, since one of the main objectives of this work is to remove the guided path, we should consider the sensing system of the mobile robot.

Perhaps Sugeno [10, 11] has done one of the pioneering researches in mobile robot navigation using fuzzy logic control. The fuzzy control rules, which he defines for the controller, were derived by modelling an expert driving action. He made a computer model of a car in microcomputer to find fuzzy rules. The speed of the designed car was constant; then, the control input to the car is only the angle of the steering angle. He used four variables as input (by modelling a human driver): 1—distance from entrance of corner; 2—distance from inner wall; 3—direction of car; 4—distance from outer wall. The output of the FLC was the movement of the steering handle. He used a supersonic sensor driven by a stepping motor for the measurement of distance and direction.

One of the disadvantages of previously designed mobile robot or AGV is using a great amount of sensors in order to reduce the ambiguity of the data, which leads to increasing

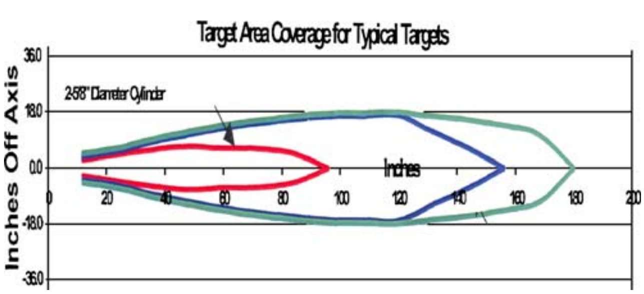


Fig. 2 Cover is by ultra-sonic sensor (www.senix.edu)

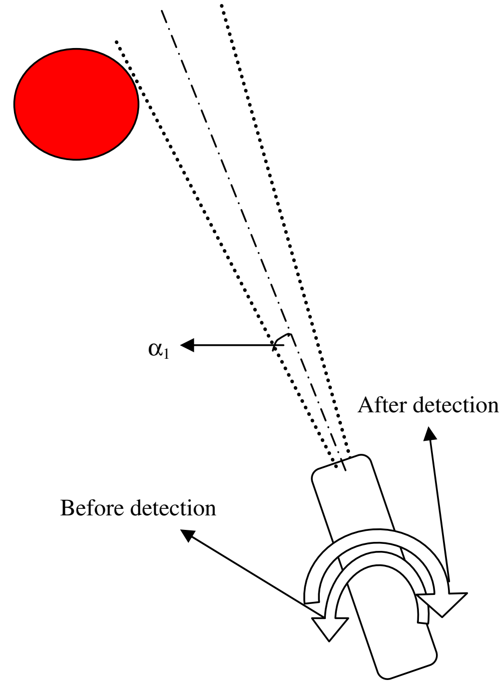


Fig. 3 Detecting the object using ultra-sonic sensor

the cost of the machine. Moreover, most of these designed mobile robot or AGV are not able to recognise the object behind them [12–15]. Therefore, it seems that the work, which has been done by Sugeno [10, 11], still is the most significant work due to its simplicity and reasonable cost. Hence, authors based our work on the Sugeno work but a new simple sensing system will be defined, which can be done in PLC.

3 A sensing system for detecting an object

In this method, in order to cover all the space around the AGV, one ultra-sonic sensor and one encoder were used. The ultra-sonic sensor was installed on a shaft, which is

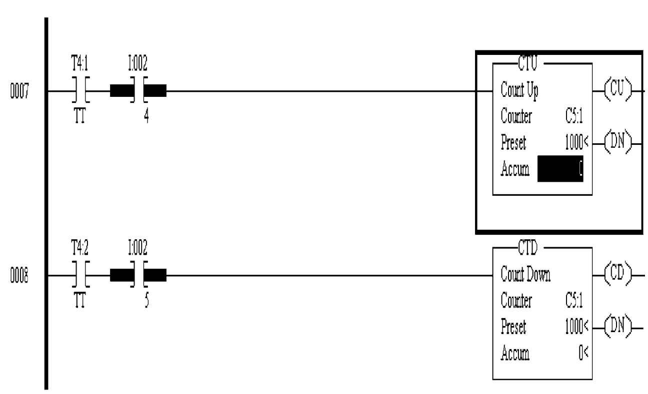


Fig. 4 Ladder diagram for counting the pulse from encoder and finding the steering angle

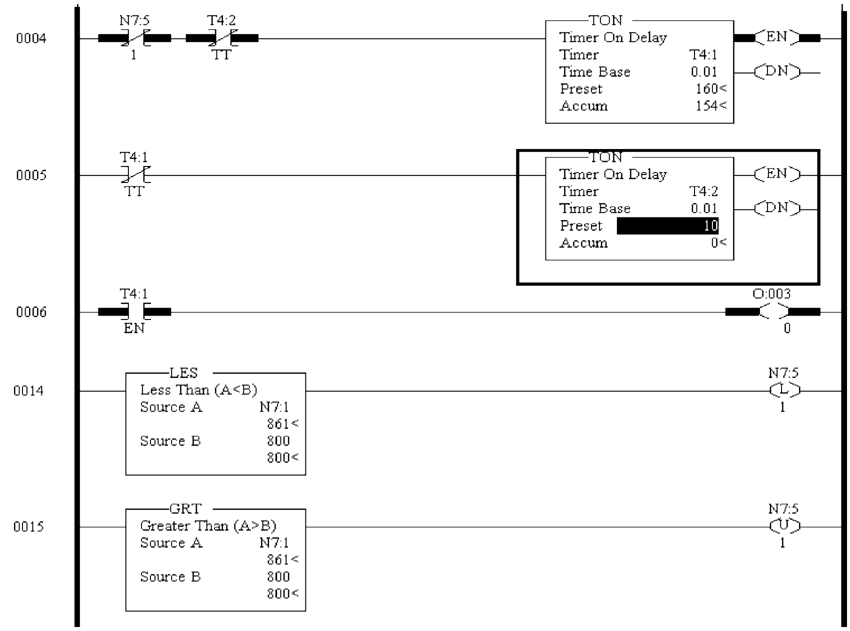


Fig. 5 Ladder diagram for rotation of the sensor

rotated by a direct current (DC) motor. An encoder was mounted on the rear shaft of the DC motor (Fig. 1). Then, when an ultra-sonic sensor detects an object, the encoder will measure the steering angle.

Actually, this simple sensing system has some important problems:

1. The ultra-sonic sensor has a detection angle. If two objects are located in the respective distance on a circular arc within the detection angle of sensor, the ultra-sonic will measure the same distance for both of them.
2. It is not possible to rotate the ultra-sonic sensor to 360° continuously since it is not wireless and the back wires of the sensor will be twisted; then, the sensor cannot rotate continuously.

