Hindawi
Complexity
Volume 2020, Article ID 3067024, 18 pages
https://doi.org/10.1155/2020/3067024

Research Article

A Novel Design of a Neural Network-Based Fractional PID Controller for Mobile Robots Using Hybridized Fruit Fly and Particle Swarm Optimization

Ghusn Abdul Redha Ibraheem,[1] Ahmad Taher Azar,[2,3] Ibraheem Kasim Ibraheem,[1] and Amjad J. Humaidi[4]

[1]*Department of Electrical Engineering, College of Engineering, University of Baghdad, Al-Jadriyah, Baghdad 10001, Iraq*
[2]*Robotics and Internet of Things Lab (RIOTU), Prince Sultan University, Riyadh, Saudi Arabia*
[3]*Faculty of Computers and Artificial Intelligence, Benha University, Banha, Egypt*
[4]*Department of Control and Systems Engineering, University of Technology, Baghdad 10001, Iraq*

Correspondence should be addressed to Ahmad Taher Azar; ahmad_t_azar@ieee.org

Received 7 October 2019; Revised 16 February 2020; Accepted 3 March 2020; Published 29 April 2020

Academic Editor: Hassan Zargarzadeh

The design of a swarm optimization-based fractional control for engineering application is an active research topic in the optimization analysis. This work offers the analysis, design, and simulation of a new neural network- (NN) based nonlinear fractional control structure. With suitable arrangements of the hidden layer neurons using nonlinear and linear activation functions in the hidden and output layers, respectively, and with appropriate connection weights between different hidden layer neurons, a new class of nonlinear neural fractional-order proportional integral derivative (NNFOPID) controller is proposed and designed. It is obtained by approximating the fractional derivative and integral actions of the FOPID controller and applied to the motion control of nonholonomic differential drive mobile robot (DDMR). The proposed NNFOPID controller's parameters consist of derivative, integral, and proportional gains in addition to fractional integral and fractional derivative orders. The tuning of these parameters makes the design of such a controller much more difficult than the classical PID one. To tackle this problem, a new swarm optimization algorithm, namely, MAPSO-EFFO algorithm, has been proposed by hybridization of the modified adaptive particle swarm optimization (MAPSO) and the enhanced fruit fly optimization (EFFO) to tune the parameters of the NNFOPID controller. Firstly, we developed a modified adaptive particle swarm optimization (MAPSO) algorithm by adding an initial run phase with a massive number of particles. Secondly, the conventional fruit fly optimization (FFO) algorithm has been modified by increasing the randomness in the initialization values of the algorithm to cover wider searching space and then implementing a variable searching radius during the update phase by starting with a large radius which decreases gradually during the searching phase. The tuning of the parameters of the proposed NNFOPID controller is carried out by reducing the MS error of 0.000059, whereas the MSE of the nonlinear neural system (NNPID) is equivalent to 0.00079. The NNFOPID controller also decreased control signals that drive DDMR motors by approximately 45 percent compared to NNPID and thus reduced energy consumption in circular trajectories. The numerical simulations revealed the excellent performance of the designed NNFOPID controller by comparing its performance with that of nonlinear neural (NNPID) controllers on the trajectory tracking of the DDMR with different trajectories as study cases.

1. Introduction

Mobile robots serve platforms with huge versatility within their environment; they are not limited to one location because they can be pushed autonomously in their own circumference. In other words, it has the capability of implementing tasks without assistance from external operators [1]. Mobile robots are unique to move freely in a predefined workplace to accomplish the preferred objectives. This skill of mobility makes the mobile robots appropriate for vast application fields in unstructured and structured surroundings. The ground mobile robot can be categorized into a wheeled mobile robot (WMR) and legged mobile robot (LMR). The WMRs are prevalent because they are tailored to particular applications with reasonably small mechanical complexities and power drain [2].

Within the last decades, many different control structures were introduced into industrial societies to handle the restrictions of the classical controllers. The PID controller which has been dominated by the industrial organizations has been changed using the concept of differentiators and integrators of fractional power. It was shown that a combination of further degrees of freedom with differentiators and integrators of fractional power provided a greater degree of flexibility and performance that would otherwise be hard (even impossible) to come by the conventional PID controllers [3, 4]. The fractional-order PID (FOPID) controllers are the generalization of widely applicable PID controllers; in recent years, they have drawn much attention from both academics and industry [3–5]. Fractional controls are less sensitive to parameter changes in a controlled system. A fractional control unit can easily achieve the isodamping property. On the other hand, incorporating an integral action in the feedback loop has the advantage of eliminating steady-state errors on account of reducing relative stability of the system. It can be concluded that by designing more general controller laws in the form $(1/s^n)$, (s^n), $n \in \mathbf{R}^+$, the feedback system with more favorable solutions between undesirable and constructive consequences of the above scenario could be attained and by combining these control actions.

Tuning of a fractional PID controller is difficult as five parameters have to be tuned, which means two more parameters compared to a traditional PID controller [3–5]. Some methods were proposed for the proper choice of the parameters' values of the PID controller. The method of Ziegler–Nichols tuning strategy was acquainted in 1942 for the parameters regulation of the PID controller coefficients; this tuning technique is utilized if the model of the system is a first order plus dead time.

In recent years, methods of optimization which are theoretically different from classical optimization have been invented. They depend on specific properties and behavior of organic herd of birds and nature-inspired and neurobiological systems. These metaheuristic procedures have been developed in the last decade and are evolving as common methods for solving numerical optimization and intricate industrial case studies. Particle swarm optimization (PSO) is a metaheuristic optimization method which depends on the motion and intellect of bird's colony behavior or fish bevy schooling. Kennedy and Eberhart initially suggested particle swarm optimization (PSO) technique in 1995 [6]. Advantages of PSO are as follows: (1) it does not need the derivative of the cost function, (2) it can be parallelized, and (3) it has fast convergence behavior.

On the other hand, the fruit fly optimization (FFO) swarm technique is one of the state-of-the-art evolutionary computation techniques based on the foraging behavior of fruit flies which was pointed out by Pan [7]. The olfactory organ of a fruit fly can collect different smells from the air and even locate the source of the food from a distance of 40 km. Subsequently, the fruit flies travel to the source of the food and use their acute visionary system to locate the food destination (minimum or maximum of the function) where their companions form a swarm and then travel in that direction. The FFO algorithm seems to be an excellent optimization algorithm; it has numerous benefits such as speed to acquire solutions, the simplicity of its structure, and ease of implementation. So, FFO was effectively used and applied in a diverse class of applications [7–9].

However, FFO algorithm suffers from some shortcomings. Firstly, there is inadequacy in the FFO algorithm concerning the searching policy, a necessary step to yield new solutions of the FFO algorithm using random information of the previous solutions. Moreover, the FFO algorithm has weak exploration ability, low convergence precision, and jumps out of the local minimum. Finally, the candidate solutions cannot be generated in a uniform manner in the domain. On the other hand, PSO experiences the premature convergence, a common phenomenon in the evolutionary methods in very sophisticated applications such as path planning and motion control of mobile robots. Also, it relies on user experience to find the optimum values of some parameters like the inertia weights and social and cognitive coefficients. Moreover, standard swarm optimization algorithms do not find the optimum solutions in a rational time [9]. Therefore, the structure of the FFO and PSO algorithms requires further improvements for attaining the optimum solutions to the real-world applications.

The motivation for the hybridization between the MAPSO and EFFO algorithms is an attempt to combine the beneficial features of MAPSO and EFFO algorithms and conduct a sequential operation for these two optimization algorithms over the progression of the process. Moreover, the hybridization between MAPSO and EFFO algorithms will overcome the limitations of the individual MAPSO and EFFO algorithms mentioned above. This hybridization will be accomplished as described later in this paper.

Many researchers have conducted research studies on motion control problem of DDMR under nonholonomic constraints, and so various kinds of controllers were demonstrated in the literature for the mobile robots to track specific trajectories. Trajectory tracking of wheeled mobile robots using hybrid visual servo equipped with onboard vision systems is described in [10]. In [11], the authors addressed the output feedback trajectory tracking problem for a nonholonomic wheeled mobile robot in the presence of parameter uncertainty, exogenous disturbances, and without

velocity measurements using fuzzy logic techniques. The work in [12] focused on the localization, kinematics, and closed-loop motion control for a DDMR. The authors of [13] developed an online nonlinear optimal tracking control method for unmanned ground systems by firstly establishing the nonlinear tracking error model for unmanned ground systems (UGSs), and then the tracking control problem for UGS was converted to a continuous nonlinear optimal control problem with the help of a symplectic pseudospectral method based on the third kind of generation function. In [14], the authors proposed a kinematic-based neural network controller for nonlinear control of the DDMR with nonholonomic constraints. In [15], an iterative learning control over a wireless network for a class of unicycle type mobile robot systems is proposed, and the study included the channel noise effect and the robustness analysis of the proposed system. In [16], a sliding mode-based asymptotically stabilizing controller law has been proposed for a mobile robot. A dynamic prediction-based model predictive control method is offered in [17] for wheeled mobile robots taking into account the tire sideslip. Fuzzy based controllers for autonomous mobile robots have been argued in [18, 19], where the work in [19] dealt with unstructured environments. The work in [20] proposed a disturbance observer based on biologically inspired integral sliding mode control for trajectory tracking of mobile robots. A time-optimal velocity tracking controller for DDMR is presented in [21]. The authors in [22, 23] investigated a model predictive control (MPC) for differential drive mobile robots. Backstepping nonlinear control has been investigated on DDMR in [24]. Recently, researchers are applying a new control paradigm named active disturbance rejection control (ADRC) [25–30] on a wide range of applications [31, 32] and particularly on DDMR [33]. The authors in the literature proposed many algorithms for tuning parameters of the FOPID controller with different applications, where [34, 35] used the genetic algorithms and [36–39] utilized PSO algorithm. Others like Rajasekhar et al. [40] applied the gravitational search optimization technique based on the Cauchy and Gaussian mutation, and El-Khazali [41] exploited the artificial bee colony algorithm. Frequency-domain methods for the design of the FOPID controllers can be found in [42]. Finally, other algorithms like GA can be used to tune the FOPID controller and more complex controllers like [43].

The contributions in this research work lie in twofold:

(1) Development of a MAPSO-EFFO algorithm: developing a modified adaptive particle swarm optimization (MAPSO) algorithm by adding an initial run phase with a massive number of particles. At the end of this initial running point, the smaller group of these fitness particles will be selected to continue with an adaptive PSO (APSO) algorithm. Moreover, the conventional fruit fly optimization (FFO) algorithm has been modified by increasing the randomness in the initialization values of the algorithm to cover wider searching space and then implementing a variable searching radius during the update phase by starting with a large radius which decreases gradually during the searching phase. Finally, adopting a hybridized MAPSO-EFFO algorithm by the serial blending of the MAPSO algorithm with EFFO one, i.e., the input to the EFFO algorithm is the output of the MAPSO. The hybridized MAPSO-EFFO technique is used for the evaluation of the parameters of the NNFOPID.

(2) New nonlinear fractional control structure: a new NNFOPID controller is proposed in this paper which employs the structure of the neural networks (NNs). With suitable arrangements of the hidden layer neurons using sigmoid nonlinear activation and linear functions in the hidden and output layers, respectively, and with appropriate connection weights between different neurons in different layers, a new class of nonlinear neural FOPID controller is obtained by approximating the fractional derivative and integral actions of the FOPID controller. The outputs of the neural networks are the control actions used to drive the motors of the DDMR.

The paper is organized in the following structure. Section 2 gives the motivation and problem statement. Section 3 presents the fractional calculus and the theoretical background of the fractional PID controllers. The kinematic model of the DDMR is introduced in Section 4. The proposed nonlinear neural conventional and fractional PID controllers for the trajectory tracking of the DDMR are explained in Section 5. Section 6 discusses the results and simulations of the designed motion controllers for the DDMR based on different trajectories. Finally, the conclusions are given in Section 7.

