Fuzzy Logic Control for an Automated Guided Vehicle

Ming Cao and Ernest Hall
Center for Robotics Research
University of Cincinnati
Cincinnati, OH 45221

ABSTRACT

This paper describes the use of fuzzy logic control for the high level control systems of a mobile robot. The advantages of the fuzzy logic system are that multiple types of input such as that from vision and sonar sensors as well as stored map information can be used to guide the robot. Sensor fusion can be accomplished between real time sensed information and stored information in a manner similar to a human decision maker. Vision guidance is accomplished with a CCD camera with a zoom lens. The data is collected through a commercial tracking device, communicating to the computer the X,Y coordinates of a lane marker. Testing of these systems yielded positive results by showing that at five miles per hour, the vehicle can follow a line and avoid obstacles. The obstacle detection uses information from Polaroid sonar detection system. The motor control system uses a programmable Galil motion control system. This design, in its modularity, creates a portable autonomous controller that could be used for any mobile vehicle with only minor adaptations.

Keywords: Fuzzy logic, vision guidance, mobile robots, motion control

1. Introduction:

Automated Guided Vehicle (AGV) is an intelligent machine that has ‘intelligence’ to determine its motion status according to the environment conditions. For an AGV to operate it must sense its environment, be able to plan its operations and then act based on this plan. This type of system must be placed in a known space from which it can determine its orientation via the use of markers (e.g. white lines, walls, obstacles, etc.). The model of the AGV itself may be easy to establish, however, the environmental model is difficult to obtain.

The running environment could be varied, such as the path orientation, road flatness, obstacle position, road surface friction, etc. There are a great many uncertainties of what conditions will emerge during its operation. Thus a new control method other than the conventional control method is demanded to manage the response of the whole system.

In the last years, fuzzy logic has been applied mobile robot and autonomous vehicle control significantly. The best arguments supporting fuzzy control are the ability to cope with imprecise information in heuristic rule based knowledge and sensor measurements.

Fuzzy logic can help design robust individual behavior units. Fuzzy logic controllers incorporate heuristic control knowledge. It is a convenient choice when a precise linear model of the system to be controlled cannot be easily found. Another advantage of fuzzy logic control is to use fuzzy logic for representing uncertainties, such as vagueness or imprecision which can not be solved by probability theory. Also, fuzzy logic offers greater flexibility to users, among which, we can chose the one that best fits the type of combination to be performed.

The University of Cincinnati Robotics Research Center has been working on improving the ability of the automated guided vehicles for several years. Based on previous experiences, a new AGV ---- Bearcat II is being built. This new robot features more agility, smaller size, digital control, high reliability, more intelligence, etc. Bearcat-II is a computer controlled mobile robot. It is designed to navigate in a changing

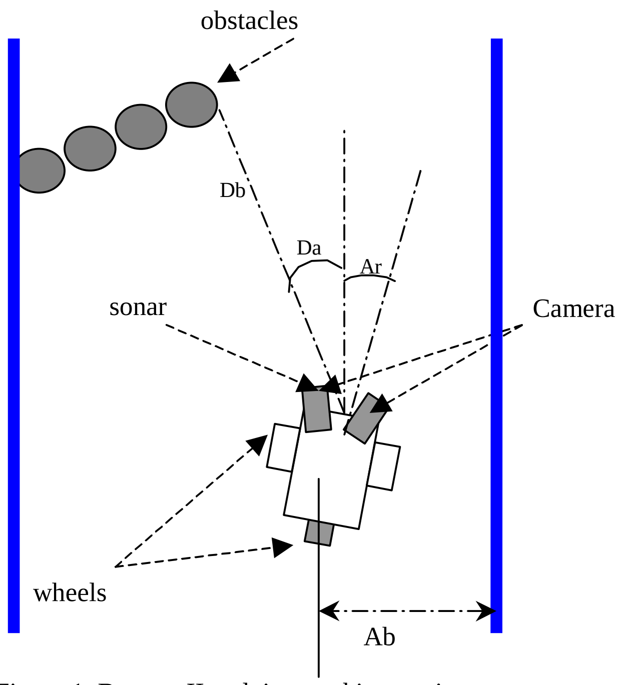

Figure 1. Bearcat II and its working environment

environment as shown in Figure 1. Generally, the road for the robot to travel is marked with white lines. The road direction may be changing while the width of the road remains stable. There will be obstacles blocking the road. It is expected that the mobile robot can pass through these obstacles without knocking them down. In short, the robot should be able to follow the changing path and avoid obstacles. In order to do this, the robot should have the ability to sense and deal with the issue of uncertainty and incomplete information about the environment in a reasonable time.

