Optimal ANFIS-FOPID with Back-Stepping Controller Design for Wheeled Mobile Robot Control

Rafik Euldji*, Noureddine Batel*, Redha Rebhi, Mohamed Redha Skender*****

** Laboratory of Advanced Electronic Systems (LSEA), Department of Electrical Engineering, Faculty of Technology, University of Yahia Fares, Medea 26000 Algeria; (e-mail: euldji.rafik@univ-medea.dz, batel.noureddine@univ-medea.dz).*
*** Department of Mechanical Engineering, Faculty of Technology, University of Yahia Fares, Medea 26000, Algeria, (e-mail: rebhi.ridha@univ-medea.dz).*
**** Research Laboratory in Electrical Engineering and Automatic (LREA), Department of Electrical Engineering, Faculty of Technology, University of Yahia Fares, Medea 26000 Algeria, (e-mail: skender.mohamed@univ-medea.dz).*

Abstract: This paper suggests a design of an optimal fractional adaptive Neuro- Fuzzy inference system (ANFIS-FOPID) controller for dealing with the dynamics of the wheeled mobile robot (WMR).Tuning parameters of the controller is a challenging task , a hybrid meta-heuristic optimization algorithm has been introduced. This evolutionary technique is known as the hybrid whale grey wolf optimizer (HWGO). Simulation results in MATLAB–Simulink environment revealed the highest efficiency of the proposed ANFIS-FOPID technique compared to the fractional proportional integral derivative (FOPID) and the adaptive Neuro- Fuzzy inference system (ANFIS), in terms of settling and rise time, overshoot, as well as steady-state error. A five shape path and zigzag path have highlighted the over performance of mentioned controller. Finally, to accomplish our work a back-stepping control technique is added for the kinematic control, a quadrifolium path was made to illustrate the capability of the mentioned controller.

Keywords: wheeled mobile robot (WMR), trajectory tracking, fractional order proportional integral derivative controller (FOPID), Adaptive neuro-fuzzy inference (ANFIS), hybrid whale grey wolf optimizer (HWGO).

1. INTRODUCTION

Nowadays the research interests related to the robotics field have growing up rapidly. Especially in designing, and controlling wheeled mobile robots (WMRs), since this lately cover a various civil and military applications such as space exploration, manufacturing, services, medical, agriculture, rescue and combat (Baskoro et al., 2020; Catalan et al., 2021; Emmi et al., 2014; Fue et al., 2020; Ikeda et al., 2018; Ravankar et al., 2021).

Currently, the tracking control problem still challenging and attracts many researchers. Each application has a specific constraint related to the internal dynamics of the robot (non-holonomic properties of WMRs). Moreover, WMRs are influenced by working conditions, such as external load disturbance, wheel slips, feedback sensors, which lead to undefined uncertainties (Cen & Singh, 2021). For that reason, developing more precise control strategies is required.

Many modern control methods have been proposed in the last decade. a sliding mode dynamic controller proposed in (Moudoud et al., 2020; Nikranjbar et al., 2018; Phuc et al., 2021; Wang, Zhou, et al., 2019; Wu et al., 2019), an online physical parameters adaptation of the robot dynamics proposed in (Martins et al., 2008), an adaptive back-stepping controller in (Binh et al., 2019), using neural network controller for dealing with unmodeled dynamics in (Bozek et al., 2020; Khnissi et al., 2020; Rossomando et al., 2011; Wang, Liu, et al., 2019), using fuzzy logic controller in (Das and Kar, 2006; Štefek et al., 2021).

The strategy of PID control has been one of the most practical and most often used technique in industry because of its simplicity form and strong robustness in large working condition. However, the fixed parameters generally degrade the control performance (Bingul and Karahan, 2018; Kumar et al., 2017; Slama et al., 2019). In order to overcome this issue, Several researches papers concentrate on designing new types of modified PID, like fusing the neural network with a conventional PID controller (NN-PID) (Pei et al., 2021; Xu et al., 2019; Ye, 2008). The approach takes the simplicity of a PID controller and the powerful capability of learning, adaptability and tackling nonlinearity of neural networks. Many researches papers use an adaptive tuning methods by combining the fuzzy logic controller with PID (Ben Jabeur and Seddik, 2021; Mai et al., 2021; Tiep et al., 2018; Zhou et al., 2019), However, the demand for new control methods becomes more intense according to applications complexity and progress.

A fractional-order proportional integral derivative (FOPID) controller has already taken over the place of the traditional PID. This controller gained an important popularity from several researchers in many areas, including robotics engineering (Bernardes et al., 2019; Erkol, 2018; Kankhunthodl et al., 2019; Mishra and Chandra, 2014; Orman et al., 2016; Zhang et al., 2020). The major reason behind this attention is that the five parameters (Kp, Ki, Kd, λ, μ) of the FOPID controller can be more adapted for designing better controllers, as being faster with less overshoot compared with the integer- order PID controller, which has only three parameters to be adjust (Bernardes et al., 2019; Erkol, 2018; Kankhunthodl et al., 2019; Mishra and Chandra, 2014; Orman et al., 2016; Zhang et al., 2020). Naturally, a controller with more parameters to be tuned means more its design becomes difficult. This makes it optimization a challenging task.

According to various studies, nature inspired algorithms have already proved their capability in scaling the FOPID controller factors. We distingue, particle swarm optimizer (PSO) (Abdelhamid et al., 2013; Rajesh, 2019), whale optimizer (WOA) (Abood and Oleiwi, 2021), grey wolf optimizer (GWO) (Verma et al., 2017), using sine-cosine algorithm (SCA) in (Bhookya and Jatoth, 2019).

The major benefit from the FOPID controllers is the rapidity in terms of rise time and settling time. On the other hand, some specific trajectory like the sharp chap one makes the robot instantaneously changing its orientation (a case of square trajectory), which leads to a large overshoot. This problem could harm the robot's actuators, due to the robot's high coupled nonlinearity, which means the FOPID controller with fixed parameter cannot handle some special trajectory, and a new controller that can deal with all types of path is required (Arpaci and Özgüven, 2011; Saxena et al., 2021).

The concept of combining fuzzy logic controller with a neural network has attracted a remarkable attention in various fields especially in control engineering (Al-Mayyahi et al., 2014; Bendary et al., 2021; El-Hasnony et al., 2020; Elsisi et al., 2021; Imen et al., 2011; Pang et al., 2021; Premkumar and Manikandan, 2014; Selma et al., 2020).The reason behind this popularity is that the adaptive neuro-fuzzy inference system (ANFIS) mixes the efficiency of fuzzy logic to deal with the uncertainties and the ability of neural networks to learn from plants/processes.

The aim of this work is to design an optimal hybrid adaptive fractional neuro-fuzzy inference controller, abbreviated as ANFIS-FOPID controller, this robust controller merges the advantages of the FOPID controller with the ANFIS controller and benefiting from the powerful hybrid whale grey wolf optimizer (HWGO), for adjusting the parameters of the proposed controller, based on the cost function namely integral square error (ISE).

This paper is organized as follows: The complete model of the wheeled mobile robot is presented in Section 2. The controller design and strategy are described in Section 3, as well as the simulation findings and discussions are summarized in Sections 4 and 5, respectively.

2. ROBOT'S MATHEMATICAL MODEL

The WMR in Fig. 1 contains two pairs of DC motors with driving wheels of radius R, where φL and φ R are the left and right rotating angles of wheels, respectively. θ is the robot orientation and L is the width of the robot body. At the distance d from the mind-point A, where (Xa,Ya) is the coordinate of A in the inertial frame (X,Y), and the coordinates of any point in the robot frame are defined by (Xr, Yr).

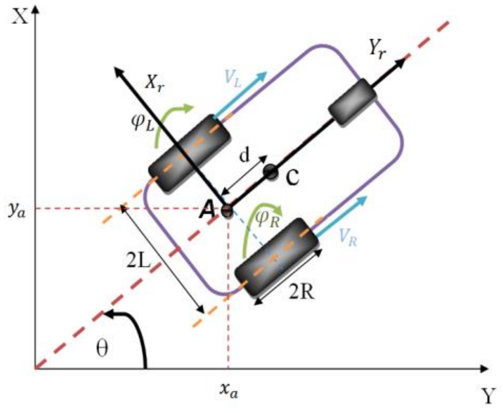

Fig. 1. Schematic design of the WMR.