2. Problem Statement

Given a nonholonomic differential drive mobile robot (DDMR) following a particular path, the tracking error occurs because of many factors like noise, disturbances, slippage, and the errors measured from sensors due to both interior and exterior causes. These issues also make the mobile robot has the difficulty to turn left or right by direction set or by using various sensors. Therefore, the DDMR kinematic model has been employed in this paper to synthesize neural network fractional-order PID (NNFOPID) controllers to regulate its speed so that it would track the required path in the plane as fast as possible with minimum mean square tracking error; this is called trajectory tracking problem. Three FOPID controllers will be designed to control the position (x and y) of the DDMR in the 2D plane and its orientation θ. Moreover, the aim of the proposed tracking FOPID controllers is reducing the energy consumed by the left and right motors of the DDMR. Given a reference path that needs to be followed by the DDMR, which consists of a set of positions in the 2D plane together with the orientation, i.e., xr, yr, and θr, the actual path consists of a set of positions x, y, and θ. Then, it is required to design a NNFOPID controller to generate the control velocities of the kinematic model of the DDMR such that the mean square error between xr and x, yr and y, and θr and θ

is minimum with minimum peak of the left and right velocities of the DDMR. In contrast to the FOPID controller, the NNFOPID controller has more capability to capture the nonlinearity of the DDMR model due to the nature of the neural network structure employed in the design where nonlinear activation functions are used with hidden layer that has adaptable parameters (NNFOPID controller parameters).

3. Fractional Control Analysis

Fractional calculus is a part of the mathematical analysis, which demonstrates the likelihood of the differential operator orders to be the complex or real number for the differentiation and integration. Generally, the form of the fractional-order operation represented by ${}_aD_b^\alpha$ is called as *differintegral operator*. The sign of α controls the action of differintegral (${}_aD_b^\alpha$) whether to be an identity operator, an integrator, or a differentiator.

The fractional integral and derivative using Grünwald–Letnikov (GL) definition follow the same procedures based on the multiderivative integer calculus. The general GL definition is stated as [42, 44]

$$ {}_a^{\mathrm{GL}}D_b^\alpha f(b) = \lim_{h\longrightarrow 0} \frac{1}{h^\alpha} \sum_{j=0}^{[(b-a)/h]} (-1)^j \binom{\alpha}{j} f(b-jh), \quad (1) $$

where $((b-a)/h)$ refers to the integer part, a and b are the start and final limit values, and h is the sampling time. The utilization of GL of (1) in the computation of the output response of any fractional-order system can be illustrated as follows. Given any fractional system expressed by the fractional-order linear constant coefficients differential equation as [44],

$$ \begin{aligned} &a_n D^{\propto_n} y(t) + a_{n-1} D^{\propto_{n-1}} y(t) + \cdots + a_0 D^{\propto_0} y(t) \\ &\quad = b_m D^{\delta_m} u(t) + b_{m-1} D^{\delta_{m-1}} u(t) + \cdots + b_0 D^{\delta_0} u(t), \end{aligned} \quad (2) $$

where $D^{\propto} = {}_0D_b^\alpha$; $a_i\ (i=0,\ldots,n)$ $b_i\ (i=0,\ldots,m)$ are constant and $\propto_n (i=0,\ldots,n)$ and $\delta_i\ (i=0,\ldots,m)$ are real numbers. Without loss of generality, the parameters $\propto$'s and δ's might be $\propto_n > \propto_{n-1} > \cdots > \propto_0$, and $\delta_m > \delta_{m-1} > \cdots > \delta_0$. Consider (2) with its right-hand side equal to $u(t)$ such that

$$ a_n D^{\propto_n} y(t) + a_{n-1} D^{\propto_{n-1}} y(t) + \cdots + a_0 D^{\propto_0} y(t) = u(t). \quad (3) $$

Recall GL definition in (1); then, by substituting (1) into (3), the numerical solution of (3) can be evaluated as [44]

$$ y_b = \frac{1}{\sum_{i=0}^{n} (a_i/h^{\alpha_i})} \left[u_b - \sum_{i=0}^{n} \frac{a_i}{h^{\alpha_i}} \sum_{j=1}^{[(b-a)/h]} w_j^{\alpha_i} y_{b-jh} \right], \quad (4) $$

and w_j^α can be evaluated in recursive manner as follows:

$$ \begin{aligned} &w_0^\alpha = 1, \\ &w_j^\alpha = \left(1 - \frac{\alpha+1}{j}\right) w_{j-1}^\alpha, \quad j = 1, 2, \ldots, \frac{(b-a)}{h}. \end{aligned} \quad (5) $$

Now reconsider (2), where its right-hand side is equal to $\breve{u}(t)$:

$$ \breve{u}(t) = b_m D^{\delta_m} u(t) + b_{m-1} D^{\delta_{m-1}} u(t) + \cdots + b_0 D^{\delta_0} u(t). \quad (6) $$

Thus, $\breve{u}(t)$ may be calculated firstly using (1), and then the output response due to $\breve{u}(t)$ is computed from the solution of (6) as

$$ y = \sum_{j=1}^{(b-a)/h} \left(\sum_{i=0}^{m} \frac{b_i}{h^{\delta_i}} * w_j^{\delta_i} \right) y_{b+h-jh}. \quad (7) $$

The FOPID controller increases the efficiency and the possibility of better system performance because of its five parameters. The differential equation of the FOPID controller with fractional power denoted as $\mathrm{PI}^\lambda\mathrm{D}^\alpha$ is described by

$$ u(t) = k_p e(t) + k_i D^{-\lambda} e(t) + k_d D^\alpha e(t). \quad (8) $$

Taking Laplace transform to (8), we have

$$ U(s) = k_p e(s) + k_i \frac{e(s)}{s^\lambda} + k_d s^\alpha e(s), \quad (9) $$

where k_d, k_p, and k_i are derivative, proportional, and integral control parameters, respectively, λ is the order of the fractional integral, and α is the fractional derivative order. It is obvious that the FOPID controller has the three standard coefficients k_d, k_i, and k_p in addition to parameters λ and α, which are fractional powers for derivative and integral actions, respectively. The values of α and λ are nonintegers with the restriction of being positive real numbers.

Discretization methods of continuous-time fractional operators have been studied widely by many researchers [45, 46]. The fundamental principle to discretize a continuous fractional-order operator s^α $(\alpha \in R)$ is to define it by what is called as the generating function $s = \omega\ (z^{-1})$. Examples of such transformations are Euler, Tustin, and Simpson transformations. A more recent transformation formula is found as a weighted interpolation between the Euler and Tustin [45]. Usually, the aforementioned transformation schemes lead to a nonrational polynomial in z. To get a rational polynomial, one may find the power series expansion (PSE) of $s = \omega\ (z^{-1})$ and then truncate the z-polynomial function (in the form of finite impulse response (FIR) filters) to compute the final approximation. The Tustin method is applied since it is more accurate compared to other transforms such as backward and forward difference as given below:

$$ s^\alpha = \left(\omega\left(z^{-1}\right)\right)^\alpha = \left(\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}\right)^\alpha. \quad (10) $$

Based on the above analysis, the nonlinear control law that drives wheels of the DDMR will be derived in detail in the next sections. Before that, a concise review of DDMR modeling will be developed.

4. Kinematic Modeling of Nonholonomic DDMR

The position of the DDMR in the world coordinates axis $\{o, x, y\}$ is illustrated in Figure 1. The kinematics model of the DDMR as shown in the figure consists of a castor wheel in the head of the cart and two driving wheels attached on one axis located at the back. The motion and the orientation of DDMR are achieved via two DC motors which form the actuators of the right and left wheels. Table 1 lists the parameters that have been used in the derivation of the DDMR kinematics.

The motion of the DDMR can be determined by the linear velocities of the right and left wheels, V_{right} and V_{left}. The angular and linear velocities of the DDMR, V_A and V_{Li}, can be described by $V_{\text{left}}(t)$ and $V_{\text{right}}(t)$ as follows [2]:

$$V_A(t) = \frac{V_{\text{left}}(t) - V_{\text{right}}(t)}{D}, \tag{11}$$

$$V_{\text{Li}}(t) = \frac{V_{\text{left}}(t) + V_{\text{right}}(t)}{2}. \tag{12}$$

The kinematics equation of the DDMR in the world frame is derived as

$$\dot{x}(t) = V_{\text{Li}}(t)\cos\theta(t), \tag{13}$$

$$\dot{y}(t) = V_{\text{Li}}(t)\sin\theta(t), \tag{14}$$

$$\dot{\theta}(t) = V_A(t). \tag{15}$$

Substituting (11) and (12) in (13)–(15), and integrating them, we get the solutions, $x(t)$, $y(t)$, and $\theta(t)$. Moreover, considering a sample interval Δt and a zero-order hold , we get the discrete representation of $x(t)$, $y(t)$, and $\theta(t)$ as follows [38]:

$$x(k) = 0.5\left[V_{\text{left}}(k) + V_{\text{right}}(k)\right]\cos\theta(k)\Delta t + x(k-1), \tag{16}$$

$$y(k) = 0.5\left[V_{\text{left}}(k) + V_{\text{right}}(k)\right]\sin\theta(k)\Delta t + y(k-1), \tag{17}$$

$$\theta(k) = \frac{1}{D}\left[V_{\text{left}}(k) - V_{\text{right}}(k)\right]\Delta t + \theta(k-1), \tag{18}$$

where $x(k)$, $y(k)$, $\theta(k)$ are the position components at the kth step of the motion and Δt is the interval between two adjacent samples. Equations (16)–(18) are used in the design of the NNFOPID controller. It should be remembered that the coordinates of the DDMR given in Figure 1 are in the global frame (world frame) and can be transformed into the local coordinate by the rotation matrix given below:

$$R = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{19}$$

The configuration error signals are acquired by this matrix to transform the DDMR from local coordinate to the global one.

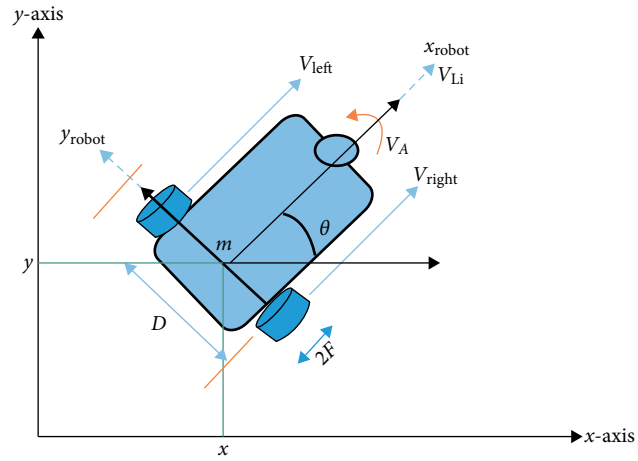

Figure 1: DDMR position representation in the world frame.

Table 1: Parameters of the DDMR model kinematics.

Symbol	Definition
m	Center point between the two rear wheels of DDMR
D	Distance between the DDMR back wheels (m)
r	Wheel radius of DDMR (m)
$V_{\text{Li}}(t)$	DDMR linear velocity (m/sec)
$V_A(t)$	DDMR angular velocity (rad/sec)
$V_{\text{left}}(t)$	Left wheel linear velocity (m/sec)
$V_{\text{right}}(t)$	Right wheel linear velocity (m/sec)
θ	Orientation of the mobile robot from the x-axis

Assumption (H1). The DDMR moves to any location in the free workplace assuming that the wheels of DDMR are ideally set up such that they have no slipping with ideal rolling.

5. The Proposed Swarm-Based Nonlinear Neural FOPID (NNFOPID) Trajectory Controller for DDMR

The two main objectives of DDMR Trajectory controller are to establish a control strategy that prevents the DDMR from drifting out of the desired track and keep its movement smoothly within a minimum error with stability. The proposed configuration of the DDMR motion control in this research consists of two main parts as explained in the next sections.