Currently three approaches to mobile robot navigation are generally discussed: model-based approaches, sensor-based approaches, and hybrid approaches. Model-based approaches need an accurate description of the environment to generate an obstacle free path. Techniques for model-based path generation include road mapping cell decomposition, and potential fields. All of these methods can generate a path from an initial point to the end point using a model of the environment. However, it is usually difficult to obtain an accurate model of a road. Sensor-based approaches execute control commands based on sensor data. A promising strategy for sensor-based approaches is the behavioral architecture, which consists of multiple behaviors, each one of which reacts to sensor input based on a particular concern of the navigation controller. Examples of behaviors include goal-attraction, wall/line-following, and obstacle-avoidance. The main advantage of sensor-based approaches is that the robot can travel safely in a changing environment, since it can react immediately to avoid obstacles detected by sensors. The major disadvantage of sensor-based approaches is that the robot may travel without a goal, even if the path to such a goal exists, such as when it losese its track.

The hybrid approaches combines model-based approaches and sensor-based approaches. It first generates a path via a model-based planner, then the path is integrated into the sensor-based controller to navigate the robot. In this way, the robot navigates along the path while avoiding obstacles unknown to the model. This approach is thus able to achieve the optimality of model-based approaches and the reactivity of sensor-based approaches.

This paper brings together much of the key research work on the overall controls of sensors and planning which were inspired during the AGV development. The paper focuses on the development of fuzzy control logic that improves the AGV's performance. Secion 2 includes the description of the design works on control, sensing technologies, sensor management and data-fusion, different styles of path planning suited for off-line or online plans and task planning. Section 3 give the fuzzy logic design. Conclusions and further works are given in Section 4.

2. SYSTEMS DEVELOPMENT

The physical structure of Bearcat II is characterized by digital control, vision guidance, ultrasonic distance sensors and emergency stop. The main components of the robot include: the central CPU, Iscan and two CCD camera as vision sensor, Plariod ultrasonic sonar, two 12V batteries as motor power source, GALIL digital controller and two DC motors. Relationships among component are illustrated in Figure 2.

