

Fuzzy Behavior-based Control of Three Wheeled Omnidirectional Mobile Robot

Nacer Hacene Boubekeur Mendil

Industrial technologies and information laboratory, Department of Electrical Engineering, Faculty of Technology,
University of Bejaia, Bejaia 06000, Algeria

Abstract: In this paper, a fuzzy behavior-based approach for a three wheeled omnidirectional mobile robot (TWOMR) navigation has been proposed. The robot has to track either static or dynamic target while avoiding either static or dynamic obstacles along its path. A simple controller design is adopted, and to do so, two fuzzy behaviors “Track the Target” and “Avoid Obstacles and Wall Following” are considered based on reduced rule bases (six and five rules respectively). This strategy employs a system of five ultrasonic sensors which provide the necessary information about obstacles in the environment. Simulation platform was designed to demonstrate the effectiveness of the proposed approach.

Keywords: Three wheeled omnidirectional mobile robot (TWOMR), autonomous navigation, obstacle avoidance, fuzzy behavior-based control, dynamic target, dynamic environment.

1 Introduction

Autonomous mobile robots are intelligent agents able to perform desired tasks in various known and unknown environments without human intervention and can react to dynamical changes of these environments. One important issue in robotics research is navigation. In a cluttered environment where the robot has to navigate successfully toward a goal while avoiding obstacles remains a challenging problem.

In recent years, omnidirectional mobile robots have attracted much attention not only as a test-bed to academic demonstrations but also as an essential component of industrial and home automation. Compared to the more common non-holonomic car-like vehicles, omnidirectional mobile robots, which are holonomic, provide superior maneuvering capability. They can perform translational and rotational motion independently and simultaneously since they are a type of 3 degrees of freedom (DOF) vehicles on a 2-dimensional plane. The ability to move in any direction, irrespective of the orientation of the vehicle, makes it an attractive option in dynamic environments^[1-6]. So they are very useful in many applications such as hospital applications^[7], rehabilitation^[8], industrial applications^[9], service robots^[10], etc.

Numerous advancements in the field of omnidirectional mobile robots have been achieved. Some of them con-

cern hardware design. Inoue et al.^[11] studied the design of omnidirectional mobile robots with an active-caster robotic drive with ball transmission. Ma et al.^[12, 13] presented a novel omnidirectional wheel mechanism called (MY wheel-II), based on a sliced ball structure. Gebre et al.^[14] explored the use of spherical ball drive system to replace the standard wheel. Zhang and Huang^[4] have proposed a novel tracked running mechanism, with which a tracked vehicle can not only achieve omnidirectional motion to improve the maneuverability of conventional tracked vehicles but also retain the cross-country capability of conventional tracked vehicles. Tavakoli et al.^[15, 16] introduced the Omniclimber, a climbing robot with high maneuverability for inspection of ferromagnetic flat and convex human made structures.

Other researchers studied the modeling and the control of omnidirectional mobile robots. A dynamic modeling of a three wheeled omnidirectional mobile robot was first derived by Watanabe et al.^[17], then, it was followed by the work of Ren and Ma^[18] with a slightly different orthogonal wheel (MY wheels-II). H. Kim and B. K. Kim^[19] presented an online minimum-energy translational and rotational velocity trajectory planning and control system on a straight-line path for three-wheeled omnidirectional mobile robots. Barreto et al.^[20] presented and discussed the implementation results of a model-predictive control scheme with friction compensation applied to trajectory following of an omnidirectional three-wheeled robot. Sharbafi et al.^[21] applied an intelligent controller to control the motion of an omnidirectional robot based on the brain-emotional-learning algorithm, which is inspired by a computational model of the limbic system in the

Research Article

Manuscript received December 25, 2017; accepted May 3, 2018; published online August 3, 2018

Recommended by Associate Editor Hong-Nian Yu

© Institute of Automation, Chinese Academy of Sciences and Springer-Verlag GmbH Germany, part of Springer Nature 2018

mammalian brain. Sira-Ramirez et al.^[22] proposed an observer-based robust linear output feedback controller for the trajectory tracking problem of an omnidirectional mobile robot.

A number of control techniques are used including fuzzy control^[23, 24], neural networks^[25, 26], genetic algorithms^[27], sliding mode^[28] and optimization methods such as ant colony optimization (ACO)^[29], particle swarm optimization (PSO)^[30], deoxyribonucleic acid (DNA) algorithms^[31], etc.

The behavior-based approach has been established as the main alternative to conventional robot control^[32]. Behavior-based architectures are bottom-up approaches inspired by biology and consist of decomposing the problem of autonomous control by task rather than by function. The two basic behavior-based control architectures include subsumption architecture^[33] and motor schemas^[34]. On the other hand, fuzzy logic is an approximate reasoning that can cope with uncertainty in information so that it can overcome behavior-based problems. The combination of the fuzzy control and behavior-based architecture has some further advantages. It can produce controllers that are robust to uncertainty and imprecision based on a set of IF-THEN rules in which the expert knowledge can be employed. The big centralized controller is reduced to distributed smaller sub-controllers. Finally, fuzzy control can be used to overcome the conflict among multiple behaviors.

Many types of research were carried out. We start with the earlier works by Izumi and Watanabe^[35–37], where a fuzzy behavior-based approach was derived. Arriving at the recent works, Selekwia et al.^[38] presented the design of a preference-based fuzzy behavior system for navigation control of robotic vehicles using the multivalued logic framework. Zein-Sabatto et al.^[39] have designed four behaviors which are integrated and coordinated to form complex robotics system. Wang and Liu^[40] proposed a fuzzy behavior-based navigation that can deal with long wall, large concave, recursive U-shaped, unstructured, cluttered, maze-like, and dynamic indoor environments.

Considering the motion of the unicycle robot or the differential drive robot which has two driving wheels^[41, 42], the motion is straightforward. To move the robot forward or backward, we give the two driving wheels the same velocity. To turn right, we increase the left wheel velocity or decrease the right wheel velocity and the inverse to turn left. The motion of the three wheeled omnidirectional mobile robot (TWOMR) is more complex; hence, most of the studies concerning omnidirectional robot control use the inverse kinematics to control the robot wheels. This paper brings some new features:

1) A reduced (a minimum number of rules and a minimum number of behaviors) fuzzy behavior-based approach for TWOMR navigation has been proposed. The robot has to track a dynamic target while avoiding

obstacles along its path. For simplicity, we have used a reduced number of behaviors based on reduced rule bases. Two fuzzy behaviors “Track the Target” and “Avoid Obstacles and Wall Following” are designed for the proposed controller.

2) The controller drives directly the robot. The controller outputs are the wheels velocities rather than using the inverse kinematics.

This paper is organized as follows. Section 2 presents a mathematical development of TWOMR modeling. Section 3 describes the navigation problem. Section 4 presents the fuzzy behavior-based controller design. The simulation results are given in Section 5. Finally, Section 6 is dedicated to the conclusion and the future work.

2 TWOMR modeling

2.1 Description of the TWOMR and its movement

The TWOMR is a holonomic robot that has the ability to move simultaneously and independently in translation and rotation. The robot is equipped with three omni-wheels equally arranged at 120 degrees on the circumference of the robot (Fig. 1).

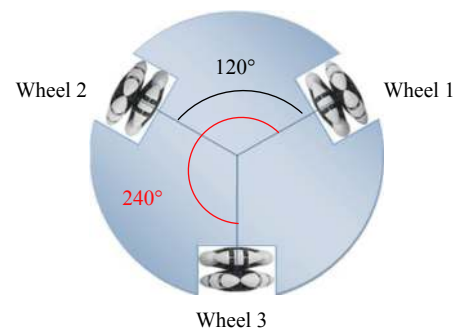


Fig. 1 Robot configuration

Each omni-wheel is mounted directly to its motor shaft so that the motor and the wheel have the same rotational axle. The distance from the center of the robot to the center of the wheel is denoted by L . By individual control of the speed of each motor, the robot is able to perform motions that robots on normal wheels cannot perform.

An omnidirectional drive system requires a minimum of three omni-wheels. Fig. 2 shows an example of an omni-wheel where the wheel is equipped with many rollers enable it to move sideways perpendicular to the normal direction of rolling (the linear velocity).

Consider the omni-drive system of Fig. 3. Assume that Wheel 1 on the right is active with the translational velocity V_{w1} while Wheels 2 and 3 on the left are inactive. Assuming no slippage, in this case, Wheels 2 and 3 will gain a velocity V_{in1} perpendicular to their normal directions of

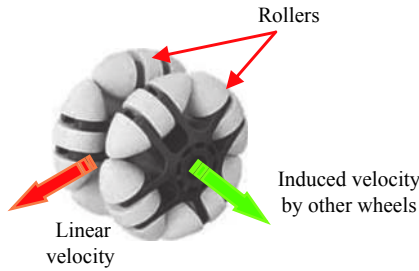


Fig. 2 Omnidirectional wheel

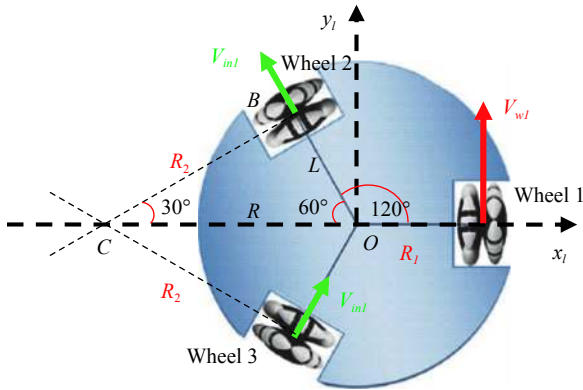


Fig. 3 Calculation of the induced velocity

rolling, outer on the Wheel 2 and inner on the Wheel 3 due to their rollers. This velocity is called the Induced Velocity. Therefore, the system will rotate about a single point C which is the intersection between the two lines perpendicular to the velocity of each wheel V_{w1} and V_{in1} .

The induced velocity can be easily calculated. From Fig. 3, in triangle OBC, we have

$$\frac{R_2}{\sin 60^\circ} = \frac{L}{\sin 30^\circ} \tag{1}$$

Therefore,

$$R_2 = \sqrt{3}L \tag{2}$$

Now, we can calculate the radius of rotation of the robot center R using Pythagoras theorem as follows:

$$R = \sqrt{R_2^2 + L^2} = 2L \tag{3}$$

Also,

$$R_1 = R + L = 3L \tag{4}$$

In this case, as depicted in Fig. 2, the robot turns around the point C, therefore the wheels of the robot turn with the same angular velocity:

$$\dot{\varphi} = \frac{V_{w1}}{R_1} = \frac{V_{in1}}{R_2} \tag{5}$$

From (5), the induced velocity can be calculated as

$$V_{in1} = \frac{R_2}{R_1} V_{w1} = \frac{V_{w1}}{\sqrt{3}} \tag{6}$$

2.2 TWOMR kinematics

Assume a global frame (x_G, y_G) which represents the environment of the robot (Fig. 4). We can also define a moving local frame (x_l, y_l) associated with the center of mass of the robot. The x_l axis is parallel to the axis of rotation of the Wheel 1. The three omnidirectional wheels are located at an angle α_i ($i = 1, 2, 3$) relative to the local frame. If we take the moving axis x_l , as starting point and count degrees in the counter-clockwise direction as positive, we have $\alpha_1 = 0^\circ$, $\alpha_2 = 120^\circ$ and $\alpha_3 = 240^\circ$. The robot's location and orientation can be represented as $X_l = [x_l, y_l, \varphi]^T$ in the local frame and $X_G = [x_G, y_G, \varphi]^T$ in the global frame, while the velocity of the robot can be written as $\dot{X}_l = [\dot{x}_l, \dot{y}_l, \dot{\varphi}]^T$ in the local frame and $\dot{X}_G = [\dot{x}_G, \dot{y}_G, \dot{\varphi}]^T$ in the global frame.

Note that, the orientation angle φ , the angle between the x_l axis with respect to the x_G (Fig. 4) axis of the robot is the same in the local and the global frames.

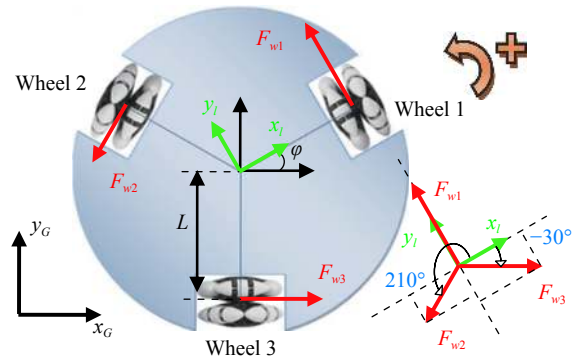


Fig. 4 Kinematic diagram of the robot

The relations of the forces in the local frame can be written geometrically as follows:

$$\begin{cases} F_{lx} = \cos(210^\circ) F_{w2} + \cos(-30^\circ) F_{w3} \\ F_{ly} = F_{w1} + \sin(-30^\circ) F_{w2} + \sin(210^\circ) F_{w3} \\ M_z = LF_{w1} + LF_{w2} + LF_{w3} \end{cases} \tag{7}$$

where F_{w1} , F_{w2} and F_{w3} are the traction forces applied to the Wheels 1, 2 and 3, respectively. F_{lx} and F_{ly} are the abscissa and the ordinates components of the resultant force in the local frame, while M_z is the moment on the robot.

Define $F = [F_{w1} \ F_{w2} \ F_{w3}]^T$ as the traction forces applied to the wheels, as shown in Fig. 4, and $F_l = [F_{lx} \ F_{ly} \ M_z]^T$ as the force and moment on

the robot in the local frame, where T stands for the transpose. Equation (7) can be written in a compact form as follows:

$$F_l = AF \tag{8}$$

where A is the geometrical matrix:

$$A = \begin{bmatrix} 0 & -\frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} \\ 1 & -\frac{1}{2} & -\frac{1}{2} \\ L & L & L \end{bmatrix}. \tag{9}$$

The instantaneous power of the force F is independent of the frame in which is defined, and is given by the dot product:

$$P = \vec{F} \cdot \vec{V} = F^T \dot{X}_w = F_l^T \dot{X}_l \tag{10}$$

where

$\dot{X}_w = [V_{w1} \ V_{w2} \ V_{w3}]^T$ represents the wheels' velocities.