Since the controller is IFLPLC, these two problems were solved. One of the first designs was using the edge of the detection angle. It means rotating the sensor in clockwise and counterclockwise with a short time interval. If the sensor rotated clockwise when it detected the object, it rotated counterclockwise and vice versa (it was programmed in PLC using ladder diagram). In this case, when the sensor detects the object, it would be on the edge of the detection angle; thereby, if we add the value of the detection angle to the measured rotary angle by encoder, it would give us the accurate steering angle. But this solution would work properly if the detection angle was constant. But Fig. 2 shows that the detection angle is not constant and also it does not follow any mathematical equation. It means that the cross section of the detection angle of sensor is varied with respect to the distance. So the

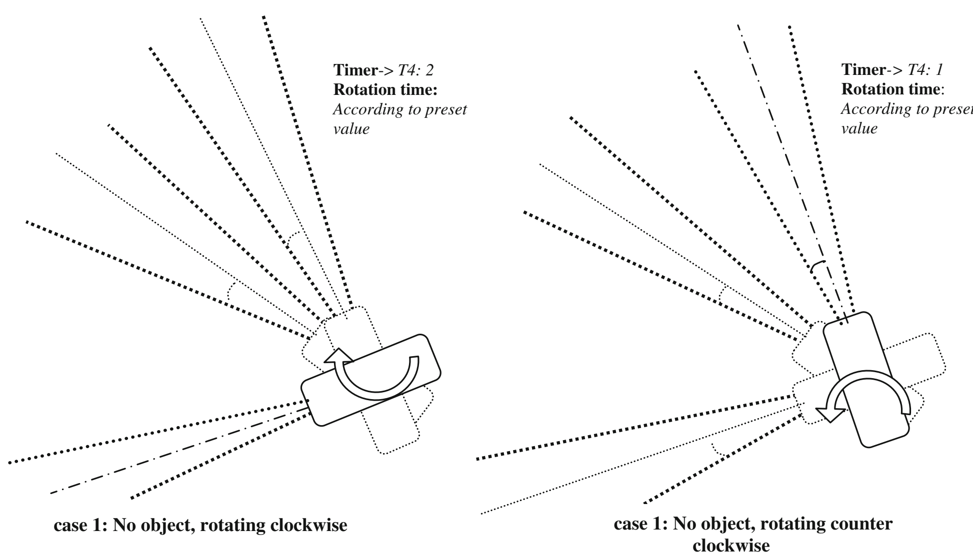


Fig. 6 Case 1: no object, rotating clockwise. Case 1: no object, rotating counterclockwise

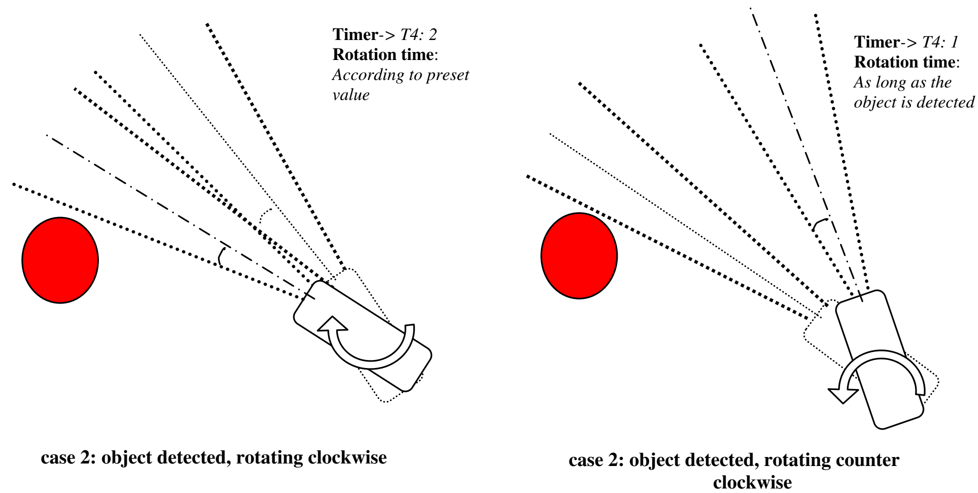


Fig. 7 Case 2: object detected, rotating clockwise. Case 2: object detected, rotating counter clockwise

edge of the detection angle is not reliable for measuring the rotary angle.

In this research, in order to cope with this problem, an ultra-sonic sensor with the small detection angle and also an object which is bigger than the detection angle were used; therefore, the effect of the detection angle will be decreased. Suppose an object is in the left side of the AGV, then in this case the sensor rotates counterclockwise (Fig. 3). As long as it detects the object, it will change its rotation direction to clockwise and simultaneously the PLC will record the analogue signal from the sensor (which is measured distance) and the encoder will send the signal (which is rotary angle). Angle α_1 is varied as mentioned before, but in this work we assume this angle is equal to 12 approximately. Ladder diagram in Fig. 4 shows how the PLC counts the pulses of the encoder

So when the number of the pulses from encoder is saved in PLC, adding 12°c(as an average) to this measured rotary angle will lead to finding the approximate (sometimes accurate) rotary angle. It is worth mentioning that the sensor will rotate counterclockwise again in a short interval to detect the object and update their inputs, which are steering angle and distance to object. If the location of the object is changed (due to the movement of AGV), the sensor will rotate counterclockwise as long as it detects the object. In other words, the time for clockwise rotation is constant and predefined but the time for counterclockwise is varied and it depends on the location of object. The rotary speed is an important concept (approximately 0.5 rounds per second) because if it is high, since the direction of the speed is reversed continuously, then it will cause vibrations, which will affect the accuracy. Also, if it is high, the counter in PLC cannot count the pulses.

Suppose the aim is to move the AGV in 80×80 cm. In rung number 14 (Fig. 5), if the measured distance (saved in N7:1) is less than 800, it means that the sensor has detected the object so that N7:5/1 is latched. If it is greater than 800, it means that measured obstacle is out of range so it should not be considered. So the N7:5/1 should be unlatched. It is worth mentioning that just the target should be in the predefined range; otherwise, we cannot trust that the measured thing is the object or not.

Moreover, if the digital output O:003/0 is fired, the motor will rotate counterclockwise and if it is not fired the motor will rotate clockwise. The fired or unfired digital output O:003/0 is related to time T4:1, which means that if, at times, the O:003/0 is fired, otherwise it is not fired. So the rotation of the motor is totally dependent on timer T4:1.