There are three essential steps to reach the mathematical model of the robot mobile: kinematic modeling, dynamic modeling, and actuator modeling, which are described as follows

2.1 Kinematic Model

This section focuses on illustrating the relationship between the linear and angular speeds of the mechanical systems without taking into account the forces affecting the motion (Mohareri et al., 2012). The linear velocity of each driving wheel is:

$$\begin{cases} v_R = R\dot{\varphi}_R \\ v_L = R\dot{\varphi}_L \end{cases} \tag{1}$$

The linear and angular velocities of the WMR are given by Eqs. (2) and (3), respectively:

$$v = \frac{v_R + v_L}{2} = R\frac{(\dot{\varphi}_R + \dot{\varphi}_L)}{2} \tag{2}$$

$$\omega = \frac{v_R - v_L}{2L} = R\frac{(\dot{\varphi}_R - \dot{\varphi}_L)}{2L} \tag{3}$$

The kinematic constraint can express by the following equations (Ben Jabeur and Seddik, 2021):

No slip constraint:

$$-\dot{x}_a \sin\theta + \dot{y}_a \cos\theta = 0 \tag{4}$$

Pur rolling constraint:

$$\begin{aligned} \dot{x}_a \cos\theta + \dot{y}_a \sin\theta + L\dot{\theta} &= R\dot{\varphi}_R \\ \dot{x}_a \cos\theta + \dot{y}_a \sin\theta - L\dot{\theta} &= R\dot{\varphi}_L \end{aligned} \tag{5}$$

The three constraint equations (4) & (5) are assembled as:

$$\Lambda(q)\dot{q} = 0 \tag{6}$$

Where:

$$(q) = \begin{bmatrix} -\sin\theta & \cos\theta & 0 & 0 & 0 \\ \cos\theta & \sin\theta & L & -\mathrm{R} & 0 \\ \cos\theta & \sin\theta - L & 0 & -R \end{bmatrix} \tag{7}$$

And:

$$\dot{q} = \left[\dot{x}_a \; \dot{y}_a \; \dot{\theta} \; \dot{\varphi}_R \; \dot{\varphi}_L\right]^T \tag{8}$$

So, the kinematic model obtained is:

$$\begin{bmatrix}\dot{x}_a\\ \dot{y}_a\\ \dot{\theta}\\ \dot{\varphi}_R\\ \dot{\varphi}_L\end{bmatrix} = \begin{bmatrix}cos\,\theta & 0\\ sin\,\theta & 0\\ 0 & 1\\ \frac{1}{R} & \frac{-L}{R}\\ \frac{1}{R} & \frac{-L}{R}\end{bmatrix}\begin{bmatrix}v\\ \omega\end{bmatrix} = \frac{1}{2}\begin{bmatrix}R\,cos\,\theta & R\,cos\,\theta\\ R\,sin\,\theta & R\,sin\,\theta\\ \frac{R}{L} & -\frac{R}{L}\\ 2 & 0\\ 0 & 2\end{bmatrix}\begin{bmatrix}\dot{\varphi}_R\\ \dot{\varphi}_L\end{bmatrix} \tag{9}$$

This may be written as:

$$\dot{q} = S(q)\eta \tag{10}$$

Where, $\eta = \begin{bmatrix}\dot{\varphi}_R\\ \dot{\varphi}_L\end{bmatrix}$ is the vector of the angular velocities of two wheels.

2.2. Dynamical Model

The purpose of dynamic modeling is to examine all different forces and energies that influence the motion of mechanical process. The motion equation of the WMRs is given by (Ben Jabeur and Seddik, 2021; Fierro and Lewis, 1995):

$$M(q)\ddot{q} + V(q,\dot{q})\dot{q} + F(\dot{q}) + G(q) + \tau_d = B(q)\tau - \Lambda^T(q)\lambda \tag{11}$$

Where, M(q) an nxn symmetric positive definite inertia matrix, $V(q,\dot{q})$ is the centripetal and coriolis matrix, $F(\dot{q})$is the surface friction matrix, G(q) is the gravitational vector, τ_d is the vector of bounded unknown disturbances including unstructured unmodeled dynamics, B(q) is the input matrix, τ is the input vector, $\Lambda^T(q)$ is the associated matrix to the kinematic constraints, and λ is the Lagrange multipliers vector. For simulation purpose and control, equation (11) should be transformed in to an alternative form, using the kinematic model (10).We have:

$$S^T(q)\Lambda^T(q) = 0 \tag{12}$$

The new matrices are express as fellow:

$$\begin{cases}\bar{M}(q) = S^T(q)\,M(q)S(q),\\ \bar{V} = S^T(q)M(q)\dot{S}(q) + S^T(q)V(q,\dot{q})S(q)\\ \bar{B} = S^T(q)B(q)\end{cases} \tag{13}$$

The reduced form of the dynamic equations is given by:

$$\bar{M}(q)\dot{\eta} + \bar{V}(q,\dot{q})\eta = \bar{B}(q)\tau \tag{14}$$

Where:

$$\bar{M}(q) = \begin{bmatrix}I_w + \frac{R^2}{4L^2}(mL^2 + I) & \frac{R^2}{4L^2}(mL^2 - I)\\ \frac{R^2}{4L^2}(mL^2 - I) & I_w + \frac{R^2}{4L^2}(mL^2 + I)\end{bmatrix}$$

$$\bar{V}(q,\dot{q}) = \begin{bmatrix}0 & \frac{R^2}{2L}m_c d\dot{\theta}\\ -\frac{R^2}{2L}m_c d\dot{\theta} & 0\end{bmatrix}, \quad \bar{B}(q) = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$$

The dynamics in Equation (14) are based only on the right and left wheel angular velocities$(\dot{\varphi}_R, \dot{\varphi}_L)$, the angular velocity of the robot $\dot{\theta}$ and the dc motor torques (τ_R, τ_L) .The equations (14) can be rewritten in a compact form:

$$\begin{cases}\left(m + \frac{2I_w}{R^2}\right)\dot{v} - m_c d\omega^2 = \frac{1}{R}(\tau_R + \tau_L)\\ \left(I + \frac{2L^2}{R^2}I_w\right)\dot{\omega} + m_c d\omega v = \frac{L}{R}(\tau_R - \tau_L)\end{cases} \tag{15}$$

Where, $m = m_c + 2m_w$ is the total mass of the robot, and $I = I_c + m_c d^2 + 2m_w L^2 + 2I_m$ is the total equivalent inertia.

2.3 Actuator modeling

Two pairs of dc motors are considered as actuators for producing the torque to control the inputs for driving the wheels of the WMR as presented by Fig 2.

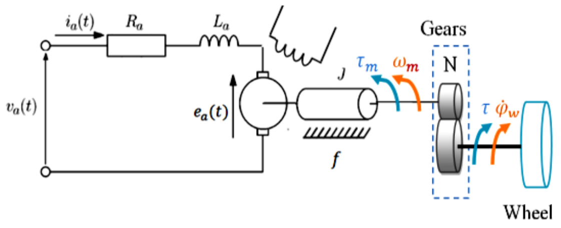

Fig. 2. Equivalent electrical scheme for dc motor.

The dynamic model of the actuators can be represented as(Ben Jabeur and Seddik, 2021)

$$\begin{cases}v_a = R_a i_a(t) + L_a\frac{di_a(t)}{dt} + e_a(t)\\ e_a(t) = K_b\omega_m(t)\\ \tau_m = j\frac{dw_m(t)}{dt} + fw_m(t) + K_t i_a(t)\\ \tau = N\tau_m\end{cases} \tag{16}$$

Where, all these parameters are explained in Table 1**.**

Since the robot motors are mechanically coupled to wheels through the gears. Therefore, each dc motor will have:

$$\begin{cases}\omega_{mR} = N\dot{\varphi}_{wR} and \tau_R = N\tau_{mR}\\ \omega_{mL} = N\dot{\varphi}_{wL} and \tau_L = N\tau_{mL}\end{cases} \tag{17}$$

For both motors, the dynamic model is expressed as:

$$\begin{cases}\frac{1}{(R + L_a P)}(e_{ar} - K_b N\dot{\varphi}_{wR}) = i_R with \tau_R = Nk_t i_R\\ \frac{1}{(R + L_a P)}(e_{al} - K_b N\dot{\varphi}_{wL}) = i_L with \tau_L = Nk_t i_L\end{cases} \tag{18}$$

The robot physical (real data) parameters are taken from (Ben Jabeur and Seddik, 2021) :

Table 1. Robot physical parameters.

