Fuzzy Behavior Based Mobile Robot Navigation

Mehdi Ravangard
Computer Engineering Department
Kharazmi University
Tehran, Iran
ravangard@khu.ac.ir

***Abstract*— In the following paper an algorithm is proposed using behavior based robotics, fuzzy logic and Braitenberg vehicles concepts to control a mobile robot. This algorithm tries to navigate the robot from a starting point to a target in the shortest possible path using two simple behaviors: goal seeking and obstacle avoidance. The algorithm is simulated on Khepera III robot in Webots simulator.**

Keywords*— *Mobile robot; Navigation; Fuzzy; Behavior

I. INTRODUCTION

Since the introduction of fuzzy logic theory to the world of science and technology, it has been used for many applications from a simple washing machine controller to train controllers and also in robotics. One of the applications of fuzzy logic theory in robotics is in mobile robot navigation. Different fuzzy methods are used to control a wheeled mobile robot [5]. One of the methods most used in controllers is Takagi-Sugeno-Kang fuzzy inference engine [2].

Behavior based robotics takes advantage of living creatures as well as human beings. It combines different behaviors to achieve a desired result. There are many ways to combine behaviors. One of the structures of behavior based robotics in wheeled mobile robot controllers is called motor schema which takes advantage of potential field method [6].

Braitenberg proposed a reactive concept in mobile robot control. He designed vehicles which use sensory-motor transformations to control vehicles with such a simple configuration [7]:

- Two motors and two light sensors.
- Left light sensor affects left motor and right light sensor affects right motor.
- Speed of each motor is proportional to the amount of sensed light.

Many complex behaviors could be obtained by using more sensors and different ways of sensor-motor connections.

The algorithm presented in this paper is a combination of three concepts above and is suitable for navigation of wheeled mobile robots. The rest of this paper is organized as follows: Section II explains behavior based robotics. Section III presents the robot's kinematic model. In section IV the proposed algorithm is described. Simulation results are given in section V. Finally section VI presents conclusion of this article.

II. BEHAVIOR BASED ROBOTICS

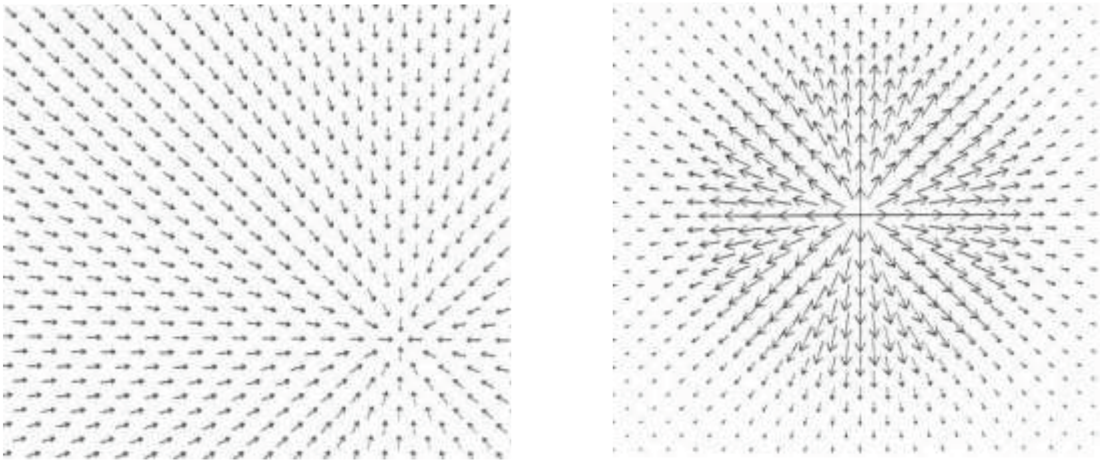
The simplest way to show the relation between stimulus and response is Stimulus-Response (SR) Diagrams as shown in Fig. 1.

As can be seen a behavior receives stimulus and generates appropriate response. This diagram could be mathematically formed as $b(s)=r$ which means behavior b generates response r for a given stimulus s [6].

