

A. First layer of the fuzzy logic controller

The ATRV-mini is equipped [7] with an array of 24 ultrasonic sensors that are around the vehicle as shown in Fig. 2. The ultrasonic sensors that are used, are manufactured by Polaroid.

After experimenting with, and testing several methods concerning sonar sensor data grouping and management, it was first decided to follow the sensor grouping in pairs as proposed in [8] (considering the ATRV-mini twelve sonar sensor groups A_i , $i=1, \dots, 12$, have been enumerated as shown in Fig. 2) and then divide the sum of the provided pair sensor data by two to determine the distance from the (potential) obstacle. However, this method gave unsatisfactory results due to the ATRV-mini specific sensor unreliability. Even in cases with obstacles present in the vicinity of the vehicle, the sensors were detecting a “free path”. To overcome this problem, a modified, simpler, sensor grouping and data management method was tested that returned much better and accurate results: The sensors were again grouped in pairs according to Fig. 2, but the minimum of their readings was considered as a distance measure from the (potential) obstacle. Each ATRV-mini sonar sensor returns data from obstacles at a maximum distance of 4 meters (experimentally verified as opposed to a different value provided by the sonar sensors manufacturer).

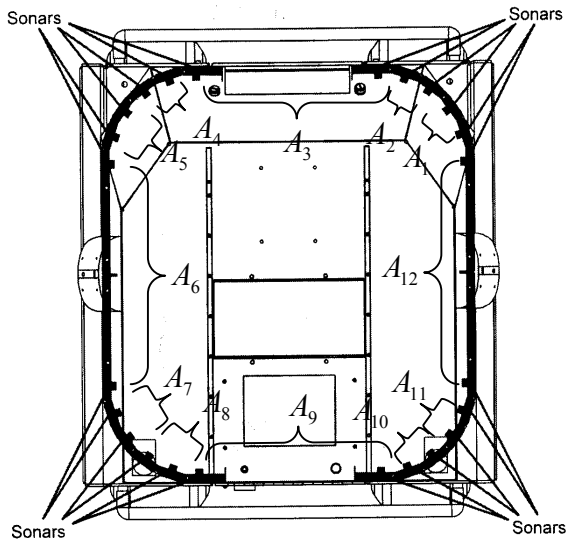


Fig. 2. Grouping of Sensors.

The form of each first layer individual fuzzy controller, including the obstacle detection module, is shown in Fig. 3. Observing Fig. 3, data from group sensors A_1, A_2, \dots, A_5 (5 inputs) and group sensors A_7, A_8, \dots, A_{11} (5 inputs) serve as inputs to the individual controllers responsible for the

calculation of the *front* and *back* collision possibilities, respectively. Data from group sensors A_5, A_6, A_7 (3 inputs) and group sensors A_{11}, A_{12}, A_1 (3 inputs) serve as inputs to calculate the *left* and *right* possibilities, respectively. The individual fuzzy controllers utilize the same membership functions to calculate the collision possibilities. The linguistic values of the variable *distance_from_obstacle* are defined to be three, *near*, *medium distance*, *away* with membership functions as shown in Fig. 4 reflecting the maximum distance of 4 meters a sonar returns accurate information about potential obstacles.

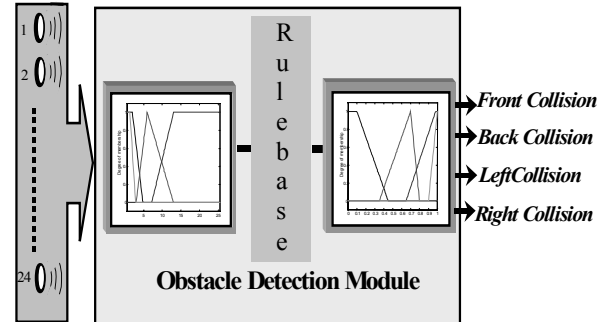


Fig. 3. Obstacle detection module.

