Fuzzy Logic Based Software Control Architecture for a Skid Steering Vehicle

L. Doitsidis, K. P. Valavanis, N. C. Tsourveloudis
Technical University of Crete
Department of Production Engineering and Management
Chania, Crete, Greece GR-73100
{ldoitsidis, kimonv, nikost}@dpem.tuc.gr

Abstract

A sensor-based software control architecture is proposed for autonomous navigation of the Real World Interface (currently iRobot) ATRV-mini skid steering vehicle. The architecture is superimposed over the existing ATRV-mini mobility software that navigates the robot. A two-layer fuzzy logic controller has been implemented: Inputs are the collision possibilities in *front*, *back*, *left* and *right* directions of movement and the heading angle error. Outputs are the rotational and translational velocities. The paper is the outgrowth of the authors previous work presented in [11].

I. INTRODUCTION

There exist several methods concerning autonomous mobile robot navigation in 2-D uncertain environments [1]-[5]. The authors have already published their results in [9]-[10], but this paper, the outgrowth of [11], differs from previous ones in that it presents a detailed architecture of a fuzzy logic controller designed to control a skid steering vehicle in general, and the ATRV-mini in particular.

Skid steering vehicles are compact, light, require few parts to assemble and exhibit agility from point turning to line driving using only the motions, components, and swept volume needed for straight line driving [6]. Skid steering vehicle motion differs from explicit steering vehicle motion in the way the skid steering vehicle turns. The wheels rotation is limited around one axis and the lack of steering wheel results in navigation determined by the speed change in either side of the skid steering vehicle. Same speed in either side results in a straight-line motion. Explicit steering vehicles turn differently since the wheels are moving around two axes.

The robot's geometric configuration in the *X-Y* plane is shown in Figure 1, where a_t is the heading angle, w is the robot width, θ corresponds to the sense of rotation and S_1, S_2 are the speeds in either side of the robot. The array of the 24 Polaroid ultrasonic sensors and their grouping is shown in
Figure 2. Given the twelve sonar sensor groups A_i, $i=1,\ldots,12$, the minimum of their readings has been considered as a distance measure from a potential obstacle. It has been experimentally verified that each ATRV-mini sonar sensor returns data from obstacles at a maximum distance of 4 meters (however data at a distance of up to 2.8 meters are much more reliable).

The derived and implemented planner is a two-layer fuzzy logic based controller that provides purely vehicle "reactive behavior" in a 2-D obstacle filled environment, with inputs readings from the 24 sonar sensors ring and angle errors, and outputs the updated rotational and translational velocities of the vehicle. Potential collisions are considered in four main directions, *front*, *back*, *left* and *right*. Put in this context, data from group sensors $A_1, A_2, \ldots, A_5$ (5 inputs) and group sensors $A_7, A_8, \ldots, A_{11}$ (5 inputs) serve as inputs to the individual controllers responsible for the calculation of the *front* and *back* collision possibilities, while 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.

.

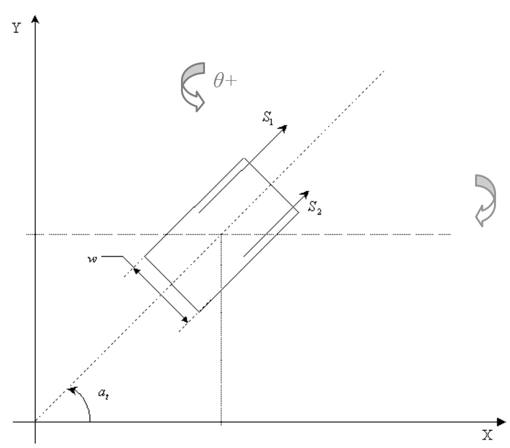

Fig. 1. Geometric configuration of the robot in the X-Y plane.

II. THE FUZZY LOGIC CONTROL SYSTEM

The two-layer fuzzy logic controller is shown in Figure 3. The first layer of the Mamdani-type controller consists of four fuzzy logic controllers responsible for obstacle detection and calculation of the collision possibilities in the four main directions, *front*, *back*, *left*, *right*.

