Fuzzy Logic Based Design of Classical Behaviors for Mobile Robots in ROS Middleware

Veli BAYAR, Bora AKAR, Uğur YAYAN, H.Serhan YAVUZ
Electrical & Electronics Engineering, Eskisehir Osmangazi University
Eskisehir, Turkey
bayar.veli@gmail.com;borakar87@gmail.com; uguryayan@gmail.com;hsyavuz@ogu.edu.tr

Ahmet YAZICI
Computer Engineering, Eskisehir Osmangazi University
Eskisehir, Turkey
ayazici@ogu.edu.tr

***Abstract*— Autonomous mobile vehicles are used in many applications to realize special tasks. These tasks involve obstacle avoidance, target reaching and/or tracking. Such vehicles include the use of artificial intelligence to assist the vehicle's operator. Fuzzy logic can be used in the design of an autonomous vehicle to improve the classical control mechanisms. Classical robot control/decision mechanisms can give imperfect results due to sensor compensation errors or calculation costs. These drawbacks can be eliminated by using a combined fuzzy inference. In this study, we have modified the mobile robot ATEKS, which is an intelligent wheelchair, by introducing three fuzzy inference systems to realize goal reaching, obstacle avoidance and a controller for combined behavior selection. Designed fuzzy control system has been implemented on Robot Operating System (ROS) under Ubuntu 12.04 operating system and tested under Gazebo simulation platform. Simulation results verified faithful behavior outputs of ATEKS.**

Keywords—ROS; Fuzzy Logic, Robot Control, Gazebo, Obstacle Avoidance, Goal Reaching, Intelligent Wheelchair

I. INTRODUCTION

Nowadays the use of mobile vehicles becomes widespread over the indoor and outdoor applications. There are several types of autonomous and semi-autonomous robots that are being developed to be used in specific purposes. Some examples include space exploration, floor cleaning, waste water treatment, even soccer playing etc. Also there are some types of mobile robots help to assist disabled or elderly people by navigating from one location to another.

In general, classical indoor mobile robot behaviors include wall following, obstacle avoiding, docking, passing doors etc. that are implemented by means of a control mechanism. When these sensor based behaviors are realized by classical robot control mechanisms, there could be sensor oriented errors or processing load problems. To overcome these problems, fuzzy logic based robot control architectures are being developed on different kind of hardware and software units and these architectures make great success [1-9].

Thongchai and Kawamura [1] represents a robot named HelpMate which is constructed with sonar sensor has a fuzzy logic control mechanism. The priority of the behavior is decided by the software that is coded with visual basic programming language. In [2], fuzzy logic based real time obstacle avoidance behavior was implemented on an embedded system to use with robot navigation. In a similar work, a fuzzy logic based obstacle avoidance behavior was implemented on a single chip microcomputer [3]. Another approach shows that a robot named OTOROB has a fuzzy logic control mechanism on its arm to avoid obstacles [4]. In this study, first the system was tested on MATLAB environment, and then it was implemented using C programming language on embedded system. There are other fuzzy logic based obstacle avoiding control systems implemented on MATLAB, like the robot named ANROV [5]. At the same time some researchers designed fuzzy logic based mobile robot control mechanisms using LABVIEW environment [6]. Another study presents a fuzzy logic based obstacle and collision avoidance systems that are implemented on soccer playing robots [7]. The system was tested on MATLAB environment, and implemented using C and C++ programming language on embedded system. In [8], a novel path following behavior that ensures variety of constraints is improved for a non-holonomic robot. Another paper cites a fuzzy logic controller that achieves robot navigation just using starting and goal point information in an unknown environment [9].

When technical literature is revised, a big percentage of the designed fuzzy logic based control mechanisms are modeled on MATLAB environment, then implemented using C/C++ programming languages, LABVIEW or embedded systems. However, we have not yet seen any study that uses Robot Operating System (ROS) [10], which is becoming very popular around the robotic applications, and fuzzy logic together to develop a control mechanism [11].

In this paper, we designed a fuzzy logic based control system for ATEKS, which is an intelligent wheelchair (a mobile robot) developed in our lab [12]. The design was based on modifying the following three basic behaviors: goal reaching behavior, obstacle avoidance behavior and a combined behavior using sonar sensors by means of fuzzy inference mechanism. The system runs on Ubuntu 12.04 OS using ROS middleware. Fuzzy Inference systems (FIS) are designed and converted to C++ codes using qtfuzzylite tool and fuzzylite library [13]. The main application which includes the fuzzy inference systems in terms of C++ codes is developed in ROS. The behaviors are tested using an exact model of ATEKS with GAZEBO simulation software [14]. The organization of the paper is given as follows. Chapter 2 gives information about ATEKS mobile robot, chapter 3 introduces the implemented behaviors, designed rules, inference systems and fuzzylite library, chapter 4 and 5 include the GAZEBO simulation test results in details and conclusion.

II. INTELLIGENT WHEELCHAIR (ATEKS)

ATEKS, shown in Fig. 1, is an intelligent wheelchair which consists of high-tech control mechanisms, low cost sensor equipment and open source software (ROS, GAZEBO and ANDROID). Moreover, it is not only a multi-purpose platform which provides a research area for mobile robots, human – machine interaction, human – robot interaction, sensor based control applications, but also a great assisting system for the disabled or elderly people in their daily lives [12].

There are three main parts of ATEKS: Intelligent Controlling Unit (AKB) which consists of a low-level controller and a high-level controller, Mission Planning Unit with Graphical User Interface (KAB) and sensors for implementing intelligent robotic behaviors.

