Proceedings of the International Conference on Information Technologies (InfoTech-2016)
20-21 September 2016, Bulgaria

A FUZZY LOGIC CONTROLLER FOR OBSTACLE AVOIDANCE AND PATH PLANNING FOR AN AUTONOMOUS FARM TRACTOR

Atanas Dimitrov[1], Violeta Vanteva[2], Mihail Mitev[3]

[1]Faculty of computer science and engineering, Burgas Free University
e-mails: [1]atanas@bfu.bg; [2]v_vanteva@abv.bg
Bulgaria
[3]iTractor" Ltd.; e-mail: michael.g.mitev@gmail.com
United Kingdom

Abstract: The goal of this research was to develop a fuzzy logic controller for an autonomous small and electrical farm tractor that moves inside a partially unknown environment. To accomplish optimal path planning by smooth obstacle avoidance, the vehicle (electrical tractor) is equipped with several different types' range finders as ultrasonic and infrared sensors and 2D LRF (laser range finder). The collected data from sensors are operated with especially custom-built PIC based fuzzy microcontroller, which generates appropriate signals to electrical motors of the vehicle.
Key words: fuzzy logic controller, ultrasonic and infrared sensors, LRF, obstacle avoidance.

1. INTRODUCTION

An autonomous small farm tractor is an intelligent robot that performs a given task by identifying the surrounded environment with the on-board sensors and reacts on the state of condition by itself instead of human. The ability to plaining motion and navigating autonomously, while avoiding any type of obstacles is a navigation problem. To accomplish the goals of this research, all the concept of building fuzzy controller is based on reactive strategy [1] – the fuzzy controller operates only with equipped and supplied from on-board sensory information.

A number of different methodologies have been proposed for achieving collision avoidance. Some of the more popular methods are based on edge-detection, certainty grids, potential fields, or combinations of these. One of the first widely used techniques for navigation and obstacle avoidance is based on the method of detecting the vertical edges of the obstacles. Borenstein and Koren [2] mentioned that this method has two major drawbacks: the huge sensitivity to the accuracy of the used ultrasonic sensors and the additional time needed to measure the obstacle after being detected. This method is inefficient in work in real-time and fast-moving robots. To overcome these limits, Borenstein and Koren developed a technique called virtual force field (Virtual Force Field - VFF), uniting the occupancy grid and the Potential Field Method (PFM). The main advantages of this developed method are: -the completely ignoring of the isolated cells from the occupancy grid, considering them as random errors; -eliminating the need of the robot to stop, when detecting an obstacle; -parallel renovation and use of occupancy grid; -possibility to use it with various type and functionality sensors.

Navigation, based on the VFF method, is suitable for detecting and avoiding obstacles, located at a relatively large distance from one another. In the case of passing through narrow corridors or between closely spaced obstacles, this method is ineffective because of the instability of the movement [3], [4] consisted in sudden changes of the movement speed.

For the preparation of a smooth motion, without the need to limit the speed of the robot, the authors suggest the use of a two-dimensional Cartesian histogram modeling of the working scene. The proposed method is known as a Vector Field Histogram (VFH) and is used for detection of a passable road between closely spaced obstacles in the scene in real-time. For this purpose, the two-dimensional histogram is transformed into a polar one.

For the indication of accessible paths serve those areas of the histogram, which have a weight number smaller than the predetermined threshold. The advantages of this method are the reduced requirements for on-board computing resources of the robot, has a high performance allowing an adequate response to the detection of unexpected obstacle and its ability to be built in a mobile robot with semi-autonomous or autonomous control. The main limitation of this method is that it can only be used for local planning in the route of the robot (local minima problem), but not for finding the optimum time, which can lead to the situation "deadlock". For reducing the risk of such situations, the authors suggested the use of an algorithm for global planning of the route or making a detailed model of the working environment [4].

Ulrich and Borenstein [5] offer an advanced version of the original VFH method. In the new method, called "VFH+", the reporting on the physical dimensions of the robot allows the use of binary polar histogram for determination of the free and possible paths. Additionally, it is added the "masked polar histogram" that takes into account the dynamics and kinematics of the robot and shows the free direction for movement, taking into account the current speed of the robot.