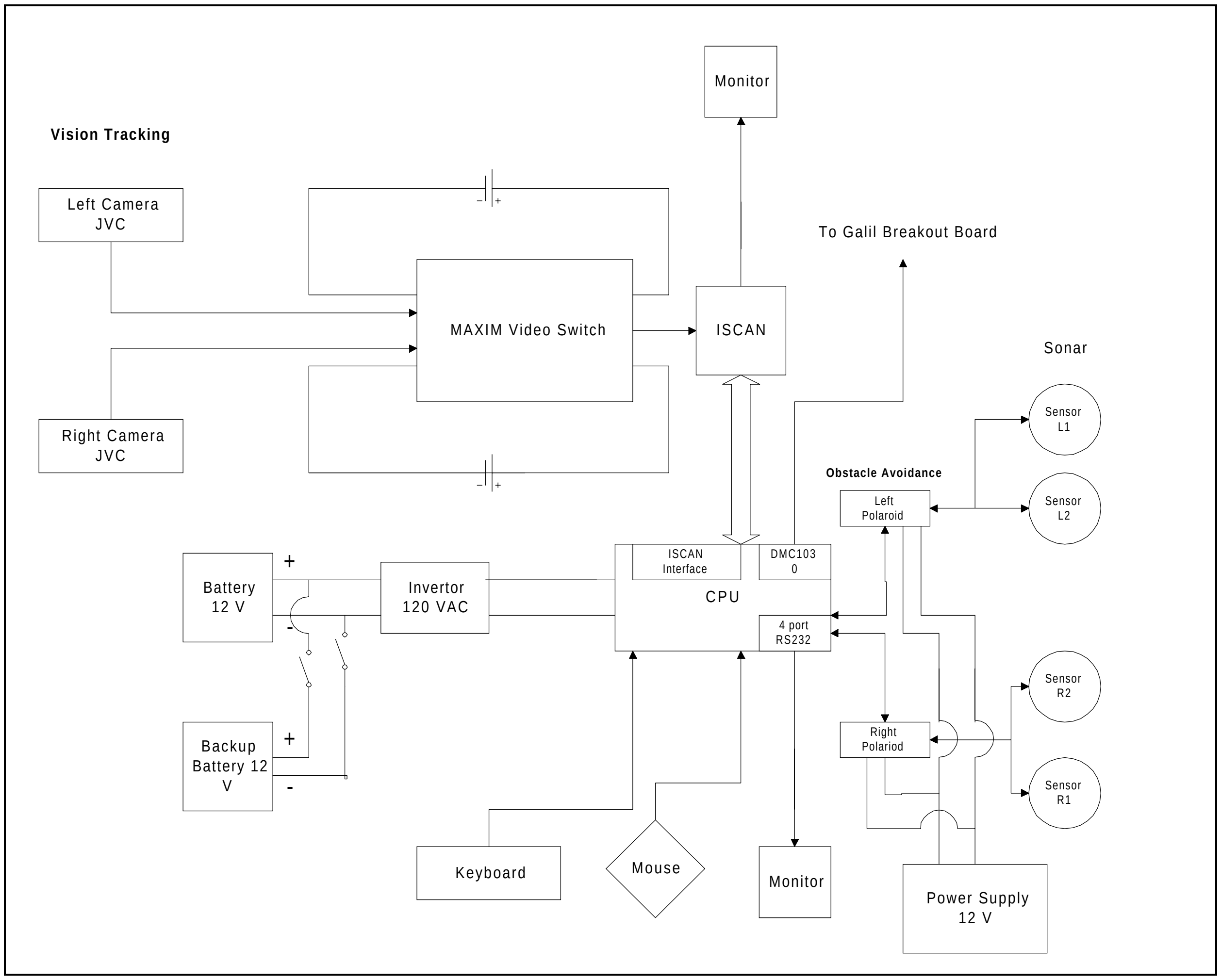

Figure 2. Block diagram of the functional components of Bearcat II

A brief description of their functions are listed below:

2.1 Systems description

Vision guidance system:

The vision guidance system helps the vehicle to know its current position along the road. It needs to be calibrated before its real functioning. The purpose of calibration is to transform the camera coordinates into robot coordinates. Two CCD cameras were used to judge the current position of the robot.

Obstacle Avoidance System: The sonar system reliably detected obstacles between 0.5 meter and 6 meters within an accuracy of 0.02 meter. The system interfaces between the Polaroid unit, Galil controller and the supervisory controller were found to be successful. The fuzzy logic controller computed correction angles. To drive the robot around the obstacles going downslope on a ramp, it was observed that the sonars detected the ground as an obstacle. This problem was taken care of by modifying the sonar algorithm such that if sonars on either sides gave identical readings, the obstacle shall be ignored.

Steering control system: The steering mechanism worked satisfactorily. Implementing the PID controller to control two identical motors on both sides of the robot. The action of turning is achieved with the speed variations of the two motors. The working status (velocity and position) is continuously reported by encoders, built in the motor.

Safety and Emergency Stop Braking System: Asafty switch and a remote safty switch are in serial connection into the motor power cirutuit. When bad situdation happens, one could press the emergency stop button or remote stop button to stop the robot immediately.

3. FUZZY LOGIC DEVELOPMENT OF BEARCAT II

There are generally two approaches in AGV fuzzy logic control:

(1) Using dead reckoning and navigation sensors to track pre-defined path.
(2) Tracking natural landmark features in the environment using external sensors.

The first tracking strategy needs precise information of the travel map as well as an accurate vehicle position status. The controller plans the next vehicle motion based on the current position and its physical constrains.

In the second tracking strategy, the controller gets information directly from external sensors. Sensors are usually used to track features of the environment and provide information directly into the controller. The controller could then use various control methods to control the vehicle reaction.

In this paper, we consider only the second tracking strategy since the Bearcat II often runs in different environment and no position strategy has been built into its control.

During its operation, Bearcat II would take care of two things:

1.It's status on the road, i.e. distance to the border of the road and the angle of its body to the orientation of the road;
2.The distance from the nearest obstacle, and the obstacle's angle relative to the vehicle body.