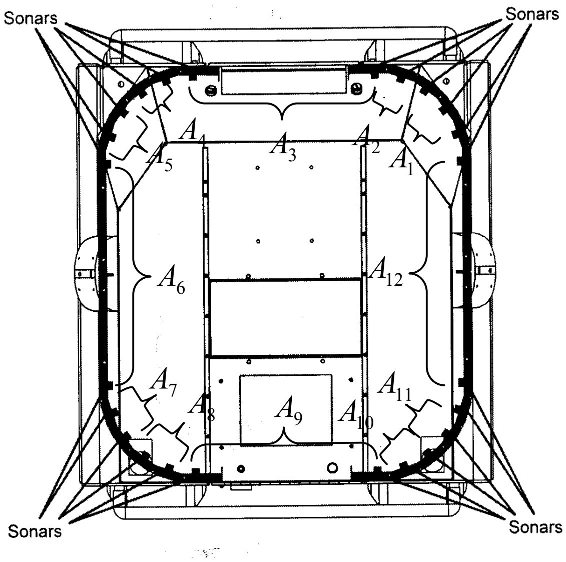

Fig. 2. Sensors grouping

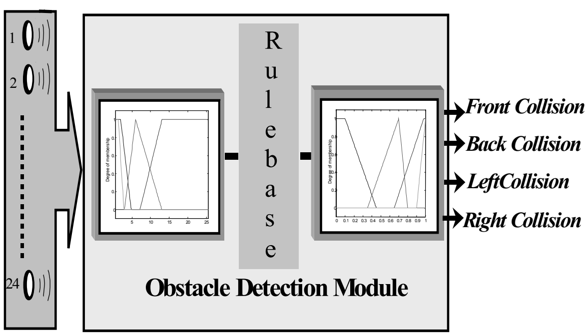

Fig.3a. Obstacle detection module

Each controller receives as inputs the sonar sensor data and returns as output the collision possibility in directions *front, back, left* and *right*. The possibilities calculated in the first layer are the input to the second layer along with the angle error (the difference between the robot heading angle and the desired target angle); the output is the updated vehicle's translational and rotational speed. 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.

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^o to 180^o. The linguistic variables that represent the angle error are: *Backwards_1, Hard_Left, Left, Ahead, Right, Hard_Right, Backwards_2.* 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.* The rotational speed (rad/sec) is described with the following linguistic variables: *right_full, right, no_rotation, left, left_full.* The linguistic variables describing each direction output variable *collision_possibility,* are *not_possible, possible, high_possibility.*

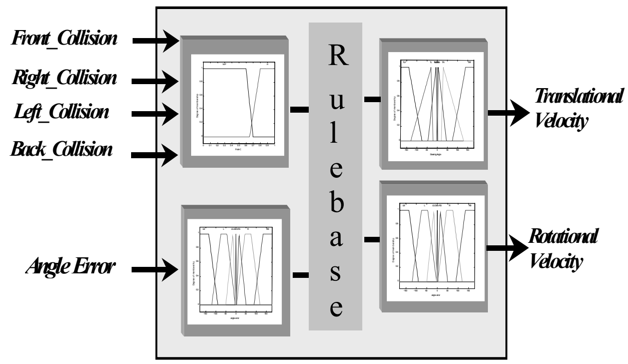

Fig. 3b.The 2nd layer of the fuzzy logic controller.

III. CONTROLLER IMPLEMENTATION

The fuzzy logic controller has been designed and implemented using C++ in an ATRV-mini manufactured by Real World Interface (RWI).

The block diagram of the main program, which is responsible for the navigation and collision avoidance of the vehicle is shown in Fig. 4.

During initialization the vehicle position is considered as the center of the environment. That position is considered as a reference point for all calculations that are taking place during the execution of the navigation program.

After the initialization the vehicle is starting to move. The sensors are scanning the environment and the data from each pair are fed to the fuzzy logic controller, which is responsible for the calculation of the translational and rotational speed of the vehicle. A check concerning the robot position relatively with the target point is then performed.

Due to the robot dimensions, the robot is considered to have reached its target when stopping inside a circle of 30 cm radius.

The program is modular, thus it can be easily modified. The program modules are presented in Fig. 5. Inside the dotted area are the modules concerning the fuzzy controller. The main program may be used regardless of the specific controller type; this refers to replacing the appropriate modules as needed. Currently we are developing modules for navigation using genetic algorithms.

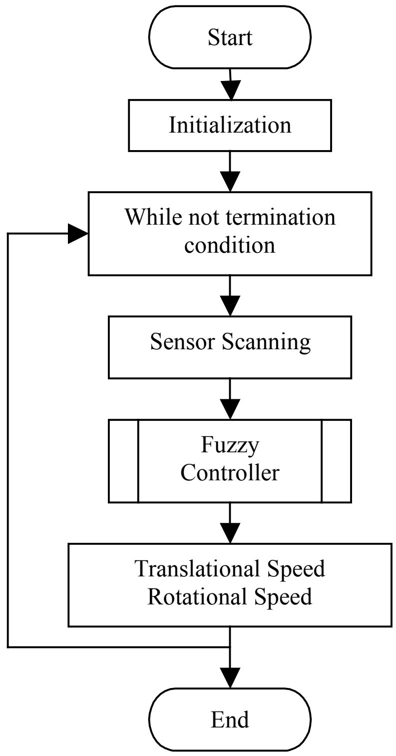

Fig. 4. Block diagram of the main program.

