Motion Control of An Autonomous Mobile Robot using Modified Particle Swarm Optimization Based Fractional Order PID Controller

Dr. Ibraheem K. Ibraheem
Electrical Engineering Department, University of Baghdad/Baghdad.
Email:ibraheemki@coeng.uobaghdad.edu.iq
Ghusn A. Ibraheem
Electrical Engineering Department, University of Baghdad/Baghdad.
Email: ghsun_eng@yahoo.com

Received on:5/1/2016 & Accepted on:22/6/2016

ABSTRACT

This paper presents a comparison between two nonlinear PID controllers, the first is the Neural based controller and the second is the nonlinear fractional order PID controller (FOPID) for a trajectory tracking control of a non-holonomic two wheeled mobile robots (2-WMR). A modified particle swarm optimization (MPSO) has been proposed in this work to tune the parameters of the nonlinear FOPID controller to design the controller so that the 2-WMR follows exactly a predefined continuous track. The kinematic model of a differential drive 2-WMR has been derived to simulate the behavior of the 2-WMR and it is used in the design and simulations of the proposed FOPID controller. From simulation and results, it can be seen that the efficiency of the proposed nonlinear FOPID controller outperforms the nonlinear integer order PID controller; this is proved by the minimized tracking error and the speed control signals obtained.

Keywords: Two wheel Mobile robot, trajectory tracking, non-holonomic systems, Factional order $PI^{\lambda}D^{\alpha}$ controller, Particle swarm optimization (PSO).

INTRODUCTION

Motion control means the policy by which the 2-WMR reaches a desired location and the implementation of this strategy. 2-WMR, as the name implies, have the bility to move around, they may walk on the ground, on the surface of bodies of water, under water, and in the air [1]. This is in contrast with fixed base robotic arms that are more common in manufacturing operations such as automobile manufactures, aircraft assemblies, electronic parts assemblies, welding, spray painting and others. An autonomous robot is automated to work without human involvement, and with the help of personified artificial intelligence can achieve and live within its surroundings.

Today's 2-WMR can move around safely in messy surroundings, understand natural speech, recognize real objects, locate themselves, plan paths, and generally think by themselves. Smart 2-WMR design employs the methodologies and technologies of smart, intellectual, and behavior-based control. It must maximize flexibility of performance subject to minimal input dictionary and minimal computational complexity. Based on the aforementioned, a strict controller has been designed to help the 2-WMR achieve its mission successfully [2].

https://doi.org/10.30684/etj.34.13A.4
2412-0758/University of Technology-Iraq, Baghdad, Iraq

PID control on the other hand, is an old strategy to control systems. The industrial control field uses it because of its simplicity of the design and best act. So that, it is worthy to improve their quality of operation. One of the methods that the engineering tries to improve the performance of the PID controller is to use the fractional order $PI^{\lambda}D^{\alpha}$ controller (FOPID), a promotion of traditional (PID) controller where the integral and derivative order are fractional instead of being integer. FOPID has two more parameters α (derivative order) and λ (integral order) which are in fractions; this increases the flexibility of the system and allows a better implementation than a classical PID controller [3]. On the other hand, Particle Swarm Optimization (PSO) is a strong swarm optimization technique based on the motion and intellectual manner of swarm of insects, such as ants, termites, and bees; a flock of birds; or a school of fish. It implements the model of social communication to solve the problem [4]. The best parameters of fractional-Order PID controller consist of proportional gain (*KP*), integral gain (*KI*), derivative gain (*KD*), fractional-order of differentiator α, fractional-order of integrator λ, all of these parameters will be tuned using our proposed modified PSO algorithm. Many algorithms were proposed to control the trajectory-tracking of a mobile robot, such as Neural-Network [5], Back-Stepping [6], Fuzzy Logic Control [7], Sliding Mode Control [8] and Wavelet Network [9].