Fig. 1. Intelligent wheelchair ATEKS

There are 10 ultrasonic sensors, 2 encoders, joystick / motor driver, a positioning unit called GAB, and Microsoft Xbox 360 Kinect sensor located on ATEKS to realize robotic behaviors. The hardware components of the ATEKS are shown in Fig. 2.

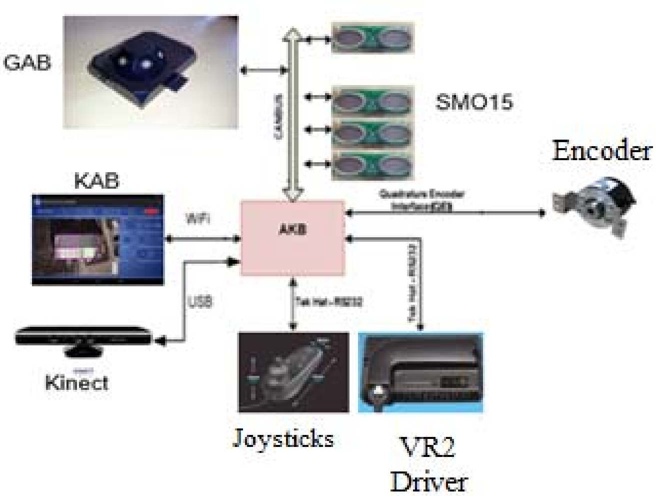

Fig. 2. ATEKS hardware components

ATEKS can detect obstacles in indoor environments by using sonar sensors called SMO15 which has a range of between 5 cm to 500 cm. The unit GAB which is shown in Fig. 3 is used with ATEKS platform to calculate position in centimeters accuracy. In fact, GAB is a part of the İÇKON system which is an indoor positioning system that uses only ultrasonic signals [15]. It has been designed as multiple cells. In each cell of this structure, a SESKON system operates [16].

Fig. 3. GAB unit

III. FUZZY LOGIC BASED CONTROL APPLICATION ON ATEKS

Rule based fuzzy inference systems can be designed for ATEKS's fuzzy logic based control that confirms goal reaching and obstacle avoidance behaviors. In this study, we used Mamdani fuzzy inference schemes [17] in our design. fuzzylite library is used to implement our design in C++ programming language. Design details are given in the following subchapters.

A. Fuzzy Inference System Design

Fuzzy inference system (FIS) consists of fuzzy logic based input – output membership functions, fuzzy rules, fuzzification and defuzzification algorithms. The general fuzzy inference system is summarized in Fig. 4. The inputs of the system are handled with fuzzy rules and outputs are produced through the fuzzy logic principles.

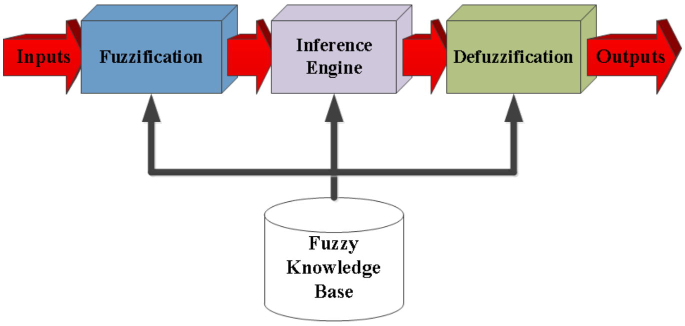

Fig. 4. A general fuzzy inference system block diagram

Our fuzzy rule based design aims to model three separate behaviors: obstacle avoidance behavior, goal reaching behavior and a combined behavior. Obstacle avoidance behavior has 3 inputs and 2 outputs. Inputs are: left, right and front sonar distances. Outputs are the linear and angular velocity that will be applied on ATEKS. The defined membership functions of inputs and outputs are summarized in Table I. In the table, “trapmf” stands for the trapezoidal membership functions with 4 variables given in parenthesis; “tripmf” stands for the triangular membership functions of three variables given in parenthesis.

TABLE I. Obstacle Avoidance Parameters

Obstacle Avoidance			
Variables		**Membership Functions and Values**	
Inputs	LeftSonar	veryNear	trapmf([-5000 0 200 600])
	FrontSonar	near	trapmf([100 1000 1500 2000])
	RightSonar	far	trapmf([1250 2000 5000 6000])
Outputs	Linear Velocity	stop	trapmf([-0.359 0 0.1 0.2])
		slow	trimf([0.1 0.2 0.5])
		fast	trapmf([0.2 0.5 1 1.36])
	Angular Velocity	rightFast	trapmf([-0.1 -0.5 -0.2 -0.074])
		rightSlow	trimf([-0.2 -0.075 0])
		stop	trimf([-0.075 0 0.075])
		leftSlow	trimf([0.002 0.07 0.2])
		leftFast	trapmf([0.075 0.2 0.5 1])

The inference system designed for goal reaching behavior has 2 inputs and 2 outputs. Inputs are the distance to the desired location and the angle between the ATEKS. Outputs are the linear and angular velocity. The defined membership functions of inputs and outputs are shown in TABLE II.