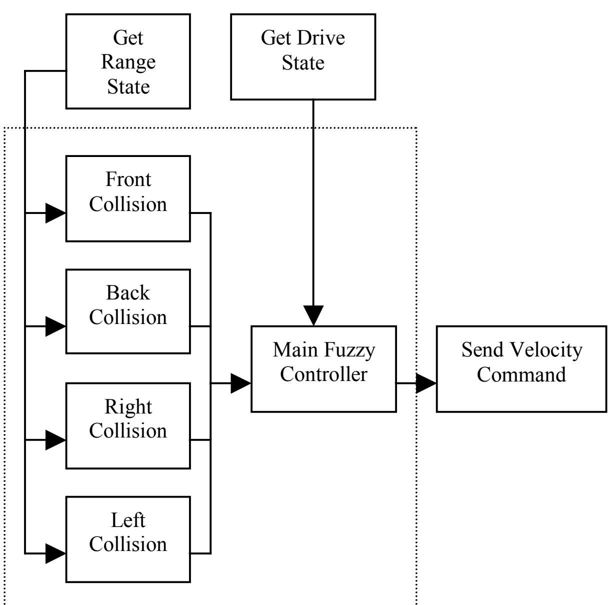

Fig. 5.Modules of the main program.

The modules developed are cooperating with mobility software developed by RWI. Mobility robot integration software is a distributed, object-oriented toolkit for building control software for single and multi-robots systems.

IV. EXPERIMENTAL RESULTS

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. 8, Fig. 13 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. 8. Fig. 9 shows the translational velocity, while the rotational velocity is given in Fig. 10. Fig. 11 presents the front collision possibility. In Fig. 12, the solid line indicates the left collision possibility while the doted 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.

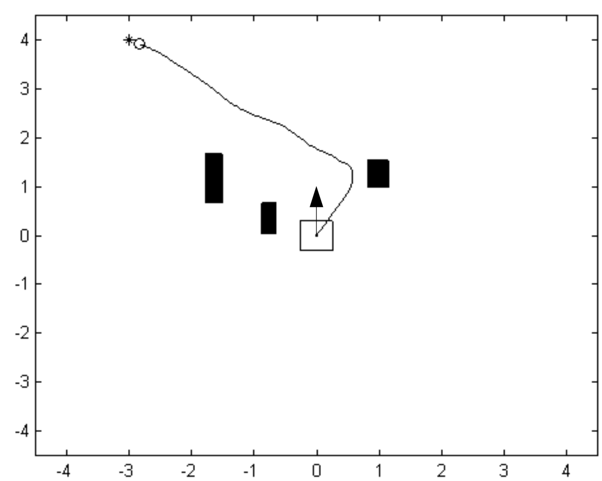

Fig. 8. Test Case 1. Environment with three obstacles and remote target point.

With the term steps we define, each time that the robot is getting sensor readings.

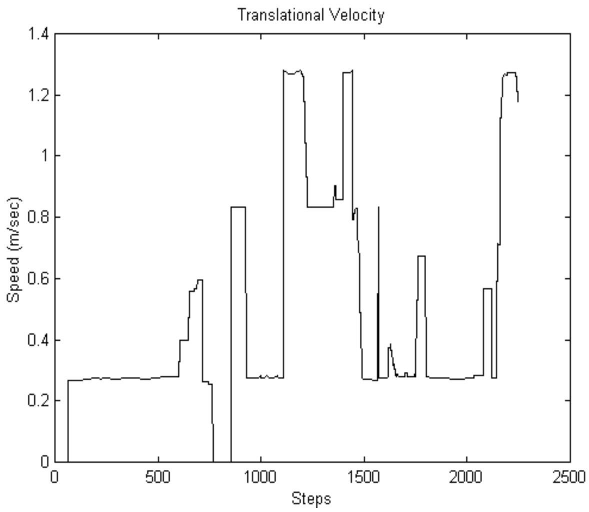

Fig. 9. Translational Velocity in Test Case 1.