The optimized VHF+ enables the smoothing of the path of the mobile robot, increasing the rapid action of the process of drawing up the histogram, eliminates the possibility of misleading targeting of the robots direction of an insurmountable obstacle, and allows a change in the behavior of the robot. Unsolved remains the problem connected to the local nature of the method. In addition, the method is useful primarily for robots with non-zero turning radius and a design shape close to the shape of a circle. For robots with an asymmetrical shape, the process of compiling a histogram is complicated because of the need to take and account of the current orientation of the robot.

Combining VFH+ algorithm to avoid obstacles with a search algorithm A* in [6] aims to overcome the problem with finding a passable way to the target destination of the robot. The method is known as VFH and it is expressed in an assessment of "consequences" if the robot chooses a particular way of circumventing an obstacle. Unlike its predecessors, the effectiveness of VFH* consists of pre-determining the new position and orientation of the robot, if you move a few steps forward.

In this paper, we propose a fuzzy logic based control system for obstacles avoidance and path planning of a small, autonomous farm tractor. At this point, the main idea is to develop the fuzzy control system for an autonomous navigation of a tractor. Because this idea is still on a conceptual level, this fuzzy controller is tested on a small mobile robot. The data for a distance to any insurmountable obstacle is obtained from three different infrared sensors, which are mounted in the front, right and left side of the robot.

The aim of this paper is to show our implementation of a simple fuzzy logic controller for autonomous obstacle collision avoidance and path planning.

2. DESIGN OF FUZZY LOGIC CONTROLLER

The conventional architecture of a fuzzy logic controller includes four main stages named: - fuzzification; - fuzzy reasoning, - defuzzification and knowledge base as shown in Fig 1.[7]

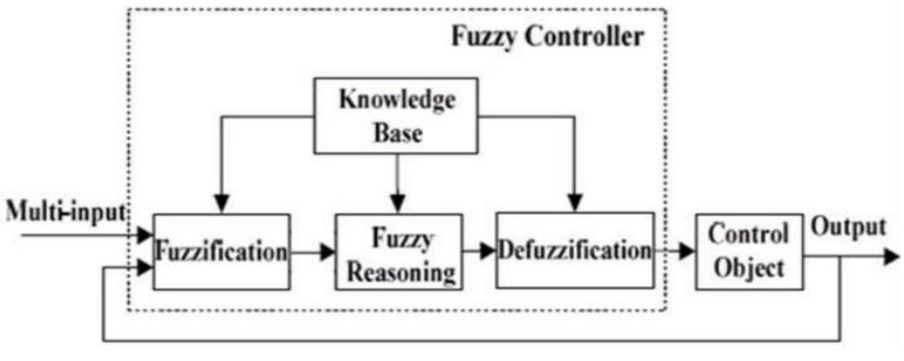


Fig. 1. Typical architecture of a fuzzy logic controller

The fuzzification process, converts the received data from sensors into input variables, known also as fuzzy variables. These fuzzy variables are passed to the fuzzy reasoning stage to be processed. The Fuzzy reasoning stage generates fuzzy results while process fuzzy variables with stored in knowledge base relevant data and control rules. To be able to use from a hardware controller, the generated fuzzy variables, must be converted into appropriate signals and this is done in the defuzzification stage.

As a first step of fuzzy logic controller design is used a fuzzy tool from MATLAB. For evaluation of the needed crisp values is used the most common technique Mamdani Method (Fig.2)