Explicitly, according to rungs 004 and rungs 005, if N7:5/1 is not fired (No object detected) and timer T4: 2 does not time, then the timer T4:1 will measure the time so the motor will rotate counterclockwise. When the timer T4:1 measures the

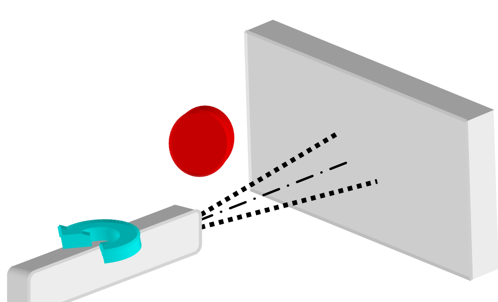


Fig. 8 Detecting the wall or any other obstacle instead of object

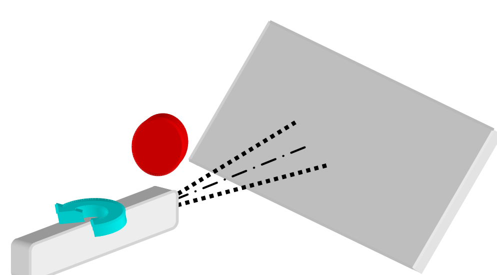


Fig. 9 Just detecting the object

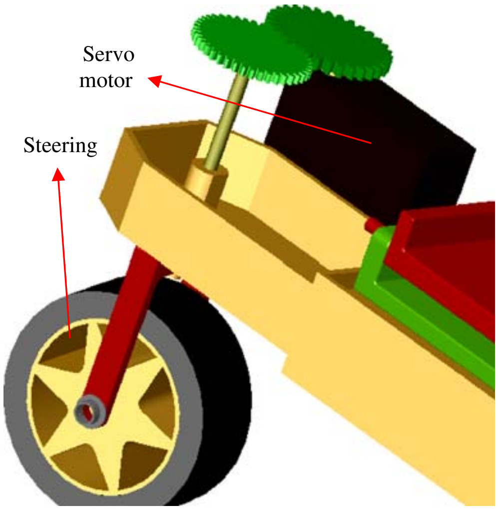


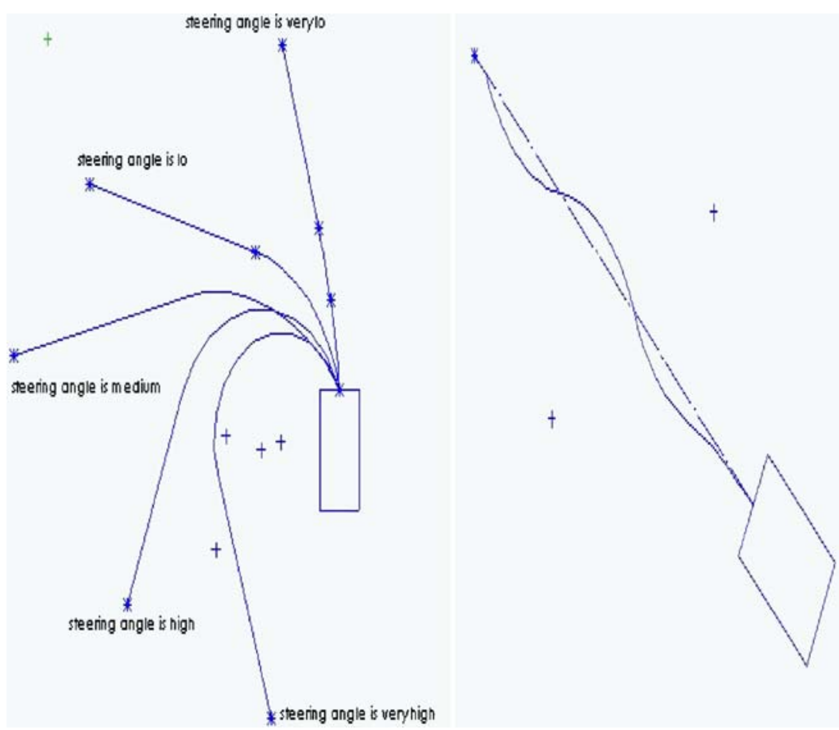
Fig. 10 Controlling the steering wheel using servomotor

predefined preset time value, the timer T4:1 will stop measuring the time so the timer T4:2 will start timing and the timer T4:1 will be turned off. Therefore, the motor will rotate clockwise until the timer T4:2 measures its predefined preset time value. When the motor rotates counterclockwise (timer T4:1 is measuring the time), if the sensor detects the object, the timer will stop and the timer T4:2 starts to time, which means that the motor rotation will be reversed suddenly from counterclockwise to clockwise.

For more explicitness, consider Figs. 6 and 7. Two cases have been proposed in these figures. First is the case when no object is detected, and the second is the case when an object is detected. As it was mentioned, the preset value is very important. The preset value for timer T4:1 should be the required time for rotating the 180° and for timer T4:2 it should be as less as possible. But it should be taken into account that if this value is very small then the counter in PLC cannot count the pulse from the encoder and also it would cause vibration. On the other hand, updating time decreases with decreasing preset time. So the movement will be smoother. So defining the preset value for timer T4: 2 is very significant.

Digital input module of the PLC was used for counting the signal from the encoder. But the digital module can accept the signal between 12 and 30 V; then, the pulses from the encoder cannot be counted by digital input since the magnitude of the pulse is not enough. Actually, TTL module for PLC is available in the market, which is able to count the 5-V signal of the encoder and define the direction. But since this module is expensive, an electronic board was made which amplifies the pulses of the channels A and B from 5 V to more than 15 V; then, the signal is accepted by PLC and the counters in PLC can count the pulses. Ladder diagram in Fig. 4 can count the signals from the encoder and find the steering angle. For instance, suppose the object is in the left side of the AGV, when timer T4:1 times, it means that the sensor rotates counterclockwise so that the counter counts up the signal which comes from the I:002/4. On the other hand, when the timer T4:2 times, it means that the sensor rotates clockwise, so that the counter counts down the signal which comes from

Fig. 11 Different action for fuzzy algorithm



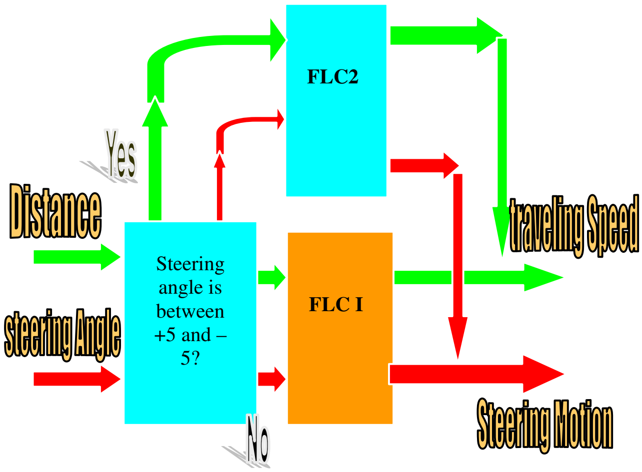


Fig. 12 Defining two actions as fuzzy algorithm

the I:002/5. Then, this angle which comes from the saved pulses in C5:1 should be added to 12°c

By using this sensing system, two problems will be encountered.

One of the problems has been illustrated in Figs. 8 and 9. Let us consider Fig. 8. One can see that, in this case, the sensor will detect the wall instead of the object, so this causes the problem for controller. In order to cope with this problem, a tilt wall was used instead of the perpendicular wall (Fig. 9).

