CEAI, Vol.26, No.1, pp. 25-35, 2024 Printed n Romania

Virtual navigation of mobile robot in V-REP using hybrid ANFIS-PSO controller

Amal Brahami*, Rabah Mellah* and Said Guermah*

**Laboratory of Design and Control of Production Systems, Faculty of Electrical Engineering and Computer Science, UMMTO, BP n°17, 15000, Tizi-Ouzou, Algeria;(e-mails: (brahami_amal@yahoo.fr) (mellah.rab@gmail.com) (saidguermah@yahoo.fr))*

Abstract: Trajectory following and obstacle avoidance are the main issues for autonomous navigation for mobile robots. *This article describes the virtual navigation of a Pioneer P3-DX differential drive mobile robot with two left and right wheels driven by DC motors and a freewheel to ensure its stability in two environments designed on the Virtual Robot Experimental Platform (V-REP). At first, we realized navigation in an environment without obstacle by tracking a target, first using the Adaptive Neuro Fuzzy Inference System (ANFIS) algorithm, then using the hybrid algorithm ANFIS-Particle Swarm Optimization (ANFIS-PSO) written in MATLAB code, and then we realized navigation in a cluttered environment that allows searching a target by avoiding obstacles with the ANFIS algorithm then with ANFIS-PSO. The objective of PSO algorithm is the minimization of the objective function which is the root mean square error between the actual and the predicted values of the ANFIS controller commands which are the left and right velocity of the mobile robot, to give optimum velocity values that will be applied to the robot for better trajectory tracking and obstacle avoiding. The Virtual Robot Experimentation Platform software V-REP was used to design our study and simulate the virtual reality of navigation and to work in synchronous mode with the codes created in MATLAB. A comparative study between navigation with ANFIS and navigation with ANFIS-PSO clearly shows the effectiveness of the PSO which ensures a good response to the system within these physical limitations (velocities, torques) and ensures optimal tracking for the virtual navigation of the mobile robot. Looking at the experimental results, it is observed that the proposed method provides a more efficient and optimal solution to the problem of target search and obstacle avoidance.*

Keywords: Mobile Robot, Pioneer P3-DX, Virtual navigation, V-REP, MATLAB, ANFIS, PSO.

1. INTRODUCTION

Advanced robots are still too expensive and can be exposed to damage during experimentation in laboratories. To avoid this, robotics researchers use the techniques of virtual reality, which is a combination of technologies that allow human interaction with the simulation created with a computer (Saiful et al., 2017). Recently, these virtual reality techniques have been developed by researchers in virtual laboratories using simulators for virtual environment design, in order to test their experiments, theories, ideas before real implementation (Ernesto et al., 2016; Faris et al., 2017). Several simulators are used in the field of robotics such as ARGoS (Pinciroli et al., 2011), Webots (Guyot et al., 2011) and V-REP (Coppelia Robotics, 2016).

Currently, mobile robots are widely used in security, industry, cleaning, medicine and many other fields (Shikha et al., 2021).

Recently, the navigation of an autonomous mobile robot uses intelligent controllers that allow the robot to search for the target and follow the shortest path, avoiding static or dynamic obstacles (Gharajeh et al., 2022). Therefore, navigation in the robotics field is divided into two main categories: local and global. Local areas are previously recognized by robots, while global areas represent areas that robots encounter for the first time. Because obstacles in global zones are unknown to robots, navigation in global zones remains one of the main problems in the robotics industry (Nursena et al., 2018).

These global and local navigation problems have been solved using several algorithms that can be classified into three categories: deterministic (e.g., fuzzy, neural network, neuro-fuzzy), non-deterministic or stochastic (e.g., Genetic Algorithm (GA), Particle Swarm Optimization (PSO), etc.) and evolutionary algorithms. The latter is the hybridization of deterministic and non-deterministic methods (Gharajeh et al., 2020).

Currently, neuro-fuzzy techniques are most widely used in mobile robot navigation because neuro-fuzzy algorithms are intelligent, knowledge-based deterministic techniques widely used for mobile robot navigation. These techniques can easily model the reasoning, uncertainty, and nonlinearity of complex environments. Autonomous collision-free navigation is defined as a set of abstract behaviors, including target finding, obstacle avoidance, etc.. (Gharajeh et al., 2020; Faris et al., 2017).

Recent studies have shown that different types of algorithms have been developed to solve the navigation problems of mobile robots. A paper (Batti et al., 2020) aims to carefully study two kinds of advanced approaches to guide a non-holonomic mobile robot to navigate in an environment cluttered with static obstacles, firstly a fuzzy logic controller (FLC) was designed, secondly a controller of adaptive neuro fuzzy inference system (ANFIS) was used to optimize the results obtained from the fuzzy controller, to validate the feasibility and efficiency of the proposed models. The V-REP and MATLAB software are used, a comparative evaluation is then made and simulation results showed that the mobile robot could successfully navigate in the environment by using two proposed approaches, but the ANFIS controller provided better results compared to the fuzzy controller (Batti et al., 2020).

In some studies, it is suggested that navigation algorithms and other optimization algorithms are combined to solve problems; a literature study proposes a hybrid GPS-ANFIS method for collision-free navigation of autonomous mobile robots. The GPS-based controller maintains the robot's navigation direction to the static or dynamic target. It uses the coordinates received from the two GPS modules on the edges of the longitudinal axis of the robot as well as the coordinates of the target to divert it from the current trajectory by making a certain angle towards the target (Gharajeh et al., 2020).

In another study, a control architecture for trajectory following while avoiding obstacles and PID controller tuning is proposed for a differential drive mobile robot (DMR). They propose a metaheuristic optimization algorithm to solve the problems path planning and controller tuning by choosing appropriate objective functions (Mubashir et al., 2020).

In another literature study, they described the navigation of a Pioneer P3-DX automated wheeled robot between obstacles using a network-tuned particle swarm optimization (PSO) algorithm. Predictive neural (FNN). From a comparative study it was found that a PSO-tuned FNN more efficient than FNN without tuning for the automated navigation of the Pioneer P3-DX wheeled robot. Additionally, they compared the PSO-optimized FNN with the previously developed PSO-optimized Fuzzy Logic Controller navigation technique to show the authenticity and real-time implementation of the PSO-optimized FNN (Anish et al., 2020).

Another literature study proposes an intelligent system for the navigation of mobile robots in different environments, using ANFIS and ACOr, at first, they used the ANFIS (Adaptive network-based fuzzy inference system) controller, in the second step , the ant colony method in a continuous environment ACOr (Ant Colony Optimization for Continuous Domains) is grafted into the second layer of the ANFIS network for hybridization, simulations of robot movements and graphical interfaces are performed under C++ language (Lazreg et al., 2019).

Our work consists in carrying out a virtual navigation of a mobile robot by using the techniques of virtual reality simulated in the virtual platform V-REP (Virtual Robot Experimentation Platform) which takes into consideration the parameters of the real environment, the commands are carried out on the MATLAB software by two methods. The first uses the neuro_fuzzy (ANFIS) algorithm (adaptive neuro-fuzzy inference system) of the Sugeno type the functions and the rules of membership are generated automatically by the adaptive techniques; the second method proposes a hybrid ANFIS-PSO algorithm to produce the best result. In the second we introduced the PSO optimization algorithm to ensure that our system (robot) remains within these physical limitations (limitation of wheels motor velocities) which are done over a velocity range so that they are limited between v_{min} and v_{max}. Then we establish a comparative study between navigation with ANFIS and ANFIS-PSO. V-REP and MATLAB communicate in synchronous mode such that V-REP receives commands from MATLAB for the navigation of the mobile robot in the virtual environment designed in order to seek the target and follow a shortest trajectory and avoid obstacles and it measures data (velocity, positions, orientation, distances to obstacles) which will be transmitted to MATLAB to realize a control loop.