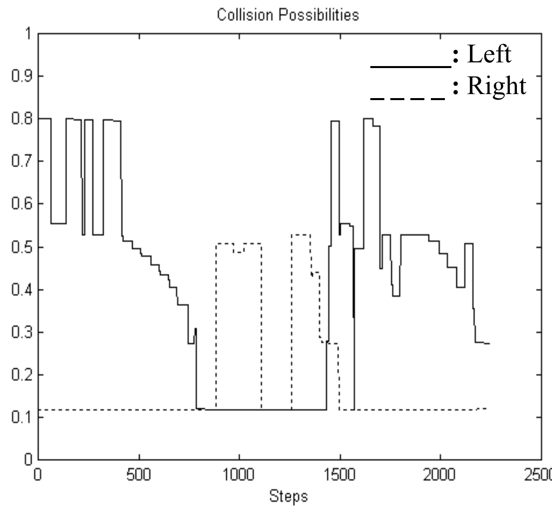

Fig. 12. Left and Right Collision Possibilities in Test Case 1.

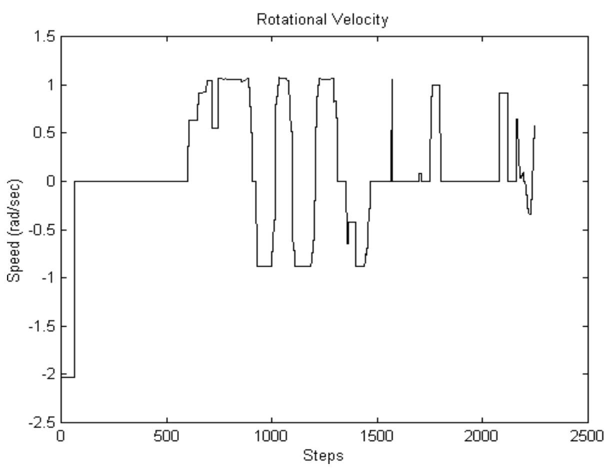

Fig. 10. Rotational Velocity in Test Case 1.

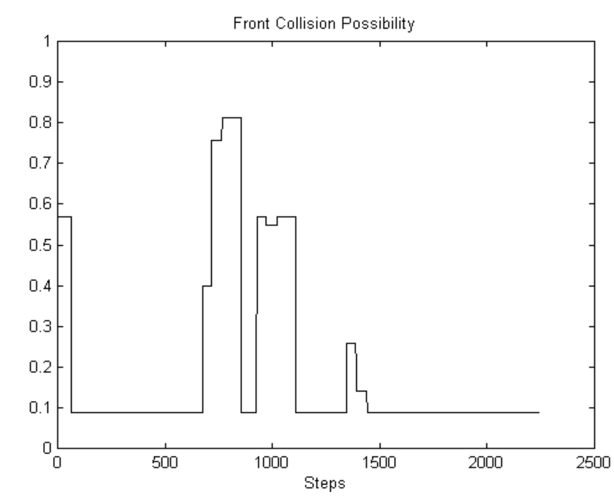

Fig. 11. Front Collision Possibility in Test Case 1.

In the second test case presented in Fig. 13, a more complicated environment with three obstacles has been tested. Fig. 14 shows the translational velocity, while the rotational velocity is given in Fig. 15. Fig. 16 presents the front collision possibility while in Fig. 17 the solid line indicates the left collision possibility while the doted the right collision possibility. In Fig. 13 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 behavior of the vehicle in each case can verified by observing the relative figures concerning the collision possibilities in each direction.

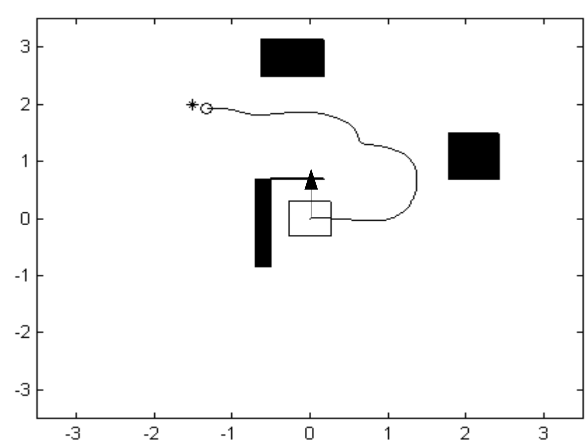

Fig 13. Test Case 2. Environment with three obstacles.