As previously described, the vehicle using one camera to track the border (white line) of the road. When one camera loses its target (white line), the other camera will be switched on to track the other side of the border. The vehicle tries to keep its position in the middle of the road by keeping its body parallel to the borderline and within a certain distance. In this design, we assumed the right camera could always capture the white border line. The angle error (Ae) defined as the angle between the road border line (its tangent if the line is circular) and the orientation of the vehicle moving direction, while the distance error (De) is defined as the distance from the center of the vehicle to the distance of the border line. When the fuzzy controller found that both Ae and De exists, it will apply the *track following rules* to generate a decision of which direction to go next. The result may be a set of feasible angle range before defuzzification, called the *desired direction*. In this way, the vehicle could run along the track and at the same time, keep its position in the center of the road.

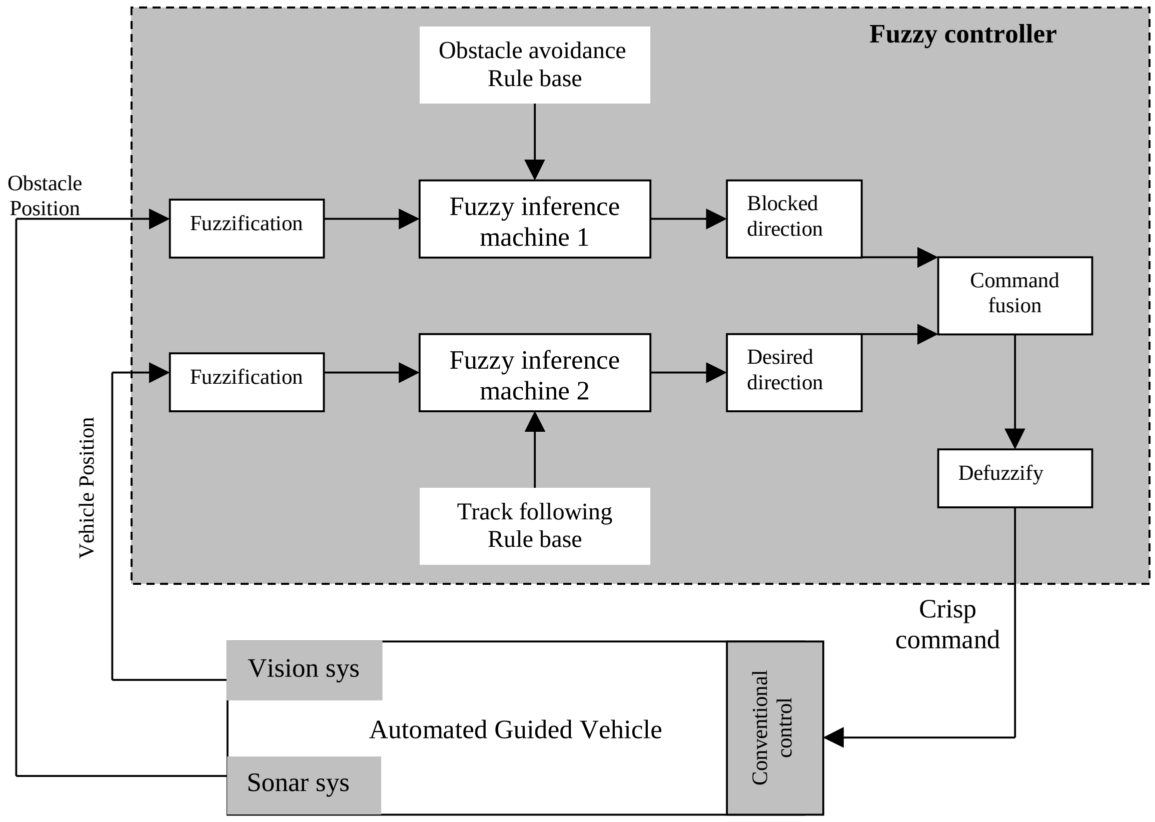

Figure 2. Fuzzy controller functional blocks

However, there are obstacles embushed along the track. They are generally put along one border of the road. The sonar detection range is from 0 meter to 6 meters. When obstacles are found, its position and distance are reported to the fuzzy controller. These data are fuzzified and then the controller applies obstacle avoidance fuzzy rules. The output of this process is the range of directions that is not available to the next motion direction, called *blocked direction*, otherwise, the mobile robot will hit the obstacles.

This may generate problems. What if the desired direction range interplayed with the blocked angle range? How to combine these two decisions to make a crisp command? This is a case where command fusion is needed.