TABLE II. Goal Reaching Behavior Parameters

Goal Reaching			
Variables		**Membership Functions and Values**	
Inputs	goalDistance	veryNear	trapmf([-10000 0 1000 3000])
		near	trimf([1000 2000 6000])
		middle	trimf([3000 6000 10000])
		far	trimf([6000 10000 20000])
		veryFar	trapmf([10000 18000 20000 30000])
	goalAngle	negativeBig	trapmf([-4.71 -3.14 -1.884 -0.942])
		negativeMiddle	trimf([-3.14 -0.942 -0.314])
		negativeSmall	trimf([-0.942 -0.314 0])
		zero	trimf([-0.314 0 0.314])
		pozitiveSmall	trimf([0 0.314 0.942]])
		pozitiveMiddle	trimf([0.314 0.942 3.14])
		pozitiveBig	trapmf([0.942 1.884 3.14 4.71])
Outputs	Linear Velocity	verySlow	trapmf([-0.25 0 0.02 0.05])
		slow	trimf([0.02 0.05 0.15])
		middle	trimf([0.05 0.15 0.3])
		fast	trimf([0.15 0.3 0.4])
		veryFast	trapmf([0.3 0.4 0.5 0.75])
	Angular Velocity	negativeBig	trapmf([-0.45 -0.3 -0.18 -0.06])
		negativeMiddle	trimf([-0.18 -0.06 -0.015])
		negativeSmall	trimf([-0.06 -0.015 0])
		stop	trimf([-0.015 0 0.015])
		pozitiveSmall	trimf([0 0.015 0.06]])
		pozitiveMiddle	trimf([0.015 0.06 0.18])
		pozitiveBig	trapmf([0.06 0.18 0.3 0.45])

There are 405 possible rules for obstacle avoidance inference system while there are 1225 rules for goal reaching behavior inference system. To reduce the calculation time, the optimal amount of rules has to be generated and selected to achieve both of the behaviors. So, the optimum rules are selected manually till we observe satisfactory results for our inference systems. qtfuzzylite tool is used to visualize results. Finally, 27 rules are chosen to model the obstacle avoidance, whereas 12 rules are selected for goal reaching behavior. The rules for goal reaching behavior is shown and summarized in TABLE III. Similarly, the obstacle avoidance behavior rules are summarized in Fig. 5 as a three dimensional structure to make sense.

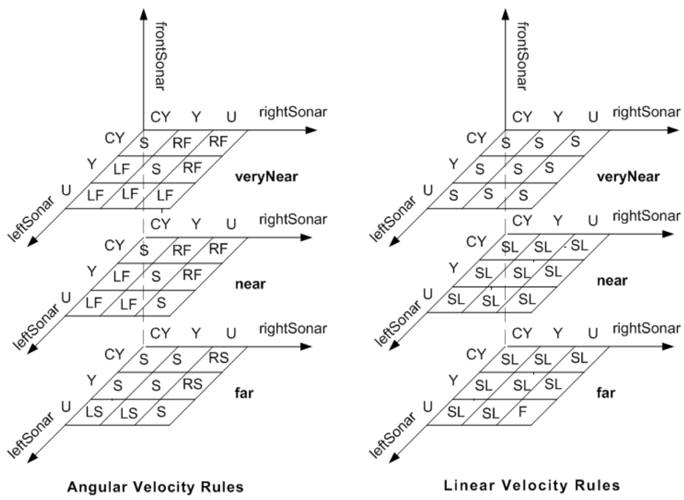

Fig. 5. Obstacle avoidance behavior rules

TABLE III. Goal Reaching Behavior Rules

Rule	Inputs		Outputs	
	Distance	Angle	Linear Velocity	Angular Velocity
1	VN	x	VS	x
2	N	x	S	x
3	M	x	M	x
4	F	x	F	x
5	VF	x	VF	x
6	x	NB	x	PB
7	x	NM	x	PM
8	x	NS	x	PS
9	x	S	x	S
10	x	PS	x	NS
11	x	PM	x	NM
12	x	PB	x	NB

In addition, another fuzzy inference system is developed for combining obstacle avoidance and goal reaching behaviors. This system has 2 inputs and one output. Inputs are selected as obstacle distances and goal distance, while the output is behavior weight. The output decides weight value which controls ATEKS's linear and angular velocities. In TABLE IV, combined behavior FIS parameters are shown.

TABLE IV. Combined Behavior Rules

Combine Behaviours			
Variables		Membership Functions and Values	
Inputs	obstacleDistance	near	trapmf([-7200 0 2000 4000])
		middle	trapmf([2000 4000 8000 12000])
		far	trapmf([8000 12000 20000 27200])
	goalDistance	near	trimf([-1120 -5.55 3000])
		middle	trimf([2500 3500 4000])
		far	trimf([3500 5000 7500])
Outputs	behaivorWeight	veriSmall	trimf ([-0.4 0 0.1])
		small	trimf([0 0.1 0.2])
		middle	trimf([0.1 0.2 0.4])
		big	trimf([0.2 0.4 0.7])
		veryBig	trapmf([0.4 0.7 1 1.1])

Combined Behavior FIS has 9 rules that shown in TABLE V. Those rules are initiatively constructed and the best are selected according to conducted tests.

TABLE V. FIS for Combined Behaviors

Rule	Inputs		Outputs
	goalDistance	obstacleDistance	behavior Weight
1	N	N	VS
2	M	N	VS
3	F	N	VS
4	N	M	S
5	M	M	S
6	F	M	S
7	N	F	M
8	M	F	M
9	F	F	M

Behavior weight (τ) affects on ATEKS linear and angular velocities related with the equations shown (1) and (2).

$$V_{applied} = (V_{GR} * \tau) + (V_{OA} * (1 - \tau)) \quad (1)$$

$$W_{applied} = (\mathrm{W}_{GR} * \tau) + (\mathrm{W}_{OA} * (1 - \tau)) \quad (2)$$

The behavior weight is inspired from the study of Abdallah and Dan [18]. Apart from that paper, 9 rules are established to control goal reaching and obstacle avoidance behaviors which are completely different.