To encode the behavioral response that the stimulus should evoke, we must create a functional mapping from stimulus space to motor space. Each active behavior produces a functional mapping between stimulus domain and response range that defines a behavioral function. This function must be defined over all relevant types of stimulus. There are two types of behavioral encoding: "discrete encoding" and "continuous functional encoding". In discrete encoding the behavioral function is defined for all relevant types of stimulus and produces responses from discrete set of all possible responses. In this case, the behavioral function consists of finite set of situation-response pairs. To represent behavioral function, rule based systems with rules in *IF antecedent Then consequent* form are used. Continuous functional encoding produces continuous response that allows the robot to have an infinite space of potential reactions. Instead of having a countable set of responses, a mathematical function transforms the sensory input to a behavioral reaction. One of the most common methods of continuous response implication is based on "Potential Field" method. This method generates a field that represents a navigational space based on an arbitrary potential function. The classic function used is Coulomb's electrostatic attraction; such that potential force has inverse relation with square of distance between the robot and objects in the environment. The goals are treated as attractors and the obstacles are treated as repulses. Potential fields for a goal and an obstacle is shown in Fig. 2.

Fig. 1. Simple SR Diagram

978-1-4673-8545-9/15/$31.00 ©2015 IEEE



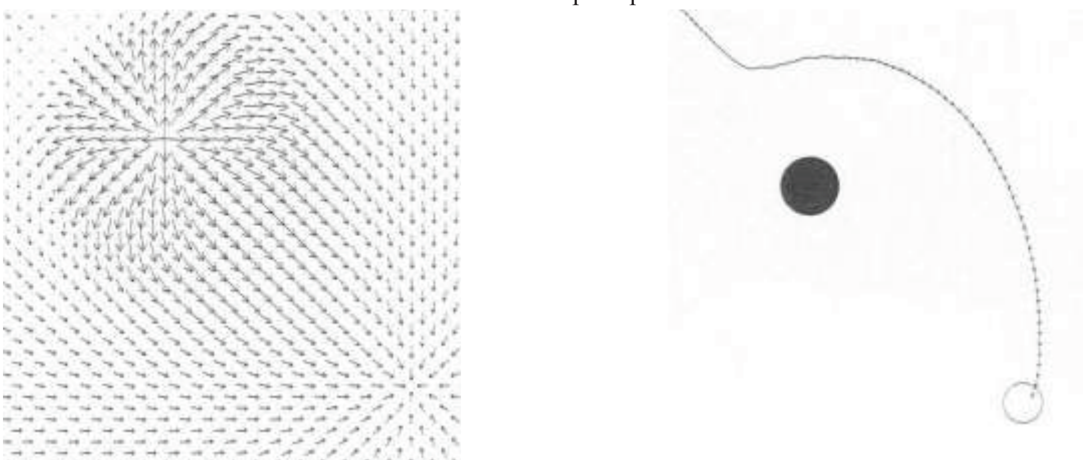
(a) (b)

Fig. 2. (a) Potential Field for a Goal (b) Potential Field for an obstacle

Vector summation calculates the path of the robot that operates in an environment with a goal and an obstacle. The result path is shown in Fig. 3. The robot moves away from the obstacle and towards the goal.

Most of robot control systems consist of multiple behaviors. In these systems more than one behavior could be activated at the same time. There should be a mechanism to determine the overall output of the system in these situations. The mechanism to eliminate these conflicts is coordination function. Generally there are two types of coordination function: competitive methods and cooperative methods. Competitive methods are generally divided into three methods: priority based coordination, action selection coordination and voting based coordination which we pass them up. Cooperative methods propose an alternative for competitive methods such as arbitration. Arbitration requires selecting a single behavioral response by means of a coordination function which is served as arbiter [6]. Behavioral fusion via vector summation could be studied under this title. In behavioral fusion via vector summation the response of each behavior is received and averaged. The average value of responses of all behaviors would be the final response.