Moreover, if the DC motor is not strong enough, inertia makes a problem. So the DC motor should be powerful enough in order to reverse its direction simultaneously as long as it receives the signal. The motor was coupled to a gearbox in order to reduce the speed of the motor;

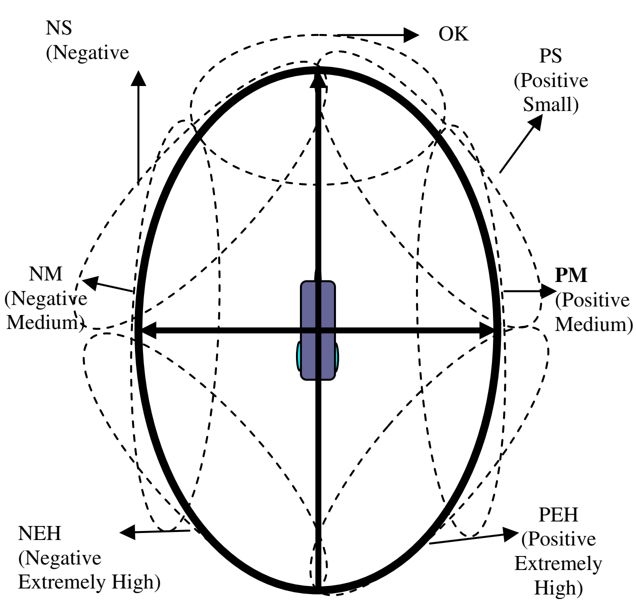


Fig. 13 Different membership function for steering angle variable (FLC1)

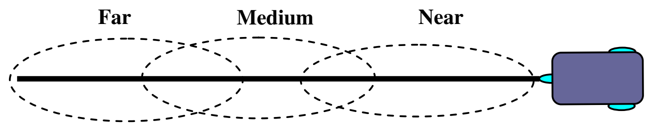


Fig. 14 Different membership functions for distance to the object variable (FLC1 and FLC2)

therefore, its rotation was an appropriate movement which did not have an effect on the machine's precision. The sensor rotation speed was 0.5 rounds per second. If we increased the speed, since the weight of the sensor was high in comparison with AGV body, vibration would be generated and AGV could not move smoothly.

Movement of AGV was provided by a DC motor controlled by pulse width modulation (PWM) method. A PWM circuit works by making a pulsating DC square wave with a variable on-to-off ratio. The average on time may be varied from 0% to 100%. Speed of AGV should be restricted; otherwise, due to the fixed scanning time, the accuracy of controlled system was reduced. To cope with this problem, this DC motor was coupled to gear box in order to reduce the speed.

For turning the front wheel, an electronic board was made to convert the analogue output into *Pulse Code Modulator*, which is acceptable, by servomotor. Basically, most of the servomotors are working in a close-loop control system, but in this work, since the aim is modelling the brain system for controlling the navigation of AGV using fuzzy logic controller, the open-loop control system has been used for servomotor intentionally. The reason is, when a driver is rotating the steering wheel in real car, he/she cannot see the wheels. Hence, in real case, our brain controls the steering angle with open-loop control system. On the whole, no feedback comes from the wheel. Actually, the brain controls the steering angle to the target, which does not depend on the wheel steering angle. Figure 10 illustrates how the servomotor controls the steering wheel.

For controlling the speed of the rear motor, an electronic board was made in order to convert the analogue output from PLC to PWM.

4 Controlling the navigation using IFLPLC

In cases where finding mathematical model of a system is too difficult, then using fuzzy logic controller will solve the problem. For example, for controlling the navigation of a model car, this problem will be encountered. Finding the mathematical model for turning the front wheel is very difficult also as we know that, in real case, the friction coefficient of the ground is not constant. So many other parameters affect the navigation of the AGV.

On the other hand, if a process is complex, it is difficult to design a fuzzy controller even from a control engineer's

1. If (distance is near) and (angle is OK) then (travel is very_low)(steering_motor is Low) (1)
2. If (distance is near) and (angle is NS) then (travel is very_low)(steering_motor is NM) (0.3)
3. If (distance is near) and (angle is NM) then (travel is very_low)(steering_motor is NH) (0.9)
4. If (distance is near) and (angle is NEH) then (travel is very_low)(steering_motor is NH) (1)
5. If (distance is near) and (angle is PS) then (travel is very_low)(steering_motor is PM) (0.3)
6. If (distance is near) and (angle is PM) then (travel is very_low)(steering_motor is PH) (0.9)
7. If (distance is near) and (angle is PEH) then (travel is very_low)(steering_motor is PH) (1)
8. If (distance is medium) and (angle is OK) then (travel is medium)(steering_motor is Low) (1)
9. If (distance is medium) and (angle is PS) then (travel is medium)(steering_motor is PM) (0.3)
10. If (distance is medium) and (angle is PM) then (travel is very_low)(steering_motor is PH) (0.5)
11. If (distance is medium) and (angle is PEH) then (travel is very_low)(steering_motor is PH) (1)
12. If (distance is medium) and (angle is NS) then (travel is medium)(steering_motor is NM) (0.3)
13. If (distance is medium) and (angle is NM) then (travel is very_low)(steering_motor is NH) (0.5)
14. If (distance is medium) and (angle is NEH) then (travel is very_low)(steering_motor is NH) (1)
15. If (distance is far) and (angle is OK) then (travel is very_high)(steering_motor is Low) (1)
16. If (distance is far) and (angle is NS) then (travel is very_high)(steering_motor is NM) (0.3)
17. If (distance is far) and (angle is NM) then (travel is very_low)(steering_motor is NH) (0.5)
18. If (distance is far) and (angle is NEH) then (travel is very_low)(steering_motor is NH) (1)
19. If (distance is far) and (angle is PS) then (travel is very_high)(steering_motor is PM) (0.3)
20. If (distance is far) and (angle is PM) then (travel is medium)(steering_motor is PH) (0.5)
21. If (distance is far) and (angle is PEH) then (travel is very_low)(steering_motor is PH) (1)

Fig. 15 Rule evaluation stage for FLC1

point of view. In such a case, it is very useful to develop a method to drive fuzzy control rules by modelling operator's control action, if there is a good reason to believe that an operator downs an excellent control [11]. So the rules, based on the procedure that happens in our brain, were developed. As it was mentioned before, this fuzzy logic controller is for reaching the target.

The fuzzy algorithm was defined which includes two actions, one for reducing or increasing the steering angle and the second one for going toward the target precisely. Figure 11 assists us to understand.

FLC1 was used to reduce the steering angle; the threshold angle was defined [8] for changing the FLC1 to FLC2, which is for going toward the object. A simple simulation of a AGV in Simulink was made and it was found that since the AGV has vibration which is due to the rotation of sensor then the AGV will not have a straight movement toward the object; therefore, another FLC was defined in order to cope with this problem. The threshold angle was ±5°, which means that if a steering angle falls between +5 and −5 then the controller will change from FLC1 to FLC2. For more explicitness, Fig. 12 was illustrated.