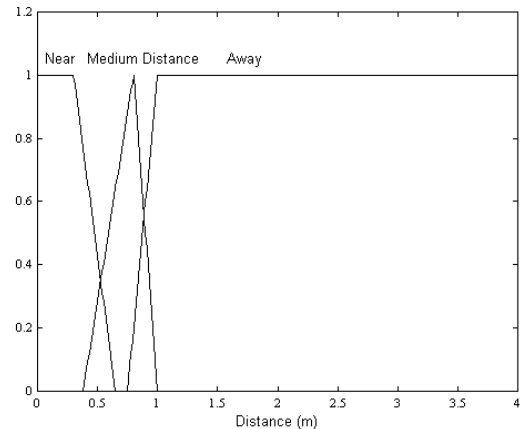


Fig. 4. Input Variable *Distance_From_Obstacle*.

The first layer output is a collision possibility in each direction taking values from 0 to 1. The linguistic variables describing each direction output variable *collision possibility*, are *not possible*, *possible*, *high possibility* (with empirically derived for best performance) membership functions as shown in Fig. 5. A part of the rules base for left collision is presented in Table I.

An example of the rules used to extract front collision possibilities is: IF A_1 is *near* AND A_2 is *near* AND A_3 is

near AND A_4 is medium_distance AND A_5 is near THEN collision_possibility is high. Similar for the back collision possibility. For left (equivalently for right collision possibilities the rule is of the form: IF A_5 is near AND A_6 is near AND A_7 is near THEN collision_possibility is high.

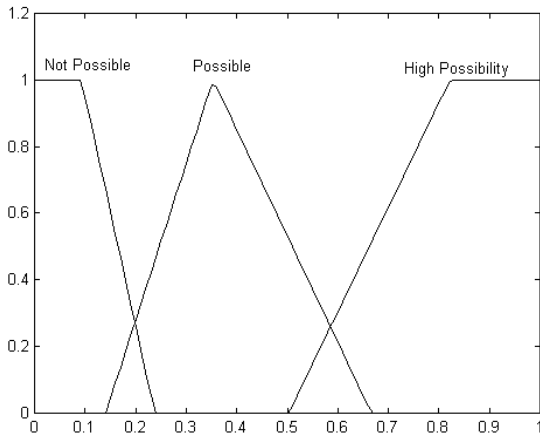


Fig. 5. Output Variable *Collision_Possibility*.

TABLE I
PART OF THE RULES BASE FOR LEFT COLLISION.

Input Variables			Output Variables
A_5	A_6	A_7	
Near	Near	Near	High Possibility
Away	Away	Away	Not Possible
Near	Away	Medium_Distance	Possible
Near	Away	Near	High Possibility

B. Second layer of the fuzzy logic controller

The second layer fuzzy controller receives as inputs the four collision possibilities in the four directions and the angle error, and outputs the translational velocity, which is responsible for moving the vehicle backward or forward and the rotational speed, which is responsible for the vehicle rotation as shown in Fig. 6.

The angle error represents the difference between the robot heading angle and the desired angle the robot should have in order to reach its target. The angle error takes values ranging from -180° to 180° . The linguistic variables that represent the angle error are: *Backwards_1*, *Hard_Left*, *Left*, *Ahead*, *Right*, *Hard_Right*, *Backwards_2* with (empirically derived from tests) membership functions as shown in Fig. 7.

The translational velocity (m/sec), which is one of the outputs of the second layer controller, is described with the following linguistic variables: *back_full*, *back*,

back_slow, *stop*, *front_slow*, *front*, *front_full*, with membership functions as in Fig. 8.

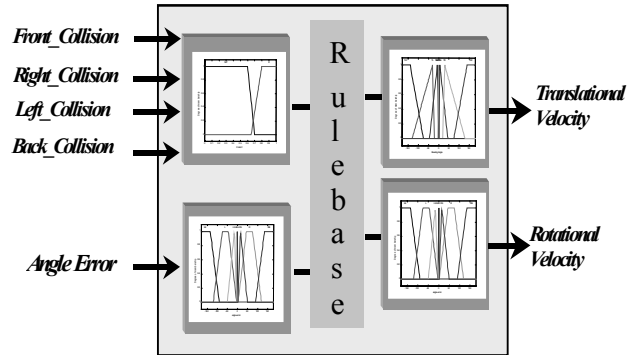


Fig. 6. Block diagram of the 2nd layer of the fuzzy logic controller.