One of the behavior based structures is “Motor Schema”. Motor schema approach is strongly motivated by biological sciences. A schema is the basic unit of behavior out of which complex actions could be constructed. It consists of the knowledge of how to act or perceive as well as computational process by which it is enacted. Motor schema based behaviors are relatively large grain abstracts reusable over a wide range of circumstances. A perceptual schema is embedded in each motor schema. Perceptual schemas provide special environmental information for a specific behavior. Perception is conducted in a need-to-know basis: perceptual algorithms propose necessary information for a special behavior. Perceptual schemas are defined recursively which means perceptual sub-schemas could extract information that will be



(a) (b)

Fig. 3. (a) Vector summation of fields given in Fig. 2 (b) Path of the robot

processed subsequently by another schema in a more behaviorally meaningful unit. This way multiple perceptual schemas could be used for a single motor schema. Each motor schema has an action vector as output that determines robot motion way in response to perceived stimulus. Fig. 4 shows a sample of designed architecture using motor schema architecture approach.

III. Kinematic Model of Mobile Robot

One of the common types of steering used for mobile robots is differential drive steering shown in Fig. 5.

Here the wheels on one side of the robot are controlled independently from the wheels on the other side. By coordinating the two different types of speeds, one can cause the robot to spin in place, move in a straight line, move in a circular path or follow any prescribed trajectory [1].

The equations of the motion for the robot steered via differential wheel speeds are now derived. Let R represent the instantaneous radius of curvature of the robot trajectory. The width of the vehicle, i.e. spacing between the wheels, is designated as W. From geometrical considerations we have:

$$v_{left} = \dot{\psi}(R - W/2) \tag{1}$$

$$v_{right} = \dot{\psi}(R + W/2) \tag{2}$$

Subtracting two above equations yields:

$$v_{right} - v_{left} = \dot{\psi} W \tag{3}$$

So we obtain for the angular rate of the robot:

$$\dot{\psi} = \frac{v_{right} - v_{left}}{W} \tag{4}$$

Solving for the instantaneous radius of curvature we have:

$$R = \frac{v_{left}}{\dot{\psi}} + \frac{W}{2} \tag{5}$$

or

$$R = \frac{v_{left}}{\frac{v_{right} - v_{left}}{W}} + \frac{W}{2} \tag{6}$$

or finally:

$$R = \frac{W}{2} \times \frac{v_{right} + v_{left}}{v_{right} - v_{left}} \tag{7}$$

This, results in the expression for the velocity along the robot's heading:

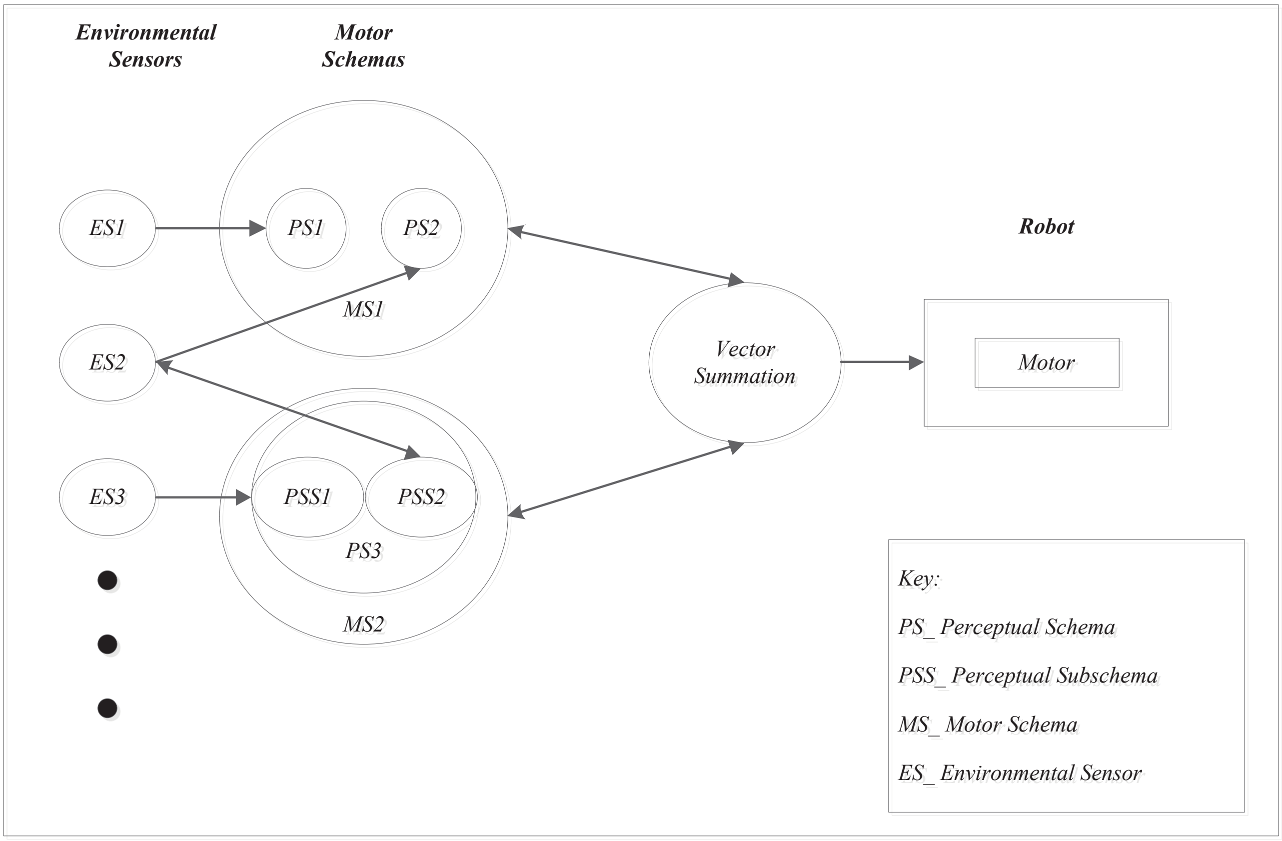


Fig. 4. Perception action schema relationship

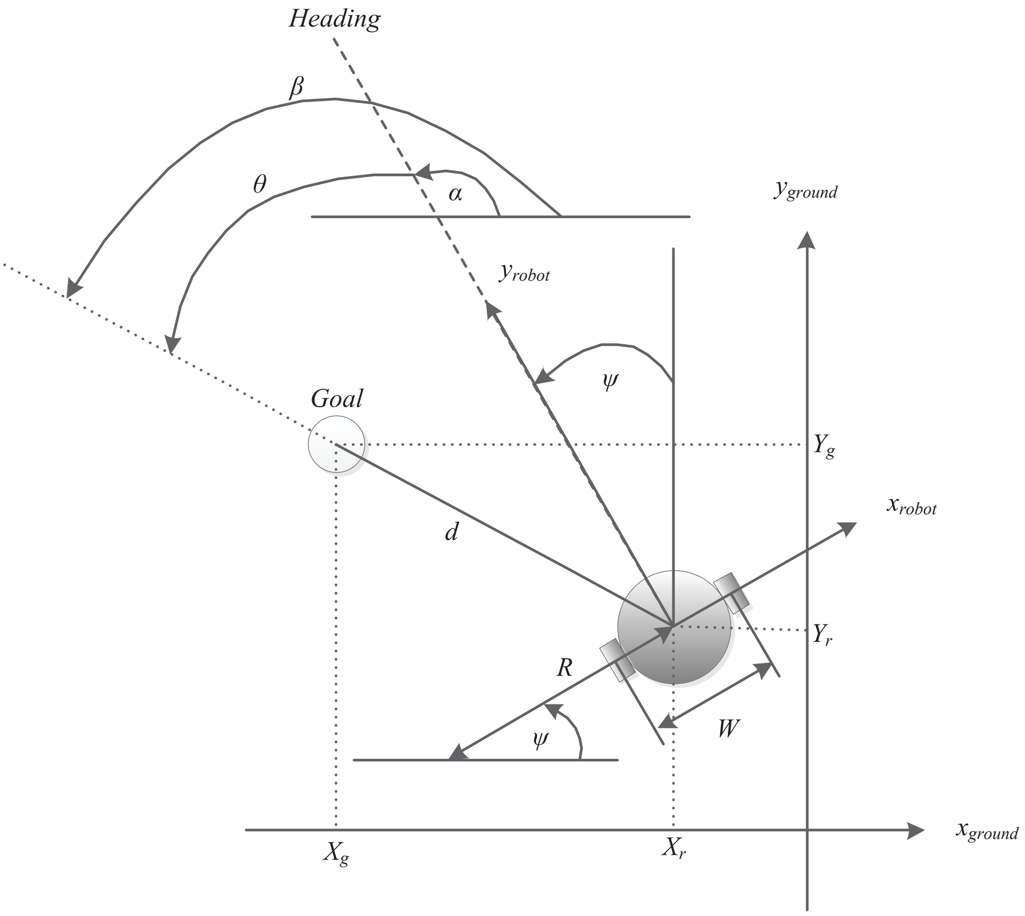


Fig. 5. Kinematic illustration of the used mobile robot

$$v_y = \dot{\psi} R = \frac{v_{right}-v_{left}}{W} \times \frac{W}{2} \times \frac{v_{right}+v_{left}}{v_{right}-v_{left}} = \frac{v_{right}+v_{left}}{2} \quad (8)$$

In summary the equations of motion in robot coordinates are:

$$v_x = 0 \quad (9)$$

$$v_y = \frac{v_{right}+v_{left}}{2} \quad (10)$$

$$\dot{\psi} = \frac{v_{right}-v_{left}}{W} \quad (11)$$

If we convert to earth coordinates these become:

$$\dot{x} = -\frac{v_{right}+v_{left}}{2} \sin \psi \quad (12)$$

$$\dot{y} = \frac{v_{right}+v_{left}}{2} \cos \psi \quad (13)$$

$$\dot{\psi} = \frac{v_{right}-v_{left}}{W} \quad (14)$$

IV. Fuzzy Behavior Based Mobile Robot Navigation

The behavioral structure of the robot is considered based on motor schema. Sugeno fuzzy model [3] is widely used in control and is appropriate to be combined with motor schema. A first order Sugeno fuzzy model is considered to be used. As each motor schema is a basic behavior, we need two behaviors (motor schemas) for robot navigation: "Goal Seeking" and "Obstacle Avoidance". So, a first order Sugeno FIS (Fuzzy Inference System) consisting two rules is required. These rules are defined as:

$$\begin{gathered} Rule1: if\ s_1\ is\ FAR\ and\ \dots and\ s_n\ is\ FAR \\ and\ |\theta|\ is\ POSITIVE\ then\ V_{left} = \\ V_{base} + V_{l\_GS}\ and\ V_{right} = V_{base} + V_{r\_GS} \end{gathered} \quad (15)$$

$$\begin{gathered} Rule2: if\ s_1\ is\ NEAR\ or\ \dots or\ s_n\ is\ NEAR \\ then\ V_{left} = V_{base} + V_{l_{OA}} \\ and\ V_{right} = V_{base} + V_{r\_OA} \end{gathered} \quad (16)$$