Substituting (8) into (10), we obtain

$$F^T \dot{X}_w = (AF)^T \dot{X}_l. \tag{11}$$

Then,

$$F^T \dot{X}_w = F^T A^T \dot{X}_l. \tag{12}$$

In (12), the same factor F^T is multiplied on the left and on the right, and this leads to

$$\begin{bmatrix} \dot{x}_G \\ \dot{y}_G \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} -\frac{2\sin(\varphi)}{3} & \frac{\sin(\varphi)}{3} - \frac{\sqrt{3}\cos(\varphi)}{3} & \frac{\sin(\varphi)}{3} + \frac{\sqrt{3}\cos(\varphi)}{3} \\ \frac{2\cos(\varphi)}{3} & -\frac{\cos(\varphi)}{3} - \frac{\sqrt{3}\sin(\varphi)}{3} & \frac{\sqrt{3}\sin(\varphi)}{3} - \frac{\cos(\varphi)}{3} \\ \frac{1}{3L} & \frac{1}{3L} & \frac{1}{3L} \end{bmatrix} \begin{bmatrix} V_{w1} \\ V_{w2} \\ V_{w3} \end{bmatrix}. \tag{19}$$

The translational and the angular velocity of the robot are given by

$$\begin{cases} V = \sqrt{\dot{x}_G^2 + \dot{y}_G^2} \\ \dot{\varphi} = \frac{V_{w1} + V_{w2} + V_{w3}}{3L}. \end{cases} \tag{20}$$

From (19) and (20), we get

$$\begin{cases} V = \frac{1}{3} \sqrt{(2V_{w1} - V_{w2} - V_{w3})^2 + 3(V_{w3} - V_{w2})^2} \\ \dot{\varphi} = \frac{V_{w1} + V_{w2} + V_{w3}}{3L}. \end{cases} \tag{21}$$

We can analyze the movement of the robot based on

$$\dot{X}_w = A^T \dot{X}_l. \tag{13}$$

Therefore,

$$\dot{X}_l = (A^T)^{-1} \dot{X}_w. \tag{14}$$

Equation (14) represents the kinematics of the TWOMR expressed in the local frame in a compact form, and it can be given in a detailed form as follows:

$$\begin{bmatrix} \dot{x}_l \\ \dot{y}_l \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} 0 & -\frac{\sqrt{3}}{3} & \frac{\sqrt{3}}{3} \\ \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} \\ \frac{1}{3L} & \frac{1}{3L} & \frac{1}{3L} \end{bmatrix} \begin{bmatrix} V_{w1} \\ V_{w2} \\ V_{w3} \end{bmatrix}. \tag{15}$$

The relation between velocities in the local frame and the global frame is given by

$$\dot{X}_G = R_l^G \dot{X}_l \tag{16}$$

where R_l^G is the rotation matrix given by

$$R_l^G = \begin{bmatrix} \cos \varphi & -\sin \varphi & 0 \\ \sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{17}$$

From (14) and (16), we get the kinematics of the robot in the global frame as following (18) and (19):

$$\dot{X}_G = R_l^G (A^T)^{-1} \dot{X}_w \tag{18}$$

(21) as follows:

- 1) If $V_{w1} + V_{w2} + V_{w3} = 0$, then $V = \text{constant}$ and $\dot{\varphi} = 0$. This corresponds to instantaneous linear movement.
- 2) If $V_{w1} = V_{w2} = V_{w3}$, then $V = 0$ and $\dot{\varphi} = \text{constant}$. This corresponds to instantaneous pure rotation.
- 3) If the two conditions mentioned above are not verified, then the movement is in instantaneous circular trajectory with translational velocity V and radius R .

Note that the three states above are instantaneous for each sample time. If we maintain the wheels velocities constant for a period of time, then the movement will remain in this state for this period of time. So we can make any motion with variable linear velocity and variable curvature radius from these three types of motion.

2.3 TWOMR dynamics

According to Newton's second law and (8), we get

$$F_G = R_l^G F_l = R_l^G A F = M \ddot{X}_G \tag{22}$$

where M is the matrix of the robot mass and inertia, given by

$$M = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix} \tag{23}$$

where $m(\text{kg})$ and $I(\text{kg}\cdot\text{m}^2)$ are the mass and the rotational inertia of the robot. Taking the time derivative of \dot{X}_w in (18), we get

$$\ddot{X}_G = \dot{R}_l^G (A^T)^{-1} \dot{X}_w + R_l^G (A^T)^{-1} \ddot{X}_w. \tag{24}$$

From (22) and (24), we get the relation of the traction force with respect to the global frame

$$F = (A)^{-1} (R_l^G)^{-1} M \left\{ \dot{R}_l^G (A^T)^{-1} \dot{X}_w + R_l^G (A^T)^{-1} \ddot{X}_w \right\}. \tag{25}$$

Now the relation between the traction force and the wheel torque $T_w = [T_{w1} \ T_{w2} \ T_{w3}]^T$ in (N·m) can be given as

$$T_w = I_w \left(\frac{\ddot{X}_w}{r} \right) + f_w \left(\frac{\dot{X}_w}{r} \right) + Fr \tag{26}$$

where I_w (kg·m²) is the moment of inertia of wheels, f_w (N·s·m⁻¹) is the viscous friction coefficient, and r (m) is the radius of the wheel. $\left(\frac{\dot{X}_w}{r}\right)$ and $\left(\frac{\ddot{X}_w}{r}\right)$ are the angular velocities and the angular accelerations of the wheels respectively.

The relation between the wheel torque and the motor torque $T_m = [T_{m1} \ T_{m2} \ T_{m3}]^T$ in (N·m) is given as follows:

$$T_m = I_m \left(n \frac{\ddot{X}_w}{r} \right) + f_m \left(n \frac{\dot{X}_w}{r} \right) + \frac{T_w}{n}. \tag{27}$$

where I_m (kg·m²) is the motor moment of inertia, f_m (N·s·m⁻¹) is the motor viscous friction, and n is the gear ratio between the wheel and the motor. $\left(n \frac{\dot{X}_w}{r}\right)$ and $\left(n \frac{\ddot{X}_w}{r}\right)$ are the angular velocities and the angular accelerations of the motors, respectively.

By substituting (26) into (27), we get

$$T_m = \left(n I_m + \frac{I_w}{n} \right) \frac{\ddot{X}_w}{r} + \left(n f_m + \frac{f_w}{n} \right) \frac{\dot{X}_w}{r} + \frac{Fr}{n}. \tag{28}$$

In (28), we have to substitute F by its value to get the relation between the motor torque T_m and the velocity of wheels \dot{X}_w , so, we substitute (25) into (28), and after simplification and arrangement, we obtain

$$T_m = \begin{bmatrix} C_1 & C_2 & C_2 \\ C_2 & C_1 & C_2 \\ C_2 & C_2 & C_1 \end{bmatrix} \ddot{X}_w + \begin{bmatrix} C_3 & -C_4 \dot{\varphi} & C_4 \dot{\varphi} \\ C_4 \dot{\varphi} & C_3 & -C_4 \dot{\varphi} \\ -C_4 \dot{\varphi} & C_4 \dot{\varphi} & C_3 \end{bmatrix} \dot{X}_w \tag{29}$$

where

$$C_1 = \frac{9L^2 n^2 I_m + 9L^2 I_w + r^2 (I + 4mL^2)}{9L^2 nr} \tag{30}$$

$$C_2 = \frac{r (I - 2mL^2)}{9L^2 n} \tag{31}$$

$$C_3 = \frac{n^2 f_m + f_w}{nr} \tag{32}$$

$$C_4 = \frac{2\sqrt{3}mr}{9n}. \tag{33}$$

We have modeled the mechanical part of the robot described by (29) that is the dynamics of the robot that connects the motor torque to the velocity of wheels. The next step is to model the electrical part of the robot motors.

For a direct current (DC) motor, we have

$$L_a \frac{di_a}{dt} + R_a i_a + k_E \left(n \frac{\dot{X}_w}{r} \right) = E \tag{34}$$

where $E = [E_1 \ E_2 \ E_3]^T$ (V) is the motor voltage input, i_a (A) is the armature current, L_a (Henries) is the armature inductance, R_a (Ω) is the armature resistance, k_E (V·s/rad) is the back electromotive force constant of the motor.

Since the inductance of the motor is omitted, as it is small and generally ignored in robot dynamics, we can neglect the motor electric circuit dynamics, $L_a \frac{di_a}{dt} = 0$, which leads to

$$R_a i_a + k_E \left(n \frac{\dot{X}_w}{r} \right) = E. \tag{35}$$

The torque T_m developed by the motor is given by the following equation:

$$T_m = k_m i_a \tag{36}$$

where k_m (N·m/A) is the motor torque constant.

From (35) and (36), we obtain

$$T_m = \frac{k_m}{R_a} E - \frac{k_m k_E n}{R_a r} \dot{X}_w. \tag{37}$$

Equation (37) can be rewritten as follows:

$$T_m = C_5 E - C_6 \dot{X}_w \tag{38}$$

where

$$C_5 = \frac{k_m}{R_a} \tag{39}$$

$$C_6 = \frac{k_m k_E n}{R_a r}. \tag{40}$$

Substituting (38) into (29), leads to:

$$\begin{bmatrix} C_1 & C_2 & C_2 \\ C_2 & C_1 & C_2 \\ C_2 & C_2 & C_1 \end{bmatrix} \ddot{X}_w + \begin{bmatrix} C_3 + C_6 & -C_4 \dot{\varphi} & C_4 \dot{\varphi} \\ C_4 \dot{\varphi} & C_3 + C_6 & -C_4 \dot{\varphi} \\ -C_4 \dot{\varphi} & C_4 \dot{\varphi} & C_3 + C_6 \end{bmatrix} \dot{X}_w = C_5 E. \tag{41}$$

Equation (41) is the dynamics of the TWOMR, it is a system of three coupled nonlinear equations, where the voltages E_i ($i = 1, 2, 3$) are the input variables while the velocities of wheels V_{wi} ($i = 1, 2, 3$) are the output variables.

3 Navigation problem formulation

Fig. 5 depicts the navigation problem of the TWOMR. The robot has to track the target T (or just reach the target in the case of static target). There are static obstacles near or on its path. The robot has

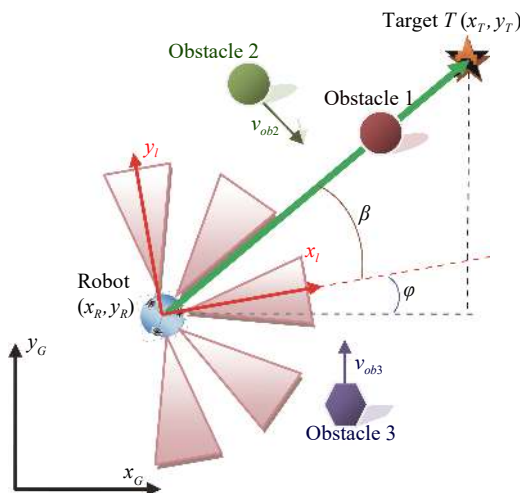


Fig. 5 Illustration of the navigation problem

to track the target T while avoiding collisions with obstacles and walls.

The angle ϕ of the x_l axis with respect to the x_G axis can be calculated as

$$\varphi = \int \dot{\varphi} dt. \tag{42}$$

The transformation of positions between the global frame and the local frame is given as

$$X_G = T_l^G X_l \tag{43}$$

where T_l^G is the transformation matrix which can be written as

$$T_l^G = \begin{bmatrix} \cos \varphi & -\sin \varphi & x_R \\ \sin \varphi & \cos \varphi & y_R \\ 0 & 0 & 1 \end{bmatrix} \tag{44}$$

where x_R and y_R are the actual coordinates of the robot in the global frame. x_l and y_l are the actual coordinates of the robot or objects in the local (robot) frame, while x_G and y_G are the actual coordinates of the robot or objects in the global frame. The position in the local frame is obtained as

$$\begin{bmatrix} x_l \\ y_l \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \varphi & \sin \varphi & -x_R \cos \varphi - y_R \sin \varphi \\ -\sin \varphi & \cos \varphi & x_R \sin \varphi - y_R \cos \varphi \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_G \\ y_G \\ 1 \end{bmatrix}. \tag{45}$$

4 Fuzzy behavior-based controller design

In classical robotic control, controllers are serial processing units where the architecture works through a cycle of Sense-Plan-Action (Fig. 6).

On the other hand, behavior-based robotics is biologically inspired, distributed, bottom-up approach. In this approach, the robot task is decomposed into several modules, called behaviors (Fig. 7). A behavior is a direct mapping of sensory inputs to a pattern of motor actions that are then used to achieve a task (stimulus-response), so, each behavior has full access to all robot sensors and processes its own command to drive the robot actuators.

The parallel structure of simple behaviors allows a real-time response with low computational cost. Basic behaviors could be “target tracking”, “obstacle avoiding”, “wall following”, etc. Behaviors with different objectives may produce conflicting actions; therefore, behavior coordination is needed to select the action that satisfies the system objective. Behavior coordination can be cooperative or competitive. In cooperative coordination (behavior

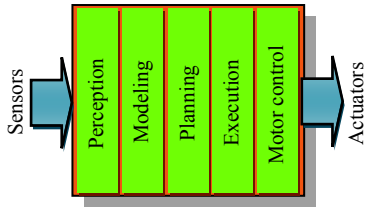


Fig. 6 Classical paradigm as horizontal functional decomposition

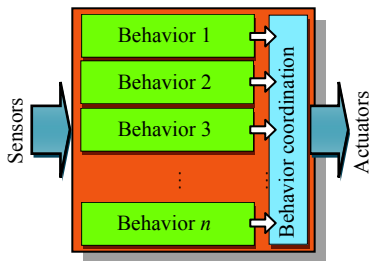


Fig. 7 Behavior-based paradigm as vertical decomposition

fusion), the behaviors are combined with a set of weights, each behavior can have the opportunity to contribute to the control output; while in competitive coordination (behavior arbitration), the behaviors compete to win the control of the robot, only one behavior's output will be valid at any time. Behavior-based control architectures are subsumption architecture^[33] and motor schemas^[34].

The theory of fuzzy logic systems is inspired by the remarkable human capacity to reason with perception-based information. Rule based fuzzy logic provides a formal methodology for linguistic rules resulting from reasoning and decision making with uncertain and imprecise information. A fuzzy controller (Fig. 8) is a static nonlinear mapping between its inputs and outputs (i.e., it is not a dynamic system). The inputs and outputs are “crisp”, i.e., they are real numbers, not fuzzy sets.

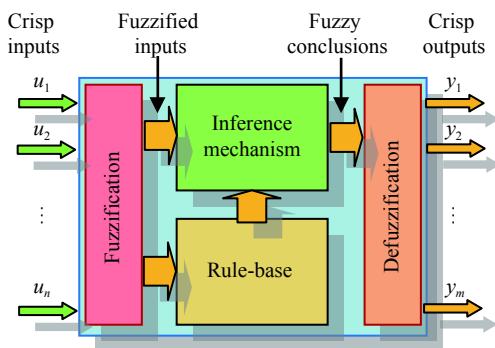


Fig. 8 General fuzzy controller

The fuzzy controller has four main components:

- 1) **The fuzzification block:** Converts the crisp inputs to fuzzy sets.
- 2) **The rule-base:** Holds the knowledge, in the form

of a set of rules, about the best way to control the system.

3) **The inference mechanism:** The inference mechanism evaluates which control rules in the rule-base are relevant at the current time to produce fuzzy conclusions, and then decides what the input to the plant should be.

4) **The defuzzification block:** Converts the fuzzy conclusions reached by the inference mechanism into the crisp outputs, which are the inputs to the plant.