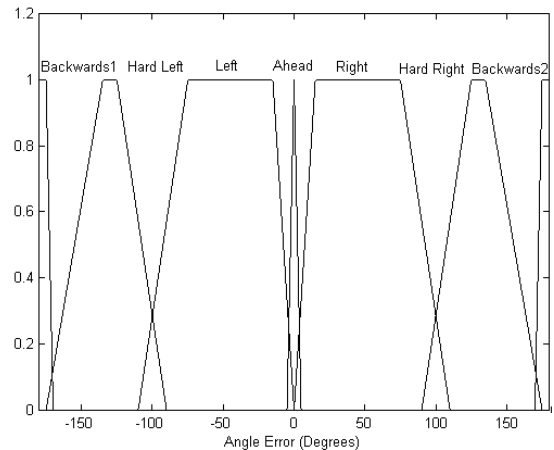


Fig. 7. Input Variable *Angle_Error*.

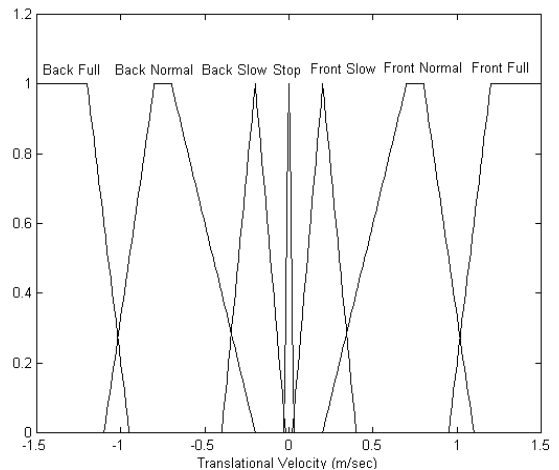


Fig. 8. Output Variable *Translational_Velocity*.

The rotational speed (rad/sec) is described with the following linguistic variables: *right_full*, *right*, *no_rotation*, *left*, *left_full* with membership functions as in Fig. 9.

An example of the rules that control the vehicle is demonstrated: IF *Front_Collision* is *Not_Possible* AND *Back_Collision* is *Not_Possible* AND *Left_Collision* is *Not_Possible* AND *Right_Collision* is *Not_Possible* AND *Angle_Error* is *Ahead* THEN *Translational_Velocity* is *Front_Full* AND *Rotational_Velocity* is *No_Rotation*

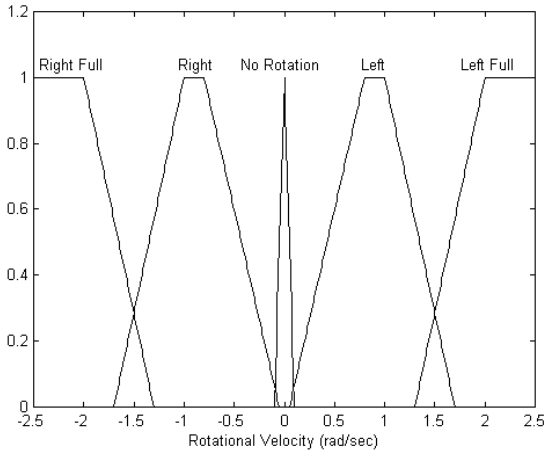


Fig. 9. Output Variable *Rotational_Velocity*.

III. RESULTS

The fuzzy logic controller has been designed and implemented using C++ in an ATRV-mini manufactured by Real World Interface (RWI). In all experiments the robot is considered to have reached its target when stopping inside a circle with radius of 30 cm. This assumption has been dictated because all calculations have been made relative to the center of the robot. So if the robot stops inside that circle it is assumed that it has reached its target.

Several scenarios in an indoor 2-D obstacle filled environment have been tested to study the robot behavior and the controller's applicability.

The arrow in Fig. 10, Fig. 15, Fig. 20 is showing the initial direction of the vehicle.