Parameters	Value and Unit
Distance between two wheels, 2 L	0.75 m
Distance of point Pc from point Po, D	0.3 m
Driving wheels radius, R	0.15 m
Mass of the mobile robot without the driving wheels and DC motors, m_c	30 kg
Mass of each driving wheel (with actuator), m_w	1 kg
Moment of inertia of the mobile robot about the vertical axis through the center of mass, I_c	15.625 kg·m^2

Moment of inertia of each driving wheel with a motor about the Wheel axis, I_w	0.005 kg·m^2
Moment of inertia of each driving wheel with a motor about the Wheel diameter, I_m	0.0025 kg·m^2
Armature winding resistance, R_a	1.6 Ω
Armature winding inductance, L_a	0.048 H
Torque constant, K_t	0.2613 N·m/A
Back emf constant, K_b	0.19 rad/s
Gear ratio, N	62.55

3. CONTROLLER DESIGN AND STRATEGY

Fig. 3 shows the schematic form of the control approach based on the dynamic and kinematic of the wheeled mobile robot. The trajectory generator supplies the desired coordinate x, y and θ.

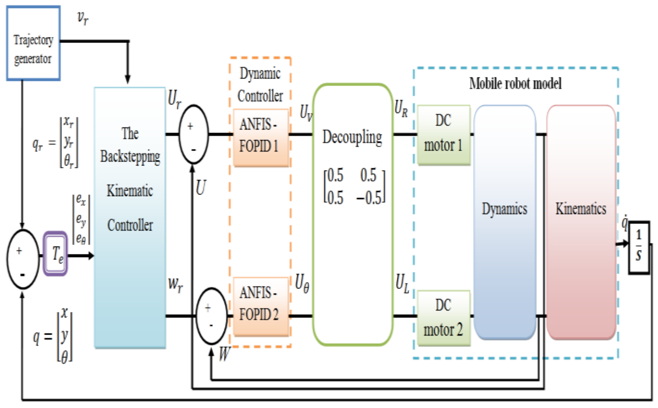
Trajectory generator
v_r
$q_r = \begin{bmatrix} x_r \\ y_r \\ \theta_r \end{bmatrix}$
The Backstepping Kinematic Controller
$\begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix}$
T_e
$q = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$
U_r
U
W_r
W
Dynamic Controller
ANFIS-FOPID 1
ANFIS-FOPID 2
U_V
U_θ
Decoupling
$\begin{bmatrix} 0.5 & 0.5 \\ 0.5 & -0.5 \end{bmatrix}$
U_R
U_L
Mobile robot model
DC motor 1
DC motor 2
Dynamics
Kinematics
$\dot{q}$
$\frac{1}{s}$

Fig. 3. Control scheme of WMR.

A back-stepping kinematic controller is detected in the external loop, here the controller examines the difference between the reference data (obtained from the bloc trajectory generator) and physical value from the robot then develop his own angular and linear velocities that are delivered into the internal loop, where two blocks of ANFIS-FOPID controller deal with the dynamics. Here, the two dynamic controllers provide their own signals of angular and linear speeds based on the values taken from the kinematic controller.

3.1 Kinematic Controller Design

One of the objectives here is to realize path tracking. The back-stepping technique has been commonly used, and it is classified as a stable tracking control law, which makes it supported for this purpose (Mai et al., 2021). The controller structure is introduced in Fig. 3, where the input error and velocity vector (vc) are:

$$\begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_r - & X \\ Y_r - & Y \\ \theta_r - & \theta \end{bmatrix} = T_e e_r \tag{19}$$

$$v_c = \begin{bmatrix} v_r \cos(e_\theta) + k_x e_x \\ w_r + k_y v_r e_y + k_\theta v_r \sin(e_\theta) \end{bmatrix} \tag{20}$$

where k_x, k_y and k_θ are the tuning parameters.

3.2 The FOPID Controller description

The fractional proportional integral derivative FOPID was first proposed in 1999 by Podlubny (Magdy et al., 2020). Introducing the two new fractional components λ and μ to the classic PID controller under the name of Fractional integrator and differentiator respectively, the operator of non-integer-order is provided in Eq. (21) (Bhookya and Kumar Jatoth, 2020).

$$D_n^t = \begin{cases} \frac{d^n}{dt^n} n > 0 \\ 1 \qquad n = 0 \\ \int_t^a dt^n n < 0 \end{cases} \tag{21}$$

Where, t and a are the lower limit and upper limit of the process and $n \in \mathbb{R}$ the constant integral differential operator.

The fractional calculus has three main definitions Riemann Liouville (RL), GrünwaldL et Nikov (GL), and Caputo. The most common definition utilized in engineering applications is the Caputo method (Bhookya and Kumar Jatoth, 2020) given in Eq. (22).

$$D_n^t f(t) = \frac{1}{\Gamma(\alpha - n)} \int_a^t \frac{f^n(\tau)}{(t-\tau)^{n-\alpha+1}} d\tau \tag{22}$$

$$for \;\; n - 1 \le n \le \alpha$$

The term s^α doesn't have an analytical solution, but it has a fractional order, which is hard to implement. Therefore, numerical solutions such as Oustaloup approximation are adopted (Bhookya and Kumar Jatoth, 2020).

$$s^\alpha \approx K \prod_{n=-N}^{N} \frac{1 + \frac{s}{\omega_{z,n}}}{1 + \frac{s}{\omega_{p,n}}} \alpha > 0 \tag{23}$$

Where, N is the number of poles/zeros.

Fig. 4 presents the parallel structure of the FOPID controller block diagram, with input E(S) and output U(S).

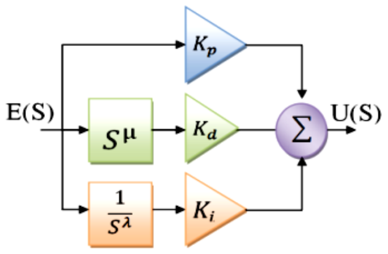

Fig. 4. Block diagram of fractional order.

The representation of the transfer function for the Fractional Order PID (FOPID) in the time domain is given by Eq. (24), and in the Laplace domain by Eq. (25).

$$u(t) = K_p e(t) + k_i D_t^{-\lambda} e(t) + K_d D_t^{\mu} e(t) \tag{24}$$

$$C(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{S^\lambda} + K_d S^\mu \tag{25}$$

where, K_p, K_i and K_d are proportional, integral and derivative gains constants, respectively, λ and μ are factional order of the integral and derivative term.

3.3 Tuning Parameters for FOPID

The WMR block diagram for tuning parameters of the Two FOPID controllers is highlighted in Fig. 5.

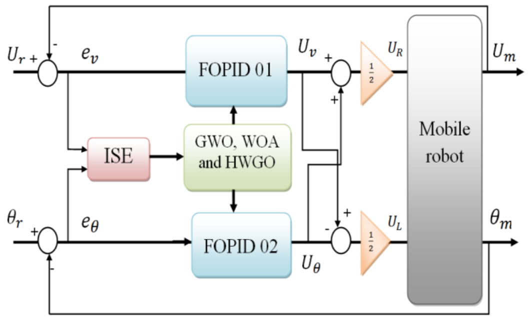

Fig. 5. Bloc diagram for parameters tuning.

The first controller treats the difference between the reference and actual velocity from the right wheel as an input where the desired velocity is constant: Ur =1 m/s.

The second controller for the left wheel treats the difference between the actual and desired orientations as input. Where the reference angle was represented by a constant value:θ_r=0.785rad.

The following equations (26) represent the relation between the input voltage of the right and left DC actuators, respectively U_R,U_L with the outputs of the first and second controller U_V,U_θ ,respectively.

$$\begin{cases} U_R = \dfrac{U_V + U_\theta}{2} \\ U_L = \dfrac{U_V - U_\theta}{2} \end{cases} \tag{26}$$

The cost function used in the study is the integral square error (ISE) which is demonstrated by equation (27):

$$ISE = \left(\int_0^\infty [e_v(t)]^2 dt + \int_0^\infty [e_\theta(t)]^2 dt\right) \tag{27}$$

where: $$\begin{cases} e_v = U_r - U_m \\ e_\theta = \theta_r - \theta_m \end{cases} \tag{28}$$

U_r is the desired velocity, U_m the actual velocity, e_v the velocity error, θ_r the desired orientation,θ_m the measured orientation, and e_θ is the orientation error.