The fuzzy inference outputs have to be digitalized to apply the control outputs on ATEKS. Because of that, the centroid method which is widely preferred and used for this purpose is selected.

B. *Fuzzy Control System with ROS*

The application which is developed to achieve obstacle avoidance and goal reaching behaviors is built on ROS, fuzzylite and GAZEBO tools. qtfuzzylite user interface is used to produce fuzzy inference system designs that are mentioned in section 3.1. ROS is the core element of these tools and has a bridge to test the results of inference systems in GAZEBO. In addition, the fuzzylite defines a bridge between the designed FIS and ROS. In Fig .6, qtfuzzylite user interface is given.

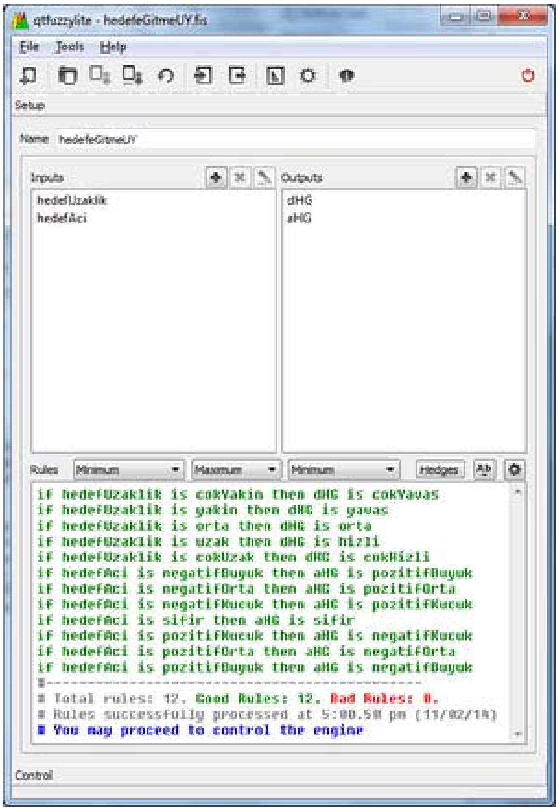

Fig. 6. qtfuzzylite user interface

Control system structure of the application is shown in Fig. 7. The inputs for the inference systems are obtained from GAZEBO where the ATEKS model runs on. These obtained data are embedded to ROS where inference algorithms execute the outputs of the fuzzy inference systems. These inputs are converted and being published with proper ros topics to angular and linear velocity for the ATEKS which it has to follow on GAZEBO environment.

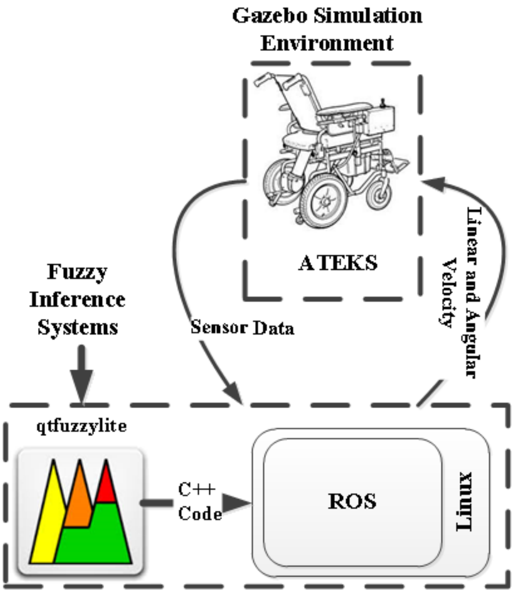

Fig. 7. Control system structure

IV. TESTS ON GAZEBO

To test the application, 20 * 20 square meters area is drawn on GAZEBO and obstacles are placed in test area which is shown in Fig. 8. Model of the ATEKS is added to the map. The model is exact the same copy of the real ATEKS itself including the sensor measurements, maximum and minimum velocity boundaries. At the beginning, the goal reaching and obstacle avoidance behaviors are tested on the map individually. Then, the goal reaching behavior including obstacle avoidance is tested within two different scenarios.

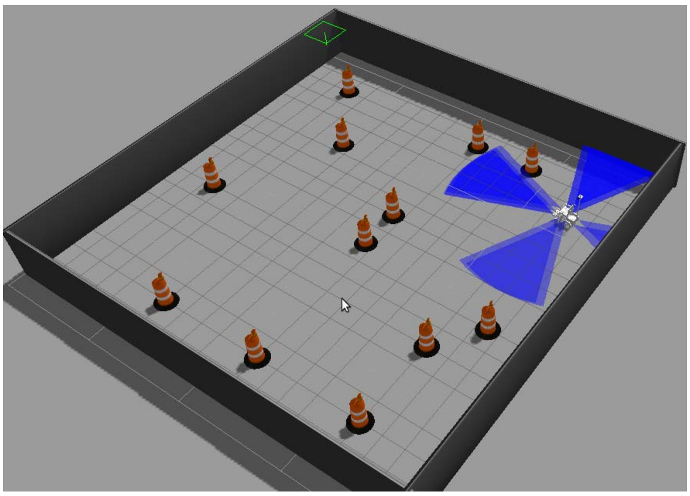

Fig. 8. GAZEBO test environment

For the obstacle avoidance behavior, the test area is shrunk into a small room which has 4 * 7 square meters. ATEKS has been trapped there. Then, the model is located in a random spot on given room. As a demonstration of ATEKS movement with the obstacle avoidance behavior, the logged position information during movement is shown in the Fig. 9. The blue square indicates the starting point of movement.