In test case 1 we examine the behavior of the vehicle in an environment with three obstacles. The test case 1 is presented in Fig. 10. Fig. 11 shows the translational velocity, while the rotational velocity is given in Fig. 12. Fig. 13 presents the front collision possibility. In Fig. 14, the solid line indicates the left collision possibility while the dotted the right collision possibility. The behavior of the vehicle is defined from the surrounding obstacles.

In the beginning the left collision possibility is high due to the obstacle in the left. The robot moves forwards and it's steering right in order to avoid the obstacle. Then it steers left and moves towards its target.

In the second test case presented in Fig. 15, a more complicated environment with three obstacles has been tested. Fig. 16 shows the translational velocity, while the rotational velocity is given in Fig. 17. Fig. 18 presents the front collision possibility while in Fig. 19 the solid line indicates the left collision possibility while the dotted the right collision possibility. In Fig. 15 we can see that the path in front of the robot is blocked. The robot uses only the rotational velocity in order to steer and avoid the obstacle. Then it moves in a curve towards its target.

The third test case considers an environment with many small obstacles. The path the vehicle is following is presented in Fig. 20. Due to the obstacles that are around the vehicle, the vehicle is forced to make a small right turn and then it escapes from the closed area. Fig. 21 shows the translational velocity, while the rotational velocity is given in Fig. 22. Fig. 23 presents the front collision possibility while in Fig. 24 the solid line indicates the left collision possibility while the dotted the right collision possibility.

The behavior of the vehicle in each case can be verified by observing the relative figures concerning the collision possibilities in each direction.

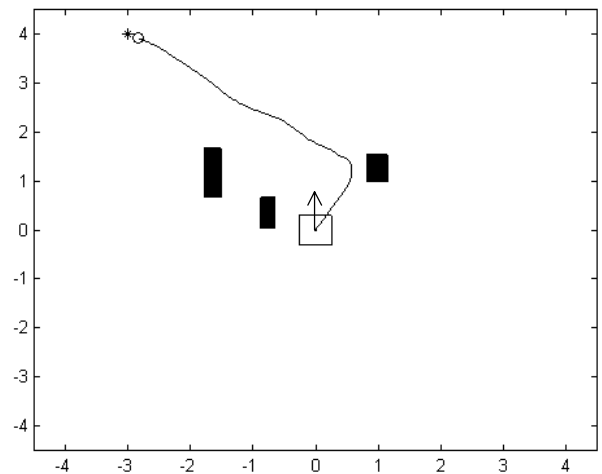


Fig. 10. Test Case 1. Environment with three obstacles and remote target point.

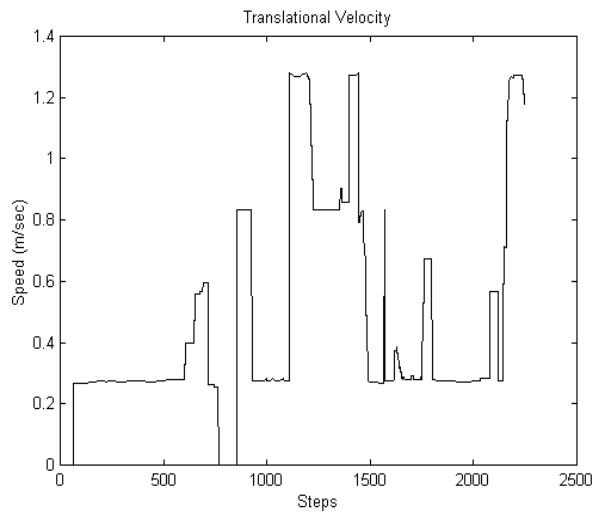


Fig. 11. Translational Velocity in Test Case 1.