4.1 FLC1 (for reducing the steering angle), FLC2 (for moving toward the object in a straight line)

Mamdani-type membership function for both controllers was selected. FLC1 has two inputs, which are steering angle and distance to the object. The steering angle variable consists of nine membership functions, which cover −180° to +180° (Fig. 13).

Also, the distance to the object is partitioned into three-membership functions as it has been illustrated in Fig. 14.

FLC rule has two outputs, which are travel motion (drive force) and steering motion (turning the front wheel). It is worth mentioning that, since servomotor rotates just about 160°, the universe of discourse for steering motion is between −80° and 80°; also, as it was explained before, an electronic board was made to convert analogue output to PWM so the universe of discourse for travel speed is between 1.2 and 1.8 which is the required analogue output to generate variable speed.

Twenty-one rules were developed based on our experience on driving the AGV to the object, as one can see in Fig. 15.

Figure 16 shows the different shapes of the membership function for FLC1.

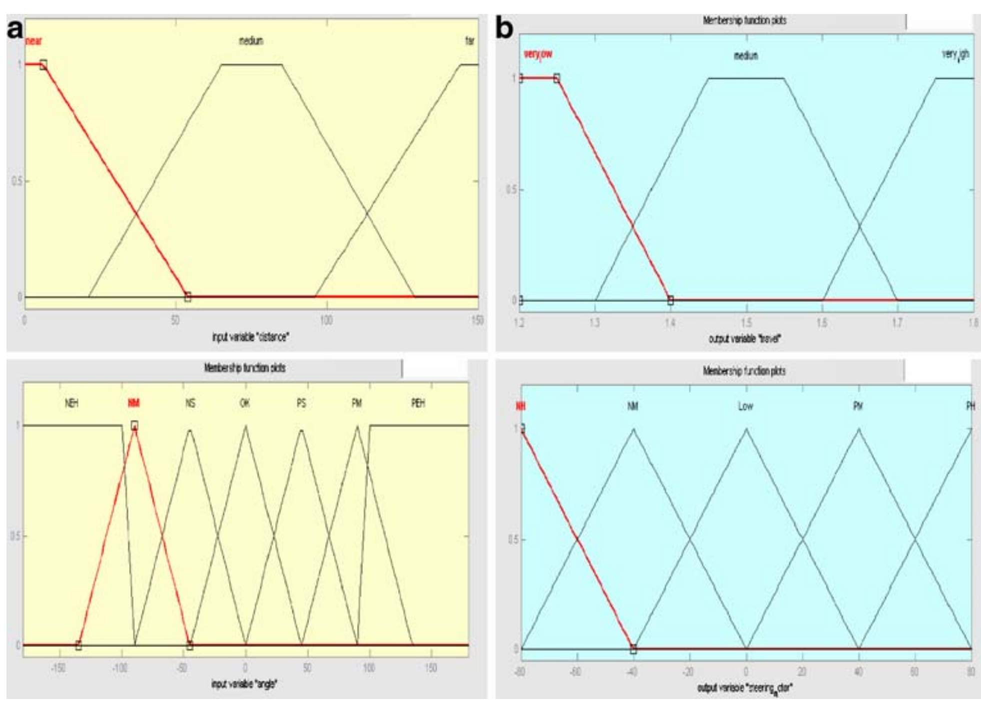


Fig. 16 Membership functions of FLC1 for input (**a**) and output (**b**) variables

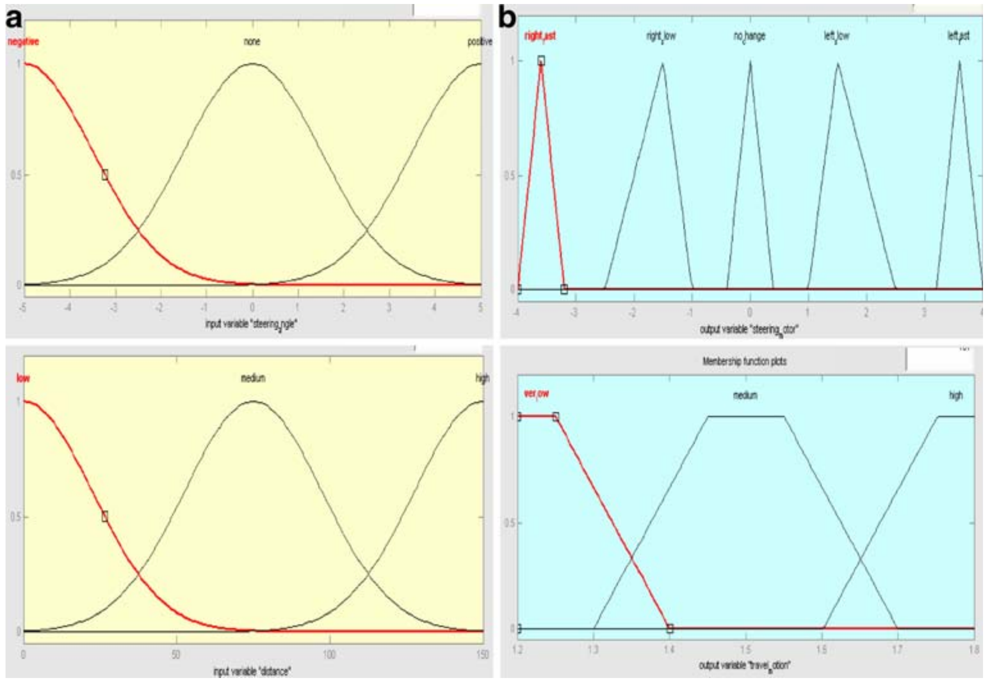


Fig. 17 Membership function for FLC2 for input (**a**) and output (**b**) variables

As long as the steering angle falls between threshold values, then the FLC2 takes control of the AGV. As one can see in Fig. 17, this controller has two inputs and two outputs similar to FLC1. One can see the rules in FLC2 and the shape of the membership function for FLC2 in Figs. 17 and 18.

Figures 19 and 20 show the fuzzy surfaces for FLC1. For more explicitness, different areas of this surface have been labelled.

It is clear that this surface should be symmetric with the zero steering angle. One can see that steering angle in FLC1 does not depend on distance's value. In area A, since the angle is more than 90°, the front wheel should rotate completely.

The surface of drive force is more complex than steering motion. This surface is also symmetric with the zero steering angle.

When the angle is extremely high, the drive force should be as less as possible (Fig. 20 area A); otherwise, the AGV will move in straight line due to the lesser friction between the wheel and the ground. The reason will be explained in the results part.