So far we have identified the input variables as:

1. The vehicle distance from the border line Ab_i,
2. The angle of the vehicle option direction to the direction of the road, $Ar_{i\,,}$
3. The distance of the obstacle to the vehicle Db_i
4. The angle of the obstacle to the current direction of the vehicle Dr_i

The output variable is identified as just one: the next motion direction of the vehicle $Aout_i$.

3.1 Fuzzy set definition:

The expected output of the fuzzy controller should be

1. The direction of the next turning (-30 degrees to +30 degrees) and
2. the distance from the nearest front obstacle.

In order to keep things simple, two basic membership functions are used: a triangle membership function and a trapezoidal membership function. The triangle membership function is used to describe common relationships, the trapezoidal membership function is used to emphasize the degree of membership with in a certain range.

The fuzzy sets could be defined as follows:

(a) The current angle corresponding to the road direction (white line direction): Ar

Fuzzy set	Ar_3	Ar_2	Ar_1	Ar	Ar1	Ar2	Ar3
Description	Leftest	Lefter	Left	Middle	Right	Righter	Rightest
Range (degree)	-30~-21	-30~-10	-18~0	-12~12	0~18	10~30	21~30

(b) The distance from the vehicle to the border line (Ab). Suppose the road has a width of 3 meters, and the vision system always use the right camera for line tracing. So, the vehicle should keep its center from the right line for a distance of 1.5 meter. The lefter and righter member function employed trap member function in order to force the vehicle move inside both road border lines.

Fuzzy set name	Ab_2	Ab_1	Ab	Ab1	Ab2
Description	Lefter	Left	Middle	Right	Righter
Range (meters)	2.2~3	1.5~3	1.4~1.6	0~1.5	0~0.8

(c) The angle of the obstacle relative to the vehicle direction. Since the next moving direction of the vehicle (output of the controller) ranges from -30 ~30 degrees, only obstacles within this range would affect further motion.(Dr)

Fuzzy set name	Dr_2	Dr_1	Dr	Dr1	Dr2
Description	Lefter	Left	Middle	Right	Righter
Range (Degree)	-30 ~-20	-30 ~ 0	-10 ~10	0 ~30	20~ 30

(d) The sonar sensor could detect the obstacles at the maximum of 6 meters. So the fuzzy set for the distance from the obstacle is (Db):

Fuzzy set name	Db1	Db2	Db3
Description	Near	Medium	Far
Range (Meters)	0~3	1~5	3~6

(e) The output angle (Aout) should range from [-30, 30] in degrees.

Fuzzy set	Aout_3	Aout_2	Aout_1	Aout	Aout1	Aout2	Aout3
Description	Leftest	Lefter	Left	Middle	Right	Righter	Rightest
Range (degree)	-30~-21	-30~-10	-18~0	-12~12	0~18	10~30	21~30

A block diagram of such a fuzzy controller is shown in Figure 3. All fuzzy sets are shown in Figure 4 and Figure 5.

3.2 Fuzzy rules:

There are two kinds of behaviors for the robot to make: 1.Track following and 2. Obstacle avoidance.

1. Rules for the track following:
   These rules take into account the value Ar and Ab. These two variables are treated as inputs into the controller. Only track following rules would involve in this fuzzy inference process. The output of this set is pre-defuzzified angle range that are available for the robot to move ($\mu_A(\mathbf{x})$). The general rule for this behavior is that: if Ar is large, turn large angle to the other side. If Ab is large, turn to the direction that could make it smaller. Fuzzy rules try to balance between them.

2. Rules for obstacle avoidance:
   This process needs Dr and Db as inputs.Only obstacle avoidance rules will be applied to this fuzzy inference process. The output of the process is also a range of pre-defuzzified angle range membership ($\mu_D(\mathbf{x})$) that are available. The general rule for obstacle avoidance behavior is: if Dr is large (obstacle is far), turn smaller angle. If Db is large, turn to the side that are empty.

Detailed fuzzy rules could be input into the simulink. Suppose Ae is the angle errorness, De is the distance errorness and Aout is the desired output angle. Some sample rules are written in *AppendixI*.