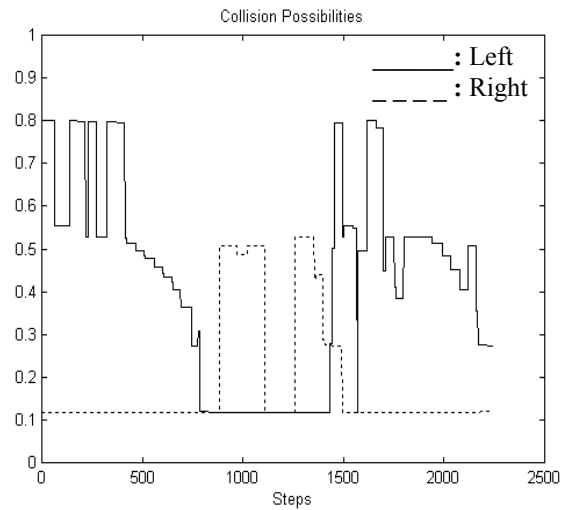


Fig. 14. Left and Right Collision Possibilities in Test Case 1.

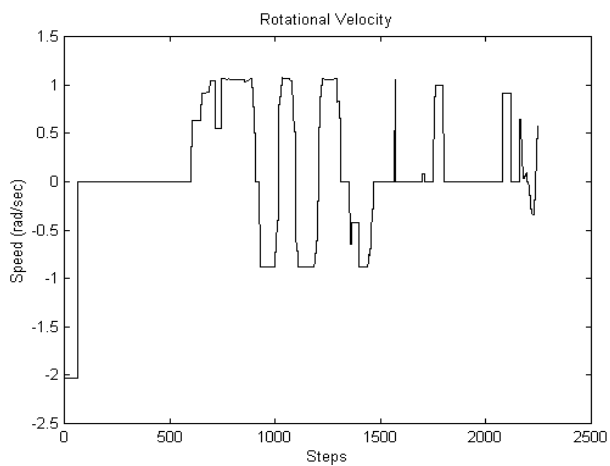


Fig. 12. Rotational Velocity in Test Case 1.

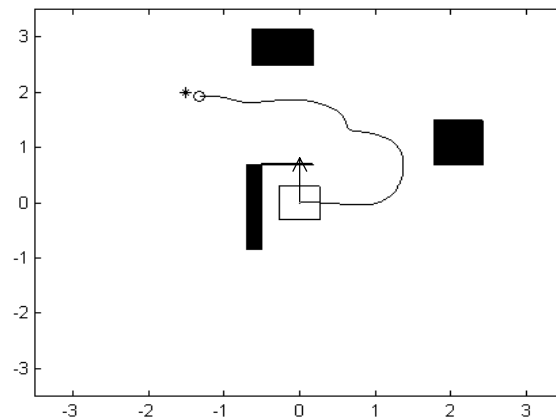


Fig. 15. Test Case 2. Environment with three obstacles.

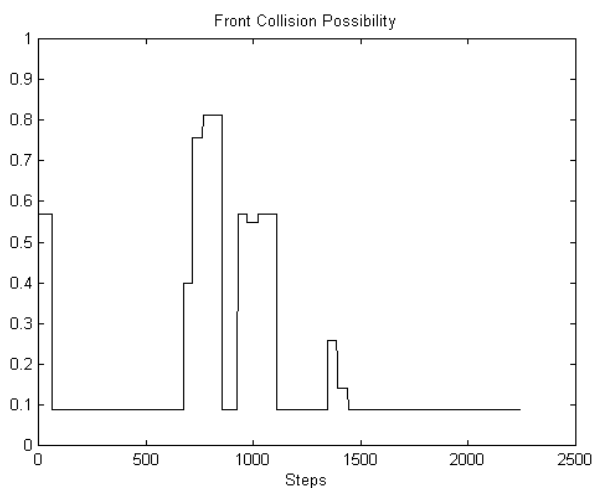


Fig. 13. Front Collision Possibility in Test Case 1.