1. If (distance is near) and (steering_angle is none) then (steering_motor is no_change)(travel_motion is ver_low) (1)
2. If (distance is near) and (steering_angle is negative) then (steering_motor is right_fast)(travel_motion is ver_low) (1)
3. If (distance is medium) and (steering_angle is negative) then (steering_motor is right_fast)(travel_motion is medium) (0.5)
4. If (distance is far) and (steering_angle is negative) then (steering_motor is right_slow)(travel_motion is medium) (1)
5. If (distance is near) and (steering_angle is positive) then (steering_motor is left_fast)(travel_motion is ver_low) (1)
6. If (distance is medium) and (steering_angle is positive) then (steering_motor is left_fast)(travel_motion is medium) (0.5)
7. If (distance is far) and (steering_angle is positive) then (steering_motor is left_slow)(travel_motion is medium) (1)
8. If (distance is medium) and (steering_angle is none) then (steering_motor is no_change)(travel_motion is medium) (1)
9. If (distance is far) and (steering_angle is none) then (steering_motor is no_change)(travel_motion is high) (1)

Fig. 18 Rule evaluation stage for FLC1

Area B is a circumstance in which AGV will move with highest speed since the steering angle is okay and the object is far from the current location of AGV.

In area C, since the AGV is near the object, then speed is less. Actually, travel speed decreases with decrease in the distance to the object.

Figures 21 and 22 show the fuzzy surfaces for FLC2. Fuzzy surface for steering motor as expected is symmetric with respect to the zero steering angle.

In portion “A” (Fig. 21), since the steering angle is acceptable (zero), then the steering motion does not need any change regardless of the distance to the object. In area “B,” since the distance to the object is not enough, the front wheel should turn with the maximum possible value in

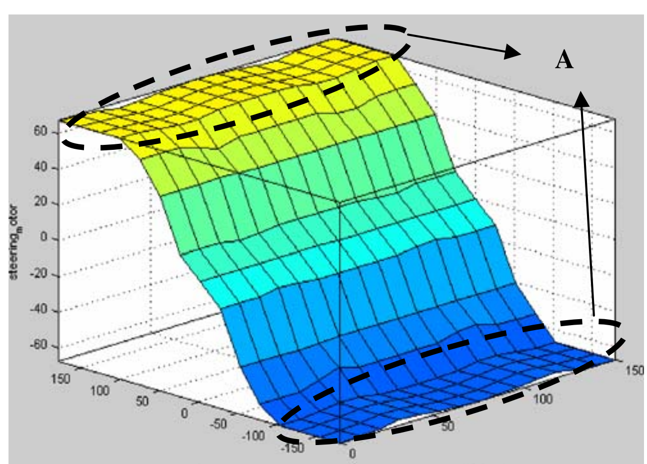


Fig. 19 Fuzzy surface for steering motion in FLC1

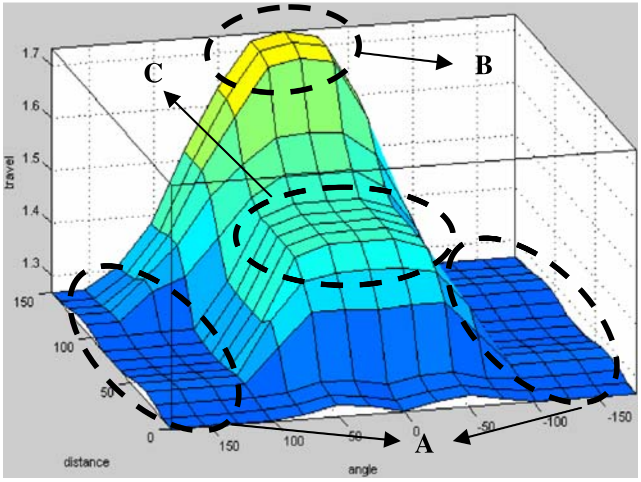


Fig. 20 Fuzzy surface for drive force in FLC1

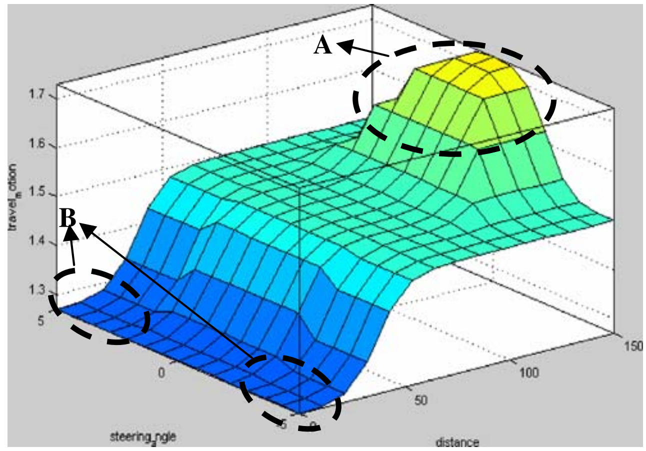


Fig. 22 Fuzzy surface for travel motion (drive force) in FLC2

order to change the direction of the AGV toward the object. In area “C,” since the AGV is far from the object, the front wheel does not need to turn with the highest value (+5 or −5) since with less turning angle the AGV has enough time to reach the object with the proper direction. It is worth mentioning that the MFs in fuzzy output for steering motion in FLC2 are discrete (Fig. 17) because, due to our experience, it seems that those actions are discrete actions.

According to Fig. 22 in area “A,” since the steering angle is okay and the distance to the object is far, then the AGV can move with the highest speed. The speed decreases with decreasing the distance to the object, when the steering angle is okay. When the steering angle is high (between +5 and −5), then the speed should be as less as possible (area “B”) in order to let the front wheel turn with its highest value to change the direction of AGV toward the object.

Moreover, two options are available to change the FLC1 to FLC2 and vice versa. First, with the help of MATLAB and the second option is with the help of PLC. Second option is recommended due to better response time. When the steering angle is calculated in PLC, then it will send correspondence to the FLC according to its value.

Figure 23 assists us to understand clearly the procedure for navigation of AGV.

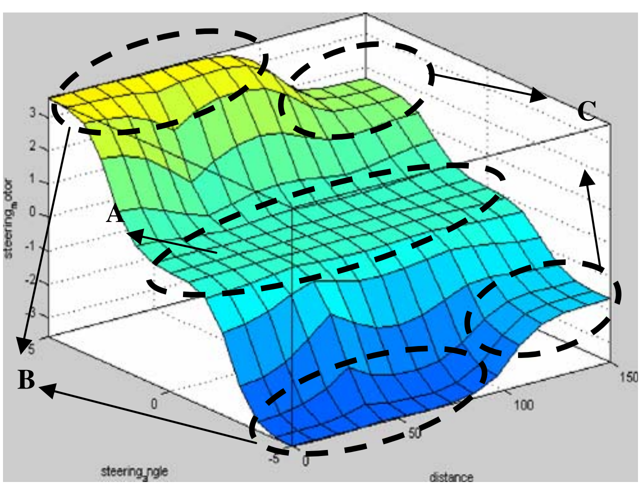


Fig. 21 Fuzzy surface for steering motion in FLC2

5 Future work

The following ideas are suggested for future research:

1. One of the most interesting ideas for future work in this area is selecting a suitable way through the factory using the analytical hierarchy process (AHP).