5.1. Nonlinear Neural FOPID (NNFOPID) Kinematic Trajectory Controller Structure. Figure 2 illustrates the structure of the NNFOPID-based kinematic trajectory controller for the DDMR. The suggested trajectory tracking control for the DDMR in this work is the NNFOPID. Mathematically, it can be derived as follows. Inspired by the work in [47] and starting from (10) and assuming $\omega = z^{-1}$, the PSE of the right-hand side of (10) is given as

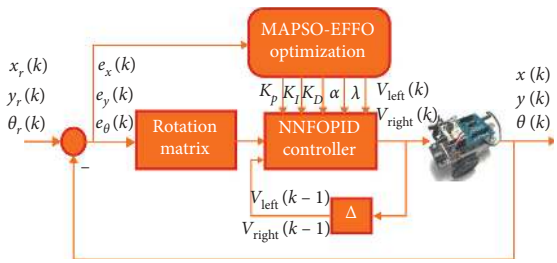

Figure 2: The NNFOPID-based trajectory tracking controller for DDMR.

$$s^{\alpha} = \left(\omega\left(z^{-1}\right)\right)^{\alpha} = \left(\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}\right)^{\alpha} = \left(\frac{2}{T}\right)^{\alpha}\sum_{i=0}^{\infty} g_i(\alpha)\omega^i, \quad |\omega| < 1. \tag{20}$$

It is mapped inside the unit circle in the z-plane, with region of convergence (ROC) $|z| > 1$, i.e.,

$$s^{\alpha} = \left(\frac{2}{T}\right)^{\alpha}\sum_{i=0}^{\infty} g_i(\alpha) z^{-i}, \quad |z| > 1. \tag{21}$$

$g(\alpha)$'s coefficients are calculated from the following equation:

$$g_i(\alpha) = \frac{1}{i!}\cdot\frac{\mathrm{d}^i}{\mathrm{d}\omega^i}\left(\frac{1-\omega}{1+\omega}\right)^{\alpha}\Bigg|_{\omega=0}. \tag{22}$$

The integral term $s^{-\lambda}$ of (9) can be treated as follows:

$$s^{-\lambda} = \frac{1}{s}\, s^{1-\lambda}. \tag{23}$$

Applying Tustin transformation of (10) on (23), we get

$$\begin{aligned} s^{-\lambda} &= \frac{T}{2}\frac{\left(1+z^{-1}\right)}{\left(1-z^{-1}\right)}\cdot\left(\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}\right)^{1-\lambda} \\ &= \left(\frac{2}{T}\right)^{-\lambda}\frac{\left(1+z^{-1}\right)}{\left(1-z^{-1}\right)}\cdot\sum_{i=0}^{\infty} g_i(1-\lambda)\, z^{-k}, \quad |z| > 1, \end{aligned} \tag{24}$$

where $g(1-\lambda)$ can be calculated using (22). Now, substituting (21) and (24) into (9) yields the general discrete-type PID controller with a long memory:

$$\begin{aligned} C(z) = K_P &+ K_D\sum_{i=0}^{\infty} g_i(\alpha)\, z^{-i} \\ &+ K_I\frac{\left(1+z^{-1}\right)}{\left(1-z^{-1}\right)}\cdot\sum_{i=0}^{\infty} g_i(1-\lambda)\, z^{-i}, \end{aligned} \tag{25}$$

where $K_P = k_p$, $K_D = (2/T)^{\alpha}k_d$, and $K_I = (2/T)^{-\lambda}k_i$. The upper limit in the of the sum in (25) cannot be considered for practical limitations. So, we calculate the sum up to L; then,

$$\begin{aligned} C(z) = K_P &+ K_D\sum_{i=0}^{L} g_i(\alpha) z^{-k} \\ &+ K_I\frac{\left(1+z^{-1}\right)}{\left(1-z^{-1}\right)}\cdot\sum_{i=0}^{L} g_i(1-\lambda) z^{-i}. \end{aligned} \tag{26}$$

Multiplying both sides of (26) by $(1-z^{-1})$ and remembering that $U(z) = C(z)\cdot E(z)$,

$$\begin{aligned} \left(1-z^{-1}\right)U(z) = &\left(1-z^{-1}\right)K_P E(z) + \left(1-z^{-1}\right)K_D\, E(z) \\ &\cdot\sum_{i=0}^{L} g_i(\alpha)\, z^{-k} + K_I\left(1+z^{-1}\right) \\ &\cdot E(z)\sum_{i=0}^{L} g_i(1-\lambda)\, z^{-k}. \end{aligned} \tag{27}$$

Then, the general difference equation relating $e(k)$ to $U(k)$ looks like

$$\begin{aligned} U(k) = U(k-1) &+ K_P\left(e(k) - e(k-1)\right) \\ &+ K_D\sum_{i=0}^{L} g_i(\alpha)\left(e(k-i) - e(k-i-1)\right) \\ &+ K_I\cdot\sum_{i=0}^{L} g_i(1-\lambda)\,\left(e(k-i) + e(k-i-1)\right). \end{aligned} \tag{28}$$

Equation (28) describes the control signal for the discrete-time FOPID controller. Then, the suggested feedback control law of the NNFOPID trajectory tracking controller for the DDMR is given as

$$\begin{aligned} V_{\text{left}}(k) &= U_1(k) = O_x(k) + O_y(k) + V_{\text{left}}(k-1), \\ V_{\text{right}}(k) &= U_2(k) = O_\theta(k) + O_y(k) + V_{\text{right}}(k-1), \end{aligned} \tag{29}$$

where $V_{\text{right}}(k-1)$ and $V_{\text{left}}(k-1)$ are the delayed signals of the control signals $V_{\text{right}}(k)$ and $V_{\text{left}}(k)$. The signals $O_x(k), O_y(k)$, and $O_\theta(k)$ correspond to the neural network's outputs of the hidden layer neurons with the sigmoid signal as an activation function and are described as

$$O_\delta(k) = \frac{2}{1+e^{-\text{net}_\delta(k)}} - 1, \tag{30}$$

for $\delta = x, y$, and θ, and $\text{net}_\delta(k)$ is given as

$$\begin{aligned} \text{net}_\delta(k) = &K_{P_\delta}\left(e_\delta(k) - e_\delta(k-1)\right) \\ &+ K_{D_\delta}\sum_{i=0}^{L} g_i(\alpha)\,\left(e_\delta(k-i) - e_\delta(k-i-1)\right) \\ &+ K_{I_\delta}\cdot\sum_{i=0}^{L} g_i(1-\lambda)\left(e_\delta(k-i) + e_\delta(k-i-1)\right), \end{aligned} \tag{31}$$

for $\delta = x, y$, and θ. Figure 3 portrays the proposed NNFOPID controller for the DDMR used in the block diagram of Figure 2.

The proposed NN structure is a multilayer perceptron (MLP) which consists of three layers. The first layer is the input layer which sums the error input signals and its past values. The second layer is the hidden layer; it consists of three neurons with nonlinear activation function; they are there to implement the three FOPID controllers needed for the x, y, and θ subsystems of the mobile robot. The

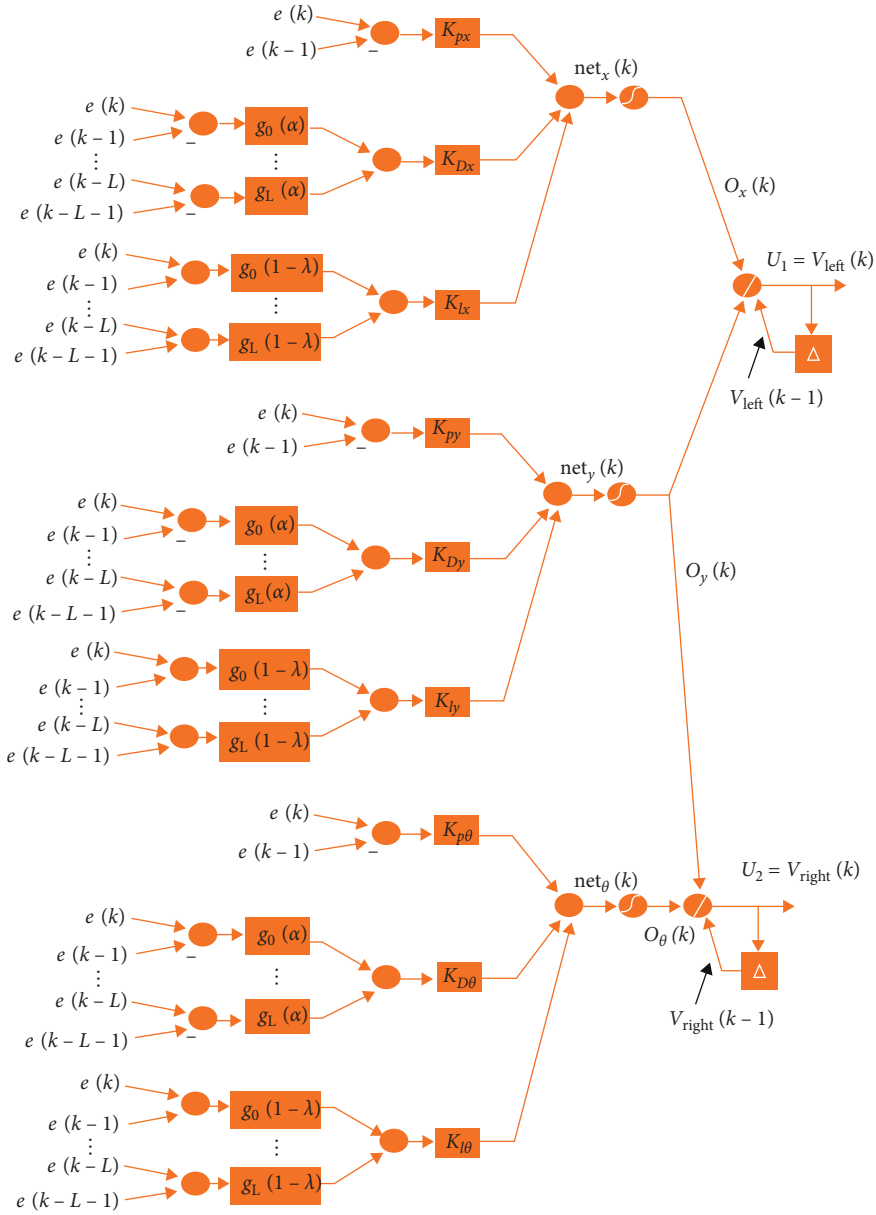

Figure 3: The proposed structure of the NNFOPID controller for DDMR.

connection weights for this hidden layer are the coefficients of the FOPID controllers, i.e., K_{P_δ}, K_{I_δ}, and K_{D_δ}, and $\delta = x, y,$ and θ. Finally, the third layer is the output layer; it consists of a single neuron with linear activation function; it adds up the three outputs of the hidden layer. The fractional derivatives are translated into a weighted sum of the past and current values of the input error signal $e_\delta(k)$ for $\delta = x, y,$ and θ as indicated by (31). So, the tuning input vector of the NNFOPID controller consists of $e_\delta(k), e_\delta(k-1)$, $e_\delta(k-2), \ldots, e_\delta(k-L-1)$, for $\delta = x, y,$ and θ. The activation function of the output layer in Figure 3 is linear. Its proposed structure is different from the NN-based FOPID controllers proposed in [48, 49], where in these works, the outputs of the neural networks are the parameters of the FOPID controller themselves, while in the proposed scheme, the structure of the neural network itself has been used as a nonlinear fractional-order PID controller where the coefficients of the fractional-order PID controller themselves are the connection weights of the NN structure, and the output of the neural networks are the control actions used to drive the motors of the DDMR. The NNFOPID neural controller parameters K_D, K_I, $K_P, \alpha,$ and λ are adjusted using the proposed MAPSO-EFFO described next.

5.2. Hybridized MAPSO-EFFO-Based Tuning Algorithm. PSO and FFO algorithms are the metaheuristic procedures which have been invented in the last decade and are evolving as common procedures for solving numerical optimization and industrial case studies. They require only function evaluations (and not the integral or derivative values). The next section presents the modified PSO and FFO algorithms,

namely, the MAPSO and EFFO and the hybridization between them to produce a hybridized algorithm with fast convergence behavior. This is to overcome the limitations of the individual PSO and FFO algorithms like premature convergence, user dependence on setting parameter values, and exploration ability. In the following section, the individual algorithms used in this work are discussed, and then the hybridization between them is introduced.