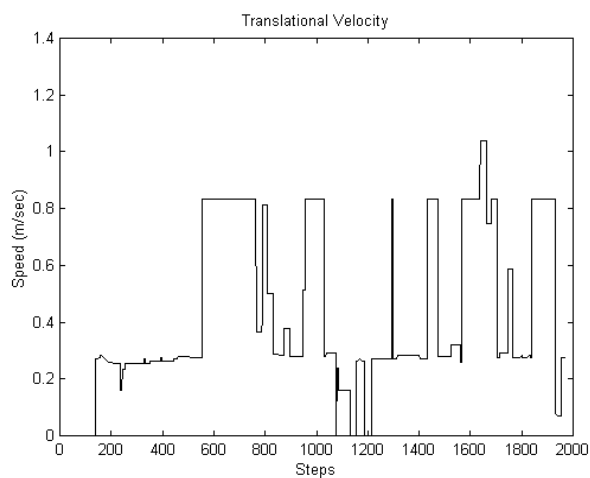


Fig. 16. Translational Velocity in Test Case 2.

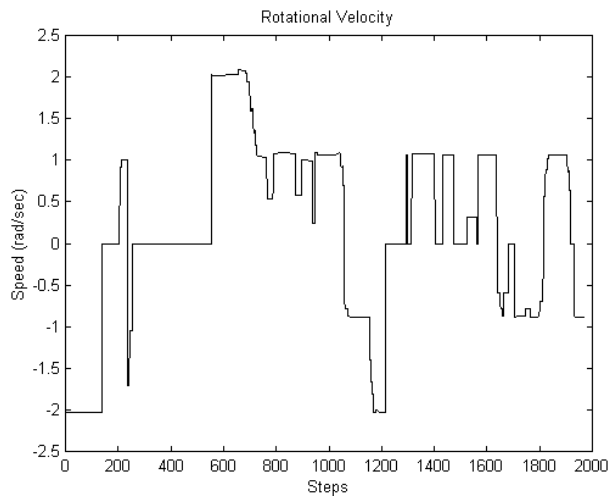


Fig. 17. Rotational Velocity in test case 2.

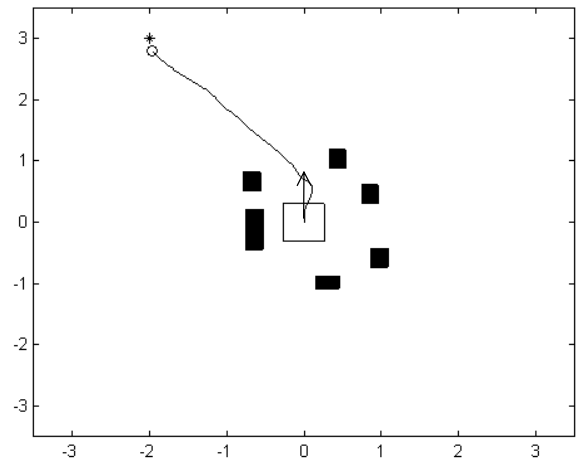


Fig. 20. Test Case 3. Environment with many small obstacles.

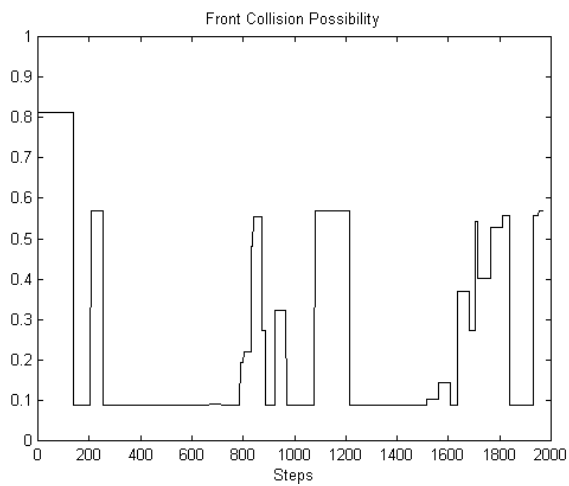


Fig. 18. Front Collision Possibility in Test Case 2.