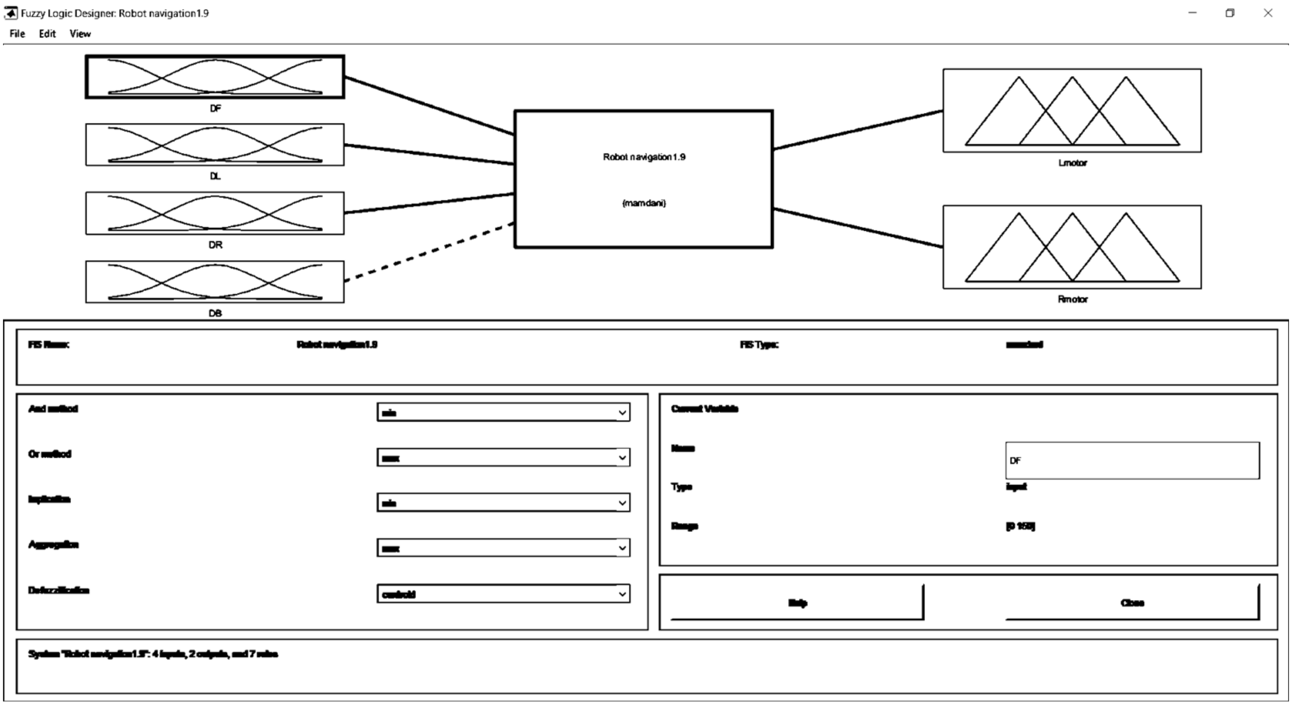
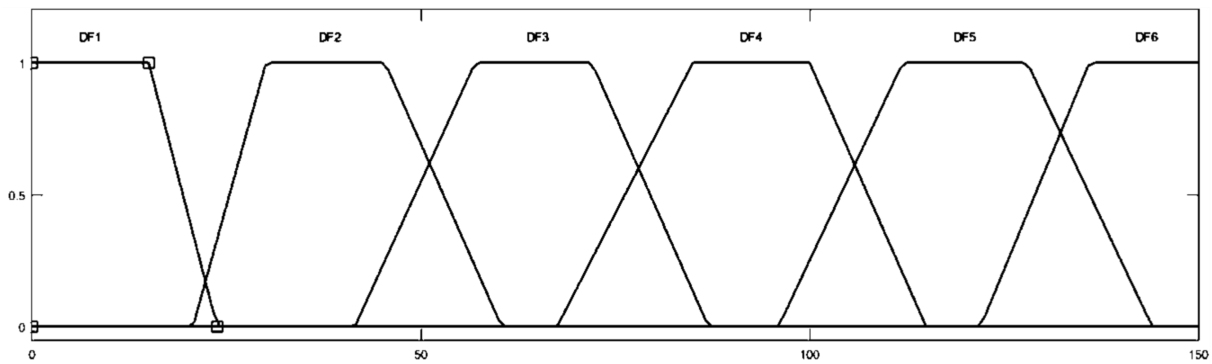


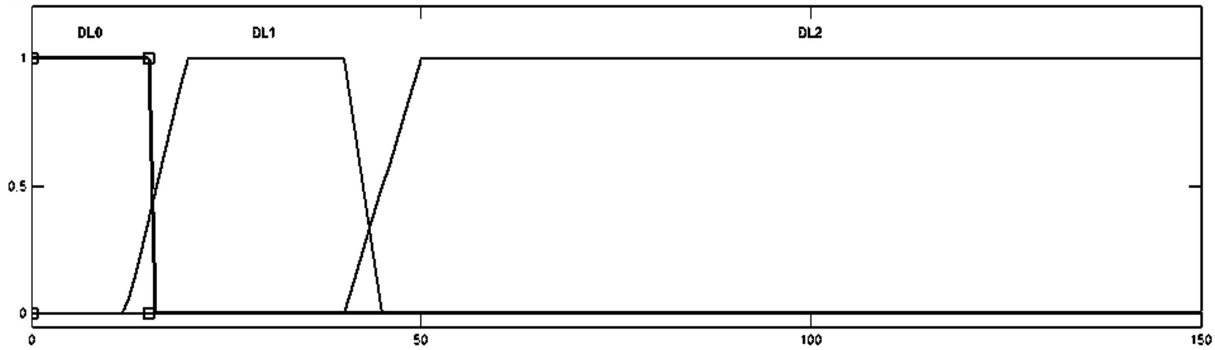
Fig. 2. Used fuzzy tool from Matlab for initial rule creation and simulations