Fuzzy behavior-based control architecture consists of a set of horizontally organized, distributed, independent fuzzy behaviors and a system of behavior coordination. Each behavior is a fuzzy logic control system that responds to its stimuli by issuing a single command that is transmitted for command coordination.

Our approach consists of a fuzzy behavior-based controller that has the following characteristics:

- 1) The controller is comprised of a minimum number of behaviors that is two behaviors,
 - a) “Target Tracking” behavior,
 - b) “Obstacle Avoidance and Wall Following” behavior.
- 2) Each behavior is composed of a set of a minimum number of fuzzy logic rules achieving a precise goal.
- 3) The output of each behavior represents the linear velocities of wheels.

The architecture of the controller is given in Fig. 9. Note that, D_{RT} refers to the distance between the robot and the target, ϕ is the bearing angle. LS is the left sensor, LFS is the left front sensor, FS is the front sensor, RFS is the right front sensor and RS is the right sensor. V_{w1} , V_{w2} and V_{w3} are the linear velocities of wheels. V_s is the sensing vector and $S = \text{Max}(V_s)$ is the switching parameter.

4.1 “Target Tracking” behavior

Target tracking behavior tends to drive the robot from a given initial position to a stationary or moving target position. Reaching a stationary target is a special case of tracking a moving target that moves with zero velocity. The block diagram of the fuzzy “Target Tracking” behavior is shown in Fig. 10. By using Mamdani fuzzy logic approach, the inputs of the fuzzy controller are the distance between the robot and the target D_{RT} and the robot bearing angle β with respect to the line which connects the actual position of the robot and the target. The outputs are the three translational velocities of the robot wheels V_{w1} , V_{w2} and V_{w3} .

The distance (in meters) between the robot and the target is given by

$$D_{RT} = \sqrt{(x_T - x_R)^2 + (y_T - y_R)^2} \tag{46}$$

where (x_T, y_T) and (x_R, y_R) are the coordinates of the target and the robot, respectively.

The bearing angle β (in radians) is given by

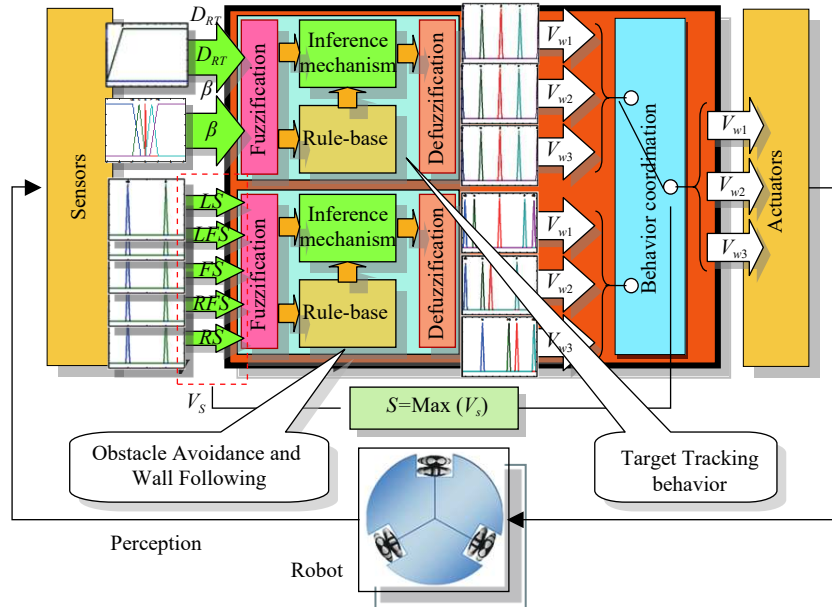


Fig. 9 Architecture of the proposed behavior-based controller

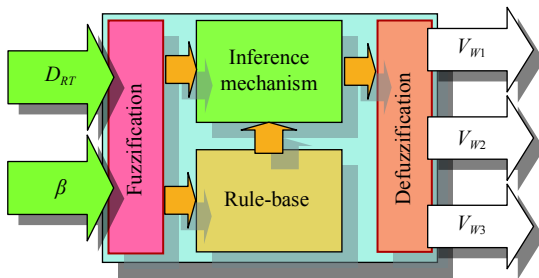


Fig. 10 Schematic diagram of the fuzzy “Target Tracking” behavior

$$\beta = \text{atan2}(y_{IT} - y_{IR}, x_{IT} - x_{IR}) \quad (47)$$

where $\text{atan2}(y, x)$ is the four-quadrant inverse tangent (arctangent) of y and x , and the result belongs to the closed interval $[-\pi, \pi]$, while $\text{atan}x\left(\frac{y}{x}\right)$ takes its values in the interval $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$.

1) The fuzzification procedure

Triangular membership functions are used to represent the fuzzy values of the input and output linguistic variables: the distance D_{RT} , the bearing angle β and the velocities of wheels V_{wi} . The labels used to express the values of D_{RT} are given in Table 1 while those of the bearing angle β and the velocities V_{wi} are given in Table 2.

The membership functions for all the terms of the input and output variables in this controller are given in Fig. 11, the degree of membership is without unit. The distance between the robot and the target D_{RT} is measured in meters while the bearing angle β is measured in radians.

2) The rule base design

As we have two inputs to the controller, then we have $2 \times 5 = 10$ possibilities, i.e., 10 possible rules, 5 rules (cor-

Table 1 Linguistic values of D_{RT}

Linguistic term	Label
Zero	Z
Far	F

Table 2 Linguistic values of the robot bearing β and the translational velocities V_{wi}

Linguistic term	Label
Negative big	NB
Negative	N
Zero	Z
Positive	P
Positive big	PB

responding to 5 values of β) for each fuzzy value of the linguistic variable D_{RT} (Z, F). But, we can reduce the five rules corresponding to the value Z of the linguistic variable D_{RT} into just 1 rule. So, the rule base of the “Target Tracking” behavior contains 6 rules as given in Table 3. To perform the “Target Tracking” behavior, the pure rotation motion is used in all rules except rule 4 which uses the linear motion.

An example of the “Target Tracking” behavior rules is:

If D_{RT} is Zero then V_{w1} is Zero and V_{w2} is Zero and V_{w3} is Zero

3) The defuzzification procedure

The “Bisector of area” method is used for the defuzzification procedure of “Target Tracking” behavior. The bisector is the vertical line that divides the region into two sub-regions of equal areas. It, sometimes but not always, coincides with the centroid line^[43]. The outputs for trans-

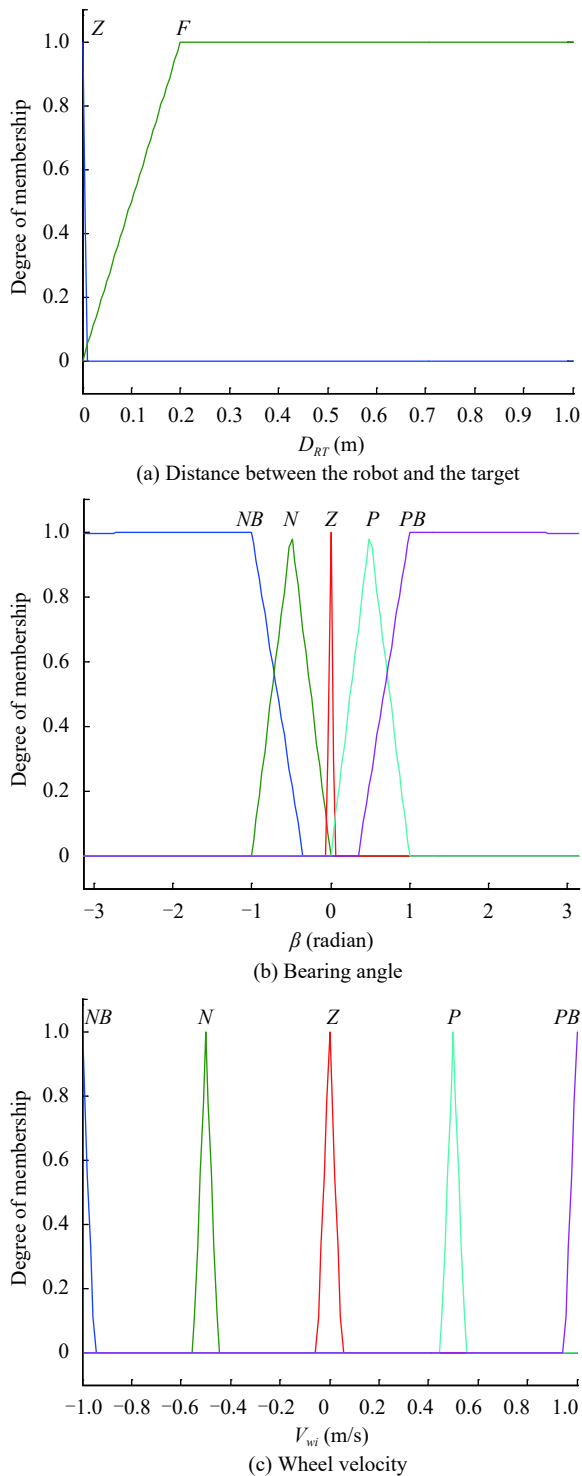


Fig. 11 Membership functions of D_{RT} , β , and V_{wi}

lational velocities are given by

$$V_{wi} = x_0 \quad \text{where} \quad \int_{\min(X)}^{x_0} \mu_A(x) dx = \int_{x_0}^{\max(X)} \mu_A(x) dx \quad (48)$$

where X is the V_{wi} universe of discourse.

Table 3 Fuzzy reduced rules for “Target Tracking” behavior

N	D_{RT}	β	V_{w1}	V_{w2}	V_{w3}
1	Z	Any	Z	Z	Z
2	F	NB	NB	NB	NB
3	F	N	N	N	N
4	F	Z	Z	NB	PB
5	F	P	P	P	P
6	F	PB	PB	PB	PB

4.2 “Obstacle Avoidance and Wall Following” behavior

This behavior combines two behaviors, “Obstacle Avoidance and Wall Following”. A wall can be considered as a large obstacle with any shape so that a wall can be a special case of multiple obstacle avoidance where the obstacles are collected together to form a wall. Because we have adopted the simplicity in our design so one behavior can be used to deal with the two problems, obstacle avoidance, and wall following. The robot is equipped with five ultrasonic sensors denoted as LS , LFS , FS , RFS and RS , mounted at the angle of 90° , 45° , 0° , -45° and -90° respectively as depicted in Fig. 12.

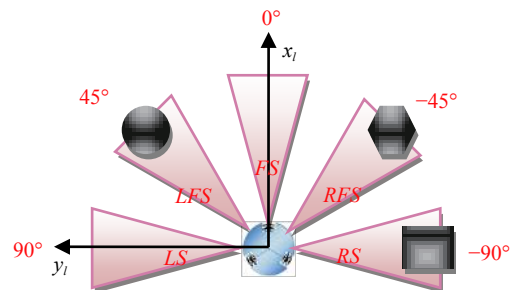
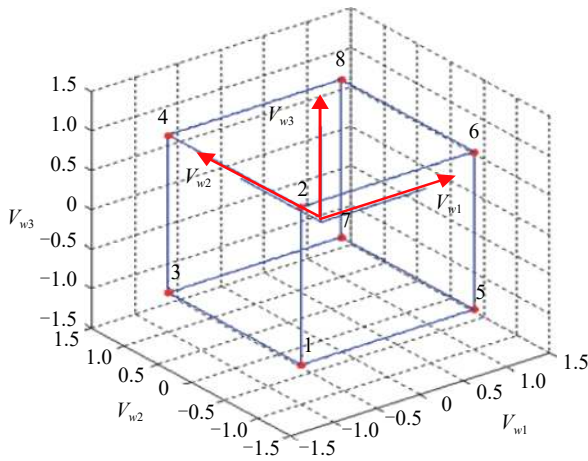


Fig. 12 Sensors configuration

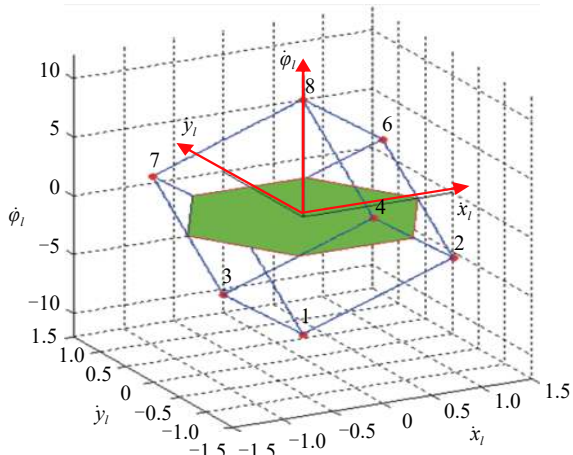
Fig. 13 shows the transformation between the frame of translational velocities of wheels and the local frame. Fig. 13(a) shows the possible linear velocities that can be taken by the wheels. They are enclosed in the blue cube. Fig. 13(b) represents the image of the linear wheels velocities in the local frame $(\dot{x}_l, \dot{y}_l, \dot{\varphi}_l)$ which appears as an inverted cube. The hexagon represents the possible velocities when $\dot{\varphi}_l = 0$ and this is more clarified in Fig. 13(c), that depicts a top view in the case of linear motion when $\dot{\varphi}_l = 0$. All possible velocities are enclosed in the green hexagon so that it can be used to calculate the limit velocity in any direction.

We define the safety distance D_s . When the relative distance to all obstacles is greater than D_s , the possibility of collision with obstacles is not checked and the robot will keep moving toward the target. If the distance with an obstacle is less than or equal to D_s , the robot starts avoiding collisions with obstacles.

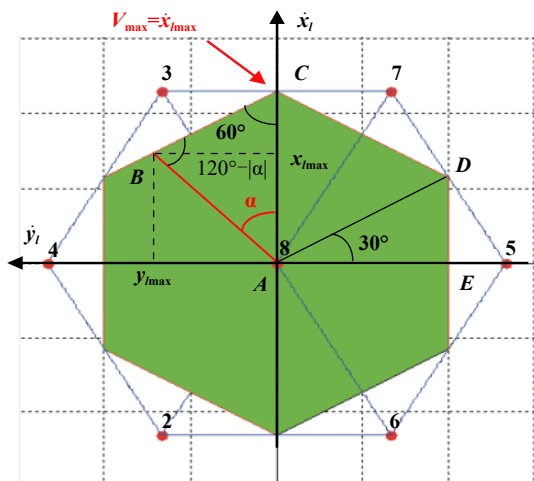
We have used the LFS sensor with 45° to calculate



(a) Frame of linear velocities of wheels



(b) Frame of local velocities



(c) A top view when the angular velocity is zero and used to calculate the safety distance D_s

Fig. 13 Transformation between the frame of linear wheels velocities and that of local velocities

the safety distance D_s (see Fig. 13(c)) because it is the nearest gap used to avoid a frontal obstacle. From Fig. 13(c), we can calculate the maximum linear velocity of the robot V_{max} , where

$$\begin{cases} V_{max} = AC = \dot{x}_l \max \\ \dot{y}_l = 0 \\ \dot{\phi}_l = 0 \text{ (linear motion).} \end{cases} \quad (49)$$

Substitute (49) into the kinematics (15), we get

$$\begin{cases} V_{max} = \frac{\sqrt{3}}{3}(V_{w3} - V_{w2}) \\ 3V_{w1} = 0. \end{cases} \quad (50)$$

Assume $V_{w \max}$ to be the maximum linear velocity of wheels. Then, $\frac{\sqrt{3}}{3}(V_{w3} - V_{w2})$ is maximum when

$$\begin{cases} V_{w3} = V_{w \max} \\ V_{w2} = -V_{w \max}. \end{cases} \quad (51)$$

From (50) and (51), we get

$$V_{max} = \frac{2\sqrt{3}}{3}V_{w \max}. \quad (52)$$

Now we can calculate any maximum velocity $V_{\alpha \max}$ situated in the hexagon (Fig. 13(c)) for any angle α ($-90^\circ \leq \alpha \leq 90^\circ$).