2. MODELING

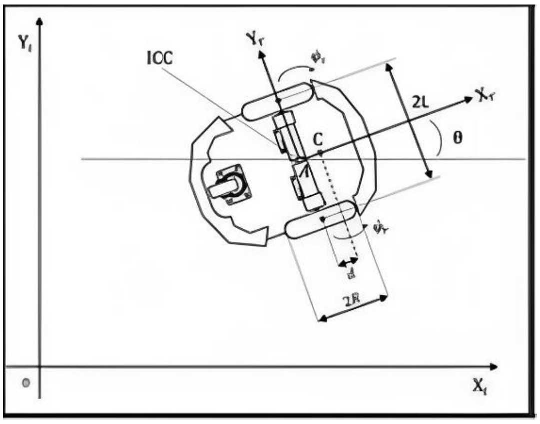

Fig. 1. Differential Drive Wheeled Mobile Robot (DDWMR) (Borenstein, 1998).

The mobile robot used in our study is of the differential drive type (unicycle) operated by two independent drive wheels (by DC motor) and has an idler wheel to ensure its stability.

For the modeling of the mobile robot in the first place it is necessary to define the coordinate systems such that $[X_I, Y_I]$ the fixed initial frame and $[X_r, Y_r]$ is a linked mobile frame to the robot, the position of t he robot is defined by a generalized coordinate vector $q = (x;\ y;\ \theta)$ (Saidi et al., 2019).

A: the origin of the mobile frame which is the center of the two wheels on the axis of the wheels.
C: robot center of gravity.
L: distance between A and the center of the wheel.
d: Distance between C and A.
θ : The angle between X_I and X_r
R: radius of each wheel.
$\dot{\varphi}_R$ and$\dot{\varphi}_L$: rotational velocities of the right and left wheels respectively.
ICC: Instant center of rotation
ρ: Radius of curvature of the robot trajectory (distance from ICC to center A).
L: distance between A and the center of the wheel.
ω: angular velocity around the ICC.

$$\begin{cases} v_R = -r\dot{\varphi}_R = (\rho + L)\omega & (1) \\ v_L = r\dot{\varphi}_L = (\rho - L)\omega & (2) \end{cases}$$

And (2) give: $\rho = L\frac{\dot{\varphi}_R - \dot{\varphi}_L}{\dot{\varphi}_R + \dot{\varphi}_L}$ which allows to locate the ICC on the axis of the wheels.

$\omega = \frac{-r\dot{\varphi}_R}{\rho + L} = -\frac{r(\dot{\varphi}_R + \dot{\varphi}_L)}{2L}$ which is the angular velocity of the robot around the ICC

If $\dot{\varphi}_R = -\dot{\varphi}_L$ the robot moves in a straight line.

If $\dot{\varphi}_R = \dot{\varphi}_L$ the robot performs a rotation on itself.

Mathematical modeling is a very important step for robot control; two types of models are generally used when ordering, namely: the kinematic model and the dynamic model.

2.1. Kinematic modeling

In the study of the kinematics, only the velocities are taken into account. The motion of a mobile robot is characterized by two non-holonomic kinematic constraints, namely: No lateral slip, rolling without slip (Dhaouadi et al., 2013).

Assumption no lateral slip: $-\dot{x}sin\theta + \dot{y}cos\theta = 0$

Non-slip rolling assumption for each wheel:

$\dot{x}cos\theta + \dot{y}sin\theta = r\dot{\varphi}$.

The kinematic behavior of the Differential Drive Wheeled Mobile Robot (DDWMR) is described by the combination of rolling constraints without wheel's slip and constraint no lateral slip.

A full derivation of the kinematic model is presented in (Dhaouadi et al., 2013; Saidi et al., 2019), such as:

$$\begin{cases} v = \frac{v_R+v_L}{2} = \frac{R(\dot{\varphi}_R+\dot{\varphi}_L)}{2} \\ w = \frac{v_R-v_L}{2L} = \frac{R(\dot{\varphi}_R-\dot{\varphi}_L)}{2L} \end{cases} \quad (3)$$

And the kinematic model is given by following system (4):

$$\begin{pmatrix} \dot{x}_A^I \\ \dot{y}_A^I \\ \dot{\theta}_A^I \end{pmatrix} = \begin{pmatrix} cos\theta & 0 \\ sin\theta & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ w \end{pmatrix} \quad (4)$$

In kinematic modeling, the command vector is composed of the linear velocity v and the angular velocity$w$. Due to non-holonomic limitation, the linear velocity v is in the direction of the robot's X_r axis. The theta angle is measured relative to the vertical Z axis, which is defined as positive pointing upward. The theta angle is zero when the forward direction of the robot chassis is aligned with the $X_r$axis of the robot; the angular velocity w angular velocity around the Instant center of rotation (ICC).

2.2. Dynamic modeling

Using the formalism of Euler Lagrange:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \left(\frac{\partial L}{\partial q_i}\right) = F + A^T(q).\lambda_k \quad (5)$$

L(q,$\dot{q}$)= T-V is the Lagrangian

T: The kinetic energy of the system.
V: The potential energy of the system.
F: The generalized force vector.
A^T : The vector of Lagrange multipliers.
q_i: The generalized coordinate and q = $[x_A; y_A; \theta_A; \varphi_R; \varphi_L]$ of size n =5.
A: Center of the wheels origin of the moving frame

The dynamic model is represented by the linear velocity and the angular velocity ,full derivation of this model is shown in (Dhaouadi et al., 2013; Saidi et al., 2019).

Moreover, it is given by system (6):

$$\begin{bmatrix} m + \frac{2I_w}{R^2} & 0 \\ 0 & I + \frac{2L^2}{R^2} I_w \end{bmatrix} \begin{pmatrix} \dot{v} \\ \dot{w} \end{pmatrix} + \begin{bmatrix} 0 & -m_c dw \\ m_c dw & 0 \end{bmatrix} \begin{pmatrix} v \\ w \end{pmatrix} = \begin{pmatrix} \frac{1}{R} & 0 \\ 0 & \frac{L}{R} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \quad (6)$$

With $u = \begin{cases} u_1 = \tau_R + \tau_L \\ u_2 = \tau_R - \tau_L \end{cases}$

m_c : Platform mass.
I_w: Moment of inertia of each wheel with the motor relative to the axis of the wheel.
m: robot mass.

3. VIRTUAL ENVIRONMENT

V-REP (Virtual Robot Experimentation Platform) is a virtual platform for designing virtual robot simulation environments. It has a remote API that facilitates data retrieval from V-REP and control command from MATLAB (Sharma et al., 2020).

V-REP contains a large number of examples, models of robots, sensors, and actuators. New models can also be designed and added to VREP to implement custom simulation experiences which guarantee a design of virtual environments very close to reality.

In this work we chose Pionner_P3DX which is a non-holonomic robot that has two driving wheels and a free wheel, it contains six ultrasound sensors in the front and six in the rear in order to ensure good measurement precession.

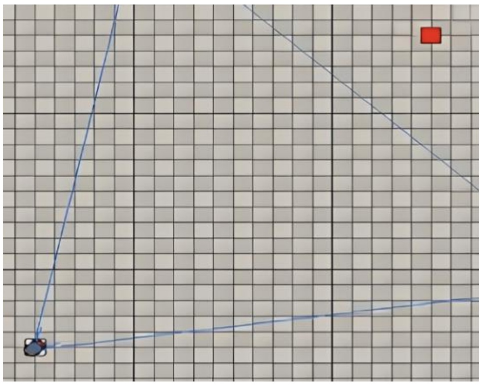

Fig. 2. Virtual environment without obstacles.

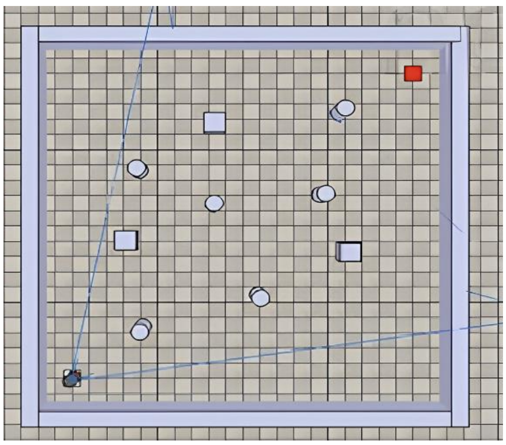

Fig. 3. Virtual environment with obstacles.