The optimization algorithms used to tune the parameters (K_p, K_I, K_d, λ, μ) are described in the next section.

3.4 Meta-heuristics Algorithms

A comparative study has been made to demonstrate the capability of the hybrid whale grey wolf optimizer (HWGO), compared to the grey wolf optimizer (GWO) and whale optimizer algorithm (WOA). Therefore, a brief description of all algorithms is given:

A. Grey wolf optimizer (GWO)

The Grey wolf optimizer was inspired by the hunting mechanism and leadership hierarchy of grey wolves in nature. Its mathematical models consist of social hierarchy, enriching prey, search for prey, attacking prey, and hunting that is described down below (Mirjalili et al., 2014; Mittal et al., 2016):

Social hierarchy: The leader of wolves is called alpha(α). Beta (β) and delta (δ) are the second and third level in the group respectively.

Encircling prey: the grey wolves encircle the prey for hunting, which can be mathematically represented as

$$d = |c \cdot x_p(n) - x(n)| \tag{29}$$
$$x(n+1) = x_p(n) - a \cdot d. \tag{30}$$

Where, x_p is the position vector of the prey, n indicates the current iteration, and x indicates the position vector of a grey wolf.

The vectors a and c are mathematically formulated as follows:

$$\vec{a}_{(.)} = 2\vec{l} \cdot \vec{r}_1 - \vec{l} \tag{31}$$
$$\vec{c}_{(.)} = 2 \cdot \vec{r}_2. \tag{32}$$

Where, r_1 and r_2 are random numbers in [0, 1] and a parameteris linearly decreased from 2 to 0 .

- *Hunting*

The following mathematical equations are developed in this regard:

$$\begin{cases} \vec{d}_\alpha = |\vec{c}_1 \cdot \vec{x}_\alpha - \vec{x}|, \\ \vec{d}_\beta = |\vec{c}_2 \cdot \vec{x}_\beta - \vec{x}|, \\ \vec{d}_\delta = |\vec{c}_3 \cdot \vec{x}_\delta - \vec{x}|. \end{cases} \tag{33.a}$$

$$\begin{cases} \vec{x}_1 = \vec{x}_\alpha - \vec{a}_1 \cdot (\vec{d}_\alpha), \\ \vec{x}_2 = \vec{x}_\beta - \vec{a}_2 \cdot (\vec{d}_\beta), \\ \vec{x}_3 = \vec{x}_\delta - \vec{a}_3 \cdot (\vec{d}_\delta), \end{cases} \tag{33.b}$$

$$x(n+1) = \frac{\vec{x}_1 + \vec{x}_2 + \vec{x}_3}{3} \tag{33.c}$$

The mathematical simulation of hunting behavior is possible after supposing that the alpha (α), beta (β), and delta (δ) have better knowledge about the potential location of prey.

- *Searching for Prey and Attacking Prey*

A is a random value in the gap $[-2a, 2a]$. When, $|A|$ <1, the wolves are forced to attack the prey. Searching for prey is the exploration ability and attacking the prey is the exploitation ability. The arbitrary values of A are utilized to force the search to move away from the prey. When, $|A|$ > 1, the members of the population are enforced to diverge from the prey.

B. Whale optimization algorithm (WOA)

This algorithm was first introduced by Mirjalili and Lewis (Mirjalili and Lewis, 2016) .Its main inspiration comes from the social behavior of humpback whales .As they have special hunting behavior, they use to hunt schools of fish in which whales create a 9-shaped net with bubbles that traps the fish and makes the whale capable of eating them with ease (Hemeida et al., 2020).This Algorithm consists of three stages.

- Encircling prey:

The humpback whales encircle the prey and modify their position towards the best solution over a course of iterations. This behavior can be mathematically represented using (29) and (30).

- Bubble-net attacking method:

Using the bubble-net strategy the humpback whales attack the prey, this is shown in the following two methods:

1. Shrinking encircling mechanism: This technique is applicable by decreasing the value of l in (31).

2. Spiral updating position: The mathematical description of the helix-shape movement of humpback whales is as follows:

$$\vec{X}(n+1) = \vec{D}' \cdot e^{bv} \cdot \cos(2\pi v) + \vec{X}^*(n) \tag{34}$$

$$\vec{D}' = \left|\vec{X}^*(n) - \overrightarrow{X(n)}\right| \tag{35}$$

Where, v is a random number in [-1], b is constant for defining the shape of the logarithmic spiral, $\vec{X}^*$is the position vector of the prey position, and $\vec{X}$is the position vector of the humpback whale.

The humpback whales swim around the prey within a shrinking circle and along a spiral-shaped path. A probability of 50% to choose either a shrinking mechanism or a helical path down below the mathematical formula:

$$\vec{X}(n+1) = \begin{cases} \vec{X}^*(n) - \vec{A} \cdot \vec{D} & \text{if } p < 0.5 \\ \vec{D}' \cdot e^{bv} \cdot \cos(2\pi v) + \vec{X}^*(n) & \text{if } p > 0.5 \end{cases} \tag{36}$$

Where, p is a random number in [0,1].

c- Search for prey:

The humpback whales search randomly for prey. Exploitation phase fellow those rules:

$$\vec{D} = \left|\vec{C} \cdot \vec{X}_{rand} - \vec{X}\right| \tag{37}$$

$$\vec{X}(n+1) = \vec{X}_{rand} - \vec{A} \cdot \vec{D} \tag{38}$$

C. Hybrid Whale grey wolf optimizer (HWGO)

The HWGO algorithm aims to improve the performance of the WOA algorithm by applying the leadership hierarchy of GWO. The next step is to apply the outcome to the attacking strategy of the WOA. the HWGO selected three best candidate solutions; the first level alpha (a), the second, and third-level in the group is beta (b) and delta (d), respectively from whole search agents, and the other search agents will modify their positions according to the position of the best search agents to improve the performance of the WOA algorithm (Korashy et al., 2019).

The mathematical model for updating the position of whales using the leadership hierarchy of GWO during optimization is represented mathematically as follows:

The updating of the position of whales using the hierarchy leadership of GWO is formulated using (33).

The updating of the position along of a spiral-shape path of humpback whales as follows:

$$\begin{cases} \vec{D}'_\alpha = \left|\vec{X}_\alpha(n) - \vec{X}\right|, \\ \vec{D}'_\beta = \left|\vec{X}_\beta(n) - \vec{X}\right|, \\ \vec{D}'_\delta = \left|\vec{X}_\delta(n) - \vec{X}\right| \end{cases} \tag{39}$$

$$\begin{cases} \vec{X}_1(n) = \vec{X}_\alpha(n) + \vec{D}_\alpha \cdot e^{bv} \cdot \cos(2\pi v), \\ \vec{X}_2(n) = \vec{X}_\beta(n) + \vec{D}_\beta \cdot e^{bv} \cdot \cos(2\pi v), \\ \vec{X}_3(n) = \vec{X}_\delta(n) + \vec{D}_\delta \cdot e^{bv} \cdot \cos(2\pi v) \end{cases} \tag{40}$$

$$\vec{X}(n+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \tag{41}$$

3.5 The ANFIS Controller descreption

The adaptive Network-Based Fuzzy Inference System (ANFIS) is a hybrid combination of artificial neural network (ANN) and fuzzy inference system (FIS). This algorithm was firstly introduced by Jang (Jang, 1993) and it is based on the first-order Sugeno fuzzy model (Buragohain and Mahanta, 2008). The ANFIS approaches integrates the best feature of neural networks and fuzzy systems by exercising learning capability of neural network while the learning ability is a advantage in terms of a fuzzy system (Bektas Ekici and Aksoy, 2011). However, this combined system offers more benefits from the aspect of a neural network. The ANFIS structure consists of if-then rules and couples of input-output data of fuzzy while for training ANFIS uses neural network's learning algorithms. Also, ANFIS apply the neural network training process to fine-tune the membership function (MF) and the associated parameter that methods the desired data sets (Bektas Ekici and Aksoy, 2011; Boyacioglu and Avci, 2010).