5.2.1. Modified Adaptive PSO (MAPSO) Algorithm. In this work, improvement is added to increase the speed and efficiency of the algorithm where the optimal solutions are found by spreading a large amount of population size (from $6M$ to $10M$, where M is the standard population size and lies between 20 and 30 particles) in the search space. Then, choosing the best (20–30 particles) from all these populations as initial populations, the best particles are the ones that have the minimum of the following fitness function:

$$\begin{aligned}\mathrm{MSE} = \frac{1}{Q}\sum_{k=1}^{Q}\left(xr(k)-x(k)\right)^2 + \left(yr(k)-y(k)\right)^2 \\ +\left(\theta r(k)-\theta(k)\right)^2,\end{aligned} \tag{32}$$

$$\begin{aligned}\mathrm{MSE} = 0.5 * \frac{1}{Q}\sum_{k=1}^{Q}\left(xr(k)-x(k)\right)^2 + \left(yr(k)-y(k)\right)^2 \\ +\left(\theta r(k)-\theta(k)\right)^2 + 0.5 * \left[U_1^2(k)+U_2^2(k)\right],\end{aligned} \tag{33}$$

where Q is the maximum number of samples, k is the current sample, xr, yr, and θr are reference values, and U_1 and U_2 are the control signals. Moreover, the social and cognitive parameters c_1 and c_2 are considered variable through the iterative process, in contrast to being constant in the standard PSO. Figure 4 illustrates this modification. The pseudocode for the calculations of the values of the optimal coefficients of the NNFOPID controller using the MAPSO is listed in Algorithm 1.

5.2.2. Enhanced Fruit Fly Optimization (EFFO) Algorithm. The process of finding food by fruit fly can be explained as follows. First, fruit flies smell the food location using the organ osphresis and then fly towards the food source. When fruit flies become in the neighboring of the food source, they use the visual sense to locate the food and the next location for the fruit flies swarm. Finally, it flies towards the food direction. The standard FFO has been explained in detail in the literature [7], and some improvements have been added in this paper to the standard EFFO as follows:

(a) Assign a range of values as an upper and lower limits for the initialization of the decision variables.

(b) A dynamic change of the search radius with some iteration has been proposed for the standard FFO algorithm to improve its performance and exclude the drawbacks of the fixed value searching radius (see Figure 5). The modification to the standard FFOA is proposed by adding an inertia weight as follows:

$$w(i) = \frac{w_{\max} - w_{\min}}{\sqrt{i}} + w_{\min}, \tag{34}$$

where w is the weight of the inertia used to adjust the radius of spread and $\delta = x,\ y,$ and θ. The pseudocode for the calculation of the optimal values of the NNFOPID controller's coefficients using the EFFO algorithm is listed in Algorithm 2.

5.3. Hybridized Swarm Optimization Algorithm: The MAPSO-EFFO Algorithm. In this paper, a hybridization between MAPSO and EFFO is proposed. The first phase utilizes MAPSO algorithm, where a particle swarm spreads in different orientations and the global best solution is obtained. The second phase includes the EFFO for updating the preceding best locations of the swarm particles. By taking advantages of the MAPSO and EFFO, the hybridized MAPSO-EFFO algorithm is illustrated in the following pseudocode (Algorithm 3).

The advantage of the proposed hybridized MAPSO-EFFO algorithm will overcome the limitations of the individual particle swarm optimization (PSO) and fruit fly optimization (FFO) algorithms like premature convergence, user dependence on setting parameter values, and exploration ability. The salient features of the hybridized MAPSO-EFFO algorithm are the simplicity of its structure and ease of implementation. Knowing that the hybridized MAPSO-EEFO algorithm does not need the derivative of the cost function and since the MAPSO and EFFO algorithms will run sequentially, the proposed hybridized MAPSO-EEFO algorithm can be parallelized, which results in fast convergence behavior. Finally, the proposed hybridized algorithm will not rely on user experience to find the optimum values of some parameters like the inertia weights and social and cognitive coefficients.

6. Simulation Results and Discussion

In this section, the simulation results of applying the NNFOPID as a kinematic controller to the DDMR given in (11)–(18) to track a certain trajectory will be presented.

6.1. Simulation Considerations. Various examinations have been done with the recommended NNFOPID-based kinematic controller for the DDMR tuned by the proposed hybridized MAPSO-EFFO algorithm. These examinations are built on the parameter values of the Eddie DDMR model [50]: $D = 0.452$ m, $r = 0.076$ m, and the sampling time is equal to 0.5 seconds.

6.2. Parameter Settings for Hybridized MAPSO-EFFO Algorithm. For simulations, comparisons between the nonlinear neural PID (NNPID) controller and nonlinear

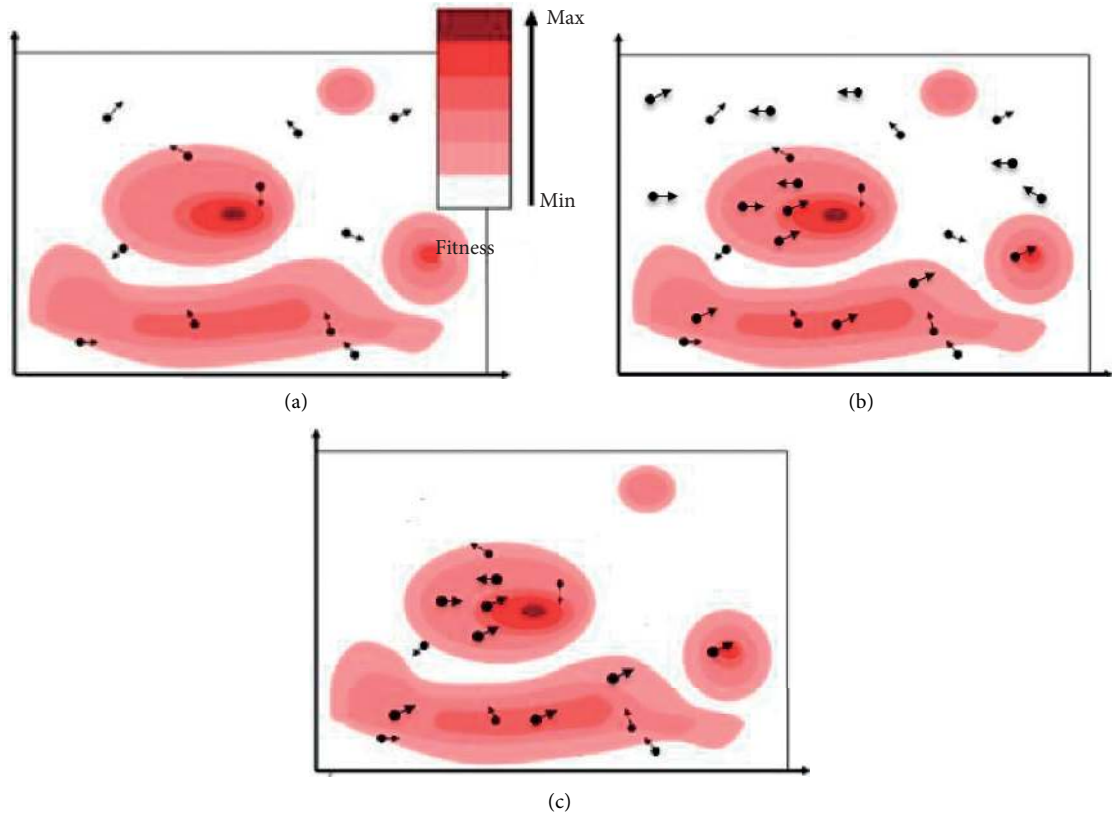

Figure 4: The proposed modification of MAPSO algorithm. (a) Standard PSO algorithm. (b) MAPSO. (c) After selecting the best pop. of the MAPSO.

(1) Initialize the swarm in the range of about ($6M$–$10M$) particles.
(2) Choose the best M swarm and randomly assign the initial location and velocities and other related parameters. Randomly assign the position values of PSO particles with zero velocities and initialize the constants $i_{\max}$, $w_{\max}$, $w_{\min}$, $c_{1\max}$, $c_{1\min}$, $c_{2\max}$, $c_{2\min}$. Set $i = 1$ and go to the step (3)
(3) **For** $j = 1, 2, \ldots, M$
Calculate the fitness function of the particle j, *i.e.*, MSE(j) using (32) or (33) to each particle j,
set local best cost = current MSE
Local best position = current position
End
Set global best MSE = min (for all local best MSE)
(4) $w(i) = w_{\max} - ((w_{\max} - w_{\min})/i_{\max})i$, $c_1(i) = c_{1\min} + ((c_{1\max} - c_{1\min})/i_{\max})i$, $c_2(i) = c_{2\min} + ((c_{2\max} - c_{2\min})/i_{\max})i$
For $\delta = x, y, \theta$
For $X = K_P, K_I, K_D, \alpha, \lambda$
For $j = 1, 2, \ldots, M$
$V_{X_{\delta,j}}(i) = w(i)V_{X_{\delta,j}}(i-1) + c_1(i) \cdot R_1[P_{\text{Best}\delta,j} - X_{\delta,j}(i-1)] + c_2(i) \cdot R_2[G_{\text{Best}} - X_{\delta,j}(i-1)]$, $X_{\delta,j}(i) = X_{\delta,j}(i-1) + V_{X_{\delta,j}}(i)$
Calculate the MSE of (32) or (33)
End
End
End
(5) Set $i = i + 1$ and go to step 4 until either iteration i reaches $i_{\max}$ or convergence is achieved.

Algorithm 1: Tuning of the optimal NNFOPID parameters for DDMR using MAPSO.

neural fractional-order PID (NNFOPID) controller are made considering the MAPSO-EFFO as a tuning algorithm for both controllers. The simulation parameters are the first population size = 150 particles, the second population size = 25 particles, $c_{1\min} = c_{2\min} = 0.25$, $c_{1\max} = c_{2\max} = 0.85$, $w_{\min} = 0.2$, $w_{\max} = 0.9$, the maximum generation size = 25 gen for EFFO, $w_{\min} = 0.1$, $w_{\max} = 0.9$, and iteration number = 100 iteration.

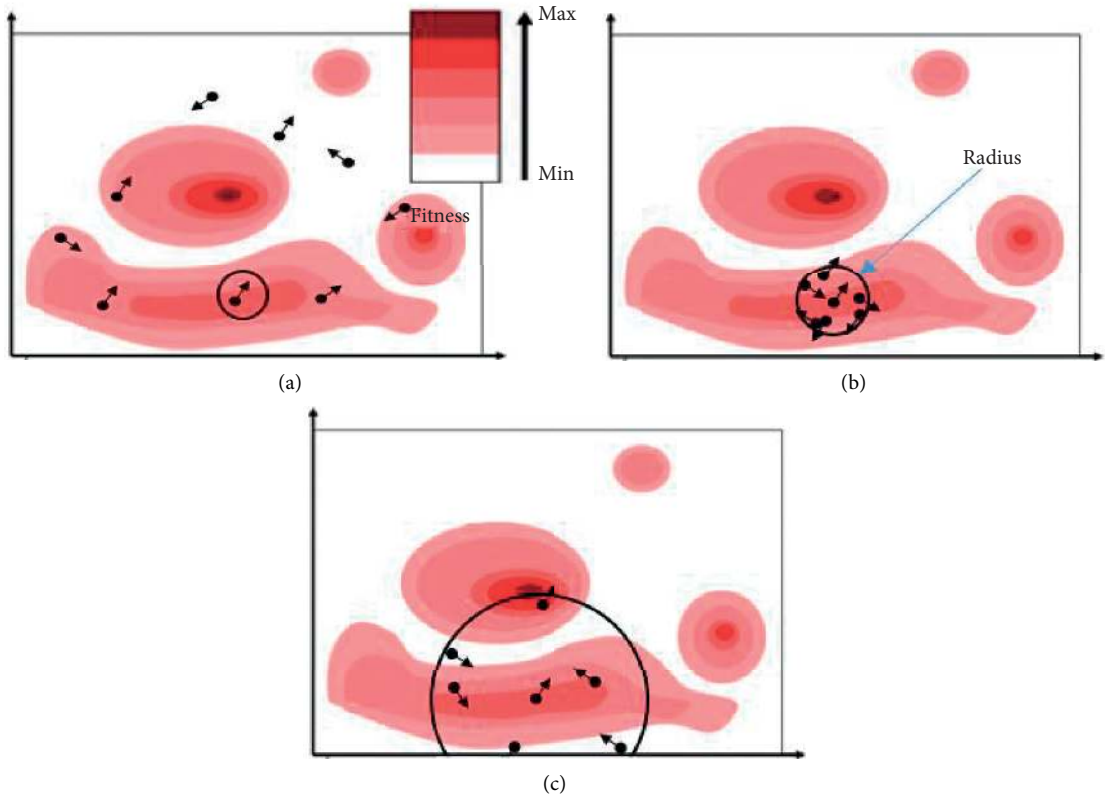

Figure 5: The proposed enhancement of EFFO algorithm. (a) Initial FFO. (b) Standard FFO. (c) Enhanced FFO (EFFO).