The navigation environments of our work are shown in below figures (2) and (3). The first contains a mobile differential drive robot (Pioneer_p3dx) and a target. The second one contains a mobile robot, static obstacles, target and walls.

4. MATLAB V-REP CONNEXION

V-REP has a remote API that allows you to control a simulation or the simulator itself remotely (for example from a real robot or another PC). Control programs can be written in C/C++, Python, Java, Lua, MATLAB, or Octave (Coppelia Robotics, 2014).

There are two possible communication modes between V-Rep and MATLAB: synchronous mode and asynchronous mode, in our case the connection in synchronous mode was chosen.

In our system, MATLAB and V-REP are used a remote API to realize virtual navigation of a mobile robot in a cluttered environment, MATLAB is used as a client and V-REP is used as a server, where MATLAB sends commands to V-REP and receives data from V-REP.

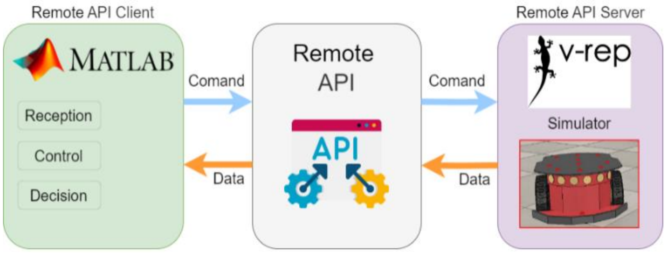

Fig. 4. Diagram of the connection between MATLAB and V-REP.

To ensure the connection between MATLAB and V-REP, the remote API must be in the same directory that we created to save all of our MATLAB codes (PSO function, ANFIS function, the main program...etc), the connection between MATALB and V-REP is ensured by command lines added to the main program as shown in the flowchart below:

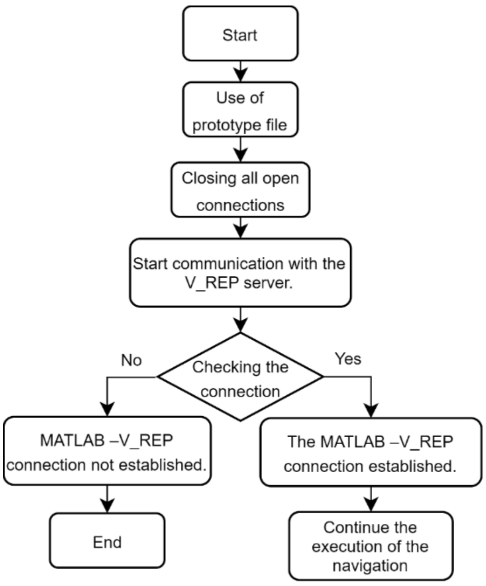

Fig. 5. MATLAB and V-REP connection flowchart.

5. NEURO FUZZY INFERENCE SYSTEM CONTROLLER (ANFIS)

The Adaptive Neuro-Fuzzy Inference System (ANFIS) belongs to hybrid neuro-fuzzy systems that combine the advantages of fuzzy logic with that of neural networks in a single network. It was proposed by Jang in 1993 (Batti et al., 2020); this technique brings the learning capabilities of neural networks to the Takagi-Sugeno type fuzzy inference system. The role of back-propagation learning based on the gradient descent method is the adjustment and optimization of the parameters of this fuzzy inference system (premise part and conclusion part of the rules).

The ANFIS system has five layers, where the adaptive nodes containing parameters are in the first and fourth layer, while the other nodes without parameters are fixed.

The structure of the ANFIS neuro-fuzzy network is composed of two parts: A premise part and a consequence part, connected to each other by a base of fuzzy rules in the form of a network. This structure consists in adapting these fuzzy rules to changing conditions. Therefore, ANFIS systems can automatically optimize and adjust membership functions using a learning algorithm (Auday et al., 2014; Pothal et al., 2015; Khati et al., 2019).

In order to minimize tracking errors, kinematic control by two ANFIS controllers was used. This yields the command vector composed of the linear velocity v and the angular velocity w. In our work, the inputs of the first controller are the position error e_x corresponds to x_1 and its derivative $\dot{e}_x$ corresponds to x_2, to calculate the command v corresponds to output u_1 as shown in the architecture below Figure 7 and the inputs of the second controller are the position error e_y corresponds to x_1 and its derivative $\dot{e}_y$ corresponds to x_2 to calculate the command w corresponds to output u_2 as shown in the architecture below Figure 7, the choice of these inputs and outputs is made because the kinematic command makes allows to calculate the command vector $[v, w]$ such that v depends on e_x and w depends on e_y as demonstrated in (Kanayama et al., 1990; Dhaouadi et al.,2013).

In our work the robot is controlled either by ANFIS tracking or ANFIS avoidance, the selection of one of the controllers is made according to the distance obtained between the robot's ultrasonic sensors and the obstacles, if the distance is less than 20 cm the robot is controlled by ANFIS avoidance if not by ANFIS tracking as shown in flowchart in Figure 11.

The ANFIS tracking regulator is composed of two regulators as shown in Figure (6). It allows calculating the commands u_1 and u_2, which are the linear velocity and the angular velocity which will be sent to the robot in order to seek the targets. Each ANFIS regulator receives two inputs x_1, x_2 which are the position error and the velocity error and gives an output u as shown in the ANFIS network structure of Figure (7). Input x_1 is associated with three fuzzy sets A_1, A_2, A_3, and input x_2 is associated with three fuzzy sets B_1, B_2, B_3. The output u is modeled by a fuzzy system of the Sugeno type, composed of the following nine rules:

Rule 1: $if\ x_1\ is\ A_1\ and\ x_2\ is\ B_1\ then$
$$u_1 = f_1(x_1, x_2) = a_1x_1 + b_1x_2 + c_1 \quad (7)$$
Rule 2: $if\ x_1\ is\ A_1\ and\ x_2\ is\ B_2\ then$
$$u_2 = f_2(x_1, x_2) = a_2x_1 + b_2x_2 + c_2 \quad (8)$$
Rule 3: $if\ x_1\ is\ A_1\ and\ x_2\ is\ B_3\ then$
$$u_3 = f_3(x_1, x_2) = a_3x_1 + b_3x_2 + c_3 \quad (9)$$
Rule 4: $if\ x_1\ is\ A_2\ and\ x_2\ is\ B_1\ then$
$$u_4 = f_4(x_1, x_2) = a_4x_1 + b_4x_2 + c_4 \quad (10)$$
Rule 5: $if\ x_1\ is\ A_2\ and\ x_2\ is\ B_2\ then$
$$u_5 = f_5(x_1, x_2) = a_5x_1 + b_5x_2 + c_5 \quad (11)$$
Rule 6: $if\ x_1\ is\ A_2\ and\ x_2\ is\ B_3\ then$
$$u_6 = f_6(x_1, x_2) = a_6x_1 + b_6x_2 + c_6 \quad (12)$$
Rule 7: $if\ x_1\ is\ A_3\ and\ x_2\ is\ B_1\ then$
$$u_7 = f_7(x_1, x_2) = a_7x_1 + b_7x_2 + c_7 \quad (13)$$
Rule 8: $if\ x_1\ is\ A_3\ and\ x_2\ is\ B_2\ then$
$$u_8 = f_8(x_1, x_2) = a_8x_1 + b_8x_2 + c_8 \quad (14)$$
Rule 9: $if\ x_1\ is\ A_3\ and\ x_2\ is\ B_3\ then$
$$u_9 = f_9(x_1, x_2) = a_9x_1 + b_9x_2 + c_9 \quad (15)$$

Such as $a_i, b_i, c_i, for\ i = 1{:}9$ are the linear consequence parameters of the fuzzy inference system.