Fractional Order Controller:

Fractional Calcuius:

Fractional order calculus is an area where the mathematicians deal with integrals and derivatives of non-integer orders. The general form of fractional order calculus represented by ${}_aD_t^{\alpha}$ and it has called as differ-integral operator. It is the combination of integration and differentiation, process commonly applied in fractional calculus [10].

$$ {}_aD_t^{\alpha} f(t) = \frac{d^{\alpha} f(t)}{[d(t-a)]^{\alpha}} \qquad \ldots(1)$$

Where *a* is starting limitation and *t* is final limitation α is the order of differ-integral.

$$ {}_aD_t^{\alpha} = \begin{cases} \frac{d^{\alpha}}{dt^{\alpha}} & if\ \alpha > 0 \\ 1 & if\ \alpha = 0 \\ \int_a^t (dt)^{-\alpha} & if\ \alpha < 0 \end{cases} \qquad \ldots(2)$$

There are several mathematical definitions used for fractional differ-integral and the most commonly used are:

- **Riemann-Liouville definition (RL).**
- **Caputo definition (C)**
- **Grunwald-Letnikov definition (GL).**

The general Riemann-Liouville definition is:

$$ {}_a^{RL}D_x^{\alpha} f(x) = \begin{cases} \frac{1}{\Gamma(-\alpha)} \int_a^x (x-t)^{-\alpha-1}\ f(t)dt & if\ \alpha < 0 \\ \frac{d^{\alpha} f(x)}{dx^{\alpha}} & if\ \alpha \in N \\ \frac{1}{\Gamma(n-\alpha)} \frac{d^n}{dx^n} \left[\int_a^x \frac{f(t)}{(x-t)^{\alpha-n+1}}\ dt \right] & if 0 < n-1 < \alpha < n \\ f(x) & if\ \alpha = 0 \end{cases} \qquad \ldots(3)$$

While that of Caputo definition is:

$$
{}_{a}^{C}D_{x}^{\alpha} f(x) = \begin{cases} \frac{1}{\Gamma(-\alpha)} \int_{a}^{x} (x-t)^{-\alpha-1} f(t)dt & if\ \alpha < 0 \\ \frac{d^{\alpha} f(x)}{dx^{\alpha}} & if\ \alpha \in N \\ \frac{1}{\Gamma(n-\alpha)} \left[\int_{a}^{x} \frac{f^{n}(\tau)}{(x-\tau)^{\alpha-n+1}} d\tau \right] n = min\{k \in N : k > \alpha\}, & if \alpha > 0 \\ f(t) & if\ \alpha = 0 \end{cases} \quad \ldots (4)
$$

The most important formula used is the Grunwald-Letnikov defined as:

$$
{}_{a}^{GL}D_{t}^{\alpha} f(t) = lim_{h \to 0} \frac{1}{h^{\alpha}} \sum_{j=0}^{\left[\frac{t-a}{h}\right]} (-1)^{j} \binom{\alpha}{j} f(t - jh) \quad \ldots (5)
$$

Fractional Order PID Controller (FOPID):

The fractional differential equation of the $PI^{\lambda}D^{\alpha}$ controller is described by:

$$
U(t) = KPe(t) + KID^{-\lambda}e(t) + KD\ D^{\alpha}e(t) \quad \ldots (6)
$$

The continuous transfer function of FOPID is obtained through Laplace transform, which is given by

$$
U(s) = KPe(s) + KI\frac{e(s)}{s^{\lambda}} + KDs^{\alpha}e(s) \quad \ldots (7)
$$

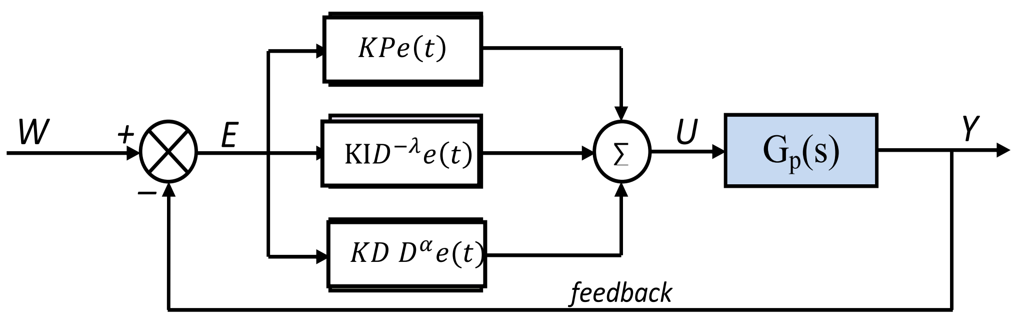


Figure (1)General closed-loop feedback control system with a FOPID controller.