(1) Initialization phase.
Assign random initializations with upper and lower limits and initialize other algorithm parameters.
$X\_\text{axis} = (\mathcal{U} + (\mathcal{U} - \mathcal{L}) * \mathcal{R}) * \mathcal{R}$
$Y\_\text{axis} = (U + (U - L) * \mathcal{R}) * \mathcal{R}$
where X_axis is the initial value of the fruit fly position in the x-direction and $\mathcal{L}$ and $\mathcal{U}$ are the lower and upper limits of X-position, respectively, $\mathcal{R} \in [0, 1]$ is a random value, Y_axis is the initial value of Y in the y-direction, and $\mathcal{L}$ and $\mathcal{U}$ are the upper and lower limits of Y, respectively. Define $i_{\max}$, Smell_best = 1. Define the number of fruit flies N, $w_{\max}, w_{\min}$. Set iteration $i = 1$.
(2) Osphresis searching process
Every individual searches about food in all directions randomly around the initial locations of the previous step using the osphresis organ to generate the next population.
For $\delta = x, y, \theta$
For $L = K_P, K_I, K_D, \alpha, \lambda,$
While $i < i_{\max}$
For $j = 1, 2, \ldots, N$
$w_{L_{\delta,j}}(i) = (w_{\max} - w_{\min})/\sqrt{i} + w_{\min}$
$X_{L_{\delta,j}}(i) = X\_\text{axis} + \mathcal{R} * w_{L_{\delta,j}}(i)$
$Y_{L_{\delta,j}}(i) = Y\_\text{axis} + \mathcal{R} * w_{L_{\delta,j}}(i)$
(3) Path construction phase.
$\text{Dis}_{L\delta,j}(i) = \sqrt{X_{L_{\delta,j}}(i)^2 + Y_{L_{\delta,j}}(i)^2}$
$L_{\delta,j}(i) = 1/\text{Dis}_{L\delta,j}(i)$
(4) Fitness calculation phase.
Calculate the fitness function by the following equations, where (32) or (33) are calculated with the updated values of the NNFOPID parameters $L_{\delta,j}(i)$, i.e.,
$\text{Sme}_{\delta L,j}(i) = \text{MSE}(L_{\delta,j}(i))$
End For
$[\text{best\_smell}, \text{best\_index}] = \min(\text{Sme}_{L\delta}(i))$
where $\text{Sme}_{L\delta}(i)$ is the vector of the concentration of smell for the fruit fly swarm, best_smell represents the smallest elements, best_index are its indices along the different dimensions of smell vectors, and min $(\text{Sme}_{L\delta}(i))$ is the minimum concentration of smell in the fly swarm.

Algorithm 2: Continued.

(5) Vision searching process
The fruit flies keep the best value of smell concentration and will use visionary sense to fly in the direction of that location according to the subsequent equations,
If best_smell < Smell_best
Smell_best = best_smell
X_axis = X [best_index]
Y_axis = Y [best_index]
End if
$i = i + 1$
End while
End For
End For

Algorithm 2: Tuning of the optimal NNFOPID parameters for DDMR using EFFO.

(1) Initialize $i_{\max}$
(2) Start MAPSO algorithm and initialize its parameters (Algorithm 1).
(3) Obtain the G_{best} position from the MAPSO for each of the NNFOPID controller parameters.
(4) Start EFFO algorithm and calculate the initial parameter values using
$X\text{\_axis}_{L_{\delta,j}} = \sqrt{G_{\text{best}}\,\text{position}^2/2}$
$Y\text{\_axis}_{L_{\delta,j}} = \sqrt{G_{\text{best}}\,\text{position}^2/2}$
(5) Follow the steps of the EFFO algorithm (Algorithm 2).
(6) Set $i = i + 1$ and go to step 5 until either iteration i reaches $i_{\max}$ or convergence is achieved.

Algorithm 3: Tuning of the optimal NNFOPID parameters for DDMR using MAPSO-EFFO

6.3. Trajectory Tracking Results

6.3.1. Study Case 1 (Circular Trajectory).

$$y_d(t) = \sin\left(\frac{t}{10}\right),$$
$$\theta_d(t) = \frac{\pi}{2} + \frac{t}{10}. \tag{35}$$

The DDMR starts from the initial posture $P(0) = [2.1, -0.1, \pi/2]$.

The actual initial posture of the DDMR is $P_d(0) = [2, 0, \pi/2]$.

The effectiveness of the proposed NNFOPID and NNPID controllers is validated in the case of tracking an orbicular trajectory. The proposed controllers for the DDMR give an excellent trajectory tracking based on hybrid MAPSO-EFFO algorithm as shown in Figure 6. The results of the simulations prove the high efficiency of the offered NNFOPID controller based on hybrid MAPSO-EFFO algorithm to yield bounded and smooth left and right wheels' velocities. When both the proposed NNFOPID and NNPID controllers are tuned using the proposed hybrid MAPSO-EFFO algorithm, the proposed NNFOPID controller presents a superior results by eliminating the overshoots in the wheel's velocities as compared to the NNPID one, this is evident from Figure 7. The plot in Figure 8 illustrates the left and right wheels' velocities of (control action) of DDMR with orbicular trajectory when $0.5 * \text{MSE} + 0.5 * (U12 + U22)$ performance index is used.

By comparing Figures 7 and 8, the difference between the two fulfillment indices (MSE and $0.5 * \text{MSE} + 0.5 * ({U_1}^2 + {U_2}^2)$) can be found. The mean velocity of the left wheel is 0.07729 and 0.07218 m/sec and the mean velocity of the right wheel is 0.1219 and 0.1165 m/sec for NNFOPID and NNPID, respectively. The maximum right and left wheels' velocities are 0.1463 m/sec and 0.08375 m/sec and 0.1983 m/sec and 0.1783 m/sec with NNFOPID and NNPID, respectively. Obviously, by adding the control action term in the performance index, it decreases the velocity amplitude for both wheels.

Figure 9 shows the convergence of the posture path and orientation motion MSEs for the DDMR model (ex, ey, $e\theta$) which in this case are (0.0000732, 0.00017, 0.000349), respectively, for the NNFOPID controller, while the MSE (ex, ey, $e\theta$) is (0.000226, 0.000371, 0.00019), respectively, for the NNPID controller.

6.3.2. Study Case 2 (Lemniscates Trajectory). The required lemniscates trajectory is drawn using the following equations:

$$x_d(t) = -0.5 * \sin\left(\frac{2\pi t}{30}\right),$$
$$y_d(t) = 0.5 * \sin\left(\frac{2\pi t}{20}\right),$$
$$\theta_d(t) = 2\tan^{-1}\left(\frac{\Delta y_d(t)}{\sqrt{(\Delta x_d(t))^2 + (\Delta y_d(tt))^2} + \Delta x_d(t)}\right). \tag{36}$$

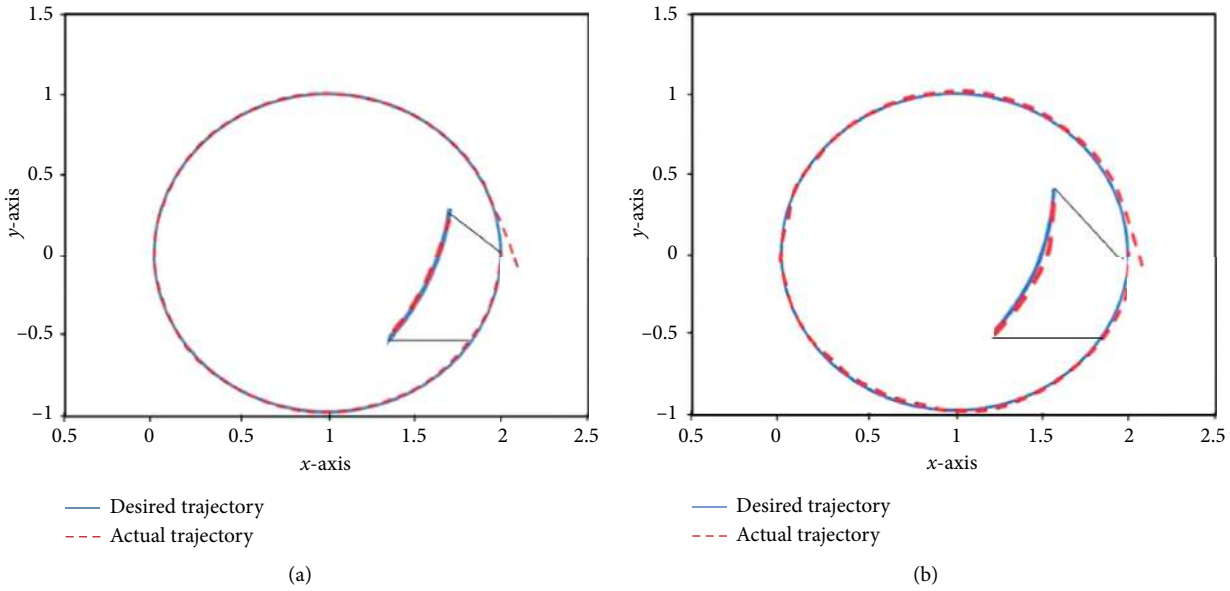

Figure 6: The desired and actual orbicular trajectory of DDMR. (a) Trajectory tracking with NNFOPID controller and (b) trajectory tracking with NNPID controller.

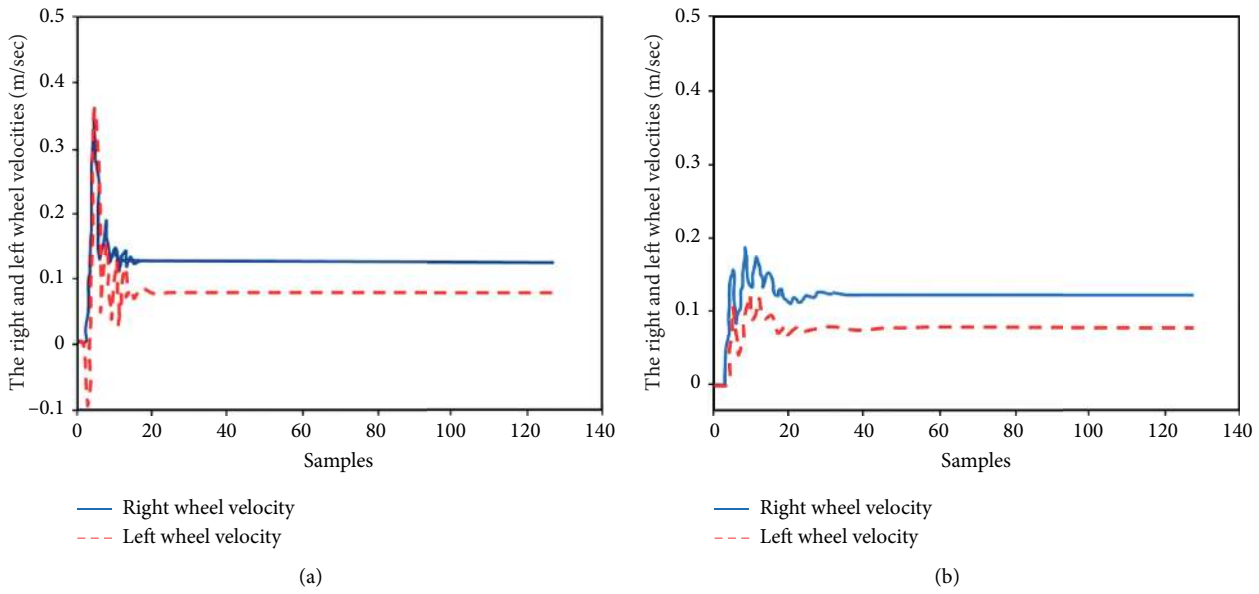

Figure 7: The left and right wheels' velocity of DDMR for circular trajectory. (a) Wheels' velocity with NNPID and (b) wheels' velocity with NNFOPID.