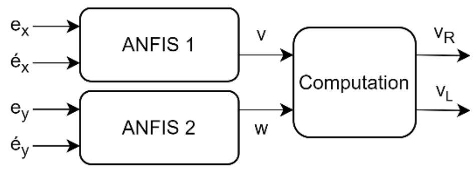

Fig. 6 ANFIS tracking.

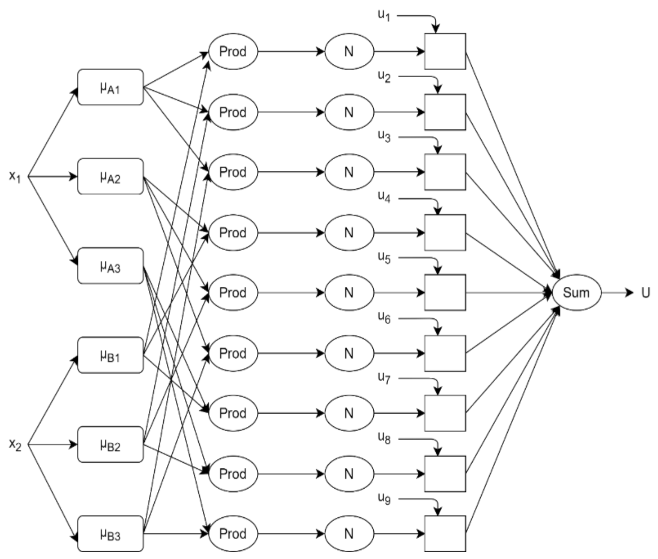

Fig. 7. structure of ANFIS tracking network.

The layers are described as follows:

Layer 1: Fuzzification

This layer contains six nodes that transform the digital inputs measured by the sensors into linguistic interpretations. Each node i of this layer is an adaptive node with a membership function with adjustable parameters:

$$O_{1,i} = \mu_{A_i}(x_1) \text{for i} = 1{:}3$$

$$O_{1,i} = \mu_{B_{i-3}}(x_2) \text{for i} = 4{:}6$$

$O_{1,i}$ Calculates its activations which are the membership degrees of the variables $x_1$and x_2 respectively to the fuzzy sets A_i and B_i represented by Gaussian functions described by:

$$\mu_{A_i}(x_1) = \exp\left(-\left(\frac{x_1 - c_i}{\sigma_i}\right)^2\right)\ for\ i = 1{:}3 \quad (16)$$

$$\mu_{B_{i-3}}(x_2) = \exp\left(-\left(\frac{x_2 - c_i}{\sigma_i}\right)^2\right)\ for\ i = 4{:}6 \quad (17)$$

Where c_i is the center of the Gaussian and σ_i the standard deviation.

Layer 2: Degree of activation

Generates the appropriate degree of activation for the rule.

$$O_{2,i} = w_i = \mu_{A_j}(x_1).\mu_{B_k}(x_2) \text{ for } i = 1{:}9\ and\ j, k = 1{:}3 \quad (18)$$

Layer 3: Normalization

Each node of this layer is a fixed node called N. Its output represents the normalized degree of activation of the ith rule.

$$O_{3.i} = \overline{w_i} = \frac{w_i}{\sum_{k=1}^{9} w_k}\ for\ i = 1{:}9 \quad (19)$$

Layer 4: Calculation of rule outputs

Each node of this layer is an adaptive node whose function is:

$$O_{4,i} = \overline{w_i} f_i = \overline{w_i}(a_ix_1 + b_ix_2 + c_i) \quad (20)$$

Where $\overline{w_i}$ is the Layer 3 output that represents the normalized activation degree of the rule and (a_i, b_i, c_i) are the adjustable output parameters of rule i. These parameters are called consequence parameters.

Layer 5: Defuzzification

This layer includes a single fixed node, which calculates the global output that is the sum of all the signals coming from layer 4: $O_{5,i} = u = \sum_{i=1}^{9} \overline{w_i} f_i = \frac{\sum_{i=1}^{9} w_i f_i}{\sum_{i=1}^{9} w_i}$ (21)

In our case, we consider that the parameters of the premise are fixed, while those of the consequence are adjusted using a learning algorithm based on the gradient descent method by minimizing the objective function $J = \frac{1}{2}e^2$, where e is the error between the current position and the desired position to find the consequence parameters.

The ANFIS avoidance regulator as shown in Figure (8) makes it possible to calculate the commands $u_1$and u_2 the linear velocity and the angular velocity respectively, which will be sent to the robot in order to avoid obstacles without collision, the ANFIS network that corresponds to this regulator shown in the structure of Figure (9). It receives three inputs x_1, x_2 and x_3 representing the distances between the obstacle and the robot which are measured by ultrasonic sensors (Front Distance (FD), Left distance (LD), and Right Distance (RD)) and gives two outputs $u_1$and u_2.

Input x_1 is associated with three fuzzy sets A_1, A_2, A_3 and input x_2 is associated with three fuzzy setsB_1, B_2, B_3, input x_3 is associated with three fuzzy setsC_1, C_2, C_3. Outputs $u_1 and\ u_2$ is modeled by a fuzzy system of the Sugeno type, composed of twenty-seven rules, and the following steps are carried out in the same way as the previous controller.

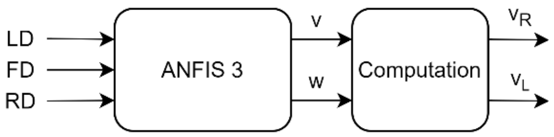

Fig. 8. ANFIS Obstacles Avoidance.

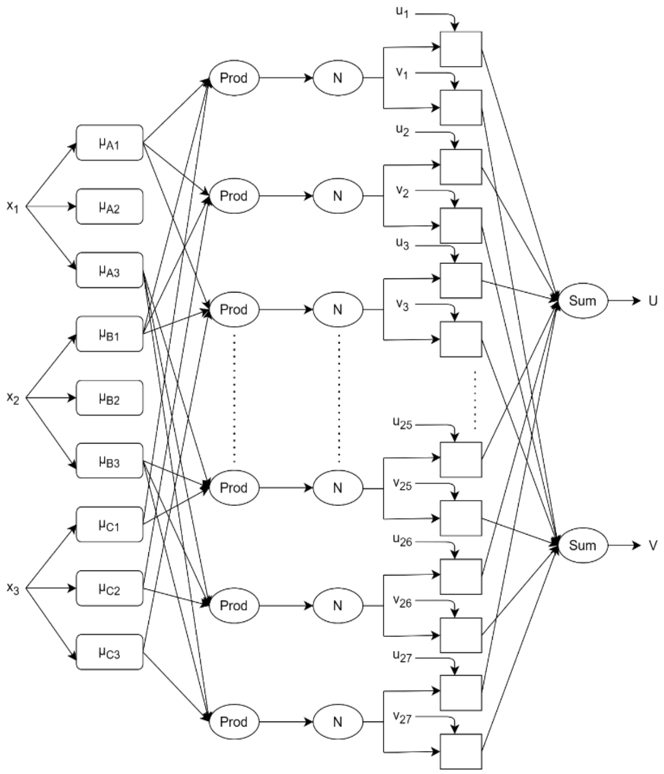

Fig. 9. Structure of ANFIS Obstacles Avoidance network.

6. PSO (PARTICLE SWARM OPTIMIZATION)

The PSO algorithm was developed imitating the behavior of birds searching for food in an unknown location, first developed in 1995 by electrical engineer Eberhard and social psychologist Kenn Q1A (Nursena et al., 2018; Allou et al., 2018), The PSO algorithm is a non-deterministic population-based optimization method (Anish et al., 2020).