where the first rule is related to goal seeking behavior and the second rule is related to obstacle avoidance behavior. As can be seen the FIS has two outputs, V_{left} and V_{right} which define the speed of left and right wheels of the robot, respectively. These outputs are used as robot's inputs to control it. The FIS inputs, s_1, …, s_n ,are proximity sensor values that are served as obstacle detectors and obviously their number is n. θ is the angle between robot's heading and the straight line which connects robot center to the target, as shown in Fig. 5. Each of V_{base}, V_{l_GS}, V_{r_GS}, V_{l_OA} and V_{r_OA} is a function that is defined as follows:

$$V_{base} = \begin{cases} 0 & d \leq d_t \\ V_c & d > d_t \end{cases} \quad (17)$$

$$V_{l\_GS} = -k\theta \quad (18)$$

$$V_{r\_GS} = k\theta \quad (19)$$

$$V_{l\_OA} = \sum_{i=1}^{n} \alpha_i s_i \quad (20)$$

$$V_{r\_OA} = \sum_{i=1}^{n} \beta_i s_i \quad (21)$$

where V_c is a constant speed and α_i, β_i and k are some constant coefficients. d is the distance of the robot from Goal and d_t is a considered threshold for this distance. Having Fig. 5 in mind, d and θ can be computed as follows:

$$d = \sqrt{(X_g - X_r)^2 + (Y_g - Y_r)^2} \quad (22)$$

$$\theta = \beta - \alpha \quad (23)$$

where α is the robot's heading angle in earth coordinate system and β can be computed through the following equation:

$$\beta = \operatorname{atan2}(Y_g - Y_r\,, X_g - X_r) \quad (24)$$

where atan2 (y, x) is in fact $\tan^{-1}$ (y/x) and the only difference is that in this function x and y signs are considered to determine the quarter in which angle is located and its output lies in (-π, π].