For simplifying, a Sugeno fuzzy model with two inputs, *x* and *y*, and one output *f* is considered. Usually, the fuzzy rules are reported as following (Bektas Ekici and Aksoy, 2011; Boyacioglu and Avci, 2010; Buragohain and Mahanta, 2008; Jang, 1993):

Rule I: if $x=A_1$ *and* $y= B_1$, then

$$f_1 = p_1 x + q_1 y + r_1 \tag{42}$$

Rule II: if $x=A_2$ *and* $y=B_2$, then

$$f_2 = p_2 x + q_2 y + r_2 \tag{43}$$

where, *f* is an output parameter (response), *p*, *q*, & r are linear parameters, and *A&B* are nonlinear parameters. Fig.6. Shows the five layers used to construct ANFIS structure while the function of each layer is described below:

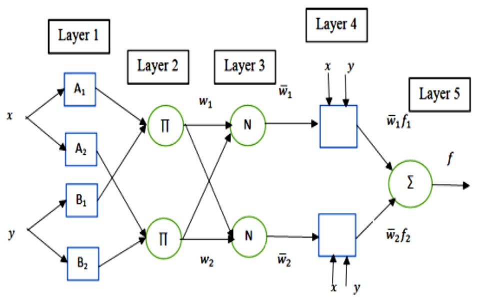

Fig. 6. The ANFIS architecture with two input parameters x and y.

Layer 1. input nodes: Every node in this layer generates membership grades to which they belong to each of the appropriate fuzzy sets using MFs.

$$\begin{cases} O_i^1 = \mu_{Ai}(x) \\ O_i^2 = \mu_{Bi}(y) \end{cases} \tag{44}$$

Where, i= 1, 2 ,x, $y$are the crisp inputs to node i, and A_i &B_i. The linguistic label (small, large, etc.) associated with the node function. μ_{Ai} , μ_{Bi} respectively.

Usually, μ_{Ai} (or μ_{Bi}) is selected as:

$$\mu_{Ai}(x) = \frac{1}{1+\left[\left(x-{c_i}/{a_i}\right)^2\right]^{b_i}} \tag{45}$$

Or

$$\mu_{Ai}(x) = exp\left\{-\left(\frac{x-c_i}{a_i}\right)^2\right\} \tag{46}$$

Here, {a_i, b_i, c_i} are the premise parameter set.

Layer 2. rule nodes: Using mathematical multiplication, the firing strength of each rule is calculated. For instance,

$$O_i^2 = w_i = \mu_{A_i}(x) \cdot \mu_{B_i}(y) \tag{47}$$

Each node output represents the firing strength of a rule.

Layer 3.average nodes: In the third layer, The ith node calculates the ratio of the ith rule firing strength to the summation of the firing strengths of all rules Consequently, as given in (48):

$$O_i^3 = \overline{w}_i = \frac{w_i}{w_1+w_2} \tag{48}$$

w_i is taken as the normalized firing strength.

Layer 4. consequent nodes: The node function of the fourth layer calculates the contribution of each i^{th} rules toward the overall output, defined as:

$$O_i^4 \quad = \overline{w}_i f_i = \overline{w}_i(p_i x + q_i y + r_i) \tag{49}$$

where, $\overline{w}_i$isthe output of the ***layer 3***, and {p_i, q_i, r_i} the parameter set.

Layer 5. *Output nodes:*The single-node computes the overall output by adding all the incoming signals from the 4th layer :

$$O_i^5 = overall\ output = \sum_i \overline{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \tag{50}$$

The final output of ANFIS can be expressed as a linear combination of consequent parameter, the output can be written as:

$$f_{out} = \overline{w}_1 f_1 + \overline{w}_2 f_2 = \frac{w_1}{w_1+w_2} f_1 + \frac{w_2}{w_1+w_2} f_2 = (\overline{w}_1 x)p_1 + (\overline{w}_1 y)q_1 + (\overline{w}_1)r_1 + (\overline{w}_2 x)p_2 + (\overline{w}_2 y)q_2 + (\overline{w}_2)r_2 \tag{51}$$

3.6 Designing the ANFIS-FOPID Dynamic Controller

The implementation of the full control strategy that handles the dynamics of the WMR based on the ANFIS-FOPID, can be divided into three steps:

The first step is represented by Fig. 7, which shows the training phase of the ANFIS controller. The ANFIS utilize their adaptive and learning capacities to learn and to predict the best required control actions based on the available data. These data are obtained from the already designed FOPID controllers (see figure 7).

The second step: the gained training data were stored in a file, from MATLAB/SIMULINK using the command ***anfisedit***, these data were taken for training.

The ANFIS training used hybrid training algorithm, with the input nodes (2, 2, 2),using the gaussmf membership function for the inputs and linear membership function for the output, each having 8 rules, the epoch length was used is 50 iterations for each sample, with, 0.01s as the Simulink sampling time.

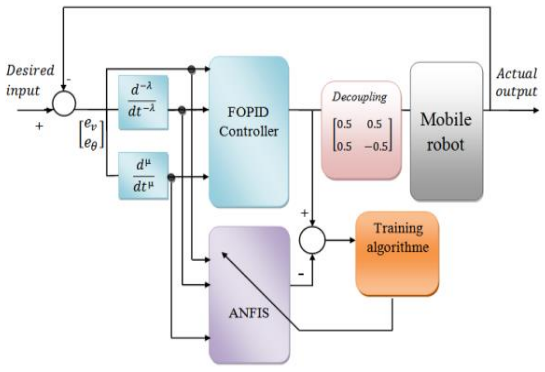

Fig. 7. The ANFIS in training phase.

At the end of the training the result was saved as a *fis* file with respect to Sugeno-style, where the root mean square error (RMSE) was found to be 0.1053 and be 0.094977 for ANFIS-FOPID1 and ANFIS-FOPID2, respectively.

Third step: after the training phase is complete and the optimal value of the data are obtained, the designed ANFIS-FOPID controller should be ameliorated by adjusting the gains, i.e., K_p, K_I, K_d, λ, μ, as shown in Fig.8.Where the final version of the controller is represented in Fig 9.

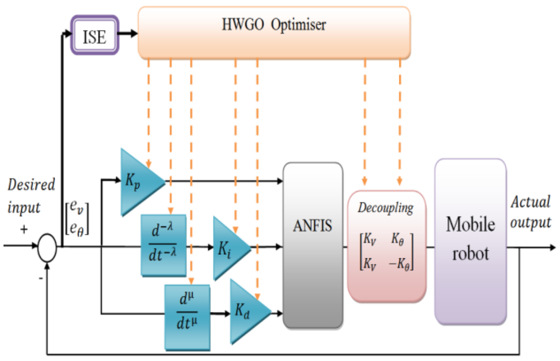

Fig. 8. Bloc diagram for parameters tuning.

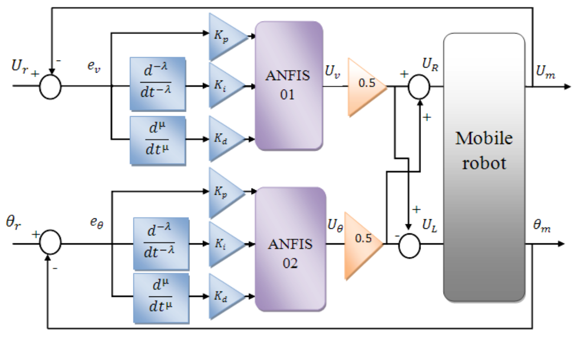

Fig. 9. Final version of the ANFIS-FOPID controller.

4. RESULTS AND DISCUSION

4.1 Convergence Curve of the Algorithms

The three *meta-heuristics* optimizations algorithms are used for the FOPID controller design. Fig. 10 presents the convergence curve obtained by these optimization's algorithms applied to FOPID and ANFIS-FOPID controllers

The obtained values of the ISE for all algorithms used in the test, are shown in Table 2.

From fig. 10 and Table 2, it quite obvious that the HWGO optimizer gives the lowest ISE value compared to the whale optimization algorithm (WOA) and the grey wolf optimizer (GWO).

Table 2. ISE values for all controllers.