This algorithm is a metaheuristic optimization method applied in our case to minimize the root mean square error (RMSE) between the current value and the predicted value of the outputs of the ANFIS controller, which was chosen in our work as the objective function (fitness) as following:

$$f_{obj} = RMSE(\%) = \left[\sum_1^k \left(\frac{v_R-\hat{v}_R}{k}\right)^2 + \left(\frac{v_L-\hat{v}_L}{k}\right)^2\right] * 100 \qquad (22)$$

In order to improve the movements of our mobile robot after several simulations with the ANFIS controller in different navigation environments proposed in our work, we noticed that the wheels velocities v_R and v_L sent to the pioneer-P3DX exceed the limit velocities of these wheels. To solve this problem we introduced a meta-heuristic PSO optimization algorithm to optimize the outputs of the ANFIS controller such that the wheels velocities and the robot velocity are limited respectively as follows $v_{min} \le v_R \le v_{max}$ *et* $v_{min} \le v_L \le v_{max}$ and $v_{min-robot} \le v_{robot} \le v_{max-robot}$ before sending them to the pionner-P3DX robot, from these conditions we brought out the following constraints (of type ≤0) used by PSO algorithm to minimize the objective function of equation (22):

$$v_{min} \le v_R \le v_{max}$$
$$\Rightarrow \begin{cases} v_R - v_{max} \le 0 & (23) \\ -v_R + v_{min} \le 0 & (24) \end{cases}$$

$$v_{min} \le v_L \le v_{max}$$
$$\Rightarrow \begin{cases} v_L - v_{max} \le 0 & (25) \\ -v_L + v_{min} \le 0 & (26) \end{cases}$$

$$v_{min-robot} \le \frac{v_R + v_L}{2} \le v_{max-robot}$$
$$\Rightarrow \begin{cases} v_R + v_L - 2v_{max-robot} \le 0 & (27) \\ -v_R - v_L + 2v_{min-robot} \le 0 & (28) \end{cases}$$

$$w_{min-robot} \le \frac{v_R - v_L}{2L} \le w_{max-robot}$$
$$\Rightarrow \begin{cases} v_R - v_L - 2Lw_{max-robot} \le 0 & (29) \\ -v_R + v_L + 2Lw_{min-robot} \le 0 & (30) \end{cases}$$

As v_R and v_L are variables to be optimized that represent the particles, the set of these particles presents a population. The first four constraints are extracted from the real wheels velocities limits, while the following four constraints are extracted from Equation (3) which represents the kinematics of the robot in order to introduce its real behavior. The PSO algorithm is as follows:

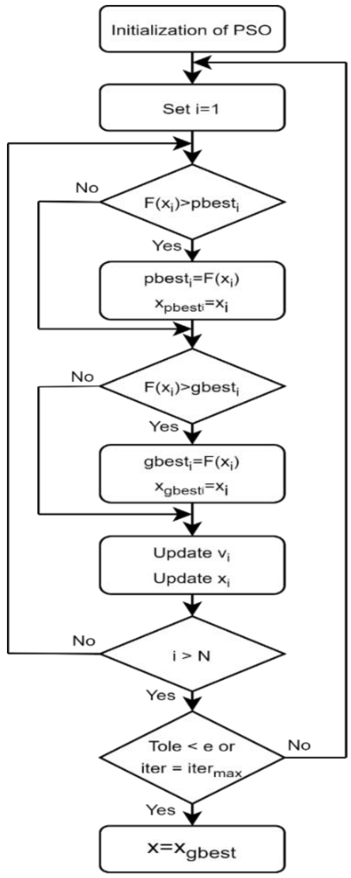

Fig. 10. Particle swarm optimization algorithm flowchart.

$$v_i(t+1) = \omega\, v_i(t) + c_1 r_1 [pbest_i(t) - x_i(t)] + c_2 r_2 [gbest_i(t) - x_i(t)] \quad (31)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (32)$$

v_i (t) the velocity of particle i at time t, x_i (t) the position of particle i at time t; ω, c_1 and c_2 ($0 \leq \omega \leq 1.2$, $0 \leq c_1 \leq 2$, et $0 \leq c_2 \leq 2$) are constants coefficients set by the user and r_1 et r_2 are random numbers drawn at each iteration ; $gbest_i(t)$ is the best solution found up to time t and $pbest_i(t)$ is the best solution found by particle i.

In order to ensure a very fast response time of the PSO algorithm and after several simulations, we set a maximum number of iterations at 10 then select the best result, this maximum number of iterations is chosen to be sufficient for the algorithm to converge to the optimal results in a very short time. The convergence time of the PSO algorithm in our simulation is 0.09 s

7. THE VIRTUAL NAVIGATION ALGORITHM PROPOSED

The proposed navigation method that uses hybrid ANFIS-PSO control is structured as follows:

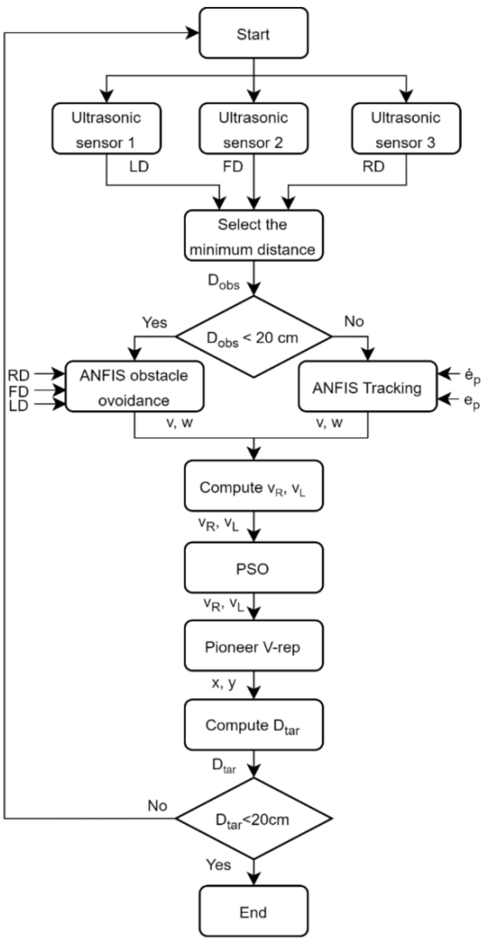

Fig. 11. Virtual navigation flowchart using the ANFIS-PSO method.

8. RESULTS AND DISCUSSION

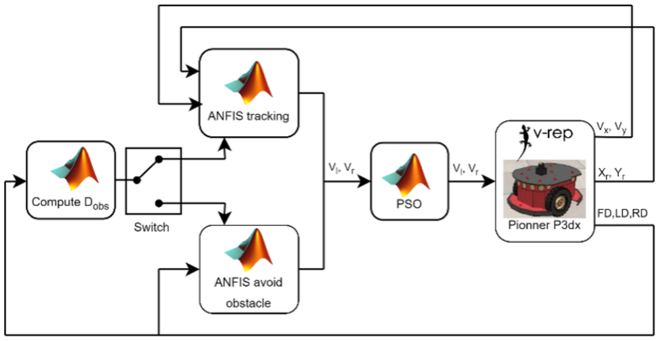

Fig. 12. Global virtual simulation diagram.

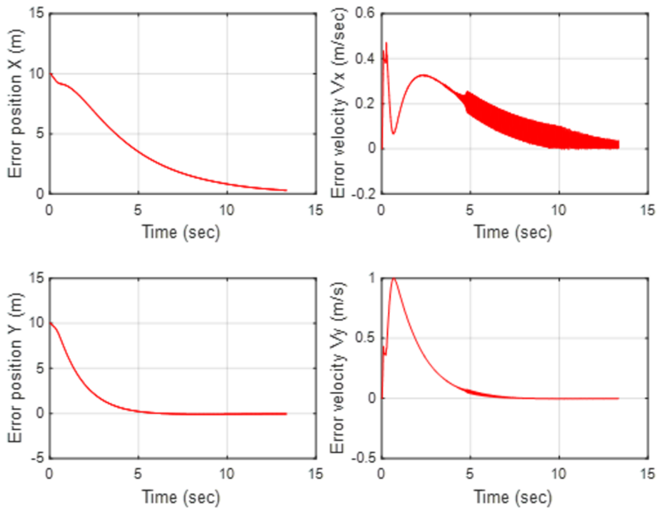

Fig. 13. Behaviors errors tracking and errors velocity using ANFIS without obstacles.