The DDMR moves from the first posture $P\ (0) = [2.1, -0.1, \pi/2]$.

The actual initial posture of the DDMR is $P_d\ (0) = [2, 0, \pi/2]$.

The efficiency of the NNFOPID controller is verified to track a lemniscates trajectory. Compared with the same case study by using an NNPID controller, the DDMR trajectory tracking obtained by the NNFOPID proposed controllers is an excellent tracking based on crossbreed MAPSO-EFFO as shown in Figure 10(a). The corresponding results using an NNPID controller are shown in Figure 10(b). The simulation results demonstrate the high success of the offered NNFOPID controllers based on hybrid MAPSO-EFFO algorithm by showing its capability to give bounded and flat velocities for the left and right wheels. Also, it dismisses all the unexpected overshoots as compared to NNPID controller with the same optimization algorithm and the same trajectory (see Figure 11). Figure 12 interprets the convergence of the posture path and orientation MSE for the DDMR model. The motion MSE (ex, ey, $e\theta$) values are (0.000049, 0.00001732, 0.0000283), respectively, for the NNFOPID controller, while the MSE (ex, ey, $e\theta$) values are

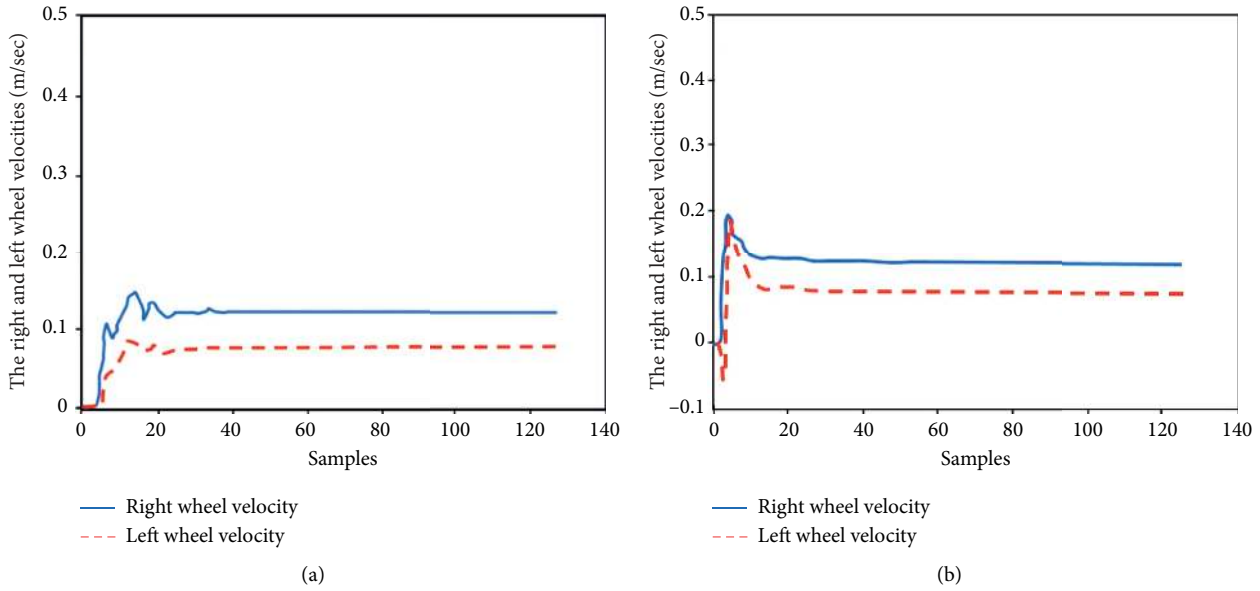

Figure 8: The left and right wheels' velocity of DDMR for circular trajectory tracking adopting $0.5 * \text{MSE} + 0.5 * (U12 + U22)$ index. (a) The left and right wheels with NNFOPID and (b) the left and right wheels with NNPID.

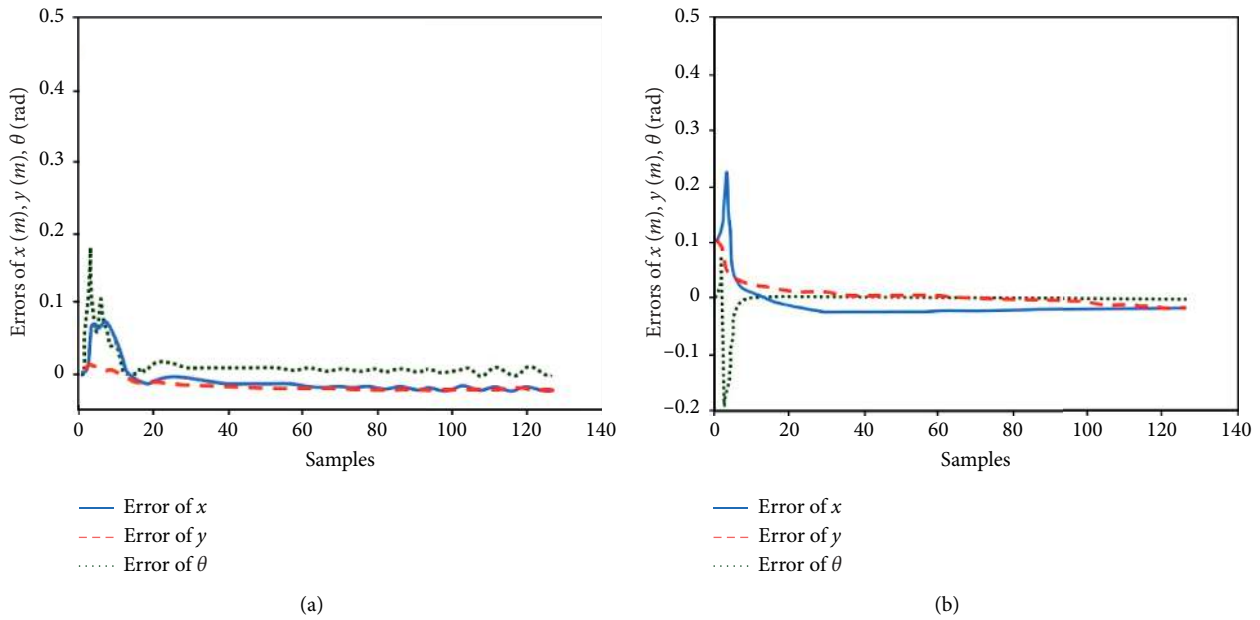

Figure 9: The error of x and y coordinates and orientation θ of DDMR in orbicular trajectory with (a) NNFOPID controller and (b) NNPID controller.

(0.000072, 0.000051, 0.00016), respectively using the NNPID controller.

6.3.3. Study Case 3 (Line Trajectory). The following equations give the required line trajectory:

$$\begin{aligned} x_d(t) &= t * 0.1, \\ y_d(t) &= t * 0.1, \\ \theta_d(t) &= \frac{\pi}{4}. \end{aligned} \tag{37}$$

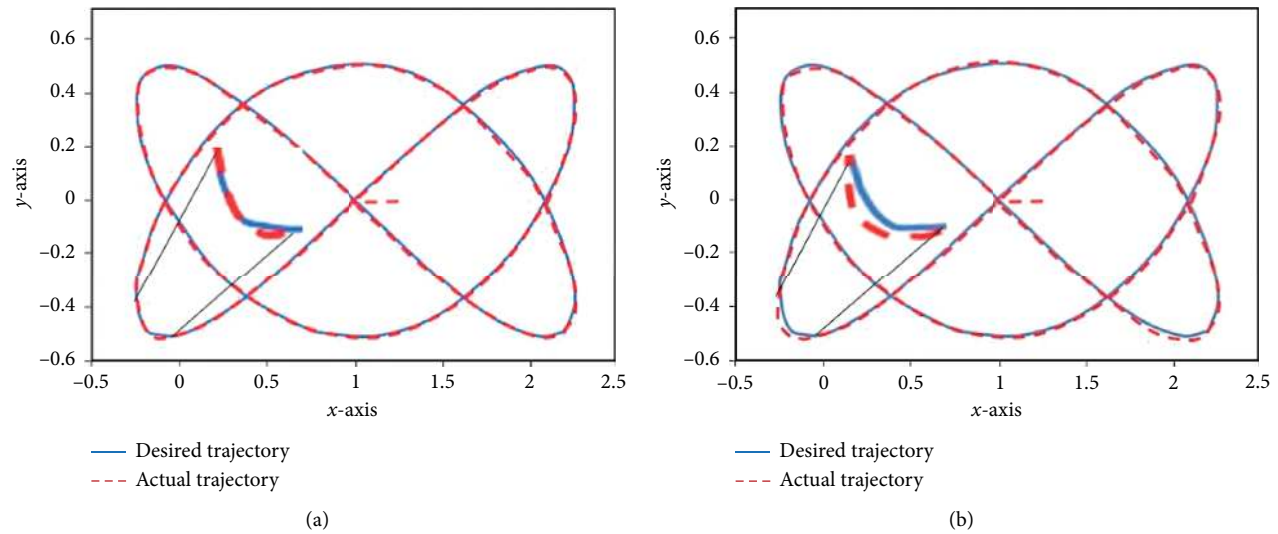

Figure 10: Desired and actual lemniscates trajectory of DDMR: (a) with NNFOPID controller and (b) with NNPID controller.

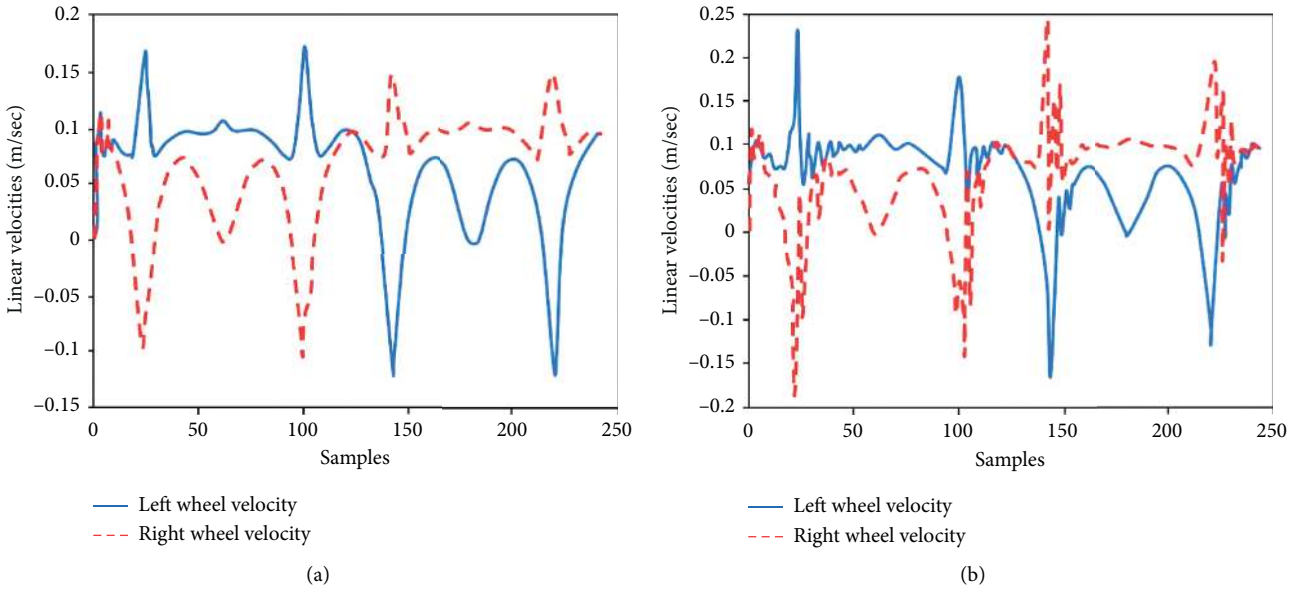

Figure 11: The left and right wheels' velocity of DDMR for lemniscates trajectory with (a) NNFOPID controller and (b) NNPID controller.