Methods	GWO-FOPID	WOA-FOPID	HWGO-FOPID	HWGO-ANFIS-FOPID
ISE value	0.0477	0.0533	0.0452	0.0268

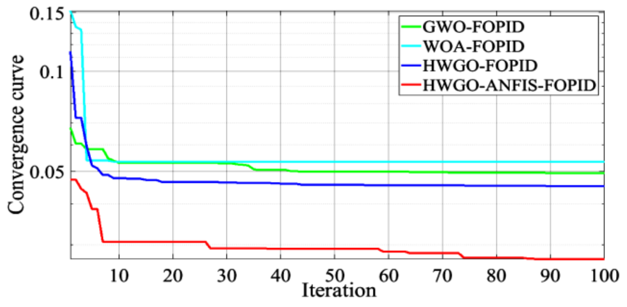

Fig. 10. Convergence curve of the algorithms.

4.2 Step Response Performance of Speed Controller

Fig 11 presents the comparative findings for the linear velocity step responses that were designed with the three controllers.

Table 3 shows the gain details of the controllers, for the speed control of the WMR, tuned by HWGO optimizer. Table.4 shows the characteristics obtained in time domain.

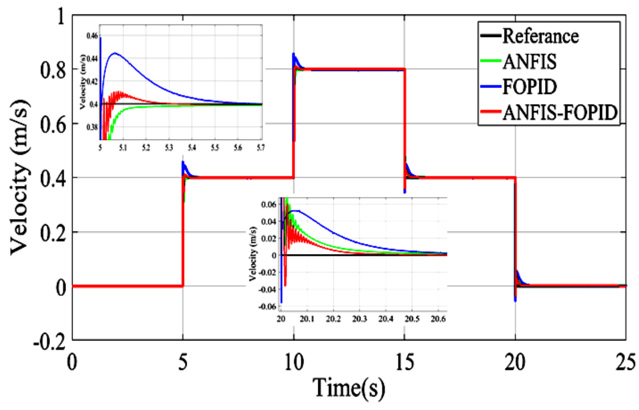

Fig. 11. Step response Comparison for velocity controllers.

Table 3.The tuned gain details of linear velocity.

Methods	Speed controller				
	K_p	K_i	K_d	λ	μ
FOPID	500	145	149	0.10	0.56
ANFIS	/	/	/	/	/
ANFIS-FOPID	2	1.01	1	0.2046	0.8674

Table 4. Performance characteristics for velocity Controller.

Methods	Characteristics of the speed controllers				
	t_r	Mp%	Peak	t_p	t_s
FOPID	0.0026	13.1268	0.4525	0.0014	0.2182
ANFIS	0.0451	0	0.3993	0.3240	0.0562
ANFIS-FOPID	0.0242	1.3500	0.4097	0.1000	0.0357

The FOPID controller only advantage is the fast rise time obtained, but a high overshoot and slow settling time.

The ANFIS-FOPID controller advantage is the best settling and rise time value with a nearly zero overshoot.

The ANFIS controller major advantage is the zero-overshoot obtained value and an acceptable settling time and rise time

4.3 Step Response Performance of Angle Controller

Fig. 11 presents the comparative findings for angle step responses that were designed using the three controllers, by applying the HWGO optimization techniques. Table 4 reveals the gain details of all controllers selected for the orientation angle control of the WMR. In contrast, the comparative results of transient response are highlighted in Table 5.

Table 5. The tuned parameters for angle control.

Methods	Angle controller parameter tuning				
	K_p	K_i	K_d	λ	μ
FOPID	63.97	26.4	81.95	0.36	0.95
ANFIS	/	/	/	/	/
ANFIS-FOPID	6.00	4.92	1.21	0.02	1

Table 6. Performance characteristics for angle control.

Methods	Transition parameter for angle controller				
	t_r	Mp%	Peak	t_p	t_s
FOPID	0.4184	3.4084	0.4136	0.9600	0.4309
ANFIS	1.6633	2.2710	0.4091	4.0512	1.6942
ANFIS-FOPID	0.3470	0.1822	0.4007	0.7466	0.3770

From Fig. 12, Table 5 and 6, we can say that the lowest amounts of peak percentage, overshoot (Mp%), rise time (t_r for 10% → 90%), and settling time (t_s for ±2% tolerance) were observed for the suggested ANFIS-FOPID controller.

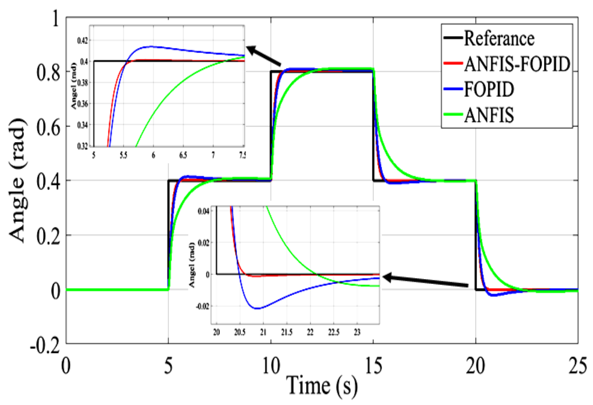

Fig. 12. Comparison of step Response for the orientation.

4.4 Five (5) chape trajetory

The five (5) shaped trajectory is a straightforward example of a non-continuous gradient trajectory. The major problem of such as paths is the sharp and a non-continuous motion, which requires a high-level control effort. here the FOPID controller gives a very high overshoot, that might produce an unstable and a disturbed motion, which could harm the DC actuators (see Fig. 14), which makes it unsupported for this kind of paths.

The ANFIS-FOPID controller has an acceptable overshoot compared to the FOPID and faster time response and lower distance error than the ANFIS , which makes it suitable for this trajectory.

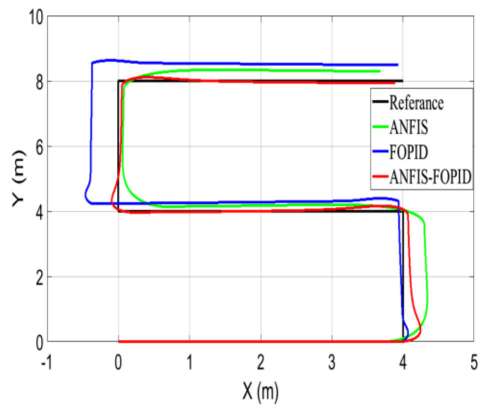

Fig. 13. Five shaped path in XY plane.

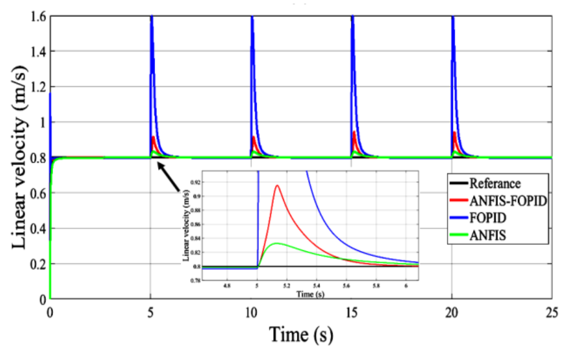

Fig. 14. Linear velocity corresponding to the path.

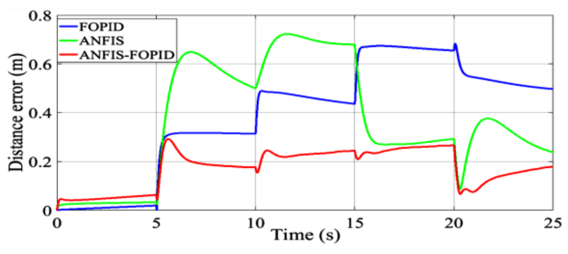

Fig. 15. Distance error of the 5 shaped path.

4.5 Zigzag trajetory

Here both linear and angular velocity changes progressivly (see Fig. 17 and 18), the ANFIS-FOPID controller felows the desired path with a minimum distance error and fast time response (error less than 0.05 m),compared to FOPID and the ANFIS.