Figure (1) illustrate the closed-loop feedback control system with FOPID. It is clear that the FOPID controller has the three parameters *KP*, *KD* and *KI*. In addition, two other parameters α and λ are needed for the derivative and integral actions respectively. The values of α and λ may be not integers with the restriction of being the positive real number. Figure (2) represent a graphical depiction for the possibilities of control using FOPID controller, the figure extends the four control possibilities of the conventional PID controller (P,PI,PD, and PID) to an area of controls in the first quadrant defined by selection of the values of α and λ equal to one[11].

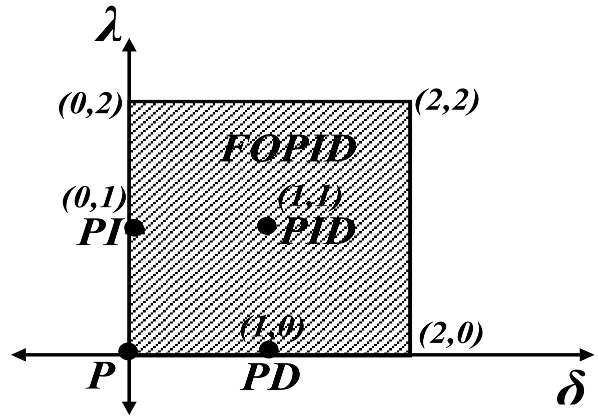


Figure (2)Generalization of the FOPID Controller: From point to an area.

Figure (3) proposed the structure of the nonlinear FOPID neural controller in this work

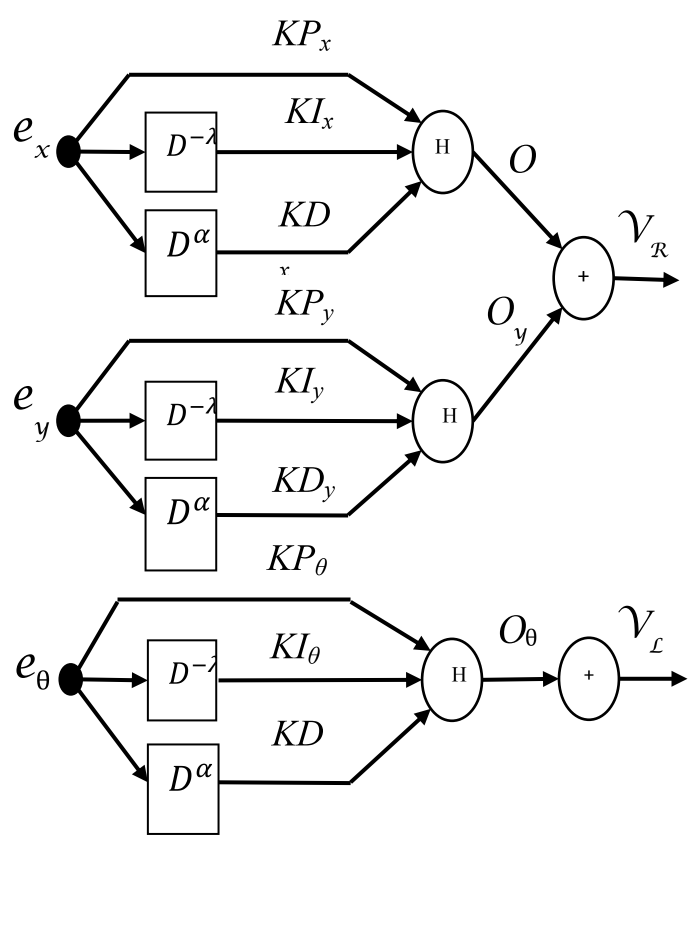


Figure (3) the nonlinear FOPID neural controller design

Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is an optimization technique based on the motion and intellectual of behavior of a colony of ants, termites, and bees. Kennedy and Eberhart originally proposed the PSO algorithm in 1995; it has been found that it is strong in solving an optimization continuous and discrete nonlinear engineering problems. Initially, each particle in the swarm has a position in solution space and it has different dimension depending on variables of the objective function. Generally, the method begins by spreading the particles randomly in the solution space and finally gathering these particles toward the best fitting solution. More specifically, each particle has position and velocity and tries to set it flying in the direction of a possible attractive region according to its own flying experience, then share social information among a swarm, so that the other members try to follow the best one (closest to optimum solution), this step represent one iteration in this process. This will continue to find the convergence until a minimum error solution obtained or terminating the process by reaching the maximum number of iterations. The mechanism of the particle when it compares its current position and velocity with the previous one and update its position and velocity is given as [3, 12]:

$$v_{j(t+1)} = w \cdot v_j(t) + C_1 \cdot rand \cdot (pbest(t) - x_j(t)) + C_2 \cdot rand \cdot (gbest(t) - x_j(t)) \quad \ldots (8)$$

$$x_j(t+1) = x_j(t) + v_j(t+1) \quad \ldots (9)$$

Where $v_{j(t+1)}$ is the speed of the j^{th} particle at (t+1) iteration, $x_j(t+1)$ is the location of the j^{th} particle at (t+1) iteration, w is the inertial weight factor (weighting function), C_1 and C_2 are random number range [0,1], that called acceleration constants called cognitive learning rate and social learning rate respectively, *rand* is the random function in the range [0,1], pbest is the individual best position of the particle, and gbest is the global best position of the swarm of the particles. The weighting function, w is responsible for dynamically adjusting the speed of the particles, so it is responsible for balancing between global and local search. Applying a large inertia weight at the start of the algorithm and decays into a smaller value through the PSO execution makes the algorithm search locally at the end and globally at the beginning of the execution. The weighting Function w is calculated as follows:

$$w(i) = w_{max} - \left(\frac{w_{max} - w_{min}}{iter_{max}}\right) iter \quad \ldots (10)$$

Where w_{max}, w_{min} are the initial and final weights, $iter_{max}$ is the maximum iteration number and $iter$ is the current iteration number [3].

The flow chart depicting the implementation of modified PSO algorithm for optimizing the parameters of the FOPID controller for the given system as shown in Figure (4):

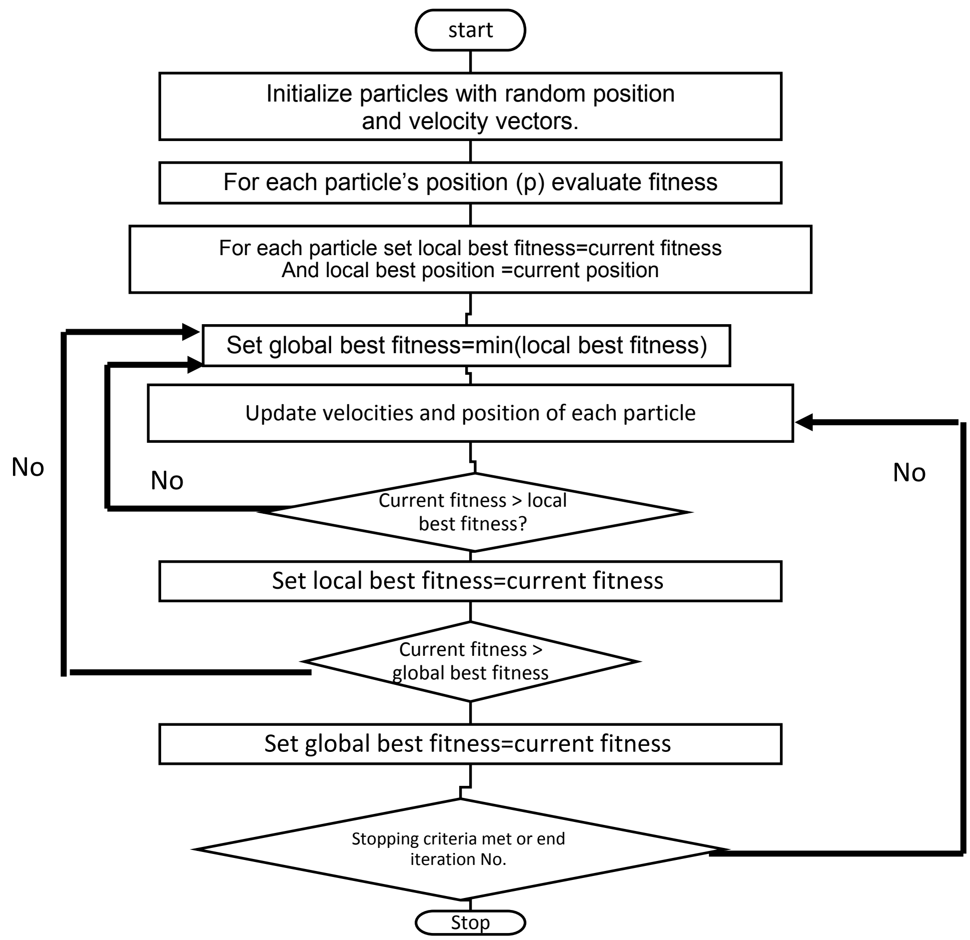


Figure (4) the flow chart of modified PSO-FPOID algorithm.