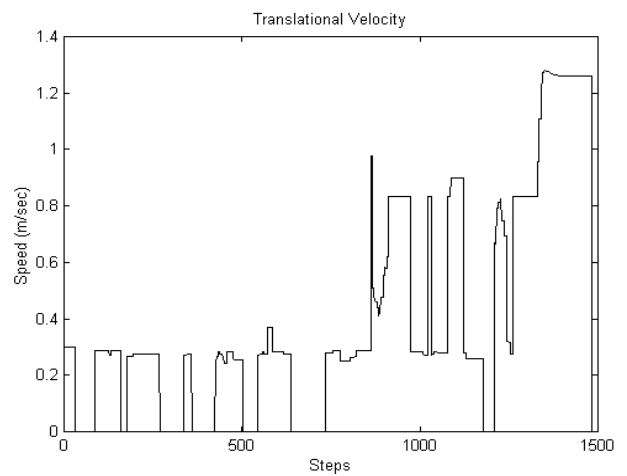


Fig. 21. Translational Velocity in Test Case 3.

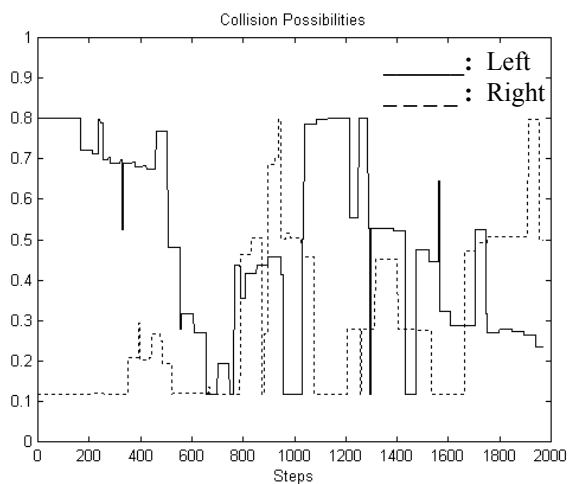


Fig. 19. Left and Right Collision Possibilities in Test Case 2.

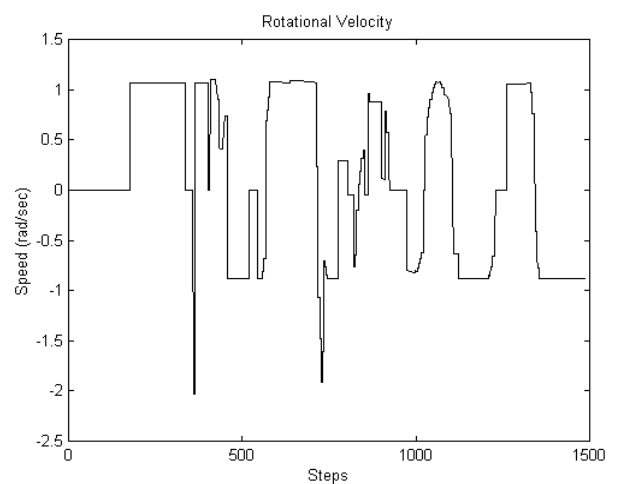


Fig. 22. Rotational Velocity in Test Case 3.

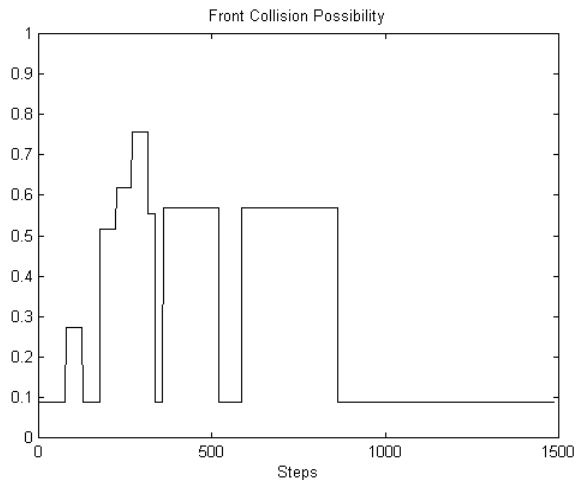


Fig. 23. Front Collision Possibility in Test Case 3.