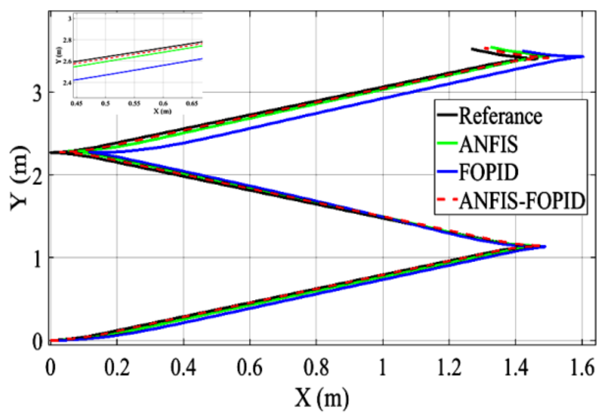

Fig. 16. Zigzag path.

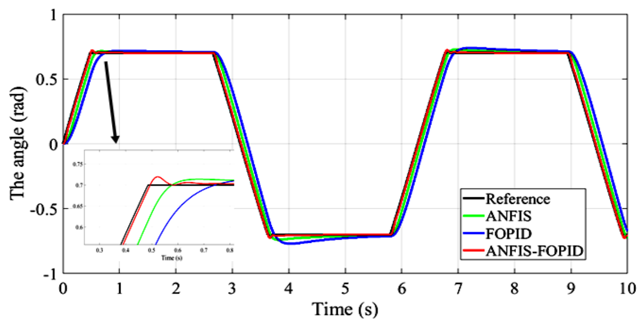

Fig. 17. The angle coresponding to the Zigzag path.

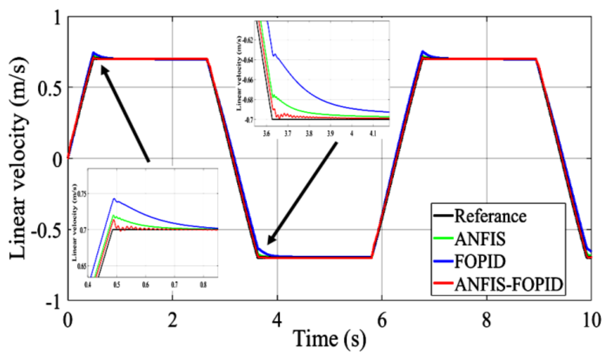

Fig. 18. The linear speed corresponding to zigzag path.

4.6 Adding the back-stepping controller in the external loop

The ANFIS-FOPID controller was designed to handle with the dynamics, which need a kinematic controller. Therefore, a back-stepping controller was proposed to guarantee a minimum distance error that already has been proved in fig.21. The error is less than 0.002 m. To illustrate the robustness of the Back-stepping combined with ANFIS-FOPID controller, a quadrifolium trajectory was selected.

In order to generate this path, the following equations were applied:

$$x_R(n) = 5 * sin^2\left(2 * \pi * \frac{t}{30}\right) * cos\left(2 * \pi * \frac{t}{30}\right) \qquad (52)$$

$$y_R(n) = 5 * cos^2\left(2 * \pi * \frac{t}{30}\right) * sin\left(2 * \pi * \frac{t}{30}\right) \qquad (53)$$

Where the gains are chosen as: k_x=10, k_y=80 and k_θ=15.

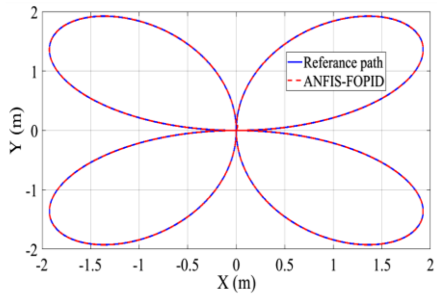

Fig.19. Quadrifolium path.

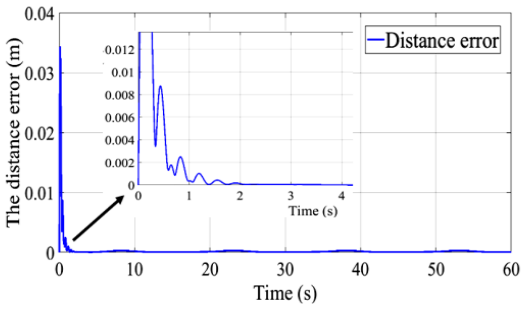

Fig. 20. Distance error of the quadrifolium path.

The influence of the added kinematic controller in external loop is highlighted in Fig.20. The minimum tracking error is less than 0.002 meter. for that purpose, we can see that the two paths are identical in Fig. 19.

5. CONCLUSIONS

The actual study shows the efficiency given by hybridizing the fractional order PID controller (FOPID) with an adaptive neuro-fuzzy inference system (ANFIS), abbreviated as ANFIS-FOPID controller. Tuning parameters of the proposed controller is challenging. Therefore, a powerful nature inspired algorithm is introduced, namely hybrid whale grey wolf optimizer (HWGO) which is a hybridization between the whale optimizer algorithm (WOA) and the grey wolf optimizer (GWO), based on the integral square error cost function. A comparative study has been made between the ANFIS-FOPID against separated FOPID and ANFIS, In terms of convergence curve obtained with lowest error values, and a minimum overshoot observed, rise time, and settling time. As well as several case studies were made for both smooth and sharp-shaped trajectories. Simulation in MATLAB environment has given the following demonstration and judgment between these three controllers:

The FOPID controller advantage is the fast rise time and settling time value obtained, this controller is ideal for smooth trajectories, but when it comes to sharp shaped trajectories, a high overshoot is observed, which could harm the dc actuators .The ANFIS controller major advantage is the zero-overshoot obtained value, but a slow settling time and rise time value (especially in controlling the angle). The ANFIS-FOPID controller mixes the two advantages of the previous controllers by giving the best settling and rise time value, with an acceptable overshoot, which makes it ideal for both smooth and sharp chapped trajectories. Moreover, a better convergence performance is accomplished when a back-stepping technique for the kinematic control is applied as external loop robot control. This practically guarantees a nearly zero tracking error, where a quadrrifolium trajectory used for test.

REFERENCES

Al-Mayyahi, A., Wang, W., & Birch, P. (2014). Adaptive neuro-fuzzy technique for autonomous ground vehicle navigation. *Robotics*, *3*(4), 349–370. https://doi.org/10.3390/robotics3040349

Amroune, M., Bouktir, T., & Musirin, I. (2018). Power System Voltage Stability Assessment Using a Hybrid Approach Combining Dragonfly Optimization Algorithm and Support Vector Regression. *Arabian Journal for Science and Engineering*, *43*(6), 3023–3036. https://doi.org/10.1007/s13369-017-3046-5

Bektas Ekici, B., & Aksoy, U. T. (2011). Prediction of building energy needs in early stage of design by using ANFIS. *Expert Systems with Applications*, *38*(5), 5352–5358. https://doi.org/10.1016/j.eswa.2010.10.021

Ben Jabeur, C., & Seddik, H. (2021). Design of a PID optimized neural networks and PD fuzzy logic controllers for a two-wheeled mobile robot. *Asian Journal of Control*, *23*(1), 23–41. https://doi.org/10.1002/asjc.2356

Bhookya, J., & Jatoth, R. K. (2019). Optimal FOPID/PID controller parameters tuning for the AVR system based on sine–cosine-algorithm. *Evolutionary Intelligence*, *12*(4), 725–733. https://doi.org/10.1007/s12065-019-00290-x

Bhookya, J., & Kumar Jatoth, R. (2020). Fractional Order PID Controller Design for Multivariable Systems using TLBO. *Chemical Product and Process Modeling*, *15*(2), 1–12. https://doi.org/10.1515/cppm-2019-0061

Binh, N. T., Tung, N. A., Nam, D. P., & Quang, N. H. (2019). An Adaptive Backstepping Trajectory Tracking Control of a Tractor Trailer Wheeled Mobile Robot. *International Journal of Control, Automation and Systems*, *17*(2), 465–473. https://doi.org/10.1007/s12555-017-0711-0

Bozek, P., Karavaev, Y. L., Ardentov, A. A., & Yefremov, K. S. (2020). Neural network control of a wheeled mobile robot based on optimal trajectories. *International Journal of Advanced Robotic Systems*, *17*(2), 1–10. https://doi.org/10.1177/1729881420916077

Buragohain, M., & Mahanta, C. (2008). A novel approach for ANFIS modelling based on full factorial design. *Applied Soft Computing Journal*, *8*(1), 609–625. https://doi.org/10.1016/j.asoc.2007.03.010