Fuzzy AND/OR operators used in this FIS are min/max functions. The membership functions of linguistic expressions in fuzzy rules are shown in Fig. 6.

As can be seen, a membership function is considered for each of inputs consisting proximity sensors values and absolute value of θ and s_{t_i} and θ_t are thresholds for each of n sensor values and $|\theta|$.

V. Simulation Results

As mentioned before, the Khepera III robot is simulated to test the proposed algorithm, but not all of its features are used. Six infra-red sensors (sensors 2 to 7 shown in Fig. 7) are used for obstacle detection. Also a GPS and a compass which didn't exist in the robot are added to it to help us find the robot's coordination and angle. The designed environment for the experiments is shown in Fig. 8.

As can be seen, a number of static obstacles are considered in different shapes and a wandering robot is considered as a dynamic obstacle.

Webots version used for this work is 7.1.0. Constant parameters introduced in previous section are as following:

$$V_c = 20000, d_t = 5cm, k = 15000, \alpha_1 = 20,$$
$$\alpha_2 = 20, \alpha_3 = 20, \alpha_4 = -20, \alpha_5 = -20,$$
$$\alpha_6 = -20, \beta_1 = -20, \beta_2 = -20, \beta_3 = -20,$$
$$\beta_4 = 20, \beta_5 = 20, \beta_6 = 20, \theta_t = 3^{\circ},$$
$$s_{t\_1} = 300, s_{t\_2} = 170, s_{t\_3} = 170,$$
$$s_{t\_4} = 170, s_{t\_5} = 170, s_{t\_6} = 300$$

The robot was located in different places in the environment and different paths were tested that showed the robot had a desired behavior. By starting the motion from the starting point, the robot tries to reach the target in a straight line which connects starting point to target and while it doesn't sense any obstacles on its path, it continues the motion. If any obstacles are sensed, after passing them, the robot again chooses the straight line which connects its current place to the target as its path. A sample path of the robot in absence of wandering robot is shown in Fig. 9.