1) For $-60^\circ \leq \alpha \leq 60^\circ$

In triangle ABC , we have

$$\frac{AB}{\sin(60^\circ)} = \frac{AC}{\sin(120^\circ - |\alpha|)}. \quad (53)$$

From (52) and (53), we obtain ($V_{max}=AC$)

$$V_{\alpha \max} = AB = \frac{V_{w \max}}{\sin(120^\circ - |\alpha|)}. \quad (54)$$

2) For $60^\circ \leq |\alpha| \leq 90^\circ$

We have

$$\frac{AB}{\sin(60^\circ)} = \frac{AC}{\sin(180^\circ - |\alpha|)}. \quad (55)$$

From (37), we get

$$V_{\alpha \max} = AB = \frac{V_{w \max}}{\sin(180^\circ - |\alpha|)}. \quad (56)$$

For the LF sensor with $\alpha = 45^\circ$

$$\begin{cases} \dot{x}_{l \max} = V_{45^\circ \max} \cos(45^\circ) = \frac{V_{w \max}}{\sin(75^\circ)} \cos(45^\circ) \\ \dot{y}_{l \max} = V_{45^\circ \max} \sin(45^\circ) = \frac{V_{w \max}}{\sin(75^\circ)} \sin(45^\circ). \end{cases} \quad (57)$$

Assume \dot{x}_{Obs} to be the maximum velocity of an obstacle in the global frame, and R_{ob} to be the radius of the circle surrounding the obstacle. When the obstacle moves toward the robot in the x_l direction, the time Δt , needed to traverse the distance $L+R_{obs}$ in y_l direction is

given by

$$\Delta t = \frac{L + R_{ob}}{\dot{y}_{lmax}}. \tag{58}$$

The relative speed between the robot and the obstacle in the x_l direction can be written as

$$\dot{x}_{lr} = \dot{x}_{lmax} + \dot{x}_{Gobs}. \tag{59}$$

Therefore, the safety distance can be written as

$$D_s = \dot{x}_{lr} \Delta t = (\dot{x}_{lmax} + \dot{x}_{Gob}) \frac{L + R_{ob}}{\dot{y}_{lmax}}. \tag{60}$$

Equation (60) can be rewritten as

$$D_s = \left(1 + \frac{\dot{x}_{Gob} \sin(75^\circ)}{V_{wmax} \sin(45^\circ)} \right) (L + R_{ob}). \tag{61}$$

We define the ‘‘Sensing Vector’’, which collects the sensory data, as follows:

$$V_s = [LS \ LFS \ FS \ RFS \ RS]. \tag{62}$$

If any sensor has detected an obstacle located at a distance less or equal to the safety distance D_s , its corresponding value gets the value ‘‘1’’, otherwise ‘‘0’’. Therefore, the ‘‘Sensing Vector’’ is a binary vector. An example is shown in Fig. 12, where

$$V_s = [0 \ 1 \ 0 \ 1 \ 1]. \tag{63}$$

We define a ‘‘Gap’’ as the free space between obstacles. The robot uses it to bypass them. A ‘‘Gap’’ can be defined as the existence of ‘‘0’’ in the ‘‘Sensing Vector’’. In the above example, there are two gaps, one on the front of the robot and the other on the left side which are corresponding to FS and LS sensors respectively.

The block diagram of the fuzzy ‘‘Obstacle Avoidance and Wall Following’’ behavior is shown in Fig. 14. The fuzzy controller inputs are the sensory data (‘‘Sensing Vector’’). The output is the three translational velocities of the robot wheels V_{w1} , V_{w2} and V_{w3} .

1) The fuzzification procedure

The linguistic values used for the sensory data are given in Table 4. We assign the label ‘‘Not Detected’’ to the value ‘‘0’’ in the ‘‘Sensing Vector’’ and the label ‘‘Detected’’ to the value ‘‘1’’.

We have adopted the linear movement to avoid obstacles using the nearest gap. We use the gap angle α (where $\alpha = 90^\circ, 45^\circ, 0^\circ, -45^\circ$ and -90° corresponding to LS, LFS, FS, RFS and RS , respectively) and the inverse kinematics to obtain the velocity values for each wheel (Fig. 15).

The inverse kinematics can be obtained from (15) as

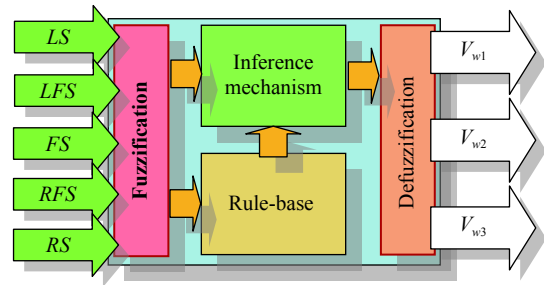


Fig. 14 Block diagram of the fuzzy ‘‘Obstacle Avoidance and Wall Following’’ behavior

Table 4 Linguistic values of sensory data

Linguistic term	Label
Not detected	ND
Detected	D

$$\begin{bmatrix} V_{w1} \\ V_{w2} \\ V_{w3} \end{bmatrix} = \begin{bmatrix} 0 & -\frac{\sqrt{3}}{3} & \frac{\sqrt{3}}{3} \\ \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} \\ \frac{1}{3L} & \frac{1}{3L} & \frac{1}{3L} \end{bmatrix}^{-1} \begin{bmatrix} \dot{x}_l \\ \dot{y}_l \\ \dot{\phi}_l \end{bmatrix}. \tag{64}$$

From Fig. 15, we calculate the components of $V_{\alpha \max}$ as

$$\begin{cases} V_{x\alpha \max} = V_{\alpha \max} \cos(\alpha) = \frac{V_w \max}{\sin(120^\circ - |\alpha|)} \cos(\alpha) \\ V_{y\alpha \max} = V_{\alpha \max} \sin(\alpha) = \frac{V_w \max}{\sin(120^\circ - |\alpha|)} \sin(\alpha). \end{cases} \tag{65}$$

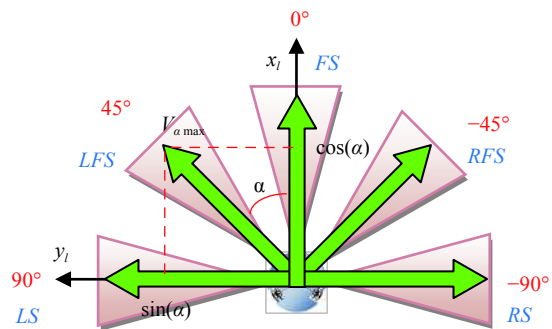


Fig. 15 Schematic diagram to calculate the velocity of each gap

From (64), we can obtain the linear velocity of wheels by putting:

$$\begin{cases} \dot{x}_l = V_{x\alpha \max} \\ \dot{y}_l = V_{y\alpha \max} \\ \dot{\phi}_l = 0 \text{ (linear motion)}. \end{cases} \tag{66}$$

Then, we get

$$\begin{bmatrix} V_{w1} \\ V_{w2} \\ V_{w3} \end{bmatrix} = \begin{bmatrix} 0 & -\frac{\sqrt{3}}{3} & \frac{\sqrt{3}}{3} \\ \frac{2}{3} & \frac{1}{3} & -\frac{1}{3} \\ \frac{1}{3L} & \frac{1}{3L} & \frac{1}{3L} \end{bmatrix}^{-1} \begin{bmatrix} \frac{V_w \max}{\sin(120 - |\alpha|)} \cos(\alpha) \\ \frac{V_w \max}{\sin(120 - |\alpha|)} \sin(\alpha) \\ 0 \end{bmatrix} \tag{67}$$

By changing values of α in (67), we obtain Table 5 which collects the linear velocities of wheels with their corresponding linguistic values. We suppose that the maximum translational velocity of wheels $V_{w \max} = 1$ (m/s).

The triangular membership functions of the sensory data and the velocities of wheels are given in Fig. 16.

2) Rule base design

We have five sensors. Hence, there are $2^5 = 32$ combinations of the distribution of obstacles between the sensors and, thus, 32 possible rules. But we can reduce this number to just 5 rules (Table 6), based on the gap search algorithm depicted in Fig. 17. Note that the rules 1, 2 and 3 are used to avoid obstacles while rules 4 and 5 are used for following walls.

The robot uses the information acquired from the sensors to look for the presence of the nearest gap or free space to bypass the obstacles in the environment. The strategy of the gap search algorithm is to find the nearest

gap to the frontal side of the robot. We can see the evolution in the flowchart by following the rule order in Table 6.

3) Defuzzification procedure

The ‘‘Bisector’’ method is used in the ‘‘Obstacle Avoidance and Wall Following’’ behavior.

4.3 Behaviors coordination

The architecture used for the proposed controller is the subsumption architecture which advocates the competitive selection of behaviors. In such architecture, a certain number of behaviors run as parallel processes. While each behavior can access all sensors, and only one behavior can have control over the robot’s actuators or driving

Table 5 Linguistic values of the velocities V_{wi}

Angle α	V_{w1}		V_{w2}		V_{w3}	
	Value	Label	Value	Label	Value	Label
-90°	-1	<i>NB</i>	0.5	<i>P</i>	0.5	<i>P</i>
-45°	-0.732	<i>N</i>	-0.268	<i>NS</i>	1	<i>PB</i>
0°	0	<i>Z</i>	-1	<i>NB</i>	1	<i>PB</i>
45°	0.732	<i>P</i>	-1	<i>NB</i>	0.268	<i>PS</i>
90°	1	<i>PB</i>	-0.5	<i>N</i>	-0.5	<i>N</i>

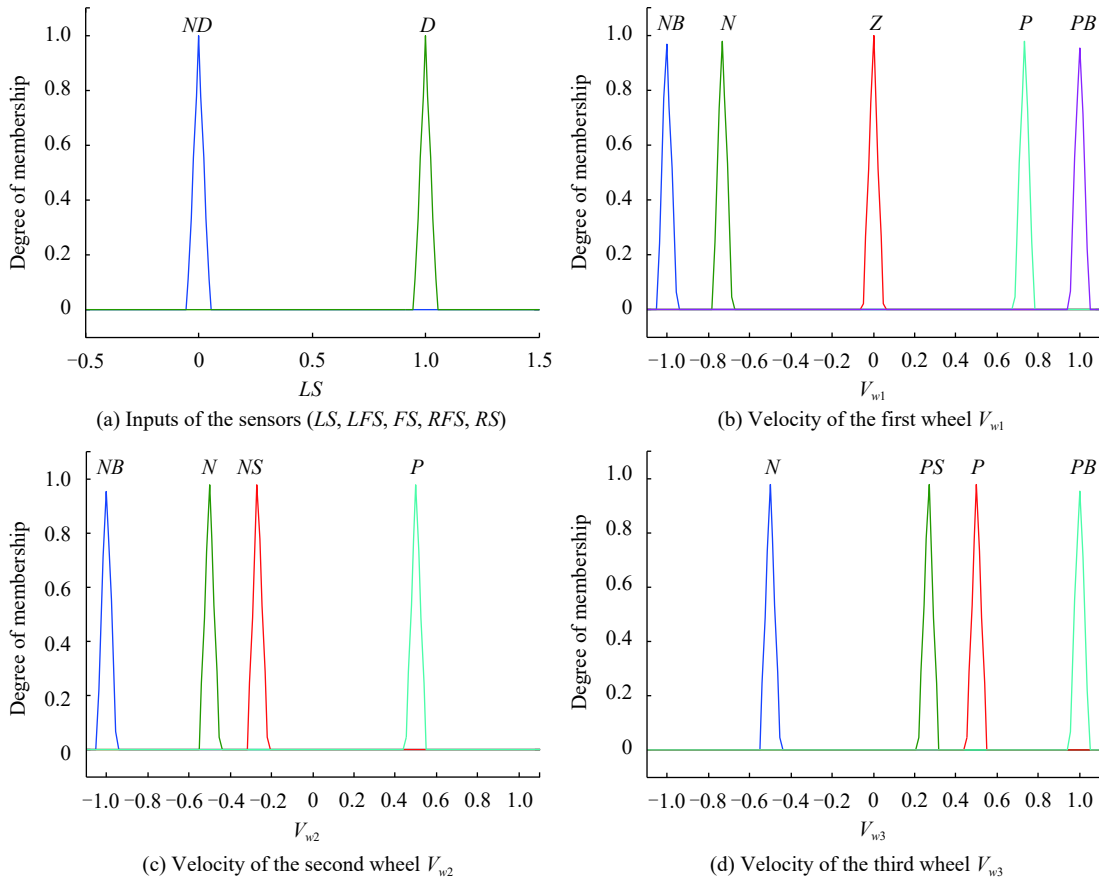


Fig. 16 Membership functions of the inputs of sensors, V_{w1} , V_{w2} , and V_{w3}

Table 6 Fuzzy rules for the “Obstacle Avoidance and Wall Following” behavior

<i>N</i>	Angle	<i>LS</i>	<i>LFS</i>	<i>FS</i>	<i>RFS</i>	<i>RS</i>	<i>V_{w1}</i>	<i>V_{w2}</i>	<i>V_{w3}</i>
1	0°	Any	Any	<i>ND</i>	Any	Any	<i>Z</i>	<i>NB</i>	<i>PB</i>
2	45°	Any	<i>ND</i>	<i>D</i>	Any	Any	<i>P</i>	<i>NB</i>	<i>PS</i>
3	-45°	Any	<i>D</i>	<i>D</i>	<i>ND</i>	Any	<i>N</i>	<i>NS</i>	<i>PB</i>
4	90°	<i>ND</i>	<i>D</i>	<i>D</i>	<i>D</i>	Any	<i>PB</i>	<i>N</i>	<i>N</i>
5	-90°	<i>D</i>	<i>D</i>	<i>D</i>	<i>D</i>	<i>ND</i>	<i>NB</i>	<i>P</i>	<i>P</i>

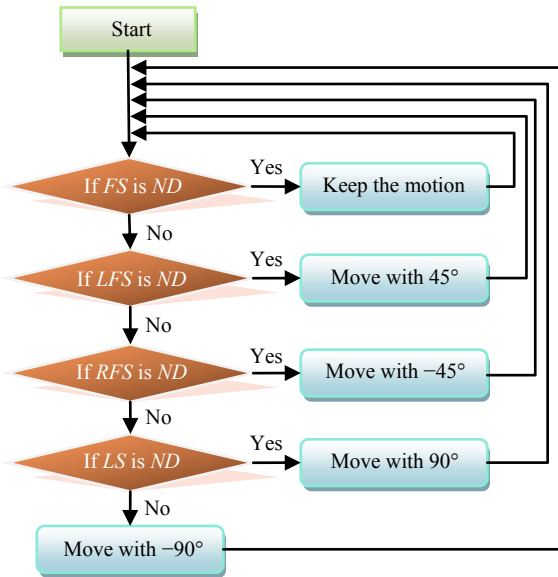


Fig. 17 Flowchart of the gap search algorithm used to construct the rule base

mechanism. Therefore, an overall controller is required to coordinate behavior selection^[44]. For integrating these basic behaviors, a behavior coordinator is designed; it uses an on-off switching schema: in each situation, one behavior is selected and is given complete control of the robot. We define the “Detection Parameter” *S* as

$$S = \text{Max}(V_s) = \begin{cases} 1, & \text{if there is an obstacle} \\ 0, & \text{otherwise} \end{cases} \quad (68)$$

where “Max” refers to the maximum element of the “Sensing Vector”, because *V_s* is a binary vector so “Max” is either 1 or 0. The flowchart (Fig. 18) of the behavior coordinator is based on two steps:

Step 1. The robot always starts with “Target Tracking” behavior (TT). If *S* = 0, there is neither obstacle nor wall at a distance less than the “Safety Distance”. The robot will continue with the “Target Tracking” behavior, otherwise, go to Step 2.