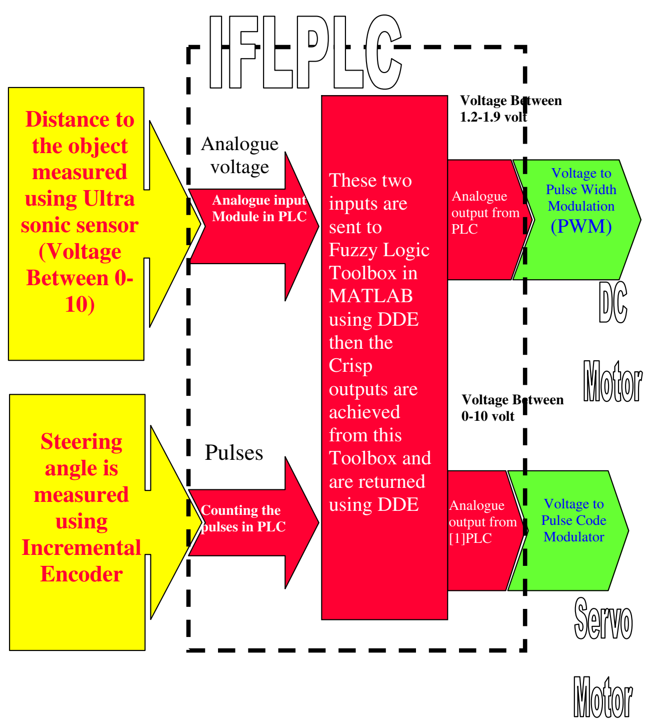


Fig. 23 Controlling the navigation using IFLPLC

Suppose an automatic guided vehicle, which is controlled by IFLPLC, wants to carry a component to machine 1. But, as one can see in Fig. 24, it faces four routs with different conditions. Since the controller is IFLPLC, the AGV can be informed about the situation of other lines and machines. For instance, although route "a" is the shortest way, it is currently occupied. So the priority of different way should be considered with respect to different criteria such as lead time, machining time of components, safety and so on. The AGV can find a suitable way with respect to these criteria using AHP.

2. One of the main drawbacks of this AGV, which was mentioned before, is wiring system. Due to the limited budget, making this AGV with wireless system was not possible. By using wireless, some other advantages will be added (Fig. 25):

    The ladder programme will be simple.
    Response time will be improved.
    Computation time will be reduced.
    There will be no limitation for movement of AGV and ultra-sonic sensor.

3. Using AGV for preventive maintenance is an interesting idea for future work. Considering this idea, this automatic guided vehicle, which is controlled by IFLPLC, can be used to read the information from each machine and send it to the PLC for preventive maintenance (PM). Since the AGV is moving through the factory, there is no need for any labour to go and read the information of each machine. This system is much more reliable compared to the convenient type of PM (Fig. 26).

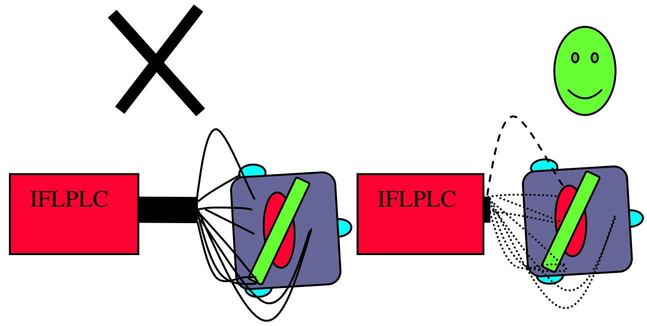


Fig. 25 Using wireless

4. This paper proposed the first stage in controlling the navigation of AGV through the factory. Further researches are required for avoiding collision, finding object which is not in the range of ultra-sonic sensor and giving the address to the AGV. Moreover, current designed AGV cannot answer the high requirements of real manufacturing environment due to the lack of scanning time and also delay in responding. However, this is stage one of this project and solving these problems will be our future work.
5. Using adaptive neuro-fuzzy inference system (ANFIS) for controlling the navigation of AGV is one of our future works. Since the controller of AGV is IFLPLC, then it is possible to move the AGV through the factory manually for the first time and then the information will be recorded in MATLAB and will be sent to the ANFIS in order to generate the appropriate fuzzy logic controller based on the new data.

So in this case the AGV is completely adaptable.