Catalan, J. M., Blanco, A., Bertomeu-Motos, A., Garcia-Perez, J. V., Almonacid, M., Puerto, R., & Garcia-Aracil, N. (2021). A modular mobile robotic platform to assist people with different degrees of disability. *Applied Sciences (Switzerland)*, *11*(15). https://doi.org/10.3390/app11157130

Cen, H., & Singh, B. K. (2021). Nonholonomic Wheeled Mobile Robot Trajectory Tracking Control Based on Improved Sliding Mode Variable Structure. *Wireless Communications and Mobile Computing*, *2021*. https://doi.org/10.1155/2021/2974839

El-Hasnony, I. M., Barakat, S. I., & Mostafa, R. R. (2020). Optimized ANFIS Model Using Hybrid Metaheuristic Algorithms for Parkinson's Disease Prediction in IoT Environment. *IEEE Access*, *8*, 119252–119270. https://doi.org/10.1109/ACCESS.2020.3005614

Elsisi, M., Tran, M. Q., Mahmoud, K., Lehtonen, M., & Darwish, M. M. F. (2021). Robust design of ANFIS-based blade pitch controller for wind energy conversion systems against wind speed fluctuations. *IEEE Access*, *9*, 37894–37904. https://doi.org/10.1109/ACCESS.2021.3063053

Erkol, H. O. (2018). Optimal PIλ Dμ controller design for two wheeled inverted pendulum. *IEEE Access*, *6*, 75709–75717. https://doi.org/10.1109/ACCESS.2018.2883504

Fue, K., Porter, W., Barnes, E., & Rains, G. (2020). An Extensive Review of Mobile Agricultural Robotics for Field Operations: Focus on Cotton Harvesting. *AgriEngineering*, *2*(1), 150–174. https://doi.org/10.3390/agriengineering2010010

Hemeida, A. M., Alkhalaf, S., Mady, A., Mahmoud, E. A., Hussein, M. E., & Baha Eldin, A. M. (2020). Implementation of nature-inspired optimization algorithms in some data mining tasks. *Ain Shams Engineering Journal*, *11*(2), 309–318. https://doi.org/10.1016/j.asej.2019.10.003

Imen, M., Mansouri, M., & Shoorehdeli, M. A. (2011). Tracking control of mobile robot using ANFIS. *2011 IEEE International Conference on Mechatronics and Automation, ICMA 2011*, *August*, 422–427. https://doi.org/10.1109/ICMA.2011.5985695

Korashy, A., Kamel, S., Jurado, F., & Youssef, A. R. (2019). Hybrid Whale Optimization Algorithm and Grey Wolf Optimizer Algorithm for Optimal Coordination of Direction Overcurrent Relays. *Electric Power Components and Systems*, *47*(6–7), 644–658. https://doi.org/10.1080/15325008.2019.1602687

Magdy, G., Shabib, G., Elbaset, A. A., & Mitani, Y. (2020). Renewable Power Systems Dynamic Security. *Power Systems*, *44*(1), 1–13.

Mai, T. A., Dang, T. S., Duong, D. T., Le, V. C., & Banerjee, S. (2021). A combined backstepping and adaptive fuzzy PID approach for trajectory tracking of autonomous mobile robots. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, *43*(3), 1–13. https://doi.org/10.1007/s40430-020-02767-8

Martins, F. N., Celeste, W. C., Carelli, R., Sarcinelli-Filho, M., & Bastos-Filho, T. F. (2008). An adaptive dynamic controller for autonomous mobile robot trajectory tracking. *Control Engineering Practice*, *16*(11), 1354–1363. https://doi.org/10.1016/j.conengprac.2008.03.004

Mirjalili, S. (2015). The ant lion optimizer. *Advances in Engineering Software*, *83*, 80–98. https://doi.org/10.1016/j.advengsoft.2015.01.010

Mirjalili, S., & Lewis, A. (2016). The Whale Optimization Algorithm. *Advances in Engineering Software*, *95*, 51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey Wolf Optimizer. *Advances in Engineering Software*, *69*, 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007

Mittal, N., Singh, U., & Sohi, B. S. (2016). Modified Grey Wolf Optimizer for Global Engineering Optimization. *Applied Computational Intelligence and Soft Computing*, *2016*, 1–16. https://doi.org/10.1155/2016/7950348

Nikranjbar, A., Haidari, M., & Atai, A. A. (2018). Adaptive Sliding Mode Tracking Control of Mobile Robot in Dynamic Environment Using Artificial Potential Fields. *Journal of Computer & Robotics*, *11*(1), 1–14.

Pei, G., Yu, M., Xu, Y., Ma, C., Lai, H., Chen, F., & Lin, H. (2021). An improved pid controller for the compliant constant-force actuator based on bp neural network and smith predictor. *Applied Sciences (Switzerland)*, *11*(6). https://doi.org/10.3390/app11062685

Poberznik, A., Leino, M., Huhtasalo, J., Jyräkoski, T., Valo, P., Lehtinen, T., Kortelainen, J., Merilampi, S., & Virkki, J. (2021). Mobile robots and rfid technology-based smart care environment for minimizing risks related to employee turnover during pandemics. *Sustainability (Switzerland)*, *13*(22). https://doi.org/10.3390/su132212809

Premkumar, K., & Manikandan, B. V. (2014). Adaptive Neuro-Fuzzy Inference System based speed controller for brushless DC motor. *Neurocomputing*, *138*, 260–270. https://doi.org/10.1016/j.neucom.2014.01.038

Rajesh, R. (2019). Optimal tuning of FOPID controller based on PSO algorithm with reference model for a single conical tank system. *SN Applied Sciences*, *1*(7), 1–14. https://doi.org/10.1007/s42452-019-0754-3

Rossomando, F. G., Soria, C., & Carelli, R. (2011). Autonomous mobile robots navigation using RBF neural compensator. *Control Engineering Practice*, *19*(3), 215–222. https://doi.org/10.1016/j.conengprac.2010.11.011

Saxena, A., Dubey, Y. M., Kumar, M., & Saxena, A. (2021). Performance Comparison of ANFIS, FOPID-PSO and FOPID-Fuzzy Tuning Methodology for Optimizing Response of High-Performance Drilling Machine. *IETE Journal of Research*. https://doi.org/10.1080/03772063.2021.1933625

Slama, S., Errachdi, A., & Benrejeb, M. (2019). Neural Adaptive PID and Neural Indirect Adaptive Control Switch Controller for Nonlinear MIMO Systems. *Mathematical Problems in Engineering*, *2019*.

https://doi.org/10.1155/2019/7340392

Štefek, A., Pham, V. T., Krivanek, V., & Pham, K. L. (2021). Optimization of fuzzy logic controller used for a differential drive wheeled mobile robot. *Applied Sciences (Switzerland)*, *11*(13). https://doi.org/10.3390/app11136023

Tiep, D. K., Lee, K., Im, D. Y., Kwak, B., & Ryoo, Y. J. (2018). Design of fuzzy-PID controller for path tracking of mobile robot with differential drive. *International Journal of Fuzzy Logic and Intelligent Systems*, *18*(3), 220–228. https://doi.org/10.5391/IJFIS.2018.18.3.220

Verma, S. K., Yadav, S., & Nagar, S. K. (2017). Optimization of Fractional Order PID Controller Using Grey Wolf Optimizer. *Journal of Control, Automation and Electrical Systems*, *28*(3), 314–322. https://doi.org/10.1007/s40313-017-0305-3

Wu, X., Jin, P., Zou, T., Qi, Z., Xiao, H., & Lou, P. (2019). Backstepping Trajectory Tracking Based on Fuzzy Sliding Mode Control for Differential Mobile Robots. *Journal of Intelligent and Robotic Systems: Theory and Applications*, *96*(1), 109–121. https://doi.org/10.1007/s10846-019-00980-9

Ye, J. (2008). Adaptive control of nonlinear PID-based analog neural networks for a nonholonomic mobile robot. *Neurocomputing*, *71*(7–9), 1561–1565. https://doi.org/10.1016/j.neucom.2007.04.014

Zhang, L., Liu, L., & Zhang, S. (2020). Design, Implementation, and Validation of Robust Fractional-Order PD Controller for Wheeled Mobile Robot Trajectory Tracking. *Complexity*, *2020*. https://doi.org/10.1155/2020/9523549