Step 2. If *S* = 1, there is an obstacle at a distance less than the “Safety Distance”. The “Target Tracking” (TT) behavior will be deactivated. The robot triggers the “Obstacle Avoidance and Wall Following” (OAWF) be-

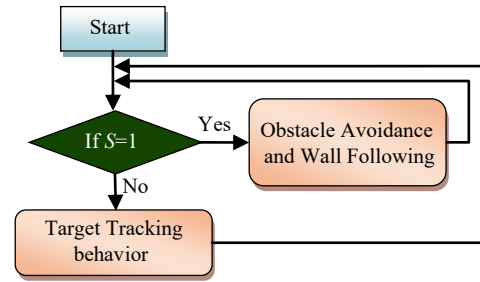


Fig. 18 Flow chart of the behavior coordinator

havior.

The flowchart can be translated by a simple equation

$$V_w = (1 - S) V_{TT} + S V_{OAWF} \quad (69)$$

where $V_w = [V_{w1} \ V_{w2} \ V_{w3}]^T$ is the whole resulted output, *V_{TT}* is the velocities vector resulted from the “Target Tracking” behavior and *V_{OAWF}* is the velocities vector resulted from the “Obstacle Avoidance and Wall Following” behavior.

5 Simulation results

To demonstrate the effectiveness of the proposed approach, simulation platform was designed using Matlab software. All simulations were executed using an AMD Athlon II P320 Dual-Core Processor with 4GB RAM running at 2.10GHz under Microsoft Windows 7. First, the simulation in the case of a static target is performed. Then, the navigation in the case of dynamic target tracking is simulated. The safety distance is taken to be *D_s* = 0.3m. Note that the robot has a green color. Its path is the solid red line, while the target is the red point circle and its path is the black dotted line. The dashed line is used for the path of the dynamic obstacles. Figs. 22–25 are constructed from many snapshots numbered in ascending order.

5.1 Static target

Static target is achieved as a special case of tracking a dynamic target moving with zero velocity.

In the first scenario, single obstacle avoidance is simulated (Fig. 19). In the first case, the robot is commanded to move from the starting point *A*(0.1, 1.4) to the target point *T* (1.4, 0.1) avoiding the obstacle located at (0.75, 0.75). In the second case, the obstacle is large.

Fig. 20 gives the simulation in the case of a cluttered static environment. The robot is commanded to move from the starting point *A*(3.7, 0.1) to the destination point *T* (0.2, 3.5). The robot starts with the “Target Tracking” behavior until meeting the first obstacle where this behavior is deactivated and the “Obstacle Avoidance and Wall Following” behavior is activated. After passing the obstacle and returning to track the target, it

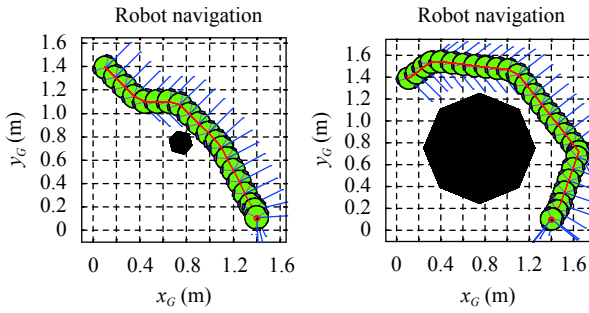


Fig. 19 Static single obstacle avoidance simulation

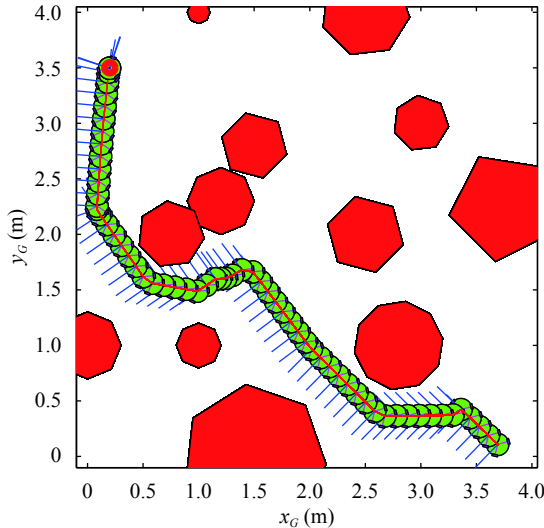


Fig. 20 Navigation in a cluttered static environment

encounters three obstacles forming a wall, so it turns left following the wall and bypasses it to continue towards the target.

Fig. 21 shows another scenario where the robot has to follow two walls to reach the target.

Fig. 22 shows the navigation in a dynamic environment where the robot has successfully reached the target. There are six obstacles in the environment, two static obstacles with black color and four dynamic obstacles

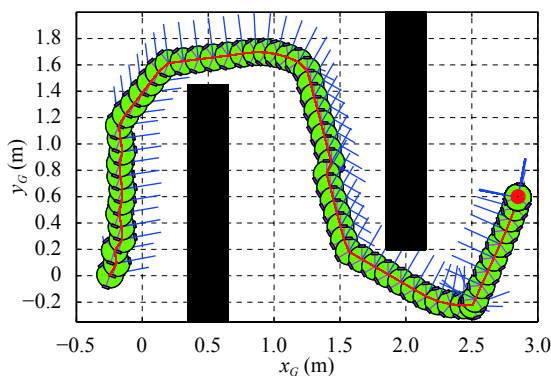


Fig. 21 Wall following

with red color. The robot is commanded to move from the starting point $A(0.25, 2.75)$ to reach the target point $T(2.75, 0.25)$ which is the red circle point (the red point at the right bottom of the snapshots).

5.2 Dynamic “Target Tracking”

For dynamic target tracking, the simulation is done for different scenarios. Note that the robot has no prior knowledge of the target path. It uses just the “Target Tracking” behavior to track the target.

In the first scenario (Fig. 23), the robot is commanded to track the dynamic target that moves in a circular path avoiding an obstacle located on this path. The robot begins with the “Target Tracking” behavior to track the target and continues with this behavior (Snapshots 1 to 6) until it reaches the target (Snapshot 7). It continues tracking the target (Snapshot 8) until it meets an obstacle which has entered the zone under the “Safety Distance” (Snapshot 9) where the robot deactivates the “Target Tracking” behavior and triggers the “Obstacle Avoidance and Wall Following” behavior to avoid the obstacle (Snapshots 9–11). After bypassing the obstacle (Snapshot 11), the “Obstacle Avoidance and Wall Following” behavior has been deactivated while “Target Tracking” behavior has triggered again to continue tracking the target (Snapshots 12 to 16).

Fig. 24 shows the second scenario where the robot has to track the dynamic target that moves in an eight-shape path.

In this scenario, there is no obstacle in the environment so that the robot navigates just with the “Target Tracking” behavior during this scenario. In Snapshots 1 to 5, the robot tracks the target until reaching it (Snapshot 6), and continues navigation in the rest of snapshots. So the robot has successfully tracked the target.

Fig. 25 shows the last example. In this more complex scenario, the robot has to track the target moving in a complex path which has the shape of a flower (the path with black color) in a dynamic environment crowded with three dynamic obstacles. These dynamic obstacles that move with different motions ensure multiple meetings with the robot in which the scenario will be more complicated. The first obstacle (with cyan color) moves in a circular path (with blue color), the second and the third obstacles (red and yellow color) move in oscillating motions in two straight lines, perpendicular to each other. Note that the robot has no prior knowledge of the motion of the target or the obstacles; the robot navigates just using the two behaviors.

The robot begins with the “Target Tracking” behavior (Snapshot 1), it continues with this behavior (Snapshot 2) until it meets the first obstacle (the yellow one, Snapshot 3) where the actual behavior is deactivated to trigger the “Obstacle Avoidance and Wall Following” behavior to avoid the obstacle.

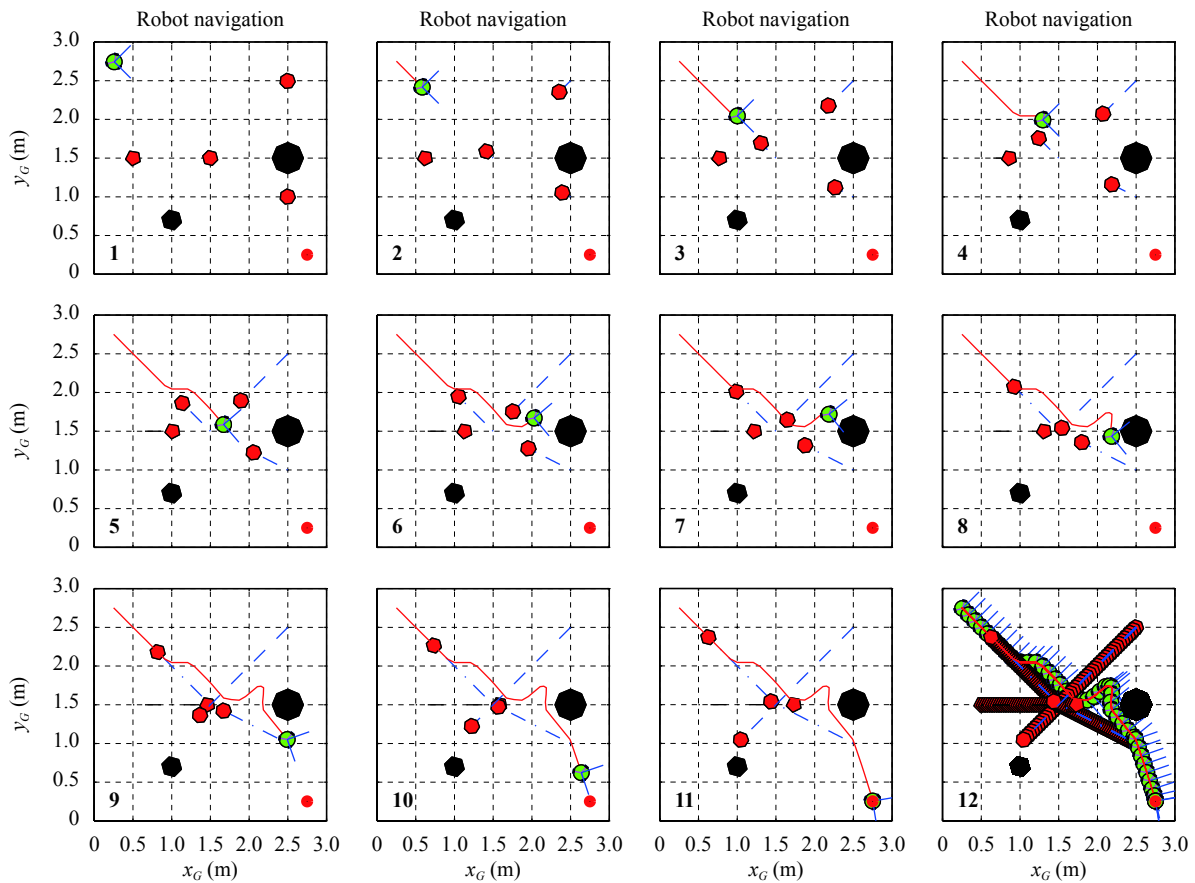


Fig. 22 Dynamic environment

After passing the obstacle and reaching the target (Snapshot 4) where the robot returns to the “Target Tracking” behavior and continues using it until it encounters two obstacles at the same time (the yellow and the red, Snapshot 5), the robot deactivates the “Target Tracking” behavior and triggers the “Obstacle Avoidance and Wall Following” behavior (Snapshot 6). It has used the gap appeared between the two obstacles for bypassing them (Snapshot 7). In Snapshot 8, the robot returned to the “Target Tracking” behavior, it has reached the target again (Snapshot 9), continuing with this behavior (Snapshots 10–13) till it detects the red obstacle entered the zone where the distance is less than the safety distance (Snapshot 14) where the “Target Tracking” behavior is deactivated to trigger the “Obstacle Avoidance and Wall Following” behavior to avoid the obstacle. In Snapshot 15, the robot returns to the “Target Tracking” behavior, it encounters the cyan obstacle (Snapshot 16), and moves to the right to avoid it (Snapshot 17), to enter a complex case by meeting the three obstacles at the same time (Snapshots 18–21), then it maintains the “Obstacle Avoidance and Wall Following” behavior to bypass the obstacles. In Snapshot 22, the robot returned to the “Target Tracking” behavior, reached the target (Snapshot 23) and continued with this behavior (Snapshot 24) until meeting the yellow obstacle (Snapshot 25),

and avoided it (Snapshot 26). For the rest two snapshots, the robot has used the “Target Tracking” behavior to track the target.

Simulation results demonstrate the effectiveness of the proposed approach where the robot has successfully tracked the target in complex scenarios in dynamic environments.

5.3 Comparison with other methods

A comparison was done with other similar approaches. First, a comparison of the number of behaviors and the number of rules is given in Table 7. Since the increase in the number of behaviors and the number of rules requires more computation, it seems from Table 7 that the proposed approach has the least number of behaviors (2 behaviors) and the least number of rules (11 rules).