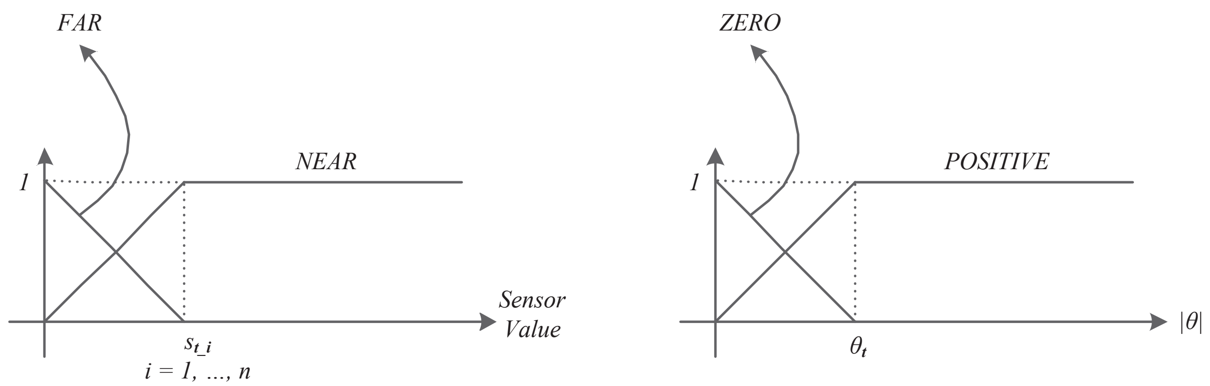


Fig. 6. FIS inputs membership functions

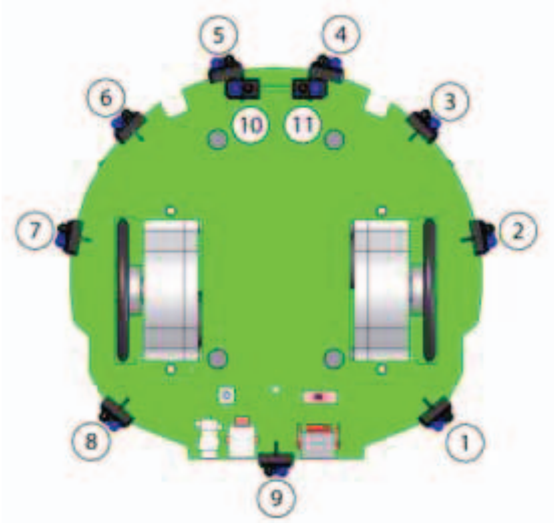


Fig. 7. Bottom view of robot's sensors

VI. Conclusion

In the current article, an algorithm is proposed to navigate the robot from a starting point to the target which is a combination of behavior based robotics, fuzzy inference systems and Braitenberg vehicles. Using this method the robot could avoid the obstacles on its path. In parts of the path with no obstacle, the robot moves to the target by passing the shortest possible path, namely straight line. These objectives, namely moving to the target and obstacle avoidance obtained by defined behaviors. The FIS was served to combine the outputs of each behavior for final output generation and also decrease the error of sensory data. The Braitenberg vehicles concept helped a lot to define the output of obstacle avoidance behavior.

The assessment of the proposed algorithm was done in Webots simulator on Khepera III robot. The obtained results from different experiments confirm desired performance of the algorithm. To improve the algorithm performance, using more proximity sensors with higher accuracy such as sonars or military sensors is suggested.