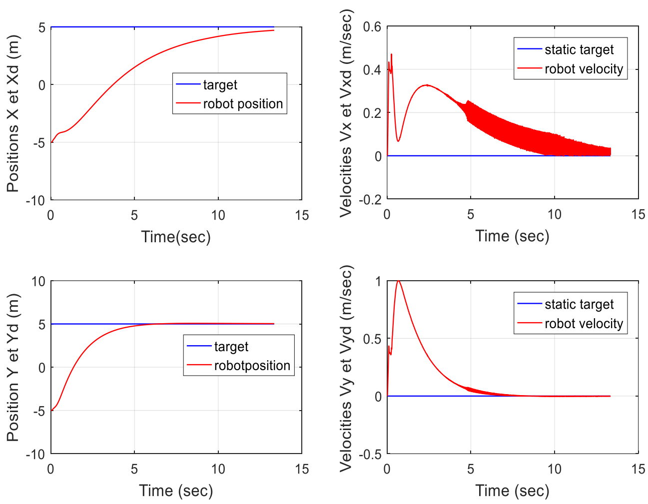

Fig. 14. Behaviors tracking and velocity using ANFIS without obstacles.

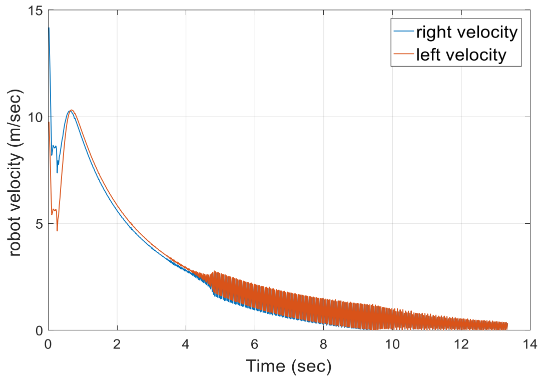

Fig. 15. Behaviors robot velocities using ANFIS without obstacles.

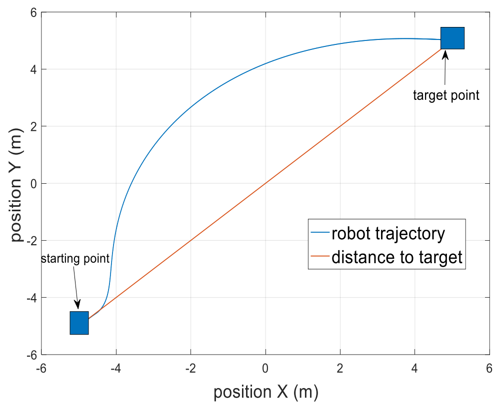

Fig.16. Path traveled by the robot using ANFIS without obstacles.

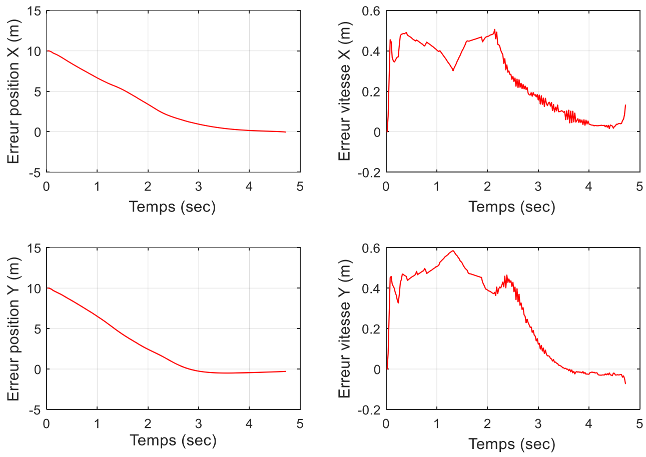

Fig. 17. Behaviors errors tracking and errors velocity using ANFIS-PSO without obstacles.

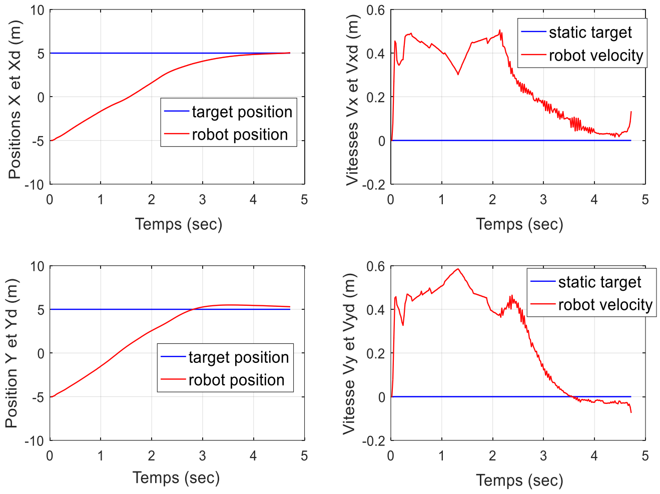

Fig. 18. Behaviors tracking and velocity using ANFIS-PSO without obstacles.

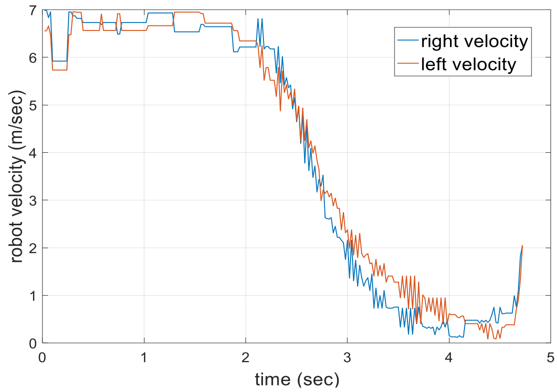

Fig. 19. Behaviors robot velocities using ANFIS-PSO without obstacles.

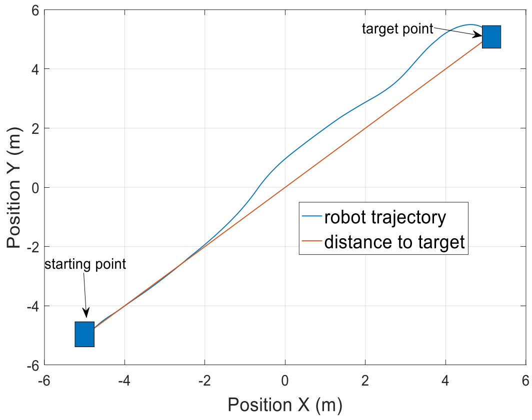

Fig. 20. Path traveled by the robot using ANFIS-PSO without obstacles.

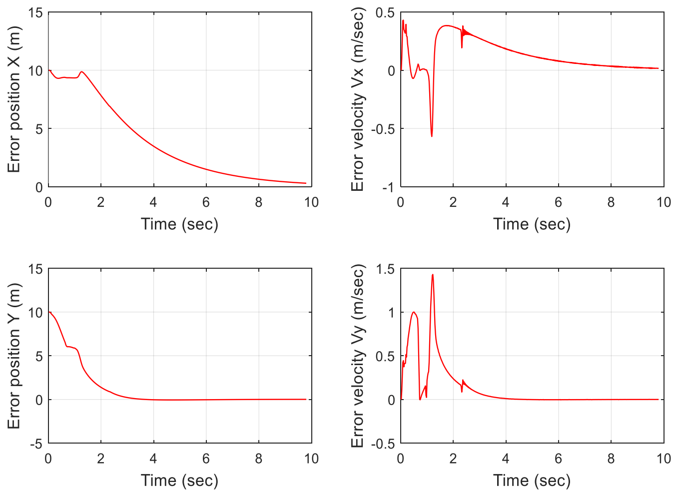

Fig. 21. Behaviors errors tracking and errors velocity using ANFIS with obstacles.