As input variables are used the measured distances from the used sensors and after defuzzification the generated crisp values are converted to signals passed to left and right track motors.

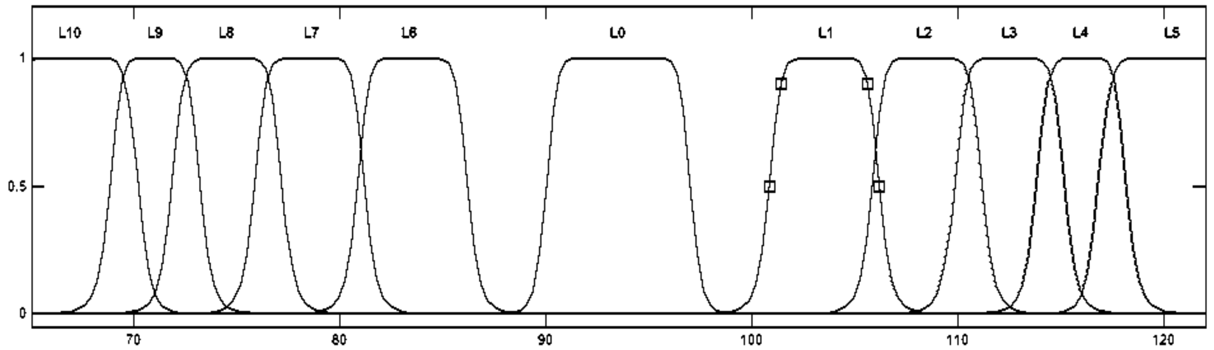
In Fig.3 are shown member functions for a fuzzy logic controller system created with used MATLAB fuzzy tool.



a/ front distance



b/ left and right distance



c/ output member function for the left and the right motor

Fig. 3. Input and output member function created with Matlab fuzzy tool.

On Fig. 3a are shown the terms used for the member function for the front distance. On the next figure (Fig. 3b) are the terms defining the member functions for right and left distance and on Fig. 3c are the terms for the output variables.

The definition of the input member functions and output member functions were defined in Table 1, Table 2 and Table 3 respectively. For the input functions is used trapezoid function for defining ranges, while output ranges are modeled with bell functions

Table 1 Front measured distance

Terms	Meaning	Range of member functions
DF1	Too Close	0 : 0 : 15 : 24
DF2	Close	21 : 30 : 45 : 60
DF3	Near	41 : 57 : 72 : 87
DF4	Medium	67 : 85 : 100 : 115
DF5	Far	96 : 112 : 128 : 144
DF6	Very Far	122 : 136 : 155 : 167

Table 2 Left or right measured distance

Terms	Meaning	Range of member functions
DL1/DR1	Too Close	0 : 0 : 15 : 15
DL2/DR2	Medium	12 : 20 : 40 : 45
DL3/DR3	Far	40 : 50 : 150 : 150

The numbers, selected as possible output values, are codes from the ASCII table, representing the small and capital English letters. They are chosen like this, because the robot, on which the fuzzy system is embedded, uses a protocol, which processes these ASCII codes and turns them into appropriate values for the PWM and direction signals.

Table 3 Output functions

Terms	Meaning	Range of member functions
L10	Maximum reverse speed	3.177 : 5.397 : 67
L9	Fast reverse speed	2.177 : 3.699 : 71
L8	Medium reverse speed	2.677 : 4.548 : 74.5
L7	Slow reverse speed	2.677 : 4.548 : 78.5
L6	Minimum reverse speed	2.677 : 4.548 : 83.5
L0	Full Stop	3.5 : 7 : 93.5
L1	Minimum forward speed	2.677 : 4.548 : 103.5
L2	Slow forward speed	2.677 : 4.548 : 108.5
L3	Medium forward speed	2.677 : 4.548 : 112.5
L4	Fast forward speed	2.177 : 3.699 : 116
L5	Maximum forward speed	3.677 : 6.247 : 120.5

In Table 4 is shown a summary of the chosen control rules for the speed of the left and the right motor depending on the desired direction of the robot's movement. When in front of the robot the distance from an obstacle is too close, right turn is prioritized. This can be seen from rule 5 in Table 4.