Fig. 26 shows a comparison of the proposed approach with two other approaches, Cherroun and Boumehratz^[45] and Mo et al.^[46], so we consider two cases, and in each case, there are three similar scenarios. In the two cases, the robot has to reach the target in a cluttered environment. We can easily see that the proposed approach (Fig. 26(c)) takes the shortest path with less curvature, while the robot in Fig. 26(b) takes a longer path by bypassing the obstacles, and hits an obstacle in the first

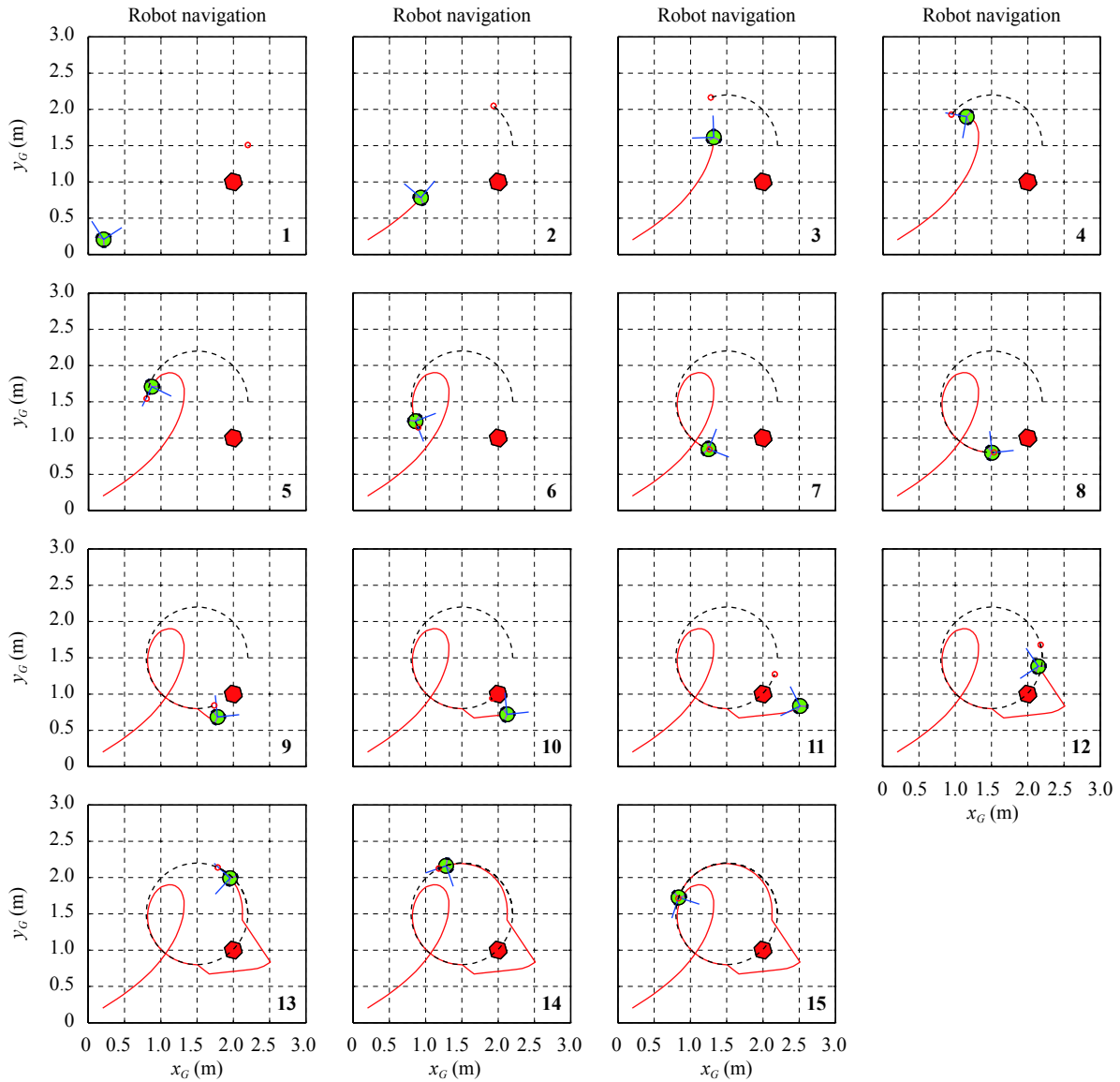


Fig. 23 Tracking a dynamic target which moves in a circular path

case. In the first scenario (Fig. 26(a)), the robot also took a long curve to bypass the obstacles which increased the length of the path. Hence, we can say that the proposed approach with a minimum number of behaviors and a minimum number of rules performs better compared to many other approaches which use more behaviors and more rules.

5.4 Tracking performance

To test the performance, the response time and the tracking error were considered. The controller step response corresponds to having the robot reaching a steady target located at a distance to the robot and staying at this location (i.e., point-to-point stabilization). The robot initially is located at the origin, and the desired target is located at 8 different positions given as follows:

$$[x_T, y_T] = \left[0.5 \cos \left(\frac{n\pi}{4} \right), 0.5 \sin \left(\frac{n\pi}{4} \right) \right], n = 0, \dots, 7. \tag{70}$$

The tracking error is given by

$$e_{tr} = \begin{bmatrix} e_x \\ e_y \end{bmatrix} = \begin{bmatrix} x_R - x_T \\ y_R - y_T \end{bmatrix} \tag{71}$$

where e_x is the tracking error in the x direction, i.e., the difference between the robot abscissa and the target abscissa while e_y is the tracking error in the y direction or the difference between the robot abscissa and the target abscissa. The tracking errors are illustrated in Fig. 27.

The results are compared with those of Huang et al.^[4], which presents a fuzzy controller for three-wheeled omnidirectional mobile robots to achieve trajectory tracking and stabilization. The fuzzy controller employs 50 fuzzy rules for tuning the proportion integral (PI) parameters.

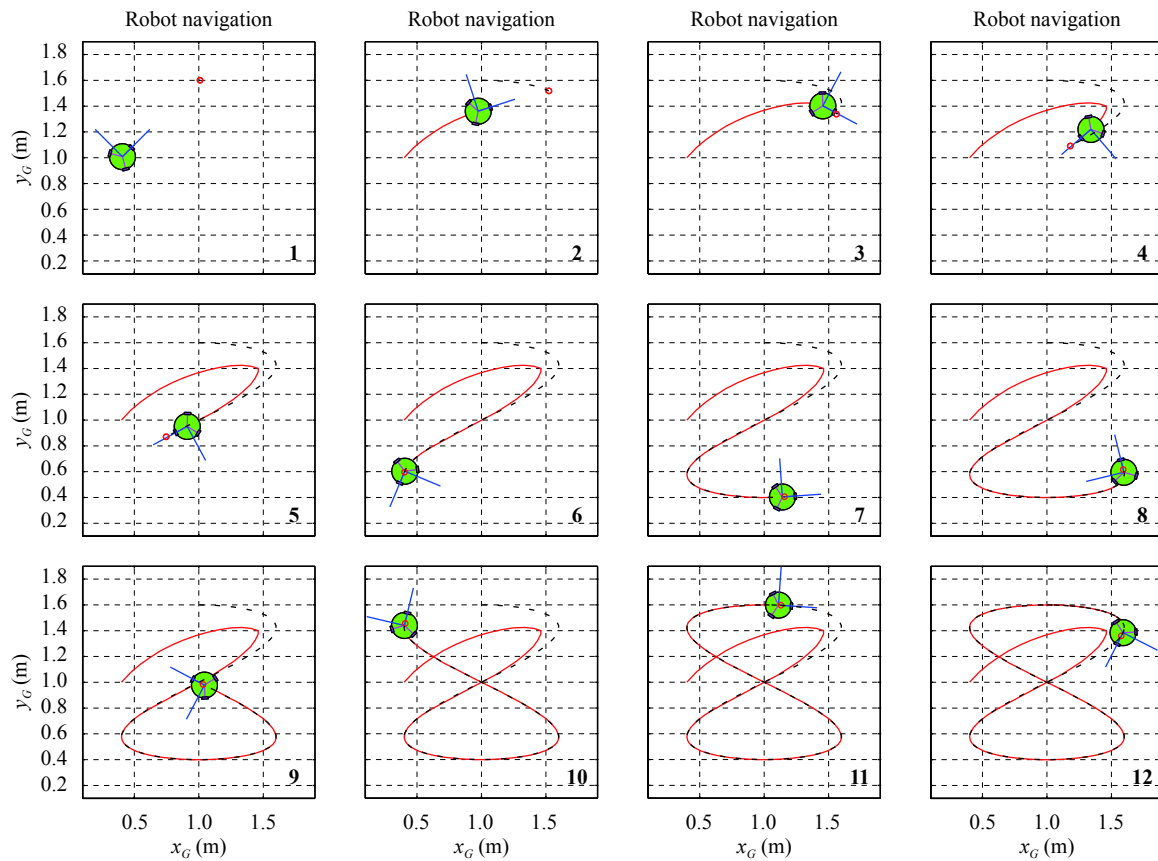


Fig. 24 Tracking a dynamic target moving in an eight-shape path

From Figs. 27 and 28, the results of Huang et al.^[1] show the robot reaches the target but not in a straight line, there is a deviation from the straight line connecting the target with the initial position of the robot (the origin), and the response time is about 4s, while in our proposed approach, the robot can reach the target in a straight line in minimum time. The response time is 0.44s and the tracking error in the x direction is less than 4mm. From the results, we can say that our proposed approach performs better than of Huang et al.^[1], furthermore, the proposed approach employs just 11 fuzzy rules compared with 50 fuzzy rules used by Huang et al.^[1]

Two other cases are depicted in Fig. 29, the first is the tracking of a target moving in a circular trajectory, and the second case is the tracking of a target moving in an eight-shape trajectory. From Fig. 29, we can see that the tracking errors are about 2 centimeters at the maximum. The robot can track either a static or a dynamic target successfully, which demonstrate the effectiveness of the proposed approach.

5.5 Failure situation

There is a failure situation where the robot may collide with obstacles. This situation occurs rarely when an obstacle moving with a velocity greater than the velocity of the robot hits the robot from its rear side (Fig. 30).

However, this situation is difficult to avoid it even for a human being, when an object hits from the rear. We can overcome this failure situation by adding more sensors to the rear side of the robot, but unfortunately, this increases the implementation cost and needs more fuzzy rules and more computations.

5.6 Evaluation of the cost of implementation

The TWOMR uses three plastic driving omni-wheels 50 mm in diameter and load capacity of 20kg, which are suitable for our design. The wheels are attached to three geared DC motors equipped with encoders to estimate the velocity of the wheels, thus by using the TWOMR kinematics, we can calculate the translational velocity of the robot in the x and y directions and the angular velocity, hence the robot position and orientation can be calculated by integration:

$$\begin{cases} x_G = x_{G,0} + \dot{x}_G \Delta t \\ y_G = y_{G,0} + \dot{y}_G \Delta t \\ \varphi_G = \varphi_{G,0} + \dot{\varphi}_G \Delta t \end{cases} \quad (72)$$

where (x_G, y_G, ϕ_G) is the actual pose, $(x_{G,0}, y_{G,0}, \varphi_{G,0})$ is the previous pose. $(\dot{x}_G, \dot{y}_G, \dot{\varphi}_G)$ is the actual velocity of the robot, and Δt is the sample time.

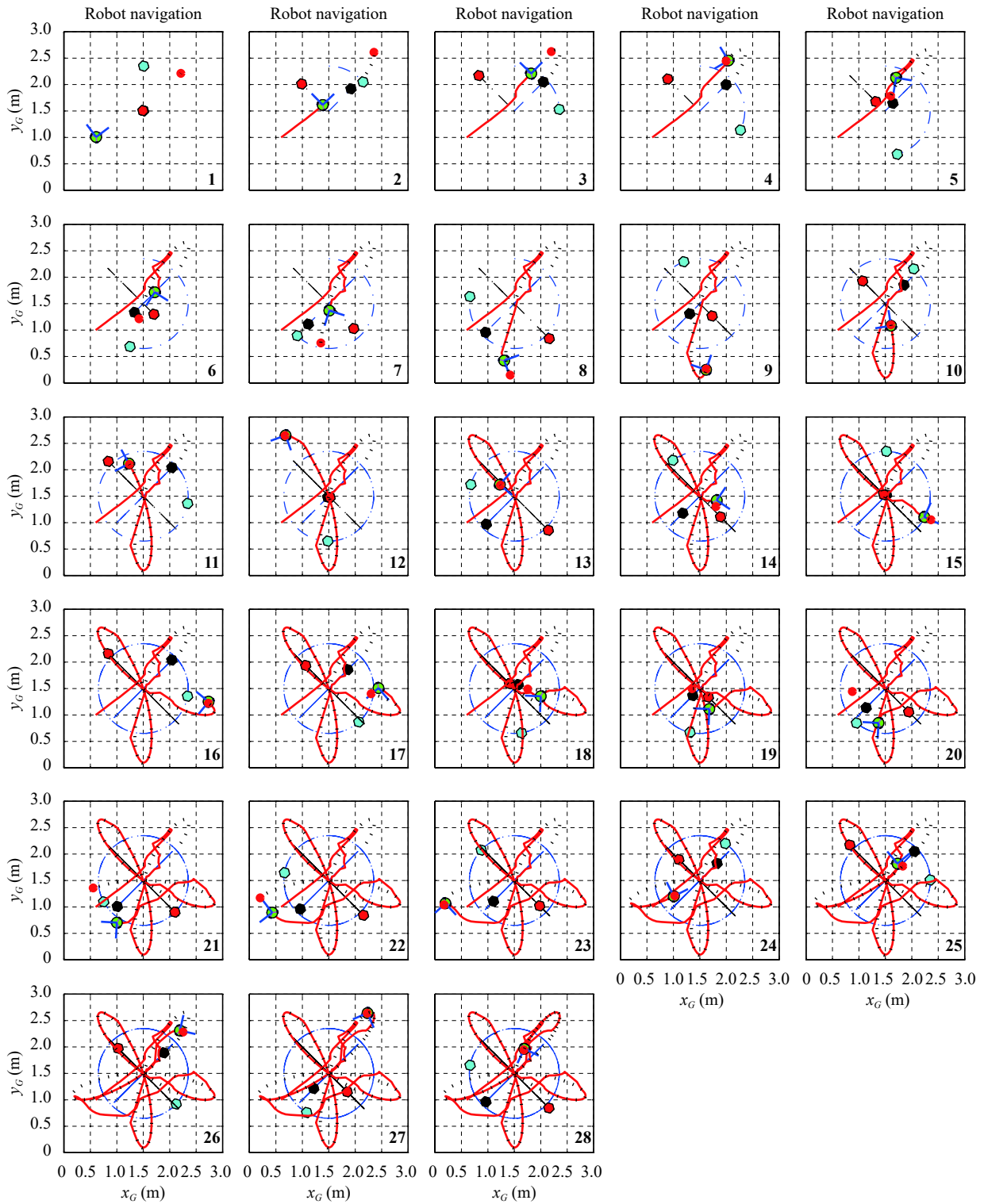


Fig. 25 Tracking a dynamic target which moves in a flower-shape path

When implementing the fuzzy logic control in a PIC controller, there were some memory as well as speed restrictions to be taken care of. However, the advantage of our approach that has a minimum number of behaviors (2 behaviors) and a minimum number of rules (11 rules) is that it does not need a large memory space and fewer

computations are needed. On the other hand, we have used Singleton fuzzification since it provides certain savings in the computations needed to implement a fuzzy system relative to, for example, “Gaussian fuzzification”, which would involve bell-shaped membership functions about input points, or triangular fuzzification, which uses