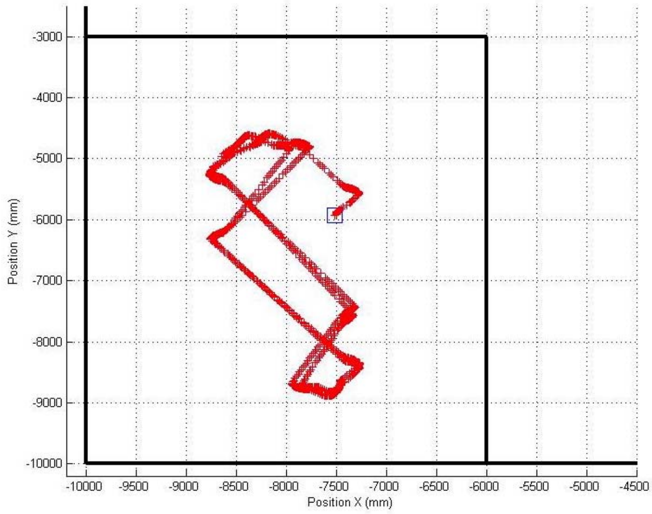

Fig. 9. Obstacle avoidance behavior position results

For the goal reaching behavior, a starting point (-9160 mm, -8400 mm) and goal point (-6400 mm, -4500 mm) are given. For this scenario, all obstacles are removed from the test area. While ATEKS moves to the goal successfully, the position and linear – angular velocity of the ATEKS is recorded to observe outputs.

The angular and linear velocities that ATEKS follows inside the test area with the goal reaching behavior are shown in Fig. 10. It is obvious that, the linear velocity of ATEKS is decreased while approaching to the goal point; the angular velocity converges to the zero and approximately settles at that point.

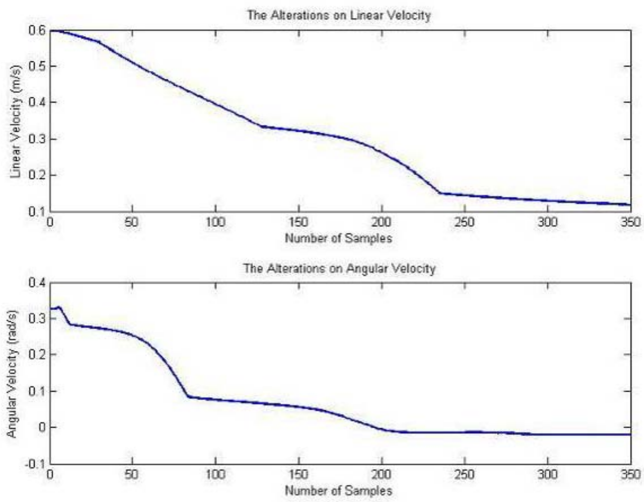

Fig.10. Angular and linear velocities of goal reaching behavior

Recorded position information during the goal reaching behavior is shown in the Fig. 11. Starting and goal points are specified as blue squares. According to the figure, it is shown that ATEKS navigates through the path and gets to the desired goal point without any oscillation.

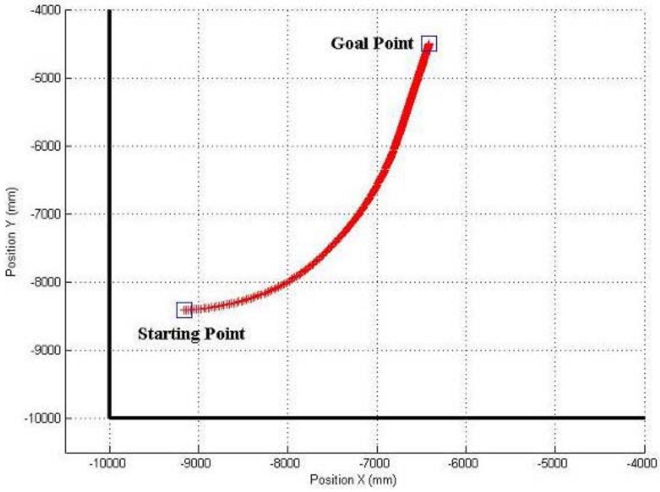

Fig.11. Goal reaching behavior position results

Goal reaching behavior including obstacle avoidance is tested with two different scenarios. The difference between two scenarios is choosing starting and goal points dissimilarly and placing obstacles in different locations on the test area. For two scenarios, all outputs and crucial intermediate outputs are recorded.

For the first scenario, the starting and goal points are (7530 mm,-7710 mm) and (-4340 mm, 6510 mm) respectively. ATEKS is placed towards the wall to put emphasize on obstacle avoidance behavior. Fig. 12 and Fig. 13 demonstrate the output velocities for obstacle avoidance and goal reaching behaviors. It is seen from the graphs that the two behaviors have different characteristics as expected.