Table 4 Rules for defining speed of left/right motor

№	Left Sensor	Front Sensor	Right Sensor	Left Motor	Right Motor
1	Far	Very Far	Far	Max Forward	Max Forward
2	Far	Far	Far	Fast Forward	Fast Forward
3	Far	Medium	Far	Medium Forward	Medium Forward
4	Far	Near	Far	Slow Forward	Slow Forward
5	Far	Close	Far	Min Forward	Min Forward
6	Far	Too Close	Far	Medium Forward	Medium Reverse
7	Far	Too Close	Medium	Medium Reverse	Slow Forward
8	Far	Too Close	Too Close	Fast Reverse	Fast Forward
9	Medium	Too Close	Medium	Slow Forward	Slow Reverse
10	Medium	Too Close	Too Close	Min reverse	Min Forward
11	Too Close	Too Close	Far	Fast Forward	Fast Reverse
12	Too Close	Too Close	Medium	Slow Forward	Slow Reverse
13	Too Close	Too Close	Too Close	Full Stop	Full Stop

To simplify the process of the defuzzification is used “singletons” method according to equation (1):

$$W_A = (\mu_i . A_{mfi}(x) + \mu_j A_{mfj}(x) / (\mu_i + \mu_j) \tag{1}$$

where: W_A is weighted average value, which is passed to the PWM system of an on-board microcontroller; μ_i and μ_j are the degrees of the membership and A_{mfi} and A_{mfj} are the X-axis centroid points from the selected output member functions, according to the rules.

3. SIMULATION

For verifying the created rules there are tests, made with the MATLAB fuzzy logic toolbox. The results from the conducted simulations are shown on the figures 4 and 5.

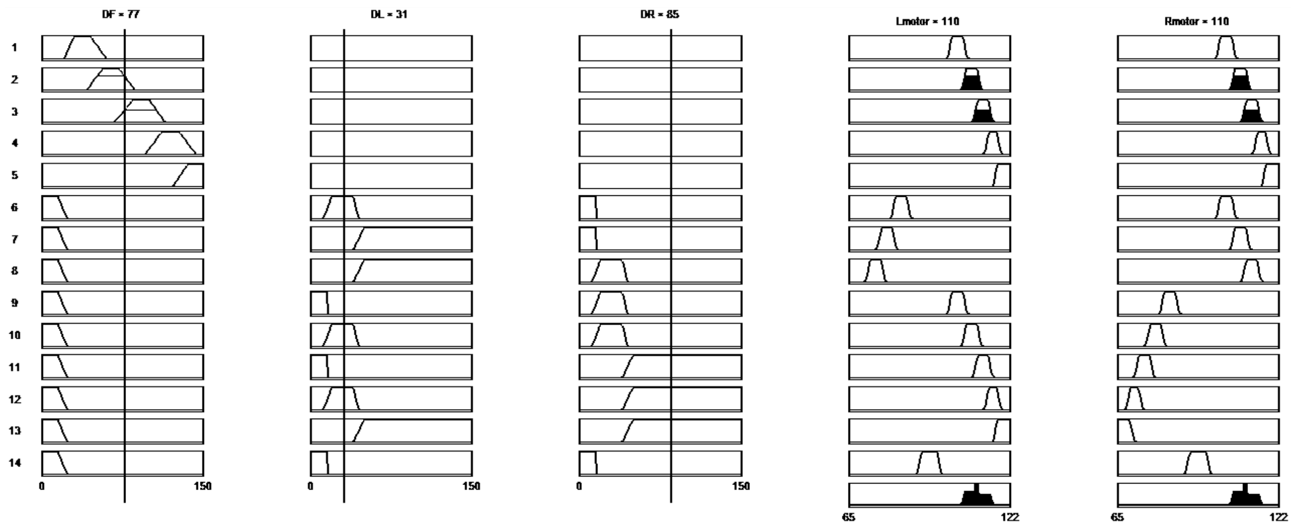


Fig. 4.Simulation results from Matlab fuzzy tool – moving in forward direction with medium speed