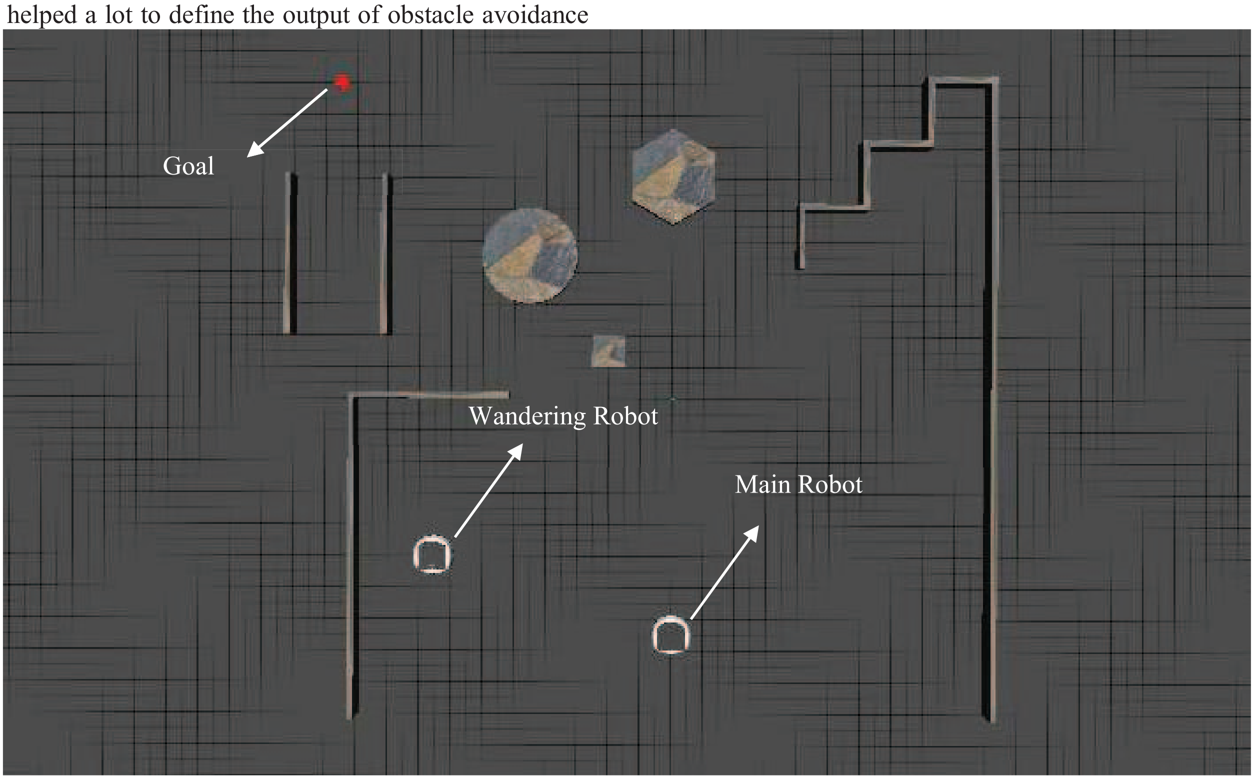


Fig. 8. Top view of the environment designed for robot motion simulation

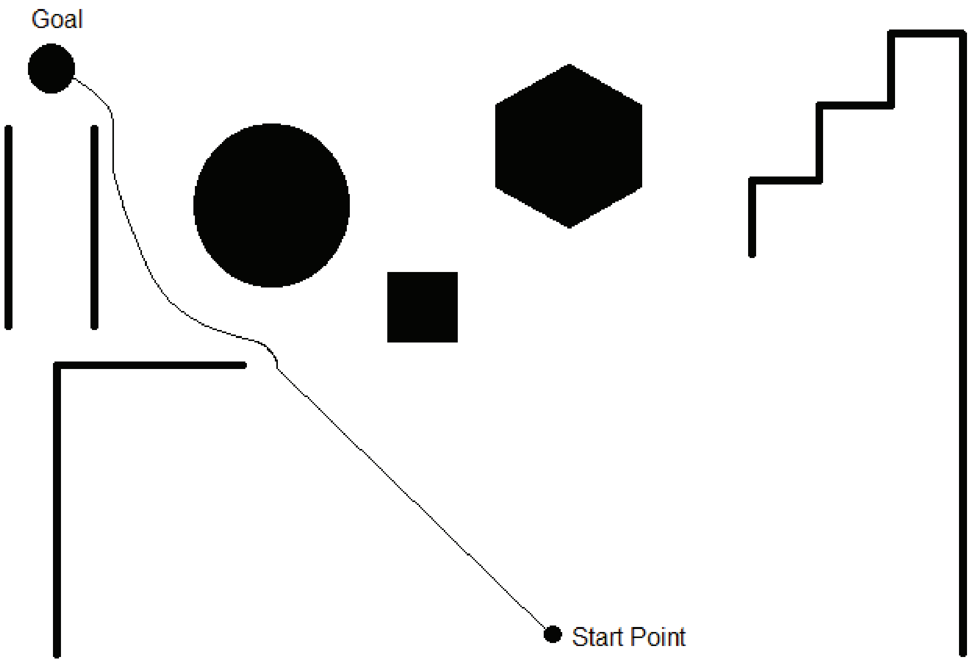


Fig. 9. Robot path in absence of wandering robot

REFERENCES

[1] G. Cook, "Mobile Robots: Navigation, Control and Remote Sensing", Wiley, Hoboken, NJ, Piscataway, NJ, 2011.

[2] H. Mo, Q. Tang and L. Meng, "Behavior-Based Fuzzy Control for Mobile Robot Navigation", Mathematical Problems in Engineering, Vol. 2013, pp. 1-10, 2013.

[3] J.S.R. Jang, C.T. Sun and E. Mizutani, Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence, Prentice Hall, Upper Saddle River, NJ, 1997.

[4] K-Team Khepera III User Manual, Version 3.5, February 2013.

[5] P. Sadeghi-Tehran and P. Angelov, Online Self-Evolving Fuzzy Controller for Au-tonomous Mobile Robots, IEEE Workshop on Evolving and Adaptive Intelligent Systems (EAIS), Paris, April 2011, pp. 100-107.

[6] R.C. Arkin, Behavior Based Robotics. MIT Press, 1998.

[7] V.Braitenberg, Vehicles: Experiments in Synthetic Psychology, MIT Press, Cambridge, MA, 1984.

[8] Webots Reference Manual, Release 7.2.4, August 2013.

[9] Webots User Guide, Release 7.2.4, August 2013.