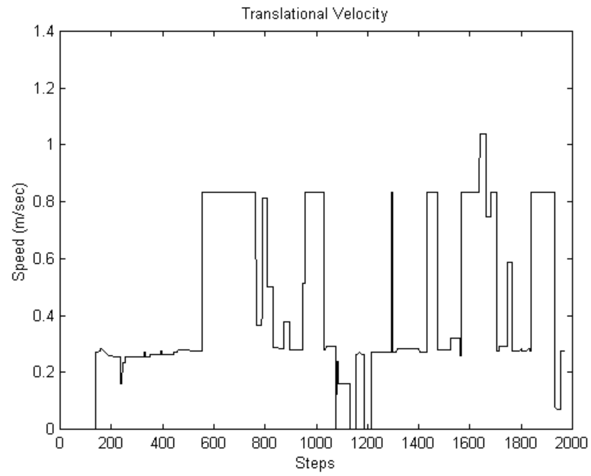

Fig. 14. Translational Velocity in Test Case 2.

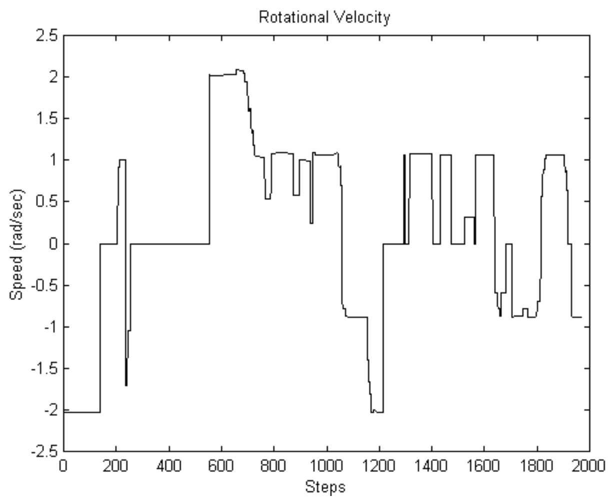

Fig. 15. Rotational Velocity in test case 2.

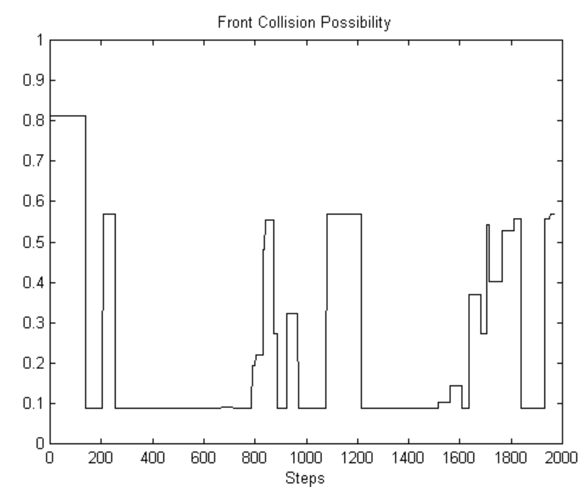

Fig. 16. Front Collision Possibility in Test Case 2.

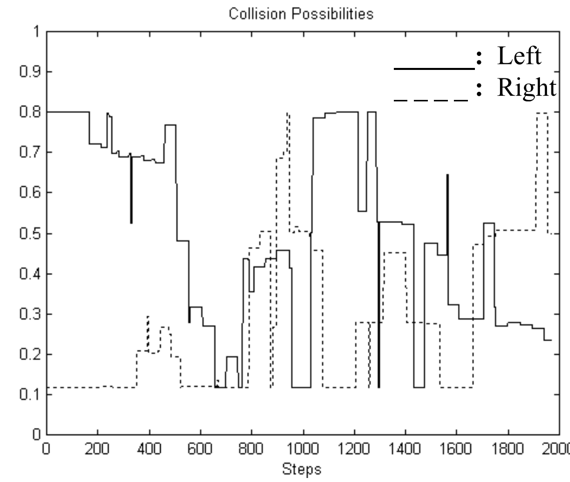

Fig. 17. Left and Right Collision Possibilities in Test Case 2.

V. 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 control architecture is modular and the fuzzy logic controller can be easily replaced with modules that can navigate the system with other methods.

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.

Currently we are developing other modules that they might navigate the vehicle using other methods, such as genetic algorithms and genetic-fuzzy algorithms.

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 EΔ 4.

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. Pratihar, 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.

[11] L. Doitsidis, K. P. Valavanis, N. C. Tsourveloudis, " Fuzzy Logic Based Autonomous Skid Steering Vehicle Navigation," *IEEE International Conference on Robotics & Automation,* Washington, DC, 2002.