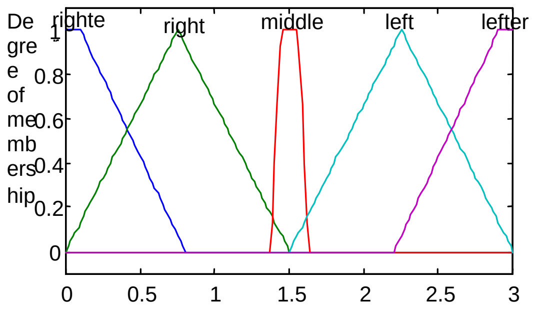

Figure 4. Input membership graphics
(1) upper left: angle input (2) upper right: distance tot the left border of the road
(3) bottom left: distance to the front obstacle (4) bottom right: angle to the front

Figure 5. Output angle membership graphics

3.3 Command fusion

As previously described, both Fuzzy Inference Rule Base(FIRB) would generate their own output sets. How to merge them into a reasonable fuzzy set?

Since the final turning angle should fall in both feasible sets, we could define the new membership value [7] as:

$$\mu_{turningangle}(x) = \mu_A(x) \text{ AND } (\text{NOT } \mu_D(x))$$

$$\mu_{turningangle}(x) = \min \{\mu_A(x), \text{NOT } \mu_D(x)\}$$

By doing so, we will get a unique fuzzy output set to be further processed by the defuzzification function.

3.4 Defuzification

The output of the fuzzy controller must be an exact angle value for the lower control component to execute. This observation demands the controller to output crisp commands. Defuzzification is such a process that converts fuzzy commands into crisp commands.

Many methods of defuzzification are available: MinMax method(used in the Mamdani fuzzy system), Mean of Maximum (MOM) Weighted sum method (used in Sugeno fuzzy inference system) Center of Gravity (Centroid) method. All of them have their own use in different situations. The MOM method calculates the average of this method and assign the fuzzy command with the highest membership degree. The MinMax method first find the minimum of membership degree for each rule, then find the max of them as defuzzification set.

However, both MinMax and MOM methods may not take into account of all information by the fuzzy command, and thus can encounter difficulty in generating commands that turns the robot smoothly.

The COG method may also have problems when applied to mobile robot[7]. However, those situations that could cause its trouble are rare. So we still decide to use the centroid method.

The formula for the centroid method is

$$z_0 = \frac{\int z\mu_c(z)dz}{\int \mu_c(z)dz}$$

Where z_0 is the final crisp output and z is the output variable of before defuzzification. $\mu_c(z)$ is the function value before defuzzification.

4. CONCLUSIONS

In perceive of the great benefits that fuzzy logic has brought to the performance of mobile robot, this paper has described a possible logic strategy for the performance improvement of a new mobile robot---BEARCAT II motion control. The paper describes component functions of the robot and developed a fuzzy logic based on two behaviors. Two fuzzy logic approaches in mobile robot control have been discussed and one is used. The developed fuzzy logic let the mobile robot have the ability to follow the track and avoid obstacles. The paper also applied an efficient command fusion technology, which helps the fuzzy controller to generate crisp command that carries information from both behavior requirements. Further works needs to be done. One can add neural networks into the fuzzy membership function approximation, path tracking approach may also be developed.

5. REFERENCES

1. E.L. Hall and B.C. Hall, Robotics: A User-Friendly Introduction, Holt, Rinehart, and Winston, New York, NY, 1985, pp. 23.
2. Z.L. Cao, S.J. Oh, and E.L. Hall, "Dynamic omnidirectional vision for mobile robots," J. of Robotic Systems, 3(1), pp. 5-17, 1986.
3. Z.L. Cao, Y.Y. Huang, and E.L. Hall, "Region Filling Operations with Random Obstacle Avoidance for Mobile Robots," Journal of Robotics Systems, 5(2),1988, pp. 87-102.
4. Nikhil D. Kelkar, A Fuzzy Controller for three Dimensional Line Following of an Unmannned Autonomous Mobile Robot, Master Thesis, 1997
5. S.J. Oh and E.L. Hall, "Calibration of an omnidirectional vision navigation system using an industrial robot," Optical Engineering, September 1989, Vol. 28, No. 9, pp. 955-962.
6. M.P. Ghayalod, E.L. Hall, F.W. Reckelhoff, B.O. Matthews and M.A. Ruthemeyer, "Line Following Using Omnidirectional Vision," Proc. of SPIE Intelligent Robots and Computer Vision Conf., SPIE Vol. 2056, Boston, MA, 1993.
7. John Yen, Mathan Pfluger, A Fuzzy Logic Based Extenssion to Payton and Rosenblatt's Command Fusion Method for Mobile Robot Navigation' Aug. 1996
8. Ollero A, A.Garcia-Cerezo and J. L. Martínez (1994). Fuzzy Supervisory Path Tracking of autonomous Vehicles.
9. Langer, D., Rosenblatt, J.K., Herbert, M., A Behaviour Based System for Off-Road Navigation, IEEE Trans. On Robotics and Automation, Vol. 10, Dec., 1994, pp 976-983.
10. Behavior-Decision Fuzzy Algorithm. Proc. IEEE International Conference on Robotics and Automation, pp. 117-122.
11. Muñoz V., A. Ollero, A. Simón and M. Prado (1994). Mobile Robot Trajectory Planning with Kinematic and Dynamic Constraints. Proc. IEEE International Conference on Robotics and Automation, vol. 4, pp. 2802-2807.
12. Ollero A., A. Simón, F. García, and V. Torres (1993). Integrated Mechanical Design of a New Mobile Robot. Proc. of the First IFAC International Symposium on Intelligent Components and Instruments for Control Applications, pp. 461-466, Pergamon Press.