Table 7 Comparison with other methods

Reference	Number of behaviors	Designation	Number of rules	Total
[39]	6	Front obstacle avoidance	27	110
		Right obstacle avoidance	27	
		Left obstacle avoidance	27	
		Goal seeking	5	
		Obstacle avoidance	9	
		Overturning avoidance	15	
[40]	4	Emergency	≥ 2	More than 77
		Avoid-obstacles	9×4	
		Move-to-point	7×2	
		Wall following	$16 + 9$	
[43]	3	Goal-seeking	2	105
		Path-searching	$9 + 27 + 9$	
		Obstacle avoidance	$27 + 4 + 27$	
[45]	2	Goal seeking	35	56
		Obstacle avoidance	21	
[45]	5	Goal seeking	35	65
		Front obstacle avoidance	8	
		Right obstacle avoidance	8	
		Left obstacle avoidance	8	
		Velocity reducing	6	
[46]	3	Goal seeking	35	89
		Obstacle avoidance	27	
		Behavior fusion	27	
Proposed approach	2	“Target Tracking”	6	11
		“Obstacle Avoidance and Wall Following”	5	

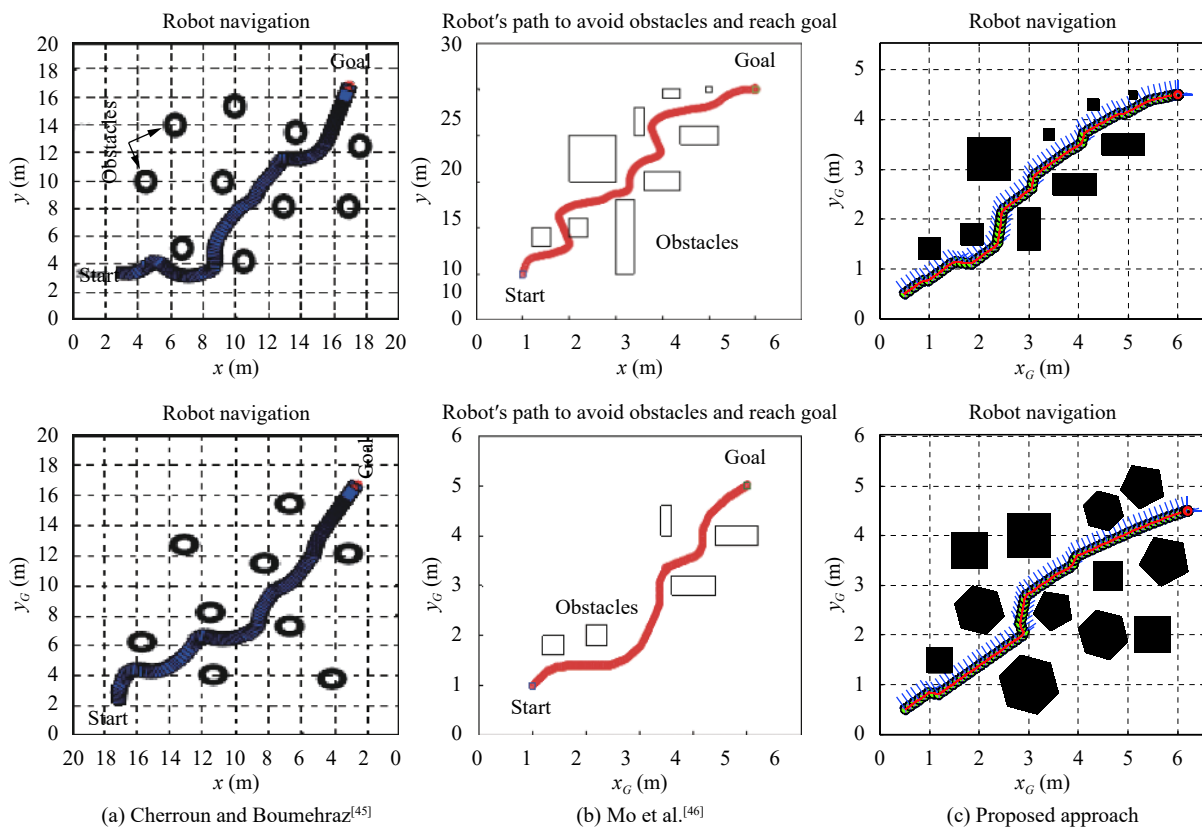


Fig. 26 Comparison between the proposed approach and two approaches

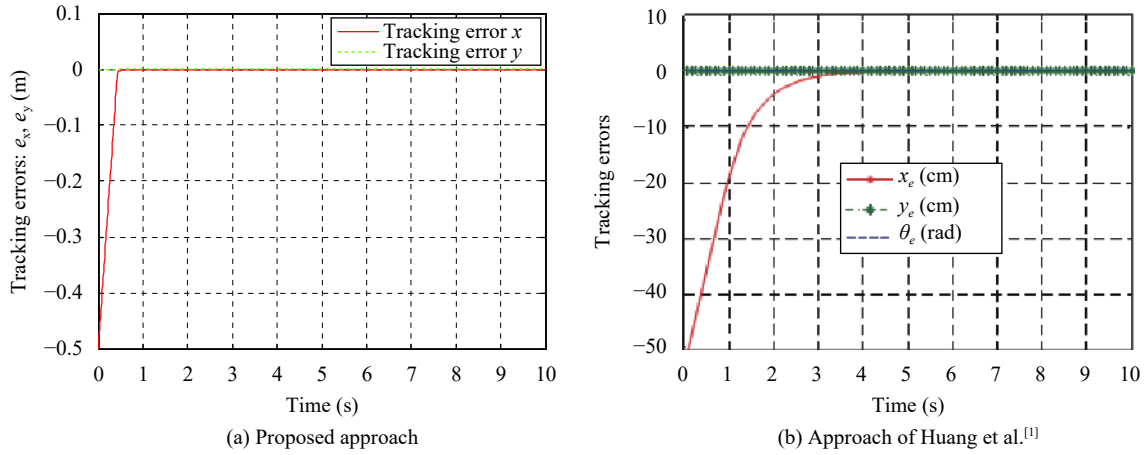


Fig. 27 Tracking errors for the target located at $[x_T, y_T] = [0.5, 0]$

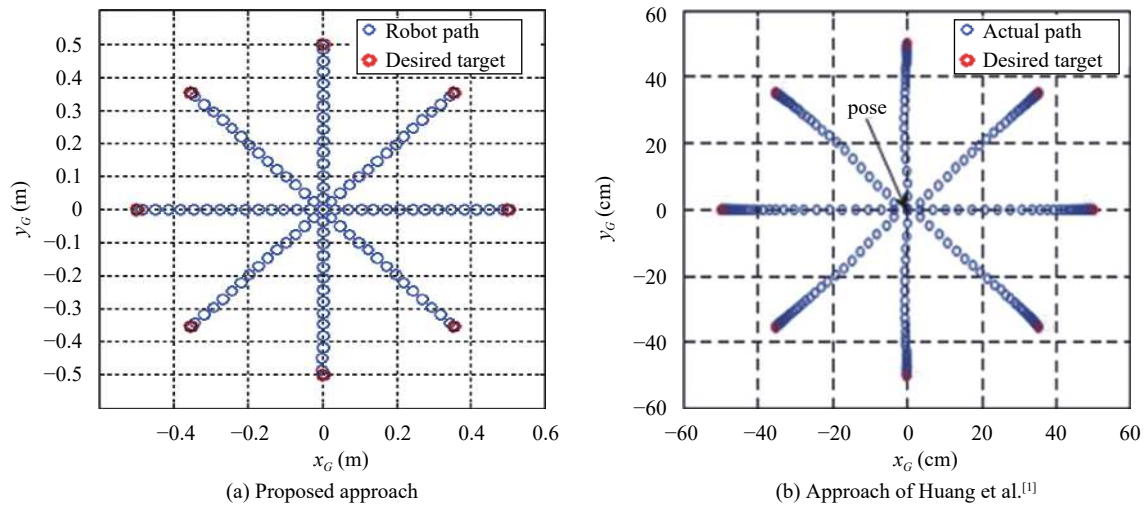


Fig. 28 Regulation performance

triangles. Hence, a first choice is to use the PIC 16F877 controller that has a low cost. But unfortunately, it could not drive three DC motors independently since it has just two independent pulse width modulation (PWM) modules. Another solution is to use a controller that has three or more PWM modules like the PIC18F4431 or the Arduino board which are suitable for our design. Our choice is PIC18F4431 which is less costly compared with the Arduino. The PIC18F4431 has the power control PWM module that supports four PWM generators and eight channels and 4 independent timers. Thus, three PWM signals can be generated independently to be applied to three H-bridges in order to convert the 12V of the battery into an average DC voltage in the motors.

The L298N dual H-bridge has two h-bridge drivers, which can drive two DC-motors. So two modules are needed, one to control two motors and one to control the third motor.

Five HC-SR04 low-cost ultrasonic sensors are used. This offers precise ranging information from roughly 2 cm to 400 cm with a ranging accuracy of 3 mm. Each HC-

SR04 module includes an ultrasonic transmitter, a receiver and a control circuit, and it has four pins: VCC (voltage at the common collector, power), Trig (trigger), Echo (receive), and GND (ground).

The main parts of the robot and their costs are given in Table 8. The total cost was evaluated at 85 USD.

6 Conclusions

In this paper, a fuzzy behavior-based approach for three-wheeled omnidirectional mobile robot navigation has been proposed. The robot has to track a dynamic target while avoiding obstacles along its path. To do so, two fuzzy behaviors “Track the Target” and “Avoid Obstacles and Wall Following” are designed based on a minimum number of behaviors and a minimum number of fuzzy rules. The outputs of the controller are the translational wheels velocities so that the controller controls the robot directly. Simulation results demonstrate the effectiveness of the proposed approach. The robot can track a dynamic target avoiding obstacles along its path success-

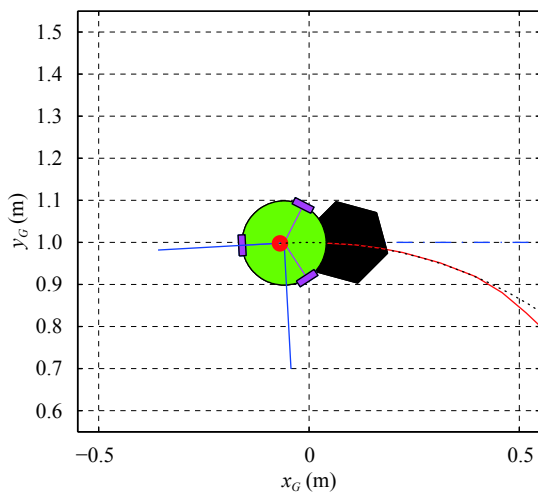
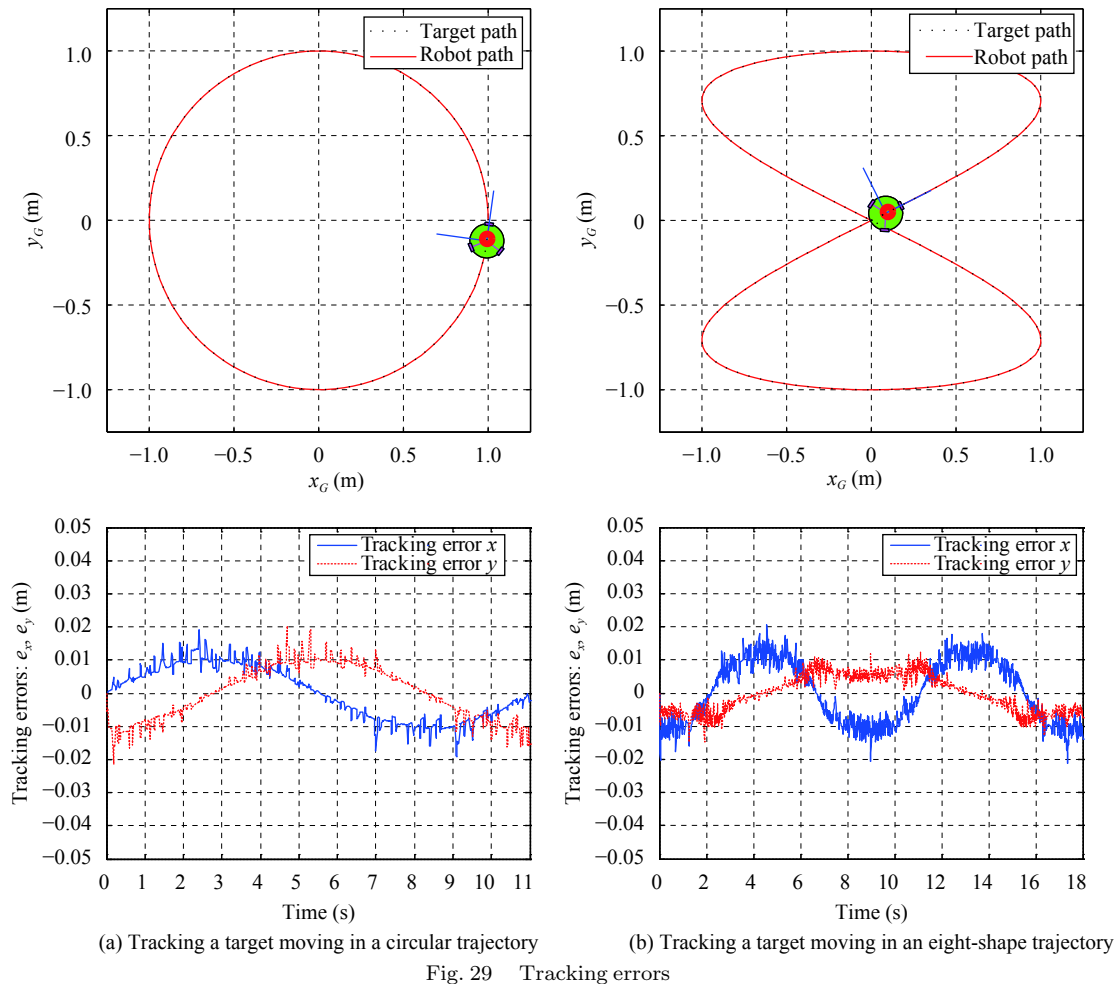


Fig. 30 Failure situation when an obstacle hits the robot from the rear side

fully. This approach can be improved by incorporating rear sensors and developing a strategy to estimate obstacle velocity to enable the robot to cope with back dynamic obstacles. Future work should look at this possibility.