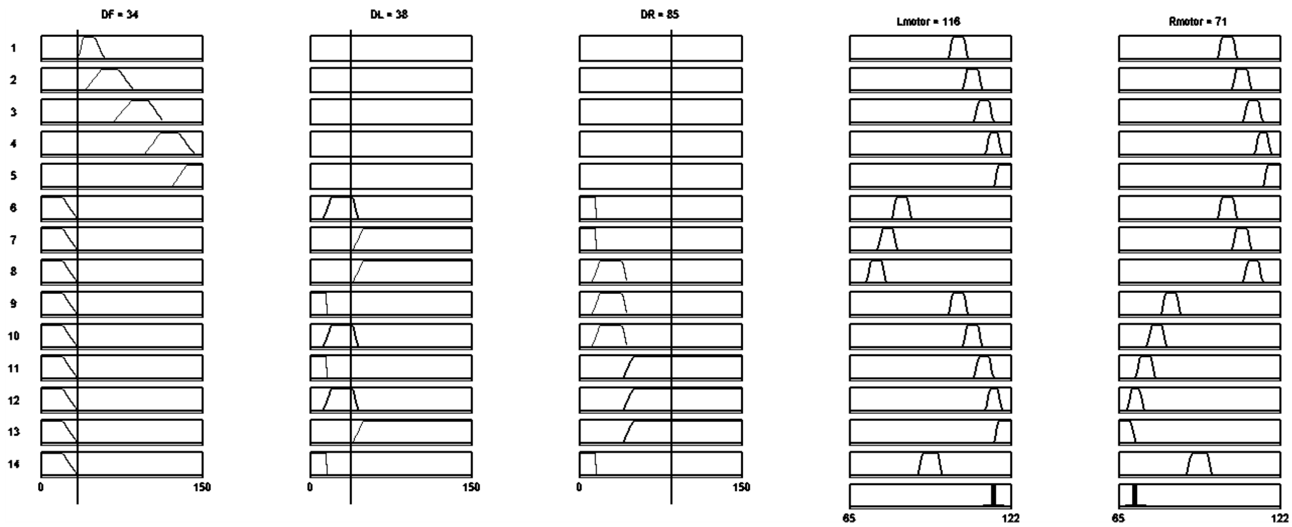


Fig. 5. Simulation results from Matlab fuzzy tool – tuning in right direction with fast speed

On Fig. 4 is shown a simulation of movement in a forward direction, where the first three columns represent the input values, respectively the measured distances from the front (DF), left (DL), right (DR) sensors. The last two columns are the output results for the left and right motors. On Fig. 5 is shown a similar simulation, when the robot has to turn right.

4. EXPERIMENTS

For verification of the created fuzzy controllers, a small tracked robot named AL1VE with 3D printed chassis is used (Fig. 6). The robot is equipped with two brushed motors and three infrared sensors with measurement ranges 20-150cm (front sensor) 10-80cm (right sensor) and 3-40cm for left sensor. Speed of the motors are controlled by PWM signals generated from custom-built microcontroller plate. The used microcontroller is PIC18F2550, which pass a PWM signal to H-Bridge drivers which control direction and speed of the motors. For communication with PC, the robot uses Bluetooth 4.0 and for powering 4 sells Lithium battery.