The DDMR starts from the initial posture $P(0) = [0, -0.2, \pi/2]$.

The actual initial posture of the DDMR is $P_d(0) = [0, 0, \pi/4]$.

A graphical comparison between the the proposed NNFOPID controller with the NNPID one is depicted in Figure 13 for the line trajectory; evidently, the proposed NNFOPID controller shows excellent performance in tracking the desired trajectory compared to the NNPID controller. Figure 14 illustrates the convergence curve of the MAPSO, EFFO, and MAPSO-EFFO algorithms for the line trajectory. As can be seen, the MAPSO-EFFO outperforms the two other remaining algorithms in terms of time and MSE. This is also evident from Tables 2–4, where they summarize the hybridized MAPSO-EFFO, MAPSO, and EFFO algorithms, respectively, used to obtain the optimum controller's parameters after 100 iterations. These tables show the MSE performance index for the NNPID controller which is higher than its corresponding index adopted by the NNFOPID controller in the three simulation case studies. Table 5 shows the MSE of the error of x and y coordinates and error of θ orientation for the NNPID and NNFOPID controllers. Table 6 tabulates the parameters of the NNFOPID controller for three case studies with the hybridized MAPSO-EFFO tuning algorithm.

6.4. Discussion. The simulations results and a comparison analysis between the NNPID and NNFOPID showed that the NNFOPID is a perfect trajectory tracking kinematic controller because of its flexibility and capability that stem

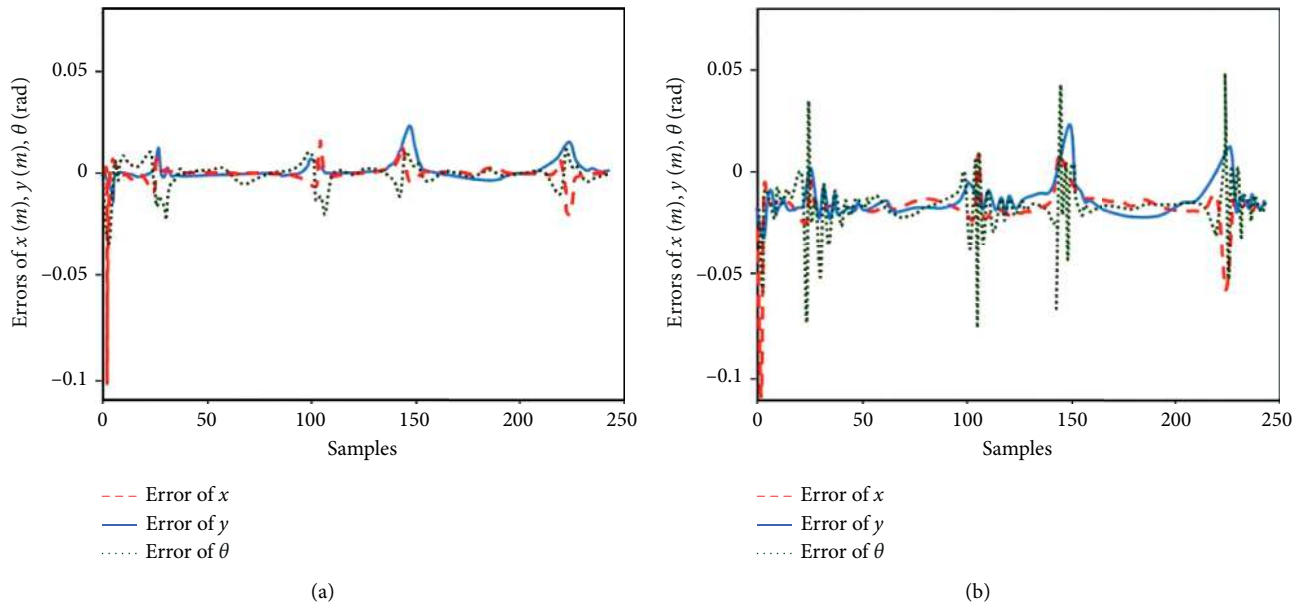

Figure 12: The error of x and y coordinates and orientation θ of DDMR for lemniscates trajectory with (a) NNFOPID controller and (b) NNPID controller.

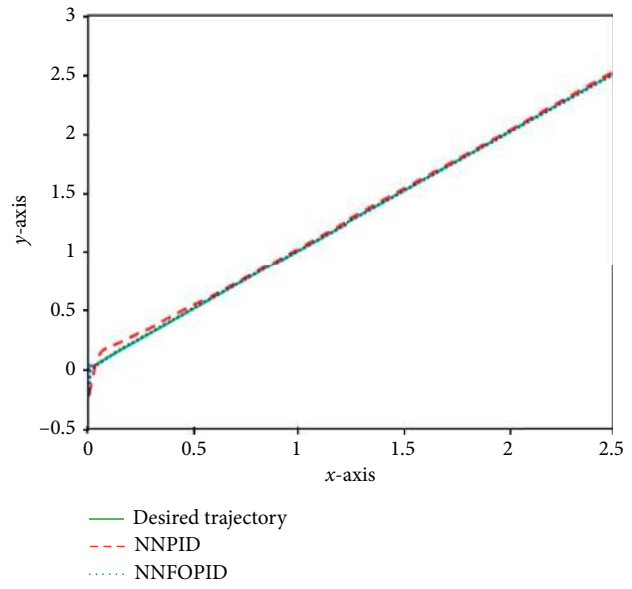

Figure 13: Desired and actual trajectories for the line trajectory.

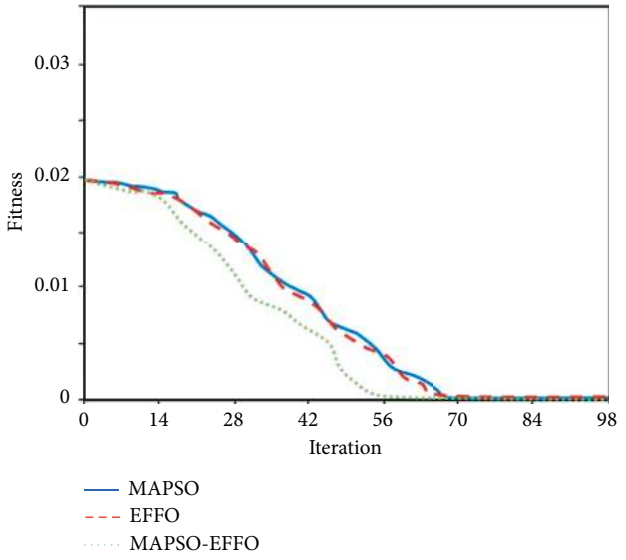

Figure 14: Convergence curve of the MAPSO, EFFO, and MAPSO-EFFO algorithms.

from the fractional power of the derivative and integral actions. The following are the advantages of the NNFOPID controller over the kinematic NNPID one:

(i) The ability to reduce the tracking MSEs for the DDMR model after 100 iterations (MSE is 0.000059 for the circular path, is 0.00073 for line path, and is 0.00009 for lemniscate path) to follow a desired continuous track.

(ii) Competence of breeding soft and perfect proper velocity control actions without sharp upswings (the linear velocity is 0.1 m/s, and the angular velocity is 0.1 rad/s for orbicular path). The linear velocity is 0.18 m/s, and the angular velocity is 0 rad/s for line path with a linear velocity of 0.04 m/s, and the angular velocity is ±0.3 rad/s for lemniscates path.

(iii) The results of the three case studies' simulation elucidate the ability of all searching methods (MAPSO, EFFO and mixture MAPSO-EFFO) to get the best parameters for the NNPID and NNFOPID kinematic controllers in spite of all the difficult challenges to control the DDMR by three

Table 2: The MSE for NNPID and NNFOPID controllers with MAPSO-EFFO.

Controller type	MSE		
	Case 1	Case 2	Case 3
NNPID	0.00079	0.0012	0.00028
NNFOPID	0.000059	0.00073	0.00009

Table 3: MSE for NNPID and NNFOPID controllers with MAPSO.

Controller type	MSE		
	Case 1	Case 2	Case 3
NNPID	0.00094	0.00134	0.00031
NNFOPID	0.00082	0.00082	0.00016

Table 4: The MSE for NNPID and NNFOPID controllers with EFFO.

Controller type	MSE		
	Case 1	Case 2	Case 3
NNPID	0.0008	0.0013	0.0003
NNFOPID	0.00071	0.0008	0.00012

Table 5: The MSE of the position and orientation using MAPSO-EFFO algorithm.

Controller type		Case studies		
		Case 1	Case 2	Case 3
NNPID	x	0.000226	0.000237	0.000072
	y	0.000372	0.00234	0.000051
	θ	0.000193	0.000563	0.00016
NNFOPID	x	0.0000598	0.000232	0.0000543
	y	0.000182	0.0000357	0.00002283
	θ	0.000474	0.00053	0.0000433

Table 6: The control parameters of NNFOPID controller by MAPSOA-EFFO.

Control parameter	Case1	Case2	Case3
K_{Px}	0.7213	0.2770	0.7706
K_{Ix}	1.1883	1.4497	8.6215
K_{Dx}	1.2108	0.5298	3.81077
λ_x	0.8284	0.8389	1.1176
α_x	1.0864	0.6049	0.0623
K_{Py}	0.1204	−0.2191	−1.1972
K_{Iy}	−1.581	−3.6139	0.5654
K_{Dy}	0.5246	−1.3169	−2.7585
λ_y	1.3507	0.3169	0.7094
α_y	1.3507	0.8025	0.0009
$K_{P\theta}$	0.6078	1.07	−3.776
$K_{I\theta}$	1.1686	1.0355	−5.8303
$K_{D\theta}$	0.3714	0.4932	−1.8916
λ_θ	0.644	0.8384	1.3287
α_θ	1.176	0.6879	0.1969

controllers in the same time (one controller for each of x, y, and θ).

(iv) The hybrid MAPSO-EFFO presents higher accuracy to give a minimum MSE, so it better tunes the control parameters of the NNPID and the NNFOPID controllers than other algorithms as noticed by the simulation results (see Tables 2–4). The reasons behind this improvement lie in two main points: firstly, instead of initializing the controllers' parameters within a space of particles of size 150, the MAPSO algorithm starts from a refined particles' space of size M which represents the best M particles of the original space. The second reason is that in the EFFO algorithm, two significant modifications have been added to the algorithm to speed up the convergence. The first modification is the inertia weight; the main advantage of the inertia weight is to accelerate the convergence rate of the fruit flies. The second modification is the initialization of the EFFO algorithm which starts from the best of the M best particles that the MAPSO has found and considers as the initial positions for the fruit flies.

(v) $0.5 * \mathrm{MSE} + 0.5 * ({U_1}^2 + {U_2}^2)$ yields a less control energy for the DDMR by eliminating the upswings and overshoots of the velocity for the left and right wheels of the DDMR and limiting the values of these control signals. This resulted in a significant advantage in reducing the energy necessary to drive the mobile robot to the required position. In addition, it avoids actuator saturation that protects the drives of the mobile robot from damage.