Table 8 Evaluation of the implementation cost

Parts	Unit cost (USD)	Quantity	Cost
Controller PIC18F4431	4.51	1	4.51
DC motors	12.59	3	37.77
L298N dual H-bridge	1.89	2	3.78
Ultrasonic sensors HC-SR04	2.50	5	12.50
Omni-wheels	4.74	3	14.22
Battery	11.59	1	11.59
Total cost			84.37

References

- [1] H. C. Huang, T. F. Wu, C. H. Yu, H. S. Hsu. Intelligent fuzzy motion control of three-wheeled omnidirectional mobile robots for trajectory tracking and stabilization. In *Proceedings of International Conference on Fuzzy Theory and its Application*, Taiwan, China, pp. 421–426, 2012.
- [2] T. Kalmár-Nagy, R. D’Andrea, P. Ganguly. Near-optimal dynamic trajectory generation and control of an omnidirectional vehicle. *Robotics and Autonomous Systems*, vol. 46, no. 1, pp. 47–64, 2004. DOI: [10.1016/j.robot.2003.10.003](https://doi.org/10.1016/j.robot.2003.10.003).
- [3] C. Ren, S. G. Ma. Generalized proportional integral ob-

- server based control of an omnidirectional mobile robot. *Mechatronics*, vol.26, pp.36–44, 2015. DOI: [10.1016/j.mechatronics.2015.01.001](https://doi.org/10.1016/j.mechatronics.2015.01.001).
- [4] Y. N. Zhang, T. Huang. Research on a tracked omnidirectional and cross-country vehicle. *Mechanism and Machine Theory*, vol.87, pp.18–44, 2015. DOI: [10.1016/j.mechmachtheory.2014.12.016](https://doi.org/10.1016/j.mechmachtheory.2014.12.016).
- [5] C. Ren, S. Ma, Y. Sun, C. Ye. A continuous dynamic modeling approach for an omnidirectional mobile robot. *Advanced Robotics*, vol.19, no.4, pp.253–271, 2015. DOI: [10.1080/01691864.2014.978372](https://doi.org/10.1080/01691864.2014.978372).
- [6] H. Kim, B. K. Kim. Minimum-energy translational trajectory planning for battery-powered three-wheeled Omnidirectional mobile robots. In *Proceedings of the 10th International Conference on Control, Automation, Robotics and Vision*, Hanoi, Vietnam, pp.1730–1735, 2008.
- [7] M. Takahashi, T. Suzuki, F. Cinquegrani, R. Sorbello, E. Pagello. A mobile robot for transport applications in hospital domain with safe human detection algorithm. In *Proceedings of IEEE International Conference on Robotics and Biomimetics*, Guilin, China, pp.1543–1548, 2009.
- [8] D. F. Luo, T. Schauer, M. Roth, J. Raisch. Position and orientation control of an omni-directional mobile rehabilitation robot. In *Proceedings of IEEE International Conference on Control Applications*, Dubrovnik, Croatia, pp.50–56, 2012.
- [9] F. Künemund, C. Kirsch, D. Heb, C. Röhrig. Fast and accurate trajectory generation for non-circular omnidirectional robots in industrial applications. In *Proceedings of the 7th German Conference on Robotics*, Munich, Germany, pp.377–382, 2012.
- [10] E. J. Jung, B. J. Yi. Study on intelligent human tracking algorithms with application to omni-directional service robots. In *Proceedings of the 10th International Conference on Ubiquitous Robots and Ambient Intelligence*, Jeju, Korea, pp.80–81, 2013.
- [11] Y. Inoue, T. Hirama, M. Wada. Design of omnidirectional mobile robots with ACROBAT wheel mechanisms. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Tokyo, Japan, pp.4852–4859, 2013.
- [12] S. G. Ma, C. Ren, C. L. Ye. An omnidirectional mobile robot: Concept and analysis. In *Proceedings of IEEE International Conference on Robotics and Biomimetics*, Guangzhou, China, pp.920–925, 2012.
- [13] C. L. Ye, S. G. Ma. Development of an omnidirectional mobile platform. In *Proceedings of IEEE International Conference on Mechatronics and Automation*, Changchun, China, pp.1111–1115, 2009.
- [14] B. A. Gebre, Z. Ryter, S. R. Humphreys, S. M. Ginsberg, S. P. Farrell, A. Kauffman, W. Capon, W. Robbins, K. Pochiraju. A multi-ball drive for omni-directional mobility: Prototyping a concept for a practical and agile omnidirectional mobility platform with the aid of 3D printing Technology. In *IEEE International Conference on Technologies for Practical Robot Applications*, Woburn, USA, 2014.
- [15] M. Tavakoli, C. Viegas, L. Marques, J. N. Pires, A. T. de Almeida. OmniClimbers: Omni-directional magnetic wheeled climbing robots for inspection of ferromagnetic structures. *Robotics and Autonomous Systems*, vol.61, no.9, pp.997–1007, 2013. DOI: [10.1016/j.robot.2013.05.005](https://doi.org/10.1016/j.robot.2013.05.005).
- [16] M. Tavakoli, C. Viegas. Analysis and application of dual-row omnidirectional wheels for climbing robots. *Mechatronics*, vol.24, no.5, pp.436–448, 2014. DOI: [10.1016/j.mechatronics.2014.04.003](https://doi.org/10.1016/j.mechatronics.2014.04.003).
- [17] K. Watanabe, Y. Shiraishi, S. G. Tzafestas, J. Tang, T. Fukuda. Feedback control of an omnidirectional autonomous platform for mobile service robots. *Journal of Intelligent and Robotic Systems*, vol.22, no.3-4, pp.315–330, 1998.
- [18] C. Ren, S. G. Ma. Dynamic modeling and analysis of an omnidirectional mobile robot. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Tokyo, Japan, pp.4860–4865, 2013.
- [19] H. Kim, B. K. Kim. Online minimum-energy trajectory planning and control on a straight-line path for three-wheeled omnidirectional mobile robots. *IEEE Transactions on Industrial Electronics*, vol.61, no.9, pp.4771–4779, 2014. DOI: [10.1109/TIE.2013.2293706](https://doi.org/10.1109/TIE.2013.2293706).
- [20] J. C. L. Barreto S., A. G. S. Conceição, C. E. T. Dorea, L. Martinez, E. R. de Pieri. Design and implementation of model-predictive control with friction compensation on an omnidirectional mobile robot. *IEEE/ASME Transactions on Mechatronics*, vol.19, no.2, pp.467–476, 2014. DOI: [10.1109/TMECH.2013.2243161](https://doi.org/10.1109/TMECH.2013.2243161).
- [21] M. A. Sharbafi, C. Lucas, R. Daneshvar. Motion control of omni-directional three-wheel robots by brain-emotional-learning-based intelligent controller. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications And Reviews)*, vol.40, no.6, pp.630–638, 2010. DOI: [10.1109/TSMCC.2010.2049104](https://doi.org/10.1109/TSMCC.2010.2049104).
- [22] H. Sira-Ramirez, C. Lopez-Urbe, M. Velasco-Villa. Real-time linear control of the omnidirectional mobile robot. In *Proceedings of the 49th IEEE Conference on Decision and Control*, Atlanta, USA, pp.4263–4268, 2010.
- [23] H. Seraji, A. Howard. Behavior-based robot navigation on challenging terrain: A fuzzy logic approach. *IEEE Transactions on Robotics and Automation*, vol.18, no.3, pp.308–321, 2002. DOI: [10.1109/TRA.2002.1019461](https://doi.org/10.1109/TRA.2002.1019461).
- [24] C. C. Shing, P. L. Hsu, S. S. Yeh. T-S fuzzy path controller design for the omnidirectional mobile robot. In *Proceedings of IECON the 32nd Annual Conference on IEEE Industrial Electronics*, Paris, France, pp.4142–4147, 2006.
- [25] Y. H. Wu, Z. F. Yuan. Motion compensation of omnidirectional wheel robot using neural networks. In *Proceedings of 6th International Conference on Intelligent Systems Design and Applications*, Ji'nan, China, pp.147–151, 2006.
- [26] Y. Jiang, S. Y. Wang, B. D. Bai. A neural-network-based robust control strategy applying to omnidirectional lower limbs rehabilitation robot during centre-of-gravity shift. In *Proceedings of IEEE International Conference on Mechatronics and Automation*, Changchun, China, pp.4907–4912, 2009.
- [27] H. C. Huang, C. C. Tsai, S. C. Lin. SoPC-based parallel elite genetic algorithm for global path planning of an autonomous omnidirectional mobile robot. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, San Antonio, USA, pp.1959–1964, 2009.
- [28] D. Xu, D. B. Zhao, J. Q. Yi, X. M. Tan. Trajectory tracking control of omnidirectional wheeled mobile manipulators: Robust neural network-based sliding mode approach. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol.39, no.3, pp.788–799, 2009. DOI: [10.1109/TSMCB.2008.2009464](https://doi.org/10.1109/TSMCB.2008.2009464).

- [29] H. C. Huang, C. C. Tsai. Design of kinematic controller based on ant colony optimization computing method for omnidirectional mobile robots. In *Proceedings of the 5th Conference on Industrial Electronics and Applications*, Taiwan, China, pp. 1287–1292, 2010.
- [30] H. C. Huang, C. C. Tsai. Particle swarm optimization algorithm for optimal configurations of an omnidirectional mobile service robot. In *Proceedings of Annual Conference*, Taiwan, China, pp. 2872–2877, 2010.
- [31] C. C. Tsai, H. C. Huang, S. C. Lin. FPGA-based parallel DNA algorithm for optimal configurations of an omnidirectional mobile service robot performing fire extinguishment. *IEEE Transactions on Industrial Electronics*, vol. 58, no. 3, pp. 1016–1026, 2011. DOI: [10.1109/TIE.2010.2048291](https://doi.org/10.1109/TIE.2010.2048291).
- [32] R. C. Arkin. *Behavior-based Robotics*, Cambridge, Massachusetts, USA: MIT Press, 1998.
- [33] R. A. Brooks. A Robust Layered Control System for a Mobile Robot, Technical Report AIM 864, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, USA, pp. 1–25, 1985.
- [34] R. C. Arkin. Motor schema-based mobile robot navigation. *The International Journal of Robotics Research*, vol. 8, no. 4, pp. 92–112, 1989. DOI: [10.1177/027836498900800406](https://doi.org/10.1177/027836498900800406).
- [35] K. Izumi, K. Watanabe. Fuzzy behavior-based tracking control for a mobile robot. In *Proceedings of the 2nd Asian Control Conference*, Seoul, Korea, vol. 1, pp. 685–688, 1997.
- [36] K. Watanabe, K. Izumi. Construction of fuzzy behavior-based control systems for a mobile robot. In *Proceedings of the Third International Symposium on Artificial Life and Robotics*, Beppu, Japan, vol. 2, pp. 518–523, 1998.
- [37] K. Izumi, K. Watanabe. Fuzzy behavior-based tracking control for an omnidirectional mobile robot. In *Proceedings of the Third International Symposium on Artificial Life and Robotics*, Beppu, Japan, vol. 2, pp. 524–527, 1998.
- [38] M. F. Selekwia, D. D. Dunlap, D. Q. Shi, E. G. Jr. Collins. Robot navigation in very cluttered environments by preference-based fuzzy behaviors. *Robotics and Autonomous Systems*, vol. 56, no. 3, pp. 231–246, 2008. DOI: [10.1016/j.robot.2007.07.006](https://doi.org/10.1016/j.robot.2007.07.006).
- [39] S. Zein-Sabatto, A. Sekmen, P. Koseyaporn. Fuzzy behaviors for control of mobile robots. *Systemics, Cybernetics and Informatics*, vol. 1, no. 1, pp. 68–74, 2003.
- [40] M. Wang, J. N. K. Liu. Fuzzy logic-based real-time robot navigation in unknown environment with dead ends. *Robotics and Autonomous Systems*, vol. 56, no. 7, pp. 625–643, 2008. DOI: [10.1016/j.robot.2007.10.002](https://doi.org/10.1016/j.robot.2007.10.002).
- [41] A. M. Rao, K. Ramji, B. S. K. Sundara Siva Rao, V. Vasu, C. Puneeth. Navigation of non-holonomic mobile robot using neuro-fuzzy logic with integrated safe boundary algorithm. *International Journal of Automation and Computing*, vol. 14, no. 3, pp. 285–294, 2017. DOI: [10.1007/s11633-016-1042-y](https://doi.org/10.1007/s11633-016-1042-y).
- [42] D. Nada, M. Bousbia-Salah, M. Bettayeb. Multi-sensor data fusion for wheelchair position estimation with unscented Kalman filter. *International Journal of Automation and Computing*, vol. 15, no. 2, pp. 207–217, 2017.
- [43] S. Naaz, A. Alam, R. Biswas. Effect of different defuzzification methods in a fuzzy based load balancing application. *IJCSI International Journal of Computer Science Issues*, vol. 8, no. 5, pp. 261–267, 2011.
- [44] L. Banjanovic-Mehmedovic, D. Lukac, M. Suljic. Biologically based behavior as inspiration for mobile robots navigations. In *Proceedings of IEEE EuroCon*, Zagreb, Croatia, pp. 1980–1987, 2013.
- [45] L. Cherroun, M. Boumehraz. Fuzzy behavior based navigation approach for mobile robot in unknown environment. *Journal of Electrical Engineering*, vol. 13, no. 4, pp. 284–291, 2013.
- [46] H. W. Mo, Q. R. Tang, L. L. Meng. Behavior-based fuzzy control for mobile robot navigation. *Mathematical Problems in Engineering*, vol. 2013, Article number 561451, 2013.



Nacer Hacene received the B.Eng. degree in automatic control from University of Mohamed Khider, Algeria in 2010 and the M.Eng. degree in automatic control and signal processing from University of Abderrahmane Mira, Algeria in 2014. He is currently a Ph.D. degree candidate in automatic control and signal processing at University of Abderrahmane Mira, Algeria

and a research fellow at the Industrial Technologies and Information Laboratory (LTII), Department of Electrical Engineering, University of Bejaia, Algeria. Currently, he is working as an assistant professor with Department of Science and Technology, Ghardaia University, Algeria.

His research interests include mobile robot control, artificial intelligence and swarm robotics.

E-mail: Hacenenacer77@gmail.com (Corresponding author)
ORCID iD: 0000-0002-8586-4590



Boubekeur Mendil received the B.Eng., M.Eng. and Ph.D. degrees from Setif University, Algeria, all in industrial control, in 1991, 1994, and 2002, respectively. Currently, he is a professor of robotics and automatic control with Electrical Engineering Department, Abderrahmane Mira University, Algeria. He is the head of the Soft-Ccomputing Research Group, LTII,

at Abderrahmane Mira University, Algeria.

His current research interests include mobile robots, soft-computing and motion control.

E-mail: bmendil@yahoo.fr