The proposed control structure using FOPID controller with modified PSO as shown in Figure (5):

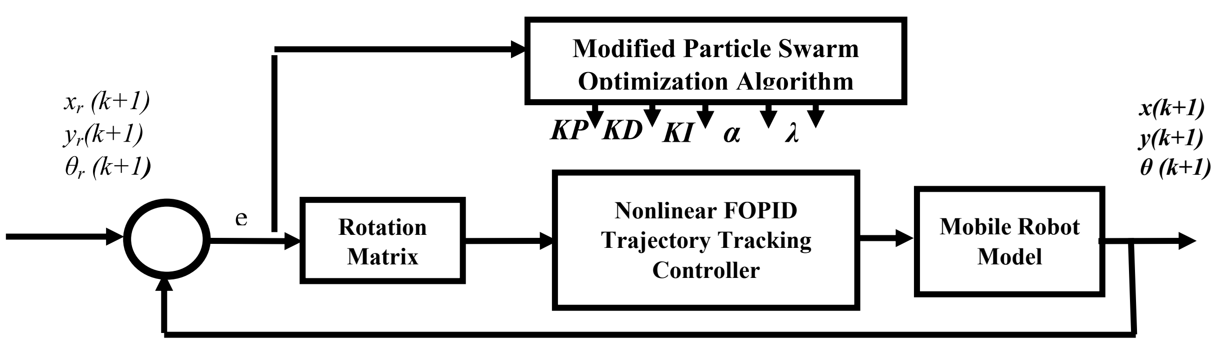


Figure (5) the structure of the proposed nonlinear FOPID path tracking controller of 2-WMR.

Wheel Mobile Robot Modeling

In the past few years, several studies have been completed in the field of 2-WMR control design, the aim is designing and implementing the driving control signal that the 2-WMR must obey to keep track of a desired track precisely and minimizing the tracking error. Tracking error of a 2-WMR can be detrimental for many reasons, it might cause collisions with obstacles, or cause the robot to fail to accomplish the task successfully, and finally it may cause an increase of the traveling distance and in turns the traveling time. [13]

Kinematic Modeling of Non-Holonomic 2-WMR

The 2-WMR, shown in Figure (6), consists of a castor wheel in the head of the 2-WMR and two driving wheels mounted on the same axis located at the back of as shown in Figure (6). The castor carries the mechanical structure and saves the platform more stable. The motion and the orientation of 2-WMR are achieved via two-DC motors which forms the actuators of the right and left wheels. The two wheels have the same radius denoted by r, and the distance between the two wheels is L [1]. The c is the center of mass of the 2-WMR and center of axis of the wheels.

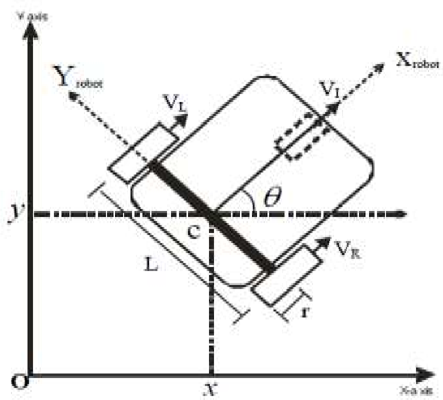


Figure (6) a non-holonomic 2WMR.