Appendix I Fuzzy rule base for the Track Following Behavior

1. IF (linedis is Ab_2) AND (angleinput is Ar_3) THEN (Aout is Aout_3)
2. IF (linedis is Ab_2) AND (angleinput is Ar_2) THEN (Aout is Aout_3)
3. IF (linedis is Ab_2) AND (angleinput is Ar_1) THEN (Aout is Aout_2)
4. IF (linedis is Ab_2) AND (angleinput is Ar) THEN (Aout is Aout_1)
5. IF (linedis is Ab_2) AND (angleinput is Ar1) THEN (Aout is Aout_1)
6. IF (linedis is Ab_2) AND (angleinput is Ar2) THEN (Aout is Aout_1)
7. IF (linedis is Ab_2) AND (angleinput is Ar3) THEN Aout is Aout_1)
8. IF (linedis is Ab_1) AND (angleinput is Ar_3) THEN (Aout is Aout_3)
9. IF (linedis is Ab_1) AND (angleinput is Ar_2) THEN (Aout is Aout_2)
10. IF (linedis is Ab_1) AND (angleinput is Ar_1) THEN (Aout is Aout_2)
11. IF (linedis is Ab_1) AND (angleinput is Ar) THEN (Aout is Aout_1)
12. IF (linedis is Ab_1) AND (angleinput is Ar1) THEN (Aout is Aout_1)
13. IF (linedis is Ab_1) AND (angleinput is Ar2) THEN (Aout is Aout1)
14. IF (linedis is Ab_1) AND (angleinput is Ar3) THEN (Aout is Aout1)
15. IF (linedis is Ab) AND (angleinput is Ar_3) THEN (Aout is Aout_2)
16. IF (linedis is Ab) AND (angleinput is Ar_2) THEN (Aout is Aout_2)
17. IF (linedis is Ab) AND (angleinput is Ar_1) THEN (Aout is Aout_1)
18. IF (linedis is Ab) AND (angleinput is Ar) THEN (Aout is Aout0)
19. IF (linedis is Ab) AND (angleinput is Ar1) THEN (Aout is Aout1)
20. IF (linedis is Ab) AND (angleinput is Ar2) THEN (Aout is Aout2)
21. IF (linedis is Ab) AND (angleinput is Ar3) THEN (Aout is Aout2)
22. IF (linedis is Ab1) AND (angleinput is Ar_3) THEN (Aout is Aout_1)
23. IF (linedis is Ab1) AND (angleinput is Ar_2) THEN (Aout is Aout_1)
24. IF (linedis is Ab1) AND (angleinput is Ar_1) THEN (Aout is Aout1)
25. IF (linedis is Ab1) AND (angleinput is Ar) THEN (Aout is Aout1)
26. IF (linedis is Ab1) AND (angleinput is Ar1) THEN (Aout is Aout2)
27. IF (linedis is Ab1) AND (angleinput is Ar_3) THEN (Aout is Aout1)
28. IF (linedis is Ab2) AND (angleinput is Ar_2) THEN (Aout is Aout1)
29. IF (linedis is Ab2) AND (angleinput is Ar) THEN (Aout is Aout1)
30. IF (linedis is Ab2) AND (angleinput is Ar1) THEN (Aout is Aout2)
31. IF (linedis is Ab2) AND (angleinput is Ar2) THEN (Aout is Aout3)
32. IF (linedis is Ab2) AND (angleinput is Ar3) THEN (Aout is Aout3)