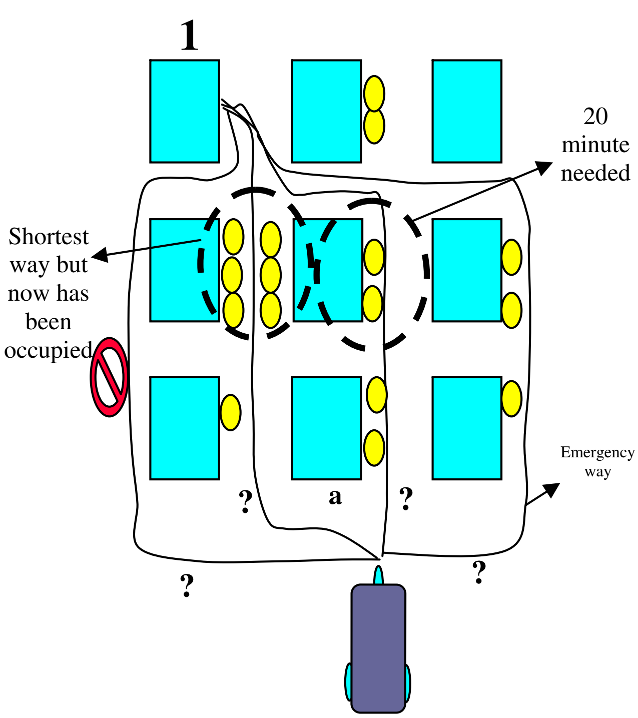


Fig. 24 Selecting suitable way through the factory using AHP

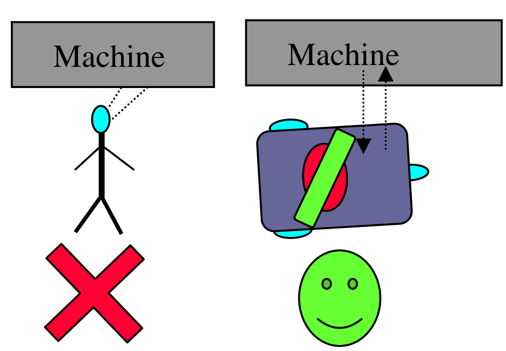


Fig. 26 Doing PM

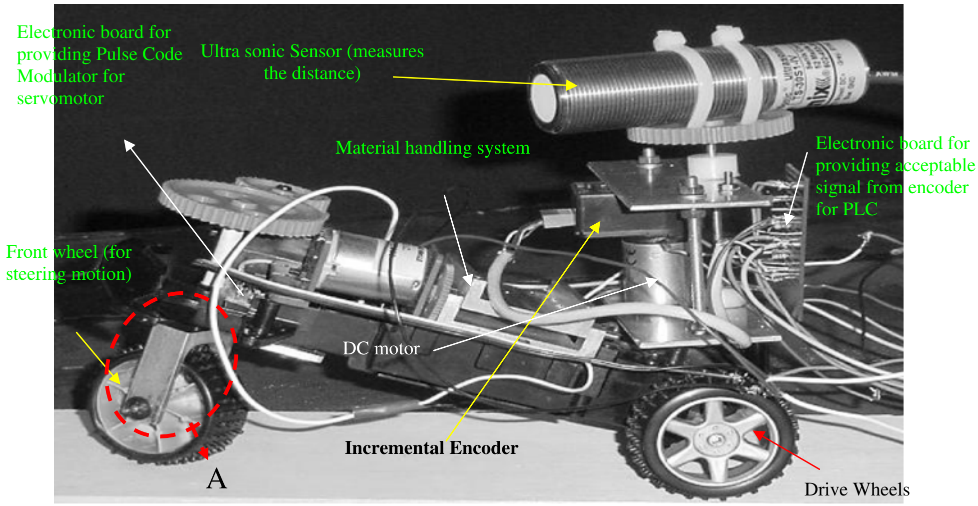


Fig. 27 An automatic guided vehicle (UMIST–Royce Lab–Industrial Control Laboratory)

6 Conclusion

Controlling the navigation of AGV through the factory is a practical example of IFLPLC. By implementing this controller, the AGV can do more work such as reading information from machine, doing online maintenance and so on. This AGV (Figs. 26 and 27) does not need any guided path. The reason for removing the guided path is to let AGV to move through the factory same as the labour. Using the new sensing system (instead of using a ring of sensors) will reduce the cost dramatically. Actually, due to the limited budget, the wireless sensing system could not be used; otherwise, the PLC programme for sensing system would be very simple.

Appendix

Integrated Fuzzy Logic Controller with Programmable Logic Controller

IFLPLC is a newly developed controller by authors in their previous work in order to increase the flexibility and improve the ability of both the PLC and FLC. In our previous work, we proposed two models for this integration and compared these two model using AHP. Since in this case (navigation of AGV) flexibility is the most important criteria and also the MF is complex, then model 2 was selected. In model 2, all the mathematical procedures are performed in MATLAB fuzzy logic toolbox. The only things that we should save in flags in PLC are crisp inputs. It means that if the PLC receives analogue signals from the sensors then it will save these analogue inputs as the integer values in the specific flags. Then, these flags are linked to crisp input variables for FLC in MATLAB using dynamic data exchange (DDE; www.mathworks.com, Mathworks Inc.). So as long as the crisp inputs, which have been saved in flags, are changed, MATLAB runs FLC to calculate and update the values with respect to the new inputs.

The crisps outputs (achieved from fuzzy logic toolbox in MATLAB) are linked to the specific flags in PLC; then, they will be transferred to the analogue output module (and they will be sent to the actuators).

This procedure should be programmed in *m* file in MATLAB. Moreover, the operator should first execute this programme.

Dynamic data exchange

MATLAB provides functions that enable MATLAB to access other Windows applications and for other Windows applications to access MATLAB in a wide range of contexts. These functions use DDE, software that allows Microsoft Windows applications to communicate with each other by exchanging data.

Applications communicate with each other by establishing a DDE *conversation*. The application that initiates the conversation is called the *client*. The application that responds to the client application is called the *server*.

When a client application initiates a DDE conversation, it must identify two DDE parameters that are defined by the server: the name of the application it intends to have the conversation with, called the *service name*, and the subject of the conversation, called the *topic*. When a server application receives a request for a conversation involving a supported topic, it acknowledges the request, establishing a DDE conversation. The combination of a service and a topic identifies a conversation uniquely. The service or topic cannot be changed for the duration of the conversation, although the service can maintain more than one conversation. During a DDE conversation, the client and server applications exchange data concerning items. An *item* is a reference to data that are meaningful to both applications in a conversation. Either application can change the item during a conversation.

References

1. Hammond G (1986) AGVs at works. IFS, UK
2. Gould L (1990) AGVs in America: an inside look. Mod Mater Handling 45(10):56–60
3. Jones JL, Flynn AM, Seiger BA (1999) Mobile robot inspiration to implementation, 2nd edn. AK Peters, Wellesley
4. Senoo S, Mino M, Funabiki S (1992) Steering control of automated guided vehicle for steering energy saving by fuzzy reasoning. In: Industry Applications Society Annual Meeting, Conference Record of the 1992 IEEE
5. Wuwei C, Mills K, Wenwu S (2004) A new navigation method for an automatic guided vehicle. J Robot Syst 21(3):129–139. doi:10.1002/rob.20004
6. Wu KH, Chen CH, Ko JM (1999) Path planning and prototype design of an AGV. J Math Comput Model 30:147–167. doi:10.1016/S0895-7177(99)00171-5
7. Lin C-H, Wang L-L (1997) Intelligent collision avoidance by fuzzy logic control. Robot Auton Syst 20:61–83. doi:10.1016/S0921-8890(96)00051-6
8. Macleod et al (1993) Navigation and control breakthrough for automated mobility. In: Proceedings of the SPIE mobile robotics VIII, Boston, MA
9. Alves and Junior (2002) Mobile ultra sonic sensing in mobile robot. In: IECON 02 [28th Annual Conference of Industrial Electronics Society
10. Sugeno M, Nishida M (1985) Fuzzy control of model car. Fuzzy Sets Syst 16:103–113. doi:10.1016/S0165-0114(85)80011-7
11. Sugeno M et al (1989) Fuzzy algorithmic control of a model car by oral instruction. Fuzzy Sets Syst 32:207–219. doi:10.1016/0165-0114(89)90255-8
12. Ramirez-Serrano A, Boumedine M (1996) Ultra sonic sensing and fuzzy logic control for navigation in unknown static environment. In: Proceedings of EUROBOT'96
13. Parasuraman S, Ganapathy V, Shririnzadeh B (2004) Mobile robot navigation in an unstructured and unknown Environment using AI Techniques. In: 7th Biennial ASME Conference Engineering Systems Design and Analysis, The University of Manchester, Manchester
14. Abdessemed F, Benmahammed K, Monacelli E (2004) A fuzzy-based reactive controller for a non-holonomic mobile robot. Robot Auton Syst 47:31–46. doi:10.1016/j.robot.2004.02.006
15. Er MJ, Tan TP, Loh SY (2004) Control of a mobile robot using generalized dynamic fuzzy neural networks. Microprocessors and Microsystems 28:491–498