From Figure (6), referring to the position of the 2-WMR in the world coordinate frame {o, x, y}, the kinematic equations of 2-WMR can be described as follows [2, 13]:

$$V_A(t) = \frac{d\theta(t)}{dt} \quad \dots (11)$$

$$V_{Li}(t) = V_A(t)R(t) \quad \dots (12)$$

$$V_{left}(t) = (R(t) + \frac{L}{2})V_A(t) \quad \dots (13)$$

$$V_{right}(t) = (R(t) - \frac{L}{2})V_A(t) \quad \dots (14)$$

$$\dot{x}(t) = V_{Li}(t)\cos\theta\,(t) \quad \dots (15)$$

$$\dot{y}(t) = V_{Li}(t)\sin\theta\,(t) \quad \dots (16)$$

$$\dot{\theta}(t) = V_A(t) \quad \dots (17)$$

Where: V_{Li} are the linear velocity, V_A are the angular velocity and R the radius of the 2-WMR instantaneous curvature trajectory. Integrating equations 15, 16 and 17 and we get [2, 13,14]:

$$x(t) = \int_0^t V_{Li}(\tau)\cos\theta\,(\tau)d\tau + x_\circ \quad \dots (18)$$

$$y(t) = \int_0^t V_{Li}(\tau)\sin\theta\,(\tau)d\tau + y_\circ \quad \dots (19)$$

$$\theta(t) = \int_0^t V_A(\tau)d\tau + \theta_\circ \quad \dots (20)$$

Where the vector $p\ (0) = (x_\circ, y_\circ\ , \theta_\circ)$ is the initial pose. Equations (18), (19) and (20) are valid and works for all kinds of wheeled 2-WMR heading in a specific path at certain velocity $V_t(t) = (x(t), y(t), \theta(t))$. For the special case of 2-WMR [13, 14]:

$$x(t) = 0.5 \int_0^t [V_{left}(\tau) + V_{right}(\tau)]\cos\theta\,(\tau)d\tau + x_\circ \quad \ldots(21)$$
$$y(t) = 0.5 \int_0^t [V_{left}(\tau) + V_{right}(\tau)]\sin\theta\,(\tau)d\tau + y_\circ \quad \ldots(22)$$
$$\theta(t) = \frac{1}{L}\int_0^t [V_{left}(\tau) - V_{right}(\tau)]d\tau + \theta_\circ \quad \ldots(23)$$

The kinematic equations 21, 22 and 23 can be convert into discrete form [13]:

$$x(k) = 0.5 \sum_{i=1}^{k} [V_{left}(i) + V_{right}(i)]\cos\theta\,(i)\Delta t + x_\circ \quad \ldots(24)$$
$$y(k) = 0.5 \sum_{i=1}^{k} [V_{left}(i) + V_{right}(i)]\sin\theta\,(i)\Delta t + y_\circ \quad \ldots(25)$$
$$\theta(k\ ) = \frac{1}{L}\sum_{i=1}^{k} [V_{left}(i) - V_{right}(i)]\Delta t + \theta_\circ \quad \ldots(26)$$

Where $x(k), y(k), \theta(k)$ are the components of the position at the k-step of the movement and Δt is the sampling interval between two adjacent samples. For the simulations, the current form of the position components can be expressed in nonlinear discrete difference equations as follows [13]:

$$x(k) = 0.5[V_{left}(k) + V_{right}(k)]\cos\theta\,(k)\Delta t + x(k-1) \quad \ldots(27)$$
$$y(k) = 0.5\big[V_{left}(k) + V_{right}(k)\big]\sin\theta\,(k)\Delta t + y(k-1) \quad \ldots(28)$$
$$\theta(k\ ) = \frac{1}{L}[V_{left}(k) - V_{right}(k)]\Delta t + \theta(k-1) \quad \ldots(29)$$

Equations (27)-(28) are used in the design of the proposed FOPID controller. Also, it is used in the simulations of the 2-WMR using MATLAB environment in time-domain. The kinematics equations 15, 16 and 17 can be represent as follows:

$$\begin{bmatrix}\dot{x}(t)\\ \dot{y}(t)\\ \dot{\theta}(t)\end{bmatrix} = \begin{bmatrix}\cos\theta\,(t) & 0\\ \sin\theta\,(t) & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}V_{Li}(t)\\ V_A(t)\end{bmatrix} \quad \ldots(30)$$

The 2-WMR can be headed to any position in a free workspace where it is assumed that the wheels of 2-WMR are ideally set up in such a way that they have ideal gently sloping with no sliding. The 2-WMR cannot make lateral movement because the number of degrees of freedom of the movement for 2-WMR is 2, the wheel must not move orthogonally to the wheel plane. Additionally, the linear velocity of the point *c* of the 2-WMR must be in the direction of the axis of symmetry (x-axis); this is referred to as the non-holonomic constraint [2,13]:

$$-\dot{x}(t)\sin\theta\,(t) + \dot{y}(t)\cos\theta\,(t) = 0 \quad \ldots(31)$$

Simulation Results

The proposed controller is tested under computer simulations with MATLAB environment. The simulations are carried out off-line by tracking a desired position (x, y) and orientation angle (θ) with an infinity shape and circle trajectories in the tracking control of the robot. The 2-WMR model parameters values are taken from [15]: M= 0.65kg, I = 0.36 kg.m2, L = 0.115 m, r = 0.033 m and sampling time is equal to 0.5 second.