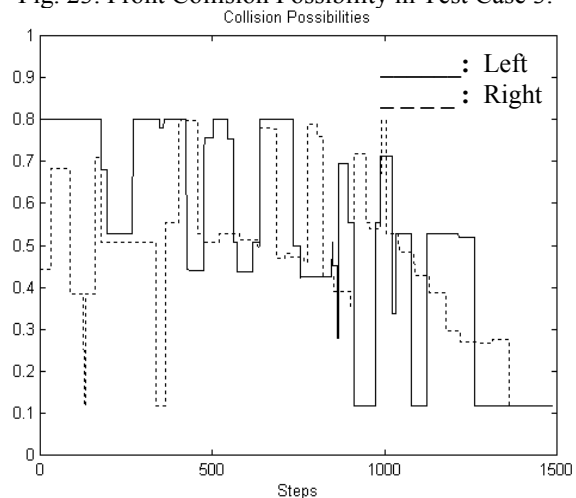


Fig. 24. Left and Right Collision Possibilities in Test Case 3.

IV. DISCUSSION AND CONCLUSIONS

We have presented a navigation system for a skid steering vehicle with the use of a two-layer fuzzy logic controller. The first layer of the fuzzy logic controller is composed of four fuzzy logic controllers. The rule base of the controllers responsible for *front* and *back* collision contains 60 rules and the rule base of the controllers responsible for *front* and *back* collision contains 57 rules. The rule base of the second layer fuzzy logic controller responsible for real-time navigation and collision avoidance contains 238 rules.

The fuzzy logic controller has performed satisfactorily. The results show that the vehicle has the ability to move in complicated environments. The controller, which is proposed in this paper, is based in the controller proposed in [9] but it is implemented in a skid steering vehicle.

Future directions of the research include the testing of

dynamic environments, and the use of other sources of information. The goal is to create an autonomous vehicle that will use for navigation and the collision avoidance combined information from visual inputs, sonars and outdoors GPS data that will guide the vehicle in remote target points.

ACKNOWLEDGEMENT

This work has been partially supported by a research grant from the GSRT, PENED-99 E44.

REFERENCES

- [1] M. B. Montaner, A. Ramirez-Serrano, "Fuzzy Knowledge-Based Controller Design for Autonomous Robot Navigation," *Expert Systems with Applications*, vol. 14, pp. 179-186, 1998.
- [2] H. Maaref, C. Barret, "Sensor-based Fuzzy Navigation of an Autonomous Mobile Robot in an Indoor Environment," *Control Engineering Practice*, vol. 8, pp. 757-768, 2000.
- [3] A. Nearchou, "Adaptive Navigation of Autonomous Vehicles Using Evolutionary Algorithms," *Artificial Intelligence in Engineering*, vol. 13, pp. 159-173, 1999.
- [4] D. K. Pratihari, K. Dep, A. Ghosh, "A Genetic-Fuzzy Approach for Mobile Robot Navigation Among Moving Obstacles," *International Journal of Approximate Reasoning*, vol. 20, pp. 145-172, 1999.
- [5] T. Hebert, N. C. Tsourveloudis, K. P. Valavanis, "Fuzzy Control of Autonomous Vehicle Utilizing Electrostatic Potential Fields," *Proceedings of the 1998 IEEE International Conference of Control Applications*, Trieste, Italy, pp. 658-662, 1998.
- [6] B. Shamah, "Experimental Comparison of Skid Steering vs. Explicit Steering for Wheeled Mobile Robot," M.Sc. Thesis, The Robotics Institute, Carnegie Mellon University, Pittsburgh Pennsylvania, 1999.
- [7] *ATRV-Mini All-Terrain Mobile Robot User's Guide*, IS Robotics, Inc., Real World Interface Division, 2000.
- [8] J. Godjevac, *Neuro-Fuzzy Controllers, Design and Application*, Lausanne, Presses Polytechniques et Universitaires Romandes, 1997.
- [9] N. C. Tsourveloudis, K. P. Valavanis, T. Hebert, "Autonomous Vehicle Navigation Utilizing Electrostatic Potential Fields and Fuzzy Logic," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 4, pp 490-497, 2001.
- [10] K. P. Valavanis, T. Hebert, R. Kolluru, N. C. Tsourveloudis, "Mobile Robot Navigation in 2-D Dynamic Environments Using Electrostatic Potential Fields," *IEEE Transactions on Systems, Man and Cybernetics*, part A, vol. 30, pp. 187-197, 2000.