7. Conclusions

This paper presented a novel nonlinear fractional control structure based on neural networks. Two tuning optimization algorithms are also introduced in the work, namely, MAPSO and EFFO. Furthermore, a hybridization between MAPSO and EFFO algorithms has been proposed to produce a new algorithm, namely, MAPSO-EFFO algorithm. It is employed to optimize controllers' coefficients of the two proposed neural network-based nonlinear PID controllers. i.e., NNPID and NNFOPID controllers. A comprehensive demonstration and assessment between these two controllers through simulations on DDMR with different trajectories as study cases showed the effectiveness and robustness of the suggested NNFOPID controller tuned by MAPSO-EFFO algorithm, which has excellent trajectory tracking, and it has the ability to generate soft and satisfactory linear and angular velocities. It has excellent performance over the NNPID controller. Moreover, the hybridized MAPSO-EFFO results in convergence behavior and enhances the performance of the individual algorithms. It has better performance than the individual MAPSO and EFFO algorithms in tuning the parameters of the NNPID and NNFOPID controllers and avoids the premature convergence of the original MAPSO and EFFO algorithms.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to thank Prince Sultan University, Riyadh, Saudi Arabia, for supporting and funding this work. Special thanks to Robotics and Internet of Things Lab (RIOTU) at Prince Sultan University, Riyadh, Saudi Arabia.

References

[1] N. Nawash, "H-infinity control of an autonomous mobile robot," M.Sc. thesis, Cleveland State University, Cleveland, OH, USA, 2005.

[2] S. G. Tzafestas, *Introduction to Mobile Robot Control*, Elsevier, Amsterdam, Netherlands, 1st edition, 2014.

[3] A. T. Azar, A. G. Radwan, and S. Vaidyanathan, *Mathematical Techniques of Fractional Order Systems*, Elsevier, Amsterdam, Netherlands, 2018, ISBN 9780128135921.

[4] A. T. Azar, S. Vaidyanatha, and A. Ouannas, "Fractional order control and synchronization of chaotic systems," *Studies in Computational Intelligence*, vol. 688, Springer-Verlag, Germany, 2017, ISBN 978-3-319-50248-9.

[5] Y. Luo and Y. Q. Chen, *Fractional Order Motion Controls*, John Wiley & Sons, Hoboken, NJ, USA, 2012.

[6] R. C. Eberhart, J. Kennedy, and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference Neural Networks IV*, pp. 1942–1948, Perth, WA, Australia, November 1995.

[7] W. T. Pan, "A new Fruit Fly Optimization Algorithm: taking the financial distress model as an example," *Knowledge-Based Systems*, vol. 26, pp. 69–74, 2012.

[8] L. Wang, Y. Shi, and S. Liu, "An improved fruit fly optimization algorithm and its application to joint replenishment problems," *Expert Systems with Applications*, vol. 42, no. 9, pp. 4310–4323, 2015.

[9] R. M. R. Allah, "Hybridization of fruit fly optimization algorithm and firefly algorithm for solving nonlinear programming problems," *International Journal of Swarm Intelligence and Evolutionary Computation*, vol. 5, no. 2, pp. 1–10, 2016.

[10] F. Yan, B. Li, W. Shi, and D. Wang, "Hybrid visual Servo trajectory tracking of wheeled mobile robots," *IEEE Access*, vol. 6, pp. 24291–24298, 2018.

[11] S. Peng and W. Shi, "Adaptive fuzzy output feedback control of a nonholonomic wheeled mobile robot," *IEEE Access*, vol. 6, pp. 43414–43424, 2018.

[12] S. K. Malu and J. Majumdar, "Kinematics, localization and control of differential drive mobile robot," *Global Journal of Researches in Engineering: H Robotics & Nano-Tech*, vol. 14, no. 1, 2014.

[13] J. Liu, W. Han, Y. Zhang, Z. Chen, and H. Peng, "Design of an online nonlinear optimal tracking control method for unmanned ground systems," *IEEE Access*, vol. 6, pp. 65429–65438, 2018.

[14] J. Velagic, N. Osmic, and B. Lacevic, "Neural network controller for mobile robot motion control," *World Academy of Science, Engineering and Technology, International Journal of Electrical and Computer Engineering*, vol. 2, no. 11, pp. 166–171, 2008.

[15] L. Huang, Q. Zhang, L. Sun, and Z. Sheng, "Robustness analysis of iterative learning control for a class of mobile robot systems with channel noise," *IEEE Access*, vol. 7, pp. 34711–34718, 2019.

[16] J. H. Lee, C. Lin, H. Lim, and J. M. Lee, "Sliding mode control for trajectory tracking of mobile robot in the RFID sensor space," *International Journal of Control, Automation and Systems*, vol. 7, no. 3, pp. 429–435, 2009.

[17] G. Bai, L. Liu, Y. Meng, W. Luo, Q. Gu, and J. Wang, "Path tracking of wheeled mobile robots based on dynamic prediction model," *IEEE Access*, vol. 7, pp. 39690–39701, 2019.

[18] R. Rashid, I. Elamvazuthi, M. Begam, and M. Arrofiq, "Fuzzy-based navigation and control of a non-holonomic mobile robot," *Journal of Computing*, vol. 2, no. 3, 2010.

[19] B. Kocaturk, "Motion control of wheeled mobile robots," *Interdisciplinary Description of Complex Systems*, vol. 13, no. 1, pp. 41–47, 2015.

[20] X. Yang, P. Wei, Y. Zhang, X. Liu, and L. Yang, "Disturbance observer based on biologically inspired integral sliding mode control for trajectory tracking of mobile robots," *IEEE Access*, vol. 7, pp. 48382–48391, 2019.

[21] H. A. Poonawala and M. W. Spong, "Time-optimal velocity tracking control for differential drive robots," *Automatica*, vol. 85, pp. 153–157, 2017.

[22] K. Worthmann, M. W. Mehrez, M. Zanon, G. K. I. Mann, R. G. Gosine, and M. Diehl, "Regulation of differential drive robots using continuous time MPC without stabilizing constraints or costs," *IFAC-PapersOnLine*, vol. 48, no. 23, pp. 129–135, 2015.

[23] S. Primatesta and B. Bona, "Motion control of mobile robots with particle filter model predictive equilibrium point control," in *Proceedings of the IEEE International Conference on*

Autonomous Robot Systems and Competitions (ICARSC), pp. 11–16, Coimbra, Portugal, April 2017.

[24] M. Kalyoncu and F. Demirbaş, "Differential drive mobile robot trajectory tracking with using pid and kinematic based backstepping controller," *Selcuk University Journal of Engineering, Science and Technology*, vol. 5, no. 1, pp. 1–15, 2017.

[25] J. Han, "From PID to active disturbance rejection control," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 3, pp. 900–906, 2009.

[26] M. Nowicki, R. Madoński, and K. Kozłowski, "First look at conditions on applicability of ADRC," in *Proceedings of the 2015 10th International Workshop on Robot Motion and Control RoMoCo 2015*, pp. 294–299, Poznan, Poland, July 2015.

[27] R. Parvathy and A. E. Daniel, "A survey on active disturbance rejection control," in *Proceedings of the 2013 International Multi-Conference on Automation, Computing, Control, Communication and Compressed Sensing*, pp. 330–335, Kottayam, India, 2013.

[28] W. R. Abdul-adheem and I. K. Ibraheem, "From PID to nonlinear state error feedback controller," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 1, pp. 312–322, 2017.

[29] I. K. Ibraheem and W. R. Abdul-adheem, "On the improved nonlinear tracking differentiator based nonlinear PID controller design," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 10, pp. 234–241, 2016.

[30] W. R. Abdul-adheem and I. K. Ibraheem, "Improved sliding mode nonlinear extended state observer based active disturbance rejection control for uncertain systems with unknown total disturbance," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 12, pp. 80–93, 2016.

[31] F. Al-Kalbani, S. M. Al Hosni, and J. Zhang, "Active Disturbance Rejection Control of a methanol-water separation distillation column," in *Proceedings of the 2015 IEEE 8th GCC Conference and Exhibition*, pp. 1–6, Muscat, Oman, February 2015.

[32] G. Z. Zheng Qing, "On practical applications of active disturbance rejection control," in *Proceedings of the Chinese Control Conference*, pp. 6095–6100, Beijing, China, July 2010.

[33] I. K. Ibraheem and W. R. Abdul-adheem, "An improved active disturbance rejection control for a differential drive mobile robot with mismatched disturbances and uncertainties," 2018, http://arxiv.org/abs/1805.12170v1.

[34] Z. Bingül and . Karahan, "Fractional PID controllers tuned by evolutionary algorithms for robot trajectory control," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 20, pp. 1123–1136, 2012.

[35] S. Padhee, A. Gautam, Y. Singh, and G. Kaur, "A novel evolutionary tuning method for fractional order PID controller," *International Journal of Soft Computing and Engineering*, vol. 3, pp. 1–9, 2011.

[36] H. Ramezanian, S. Balochian, and A. Zare, "Design of optimal fractional-order PID controllers using particle swarm optimization algorithm for automatic voltage regulator (AVR) system," *Journal of Control, Automation and Electrical Systems*, vol. 24, no. 5, pp. 601–611, 2013.

[37] A. Al-Mayyahi, W. Wang, and P. Birch, "Design of fractional-order controller for trajectory tracking control of a non-holonomic autonomous ground vehicle," *Journal of Control, Automation and Electrical Systems*, vol. 27, no. 1, pp. 29–42, 2016.

[38] I. K. Ibraheem and G. A. Ibraheem, "Motion control of an autonomous mobile robot using modified particle swarm optimization based fractional order PID controller," *Engineering And Technology Journal, Part (A)*, vol. 34, no. 13, pp. 2406–2419, 2016.

[39] A. T. Azar and F. E. Serrano, "Fractional order sliding mode PID controller/observer for continuous nonlinear switched systems with PSO parameter tuning," in *The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2018). AMLTA 2018, Advances in Intelligent Systems and Computing*, A. Hassanien, M. Tolba, M. Elhoseny, and M. Mostafa, Eds., Springer, Cham, Switzerland, 2018.

[40] C. Li, N. Zhang, X. Lai, J. Zhou, and Y. Xu, "Design of a fractional-order PID controller for a pumped storage unit using a gravitational search algorithm based on the Cauchy and Gaussian mutation," *Information Sciences*, vol. 396, pp. 162–181, 2017.

[41] A. Rajasekhar, R. K. Jatoth, and A. Abraham, "Design of intelligent PID/PIλDμ speed controller for chopper fed DC motor drive using opposition based artificial bee colony algorithm," *Engineering Applications of Artificial Intelligence*, vol. 29, pp. 13–32, 2014.

[42] R. El-Khazali, "Fractional-order PIλdμ controller design," *Computers & Mathematics with Applications*, vol. 66, no. 5, pp. 639–646, 2013.

[43] I. K. Ibraheem, "Anti-disturbance compensator design for unmanned aerial vehicle," *Journal of Engineering*, vol. 26, no. 1, pp. 86–103, 2020.

[44] Y. Q. Chen, I. Petr, and D. Xue, "Fractional order control-a tutorial," in *Proceedings of the American Control Conference*, pp. 1397–1411, St. Louis, MO, USA, 2009.

[45] M. A. Al-Alaoui, "Al-Alaoui operator and the new transformation polynomials for discretization of analogue systems," *Electrical Engineering*, vol. 90, no. 6, pp. 455–467, 2008.

[46] R. S. Barbosa and J. A. T. Machado, "Implementation of discrete-time fractional- order controllers based on LS approximations," *Acta Polytechnica Hungarica*, vol. 3, no. 4, pp. 5–22, 2006.

[47] L. T. Rasheed and A. S. Al-Araji, "Design of a nonlinear fractional order PID neural controller for mobile robot based on particle swarm optimization," *Journal of Engineering Technology*, vol. 34, no. 12, pp. 2318–2333, 2016.

[48] B. Ou, L. Song, and C. Chang, "Tuning of fractional PID controllers by using radial basis function neural networks," in *Proceedings of the IEEE International Conference on Control and Automation (ICCA)*, pp. 1239–1244, Xiamen, China, June 2010.

[49] S. Wen-juan and T. Wei, "A neural network fractional order PID controller for FOLPD process," in *Proceedings of the 35th Chinese Control Conference*, pp. 10459–10463, Chengdu, China, July 2016.

[50] Internet website, "Robotics with the Eddie Mobile Robot text manual," 2014, http://parallax.com/.