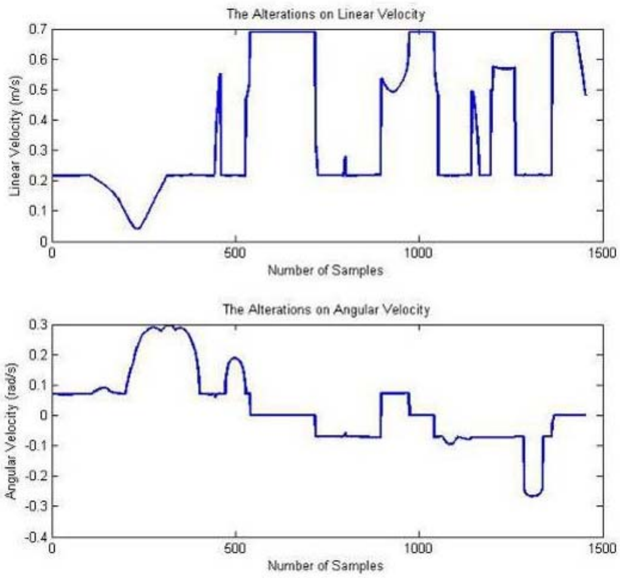

Fig. 12 Obstacle avoidance behavior velocities for scenario I

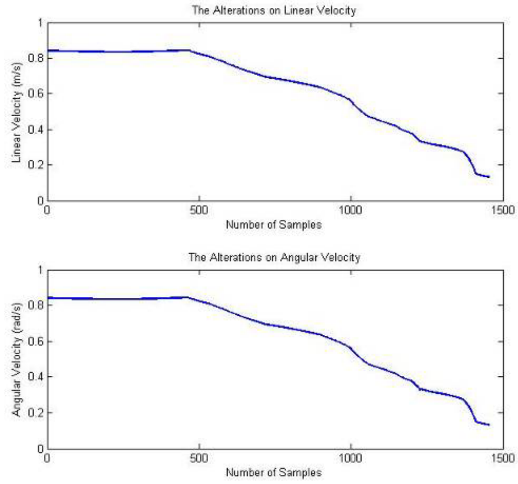

Fig. 13. Goal reaching behavior velocities for scenario I

The output velocities are combined by using τ coefficient which is shown in Fig 14.

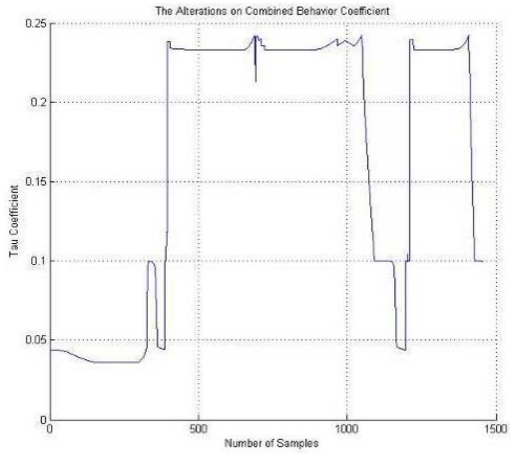

Fig. 14. τ coefficient variation for scenario I

Combining two behaviors' velocity outputs with calculated τ coefficient gives the final velocities to drive ATEKS are shown in Fig 15.

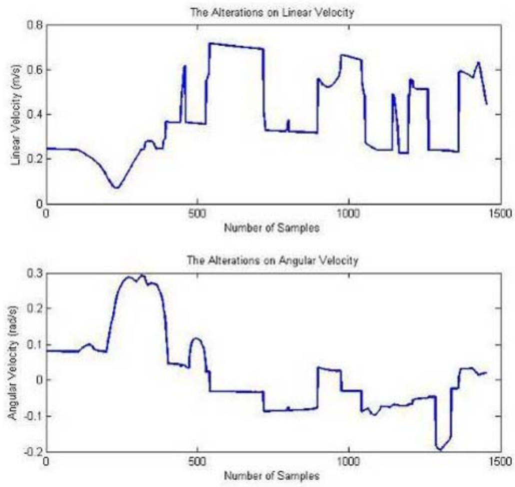

Fig. 15. Combined behavior velocities for scenario I

It is obvious that the obstacle avoidance behavior is dominant to goal searching behavior due to obstacles placed in test area. According to applied linear and angular velocities, ATEKS navigates to the goal point through obstacles shown in Fig. 16. ATEKS arrives the goal at the point (-4391 mm, 6608 mm). 11 cm Euclidian distance error is measured to given goal point at the beginning of movement.

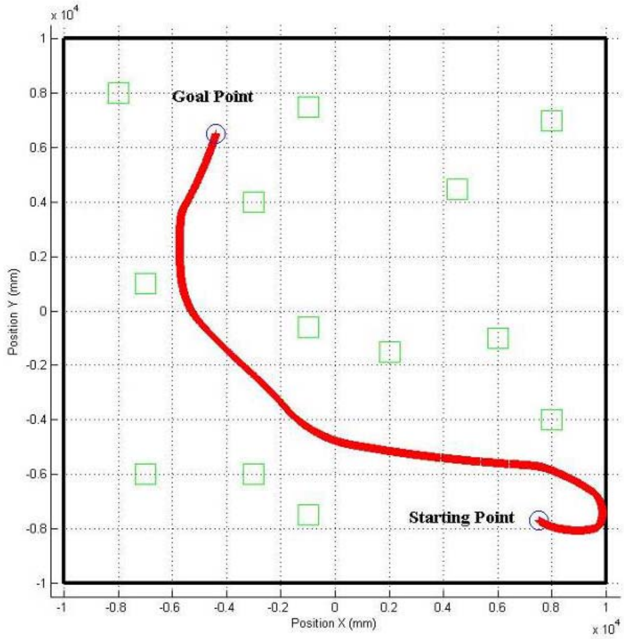

Fig. 16. ATEKS position information of scenario I

For the second scenario, the starting and goal points are (-5160 mm, -8460 mm) and (6890 mm, 7730 mm) respectively. Similarly, different characteristics for each behavior output velocities are produced from fuzzy inference systems shown in Fig. 17 and Fig 18.