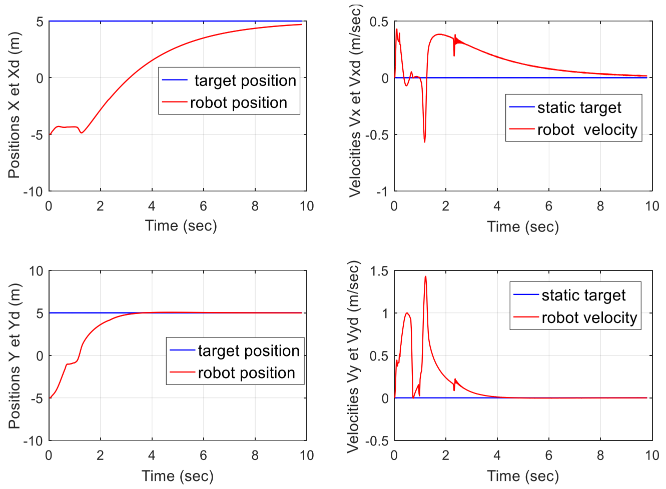

Fig. 22. Behaviors tracking and velocity using ANFIS with obstacles.

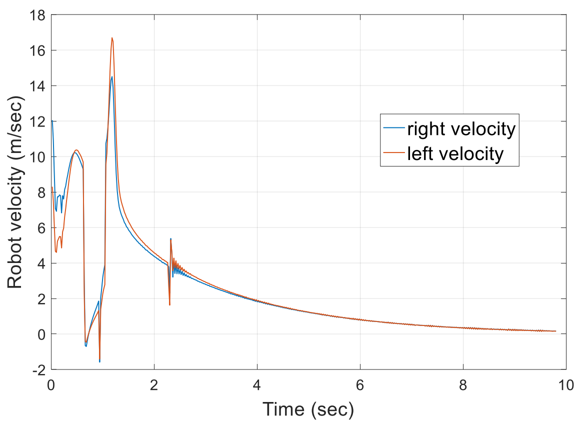

Fig. 23. Behaviors robot velocities using ANFIS with obstacles.

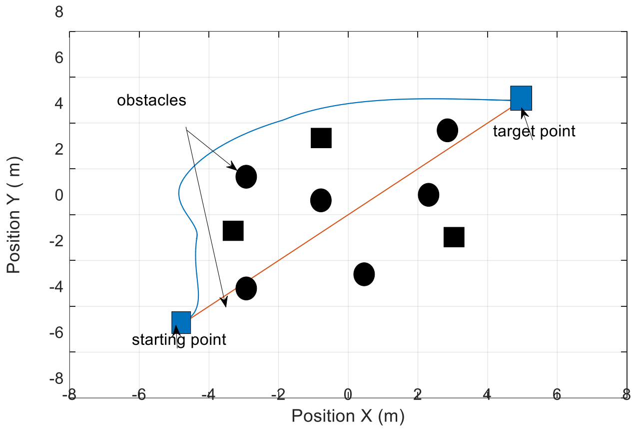

Fig. 24. Path traveled by the robot using ANFIS with obstacles.

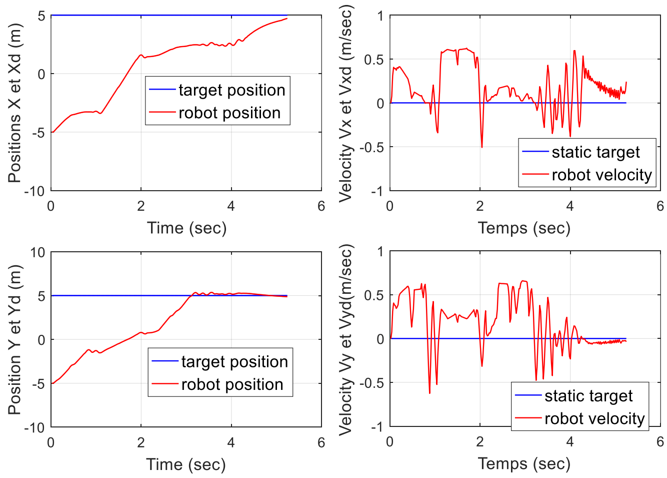

Fig. 25. Behaviors errors tracking and errors velocity using ANFIS-POS with obstacles.

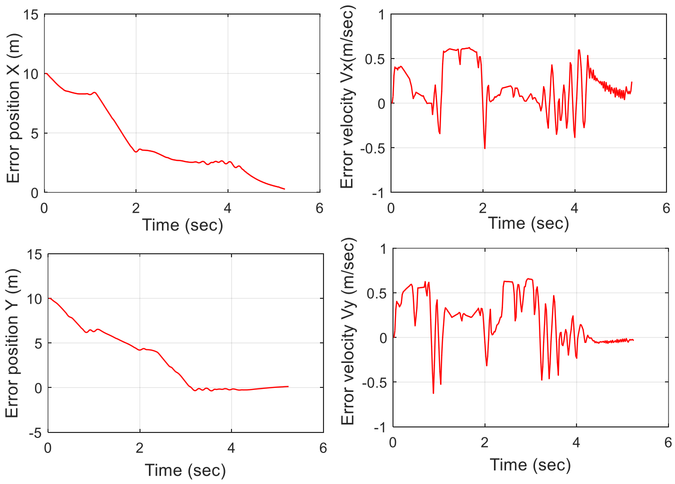

Fig. 26. Behaviors tracking and velocity using ANFIS-PSO with obstacles.

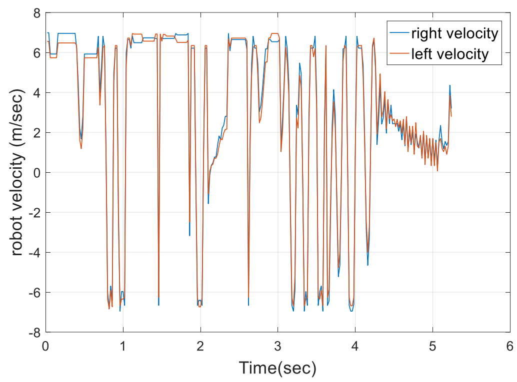

Fig. 27. Behaviors robot velocities using ANFIS-PSO with obstacles.

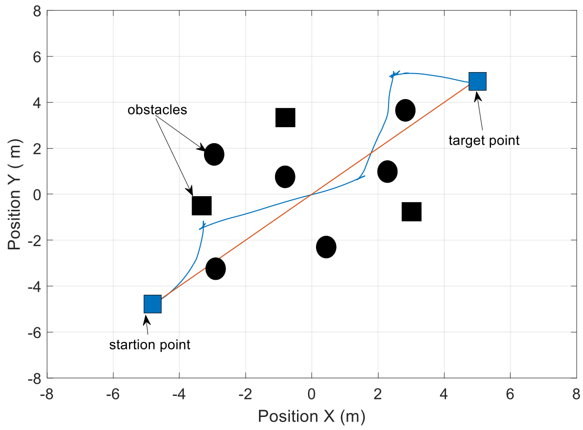

Fig. 28. Path traveled by the robot using ANFIS-PSO with obstacles.

The control of the P3-DX differential drive mobile robot has been carried out by two methods. In the first we used two ANFIS controllers and in the second an ANFIS-PSO hybrid controller, written in MATLAB code in order to calculate the commands that will be sent to the robot on the designed V-REP virtual platform which contains the target, obstacles and a robot. We have designed on V-REP two virtual simulation environments in order to test our algorithms by taking into account the real physical constraints of the robot. The first case is to navigate the robot from a starting point to a target without obstacles, while the second case is to navigate from a starting point to a target avoiding several obstacles. The parameters of the robot, regulators are declared in the MATLAB program.

Our simulation results by the two methods ANFIS and ANFIS-PSO are presented in the before figures, allowing us to develop a comparative study and to show the effectiveness of each method.