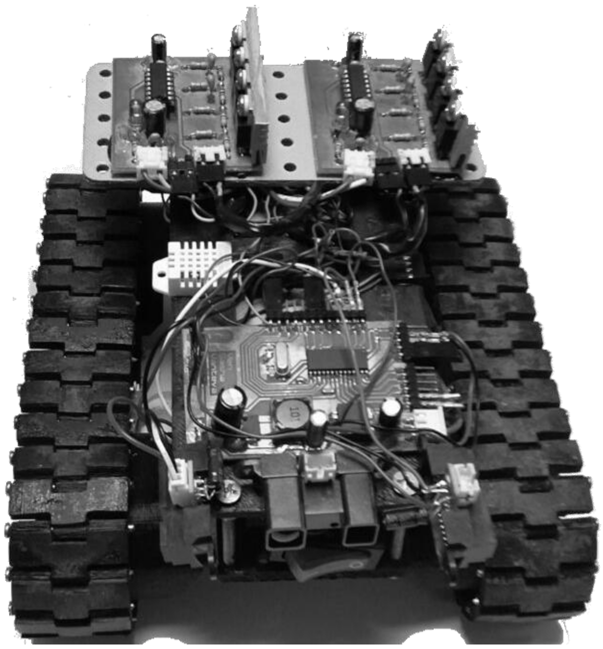
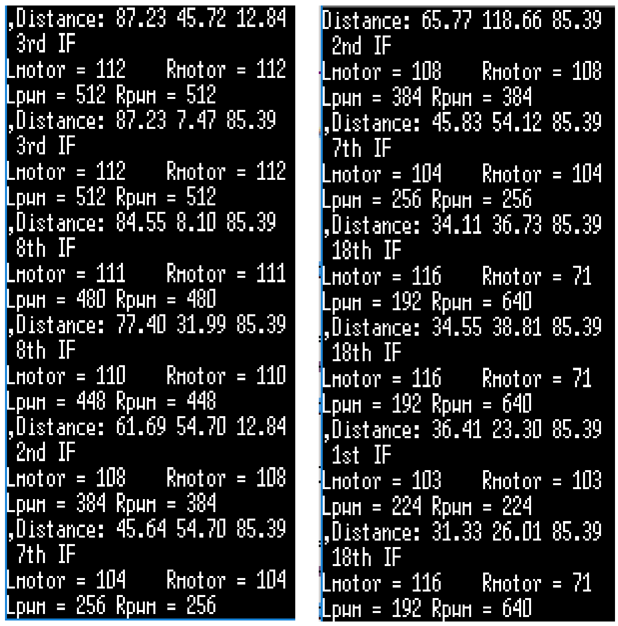


Fig. 6. Used mobile robot AL1VE for carried experiments



a/ moving forward *b) turning right*

Fig. 7. Experimental result with mobile robot ALIVE

In Fig. 7 are shown received data from robot according to the shown in fig.4 and Fig 5 simulation results. In line Distance, the first value is from the front sensor, the second is from left and the third is from the right sensor. In Fig.7a the robot moves in a forward direction with different speed according to the measured distance. In Fig. 7b, the robot is turning in right, because a maximum distance is measured from the right sensor.

5. CONCLUSIONS AND FUTURE WORK

In this paper is described the design of a fuzzy logic based controller, suitable for autonomous obstacle avoidance and path planning. The realized controller can control independently the speed and direction movement of a mobile robot. For verification of the results are made simulations and real experiments with a small 3D printed mobile robot, named AL1VE. In this case are used infrared sensors and because for this type of light source, the obstacles with transparent material are undetectable, in future work it is planned to upgrade the system with two other different types of sensors – 2D laser range finder (LRF) and ultrasonic sensors. Another thing, which should be optimized, is the rules of the fuzzy system.

REFERENCES

[1] RUSU, C. G., BIROU, I. T. (**2010**) Obstacle Avoidance Fuzzy System for Mobile Robot with IR Sensors. *10th, International Conference on Development and Application Systems*, Suceava, Romania, May 27-29, pp. 25-29

[2] Borenstein, J., Koren, Y. (**1988**) High-speed obstacle avoidance for mobile robots. *In Intelligent Control, Proceedings., IEEE International Symposium on*, Arlington, 24-26 Aug, pp. 382-384.

[3] Borenstein, J. and Koren, Y. (**1990**) Real-time Obstacle Avoidance for Fast Mobile Robots in Cluttered Environments. *In IEEE International Conference on Robotics and Automation*, Cincinnati, Ohio, May 13-18, pp.572-577

[4] Borenstein, J. and Koren, Y. (***1991***) The vector field histogram - fast obstacle avoidance for mobile robots. *IEEE Journal of Robotics and Automation*, vol. 7, no. 3, pp. 278-288

[5] Ulrich, I. and Borenstein, J. (**1998**) VFH+: Reliable Obstacle Avoidance for Fast Mobile Robots *In Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, Leuven, Belgium, May 16-21

[6] Ulrich, I. and Borenstein, J. (**2000**) VFH*: Local Obstacle Avoidance with Look-Ahead Verification, San Francisco, CA, April 2000, pp. 2505-2511

[7] Xi Li, Byung-Jae, Choi (**2013**) Design of Obstacle Avoidance System for Mobile Robot using Fuzzy Logic Systems, *International Journal of Smart Home*, Vol. 7, No. 3, May, pp. 321-328