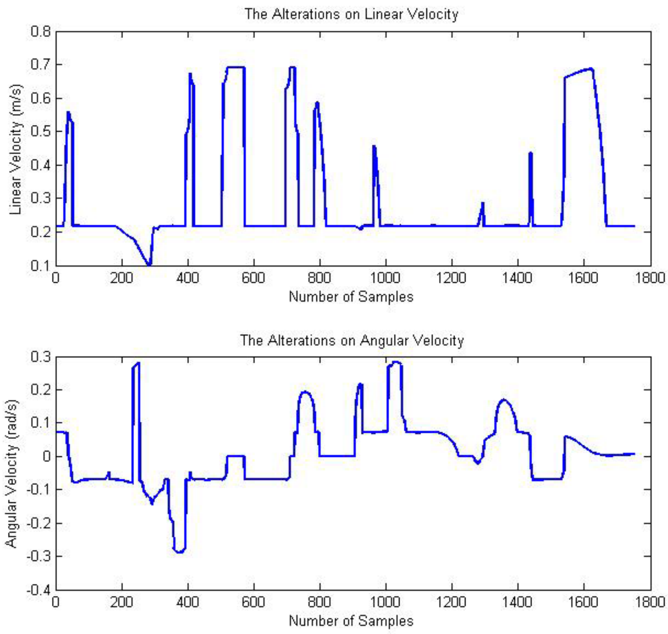

Fig. 17 Obstacle avoidance behavior velocities for scenario II

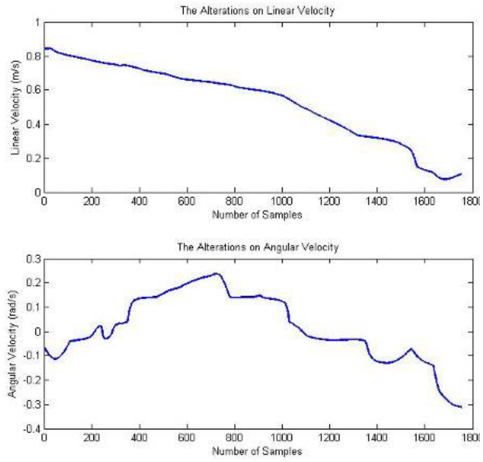

Fig.18 Goal reaching behavior velocity values for scenario II

While observing output velocities, τ coefficient is calculated from fuzzy inference system is applied to these velocities. The alterations on calculated coefficient is shown in Fig. 19.

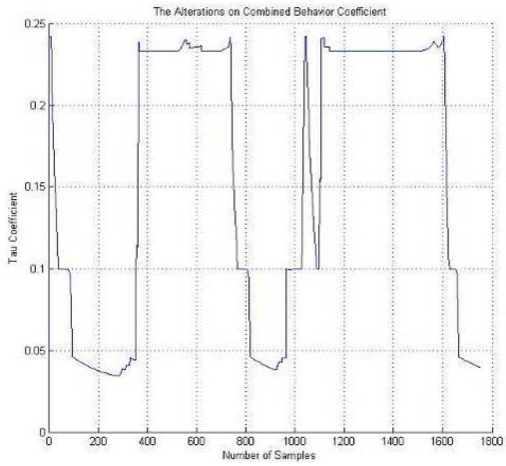

Fig. 19. τ coefficient variation for scenario II

The obstacle avoidance is dominant as desired shown in Fig. 20 due to obstacles in test area.

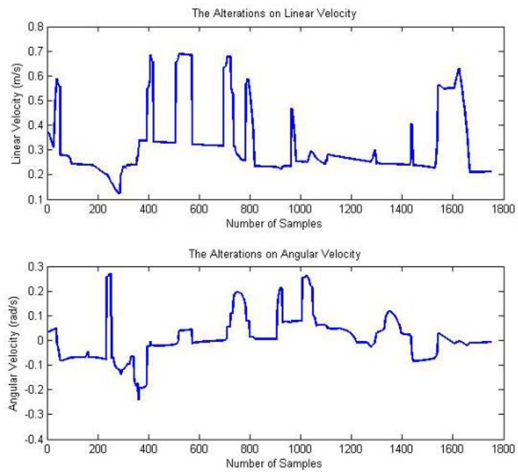

Fig. 20. Combined behavior velocities for scenario II

According to Scenario II final outputs, ATEKS can stil drive through obstacles and walls in a convenient way shown in Fig. 21. ATEKS position information at the goal point is given as (6817 mm, 7682 mm) with the error of almost 9 cm Euclidian distance from the desired goal point.