The simulation carried out in the first environment without obstacles Figure (2) gives the results represented in Figures (13) (14) (15) (16) by the ANFIS method and Figures (17) (18) (19) (20) by the ANFIS-PSO method, it was found that the two methods ensure that the robot reaches the target. Nevertheless, the robot is faster by the ANFIS–PSO method where it crosses a short distance as shown by Figures (16) (20), and in a short time as shown by Figures (14) (18). The response time in Figures (13) (17) which presents the tracking error is better by the ANFIS-PSO method; Figures (15) (19) represent the velocities of the left and right wheels, clearly show the effectiveness of introducing the PSO algorithm which optimizes the velocities of the wheels while respecting all the physical constraints of the robot mentioned in Equations (30:37); according to this analysis we observe that the ANFIS-PSO algorithm gives good results compared to the ANFIS algorithm .

For the simulation carried out in the second environment with obstacles, Figure (3) gives the results represented in Figures (21) (22) (23) (24) by the ANFIS method and Figures (25) (26) (27) (28) by the ANFIS-PSO method, in this step it is found that the two methods ensure that the robot tracks the target while avoiding obstacles, which shows the robustness of our algorithms. However, the robot is faster by the ANFIS-PSO method such that it reacts to obstacle avoidance in an optimal way by following a shortest path as shown in Figures (24) (28) in a short time as shown in Figures (22) (26). However, the response time in the Figures (21) (22) which presents the tracking error is better by the ANFIS-PSO method; Figures (23) (27) represent the velocities of the left and right wheels, clearly show the effectiveness of introducing the PSO algorithm which optimizes the velocities of the wheels while respecting all the physical constraints of the robot mentioned in Equations (30:37) as well as the optimization of obstacle avoidance by choosing a shortest path which means that the algorithm ANFIS-PSO is more robust; according to this analysis, it can be seen that the ANFIS-PSO algorithm gives good results in terms of robustness, response time, distance traveled, efficiency compared to the ANFIS algorithm.

9. CONCLUSION

In this article we presented a virtual navigation of a mobile robot using the virtual environment design software V-REP. We applied intelligent controls based on neuro-fuzzy regulators written in MATLAB code.

First, we designed the virtual environments on the V-REP platform in which we will perform our virtual navigation of the P-3DX mobile robot, then we developed a connection in synchronous mode between MATLAB and V-REP to ensure the data recovery from V-REP and information processing on MATLAB to calculate the commands that will send the commands to the robot in real time.

The control tests are carried out using two methods, the first with two ANFIS controllers and the second with a hybrid ANFIS-PSO controller. The results of our work show the effectiveness of the two methods: they realize a good target tracking while avoiding static obstacles; and we observe also that the ANFIS-PSO hybrid controller gives better results compared to the ANFIS method.

As additional work, we suggest doing other navigation in more complex environments, for example n presence of dynamic obstacles, in maze environment, and trajectory planning.

REFERENCES

Allou S, Zennir Y: A Comparative Study of PID-PSO and Fuzzy Controller for Path Tracking Control of Autonomous Ground Vehicles. In Proceedings of the

15th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2018) - Volume 1, pages 296-304.

Anish P,Vikas SP, Ehtesham HMd, Dayal RP.V-REP-based navigation of automated wheeled robot between obstacles using PSO-tuned feedforward neural network. *Journal of Computational Design and Engineering*, 2020, 7(0), 1–8.

Auday AM, William W, Phil B,.Adaptive Neuro-Fuzzy Technique for Autonomous Ground Vehicle Navigation. *Robotics* 2014, 3, pp. 349-370.

Batti H, Ben Jabeur C, Seddik H: Autonomous smart robot for path predicting and finding in maze based on fuzzy and neuro-Fuzzy approaches. *Asian J Control*. 2020, 1–10.

Borenstein J. Experimental results from internal odometry error correction with the OmniMate mobile robot. *IEEE Trans Rob Autom*. 1998;14(6): pp. 963–969.

Coppelia Robotics GmbH, "V-REP simulator," 2016. [Online]. Available: *http://www.coppeliarobotics.com*

Dhaouadi R, Hatab AA: Dynamic modelling of differentiai-drive mobile robots using lagrange and newton-euler methodologies: A unified framework, Advances in *Robotics & Automation*, vol. 2, pp. 1-7, 2013.

Ernesto F, Gonzalo F, Emmanuel P, Héctor V, Sebastian D.: Teaching control in mobile robotics with V-REP and a Khepera IV library. 2016 *IEEE Conference on Control Applications* (CCA) Part of 2016 *IEEE Multi-Conference on Systems and Control* September 19-22, 2016. Buenos Aires, Argentina.

Faris G,Fabregas E,Peralta E,Torres E,Dormido S.A Khepera IV library for robotic control education using V-REP. *IFAC PapersOnLine* 50-1 (2017) 9150–9155.

Gharajeh MS and Jond HB.An intelligent approach for autonomous mobile robots path planning based on adaptive neuro-fuzzy inference system. *Ain Shams Engineering Journal* 13 (2022) 101491.

Gharajeh MS and Jond HB. : Hybrid Global Positioning System-Adaptive Neuro-Fuzzy Inference System based autonomous mobile robot navigation. *Robotics and Autonomous Systems* 134 (2020) 103669.

Guyot L, Heiniger N, Michel O, and Rohrer F, "Teaching robotics with an open curriculum based on the e-puck robot, simulations and competitions," *in Proceedings of the 2nd International Conference on Robotics in Education*. Vienna, Austria, 2011.

Kanayama Y, Kimura Y, Miyazaki F, and Noguchi T, "A stable tracking control method for an autonomous mobile robot," in Robotics and Automation, 1990. *Proceedings. 1990 IEEE International Conference on*, 1990, pp. 384-389.

Khati H, Talem H ,Mellah R, Bilek A. "Neuro-fuzzy control of bilateral teleoperation system using FPGA." *Iranian Journal of Fuzzy Systems* vol. 16, N°. 6, pp. 17-32, 2019.

Lazreg M, Benamrane N: Intelligent System for Robotic Navigation Using ANFIS and ACOr. *Applied Artificial Intelligence Journal*.2019, 1-20.

Mubashir UI, Ameer TK, Shuai L.Bio-inspired BAS: Run-time Path-planning And The Control of Differential Mobile Robot. *EAI Endorsed Transactions on AI and Robotics Journal*.2020, 1-10.

Nursena B, Mehmet B, Mehmet K. PSO Based Path Planning Approach for Multi Service Robots in Dynamic Environments. 2018 *International Conference on Artificial Intelligence and Data Processing* (IDAP).

Pinciroli C, Trianni V, and O'Grady R, "ARGoS: A modular, ultiengine simulator for heterogeneous swarm robotics," in *International Conference on Intelligent Robots and Systems* (IROS), Sept 2011, pp. 5027–5034.

Pothal JK, Dayal RP. : Navigation of multiple mobile robots in a highly clutter terrains using adaptive neuro-fuzzy inference system. *Robotics and Autonomous Systems*. 2015, 1:28.

Saidi Y, Nemra A & Tadjine M: Robust mobile robot navigation using fuzzy type 2 with wheel slip dynamic modeling and parameters uncertainties. *International Journal of Modelling and Simulation*. 2019, 1-24.

Saiful Azimi M, Shukri ZA and Zaharuddin M.: Virtual Reality based Mobile Robot Navigation in Greenhouse Environment. *Advances in Animal Biosciences: Precision Agriculture* (ECPA) 2017, (2017), 8:2, pp 854–859.

Sharma K Sh, Manivannan PV: Stereo Image Partitioning Based Fuzzy Logic Controller for Real-time Obstacle Detection and Avoidance. *International Journal of Mechanical Engineering and Robotics Research* Vol. 9, No. 8, August 2020.

Shikha S, Abhishek KK, Anish P, Vikas SP, and Nitin S: Sunflower Optimization Algorithm Based Steering Angle Controlled Motion Planning of Two-Wheeled Pioneer P3-DX Robot in V-REP Scenario. *AIP Conference Proceedings* 2341, 020003 (2021).