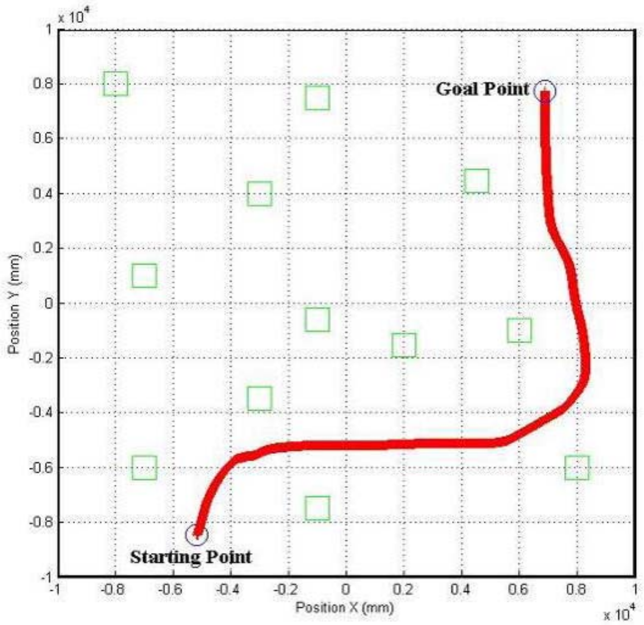

Fig. 21. ATEKS Position Information of Scenario II

V. RESULTS AND FUTURE WORK

In this paper, we designed fuzzy inference mechanisms to realize three basic behaviors of ATEKS. Different from the similar applications, we introduced a fuzzy logic based robot control system running on ROS architecture. First, fuzzy inference systems for goal reaching and obstacle avoidance behaviors were designed and tested individually, and then the outputs of these two behaviors were combined with a weight coefficient which is the output of another fuzzy inference system related with behaviors. We tested the designed inference systems on GAZEBO environment. The results demonstrated that ATEKS can perform mentioned behaviors successfully.

As a future work, we plan to realize more complex behaviors, such as door passing, docking and wall following behaviors by using fuzzy logic and adopt these behaviors to be used with the basic ones presented in this paper to make a full fuzzy logic based mobile robot control.

ACKNOWLEDGMENTS

This paper is supported by İnovasyon Mühendislik, Teknoloji Geliştirme, Danışmanlık, San. ve Tic. Ltd. Şti.

REFERENCES

[1] S. Thongchai, K. Kawamura, “Application of Fuzzy Control to a Sonar-Based Obstacle Avoidance Mobile Robot”, Proceedings of the 2000 IEEE International Conference on Control Applications Anchorage,Alaska, USA, September 25-27, 2000.

[2] S. Siddaiyan, R. W. Arokiasamy, “DVFH – VFH*: Reliable Obstacle Avoidance for Mobile Robot Navigation Coupled with A* Algorithm Through Fuzzy Logic and Knowledge Based Systems”, 2012 International Conference on Computer Technology and Science (ICCTS), IPCSIT vol. 47 IACSIT Press, Singapore, 2012.

[3] W.Ling, “Fuzzy Control of Obstacle Avoidance for Mobile Robot Based on Single Chip Microcomputer”, International Conference on Artificial Intelligence and Software Engineering (ICAISE), 2013.

[4] M. Mariappan, T. Ganesan, V. Ramu, “Fuzzy Controlled Obstacle Avoidance System (OAS) for OTOROB Flexible Robotic Arm Safety System”, International Journal of Robotics Research and Development (IJRRD), Vol.1, Issue 1 Dec 2011 1-20.

[5] C. Y. Wong, U. Qidwai,”Intelligent Sensor Network for Obstacle Avoidance Strategy”, IEEE 2005.

[6] P. Ponce, A. Molina, R. Mendoza, “Wheelchair and Virtual Environment Trainer by Intelligent Control”, INTECH, 2012

[7] M. H. Araghi, M. B. Menhaj, “A Combined Collision Cone and Fuzzy Based Obstacle Avoidance Algorithm for the Small Size Soccer Robots”, In Proceedings of ARCS, 2008, 151-158.

[8] On S.,Yazıcı A., A reference governor approach for path tracking of nonholonomic mobile robot, INISTA 2011, International Symposium on Innovations in Intelligent Systems and Applications, 15-17June 2011, İstanbul Turkey, pp. 550-554.

[9] Parlaktuna O., Bakla B., Özkan M., Yazıcı A., "Bulanık Mantık Yöntemiyle Gezgin Robot Navigasyonu", 16-18 Mayıs 2005, SİU 2005 (IEEE 13. Sinyal İşleme ve İletişim Uygulamaları), 2005

[10] Robot Operating System, www.ros.org

[11] X. Li, B. J. Choi, “Design of Obstacle Avoidance System for Mobile Robot using Fuzzy Logic Systems”, International Journal of Smart Home, Vol. 7, No. 3,May, 2013

[12] B. Akar, U. Yayan, H. Yücel, V. Bayar, A. Yazici, ”Akıllı Tekerlekli Sandalye Kontrolü İçin Sonlu Durum Makinesi Tasarımı”, TOK 2013

[13] Juan Rada-Vilela. fuzzylite: a fuzzy logic control library, 2014. URL http://www.fuzzylite.com

[14] Gazebo Simulator, http://gazebosim.org

[15] C. Yeniceri, T. Tuna, A. Yazici, H. Yucel, U. Yayan, V. Bayar, “A Smart Solution for Transmitter Localization”, Innovations in Intelligent Systems and Applications (INISTA), pp 1-5, 2013

[16] A. Yazici, U. Yayan and H. Yucel, “An ultrasonic based indoor positioning system,” Innovations in Intelligent Systems and Applications (INISTA), pp.585-589, 2011.

[17] T. J. Ross, “Fuzzy Logic with Engineering Applications”, McGraw-Hill, 1995.

[18] K. B. Abdallah, Z. Q. Dan, “A Fuzzy Logic Behavior Architecture Controller for a Mobile Robot Path Planning in Multi-Obstacles Environment”, Research Journal of Applied Sciences, Engineering and Technology 5 (14); 3835-3842, 2013