The proposed nonlinear FOPID neural controller has been designed by finding the optimum set of the five parameters by the proposed modified PSO. The designed FOPID together with the conventional nonlinear PID neural controllers are applied to 2WMR with kinematic model. The first step of the design procedure is to set the values of the following parameters of the proposed modified PSO algorithm: Population size of particles is equal to 25 and iteration number is equal to 100. It is shown from the model of the 2WMR that it is MIMO system, with input of V_l and V_r and output [$x(k)$ $y(k)$ $\theta(k)$]. Now, to design a motion controller for the 2WMR, it required to design a single PID or FOPID controller for reach output to track the desired reference input. At the end, the design comes to three conventional PID controllers and each controller have three parameters so the number of weights in each particle is a vector of nine parameters. While with a FOPID controller, the total number of weights to be adjusted or tuned in each particle is a vector of fifteen parameters because each controller have 5 parameters to tune. The acceleration constants C_1 and C_2 are equal to 1.4, r_1 and r_2 are random values between 0 and 1. Two case studies are used in this paper.

Case Study 1:

The required infinity path which has explicitly continuous gradient with rotation changing radius. This path can be express as [14]:

$$x_r(t) = 1.75 + 1.75 * \sin\left(\frac{\pi t}{25}\right) \quad \ldots (32)$$

$$y_r(t) = \sin\left(\frac{2\pi t}{25}\right) \quad \ldots (33)$$

$$\theta(t) = 2\tan^{-1}\left(\frac{\Delta y_r(t)}{((\Delta x_r(t)^2 + \Delta y_r(t)^2)^{0.5} + \Delta x_r(t)^2)}\right) \quad \ldots (34)$$

The initial situation of the robot model starts with p (0) = [1.75, -0.25, π/ 2]. The 2-WMR path tracking with the proposed FOPID controller is shown in Figure (7). This Figure explains the position, orientation, pathing performance. The simulation results demonstrated the activity of the FOPID controller by showing its ability to generate small, smooth values of the control input speeds for right and left wheels without an acute shape as shown in Figure (8), that the amplitude of the controls design is damped more than in the conventional PID controller.

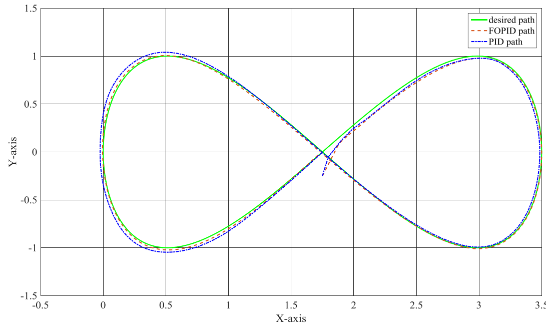
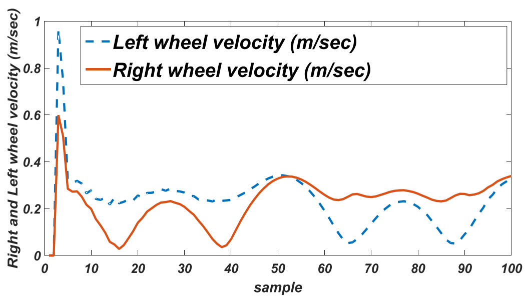
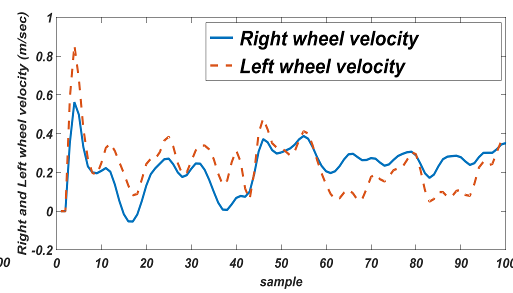
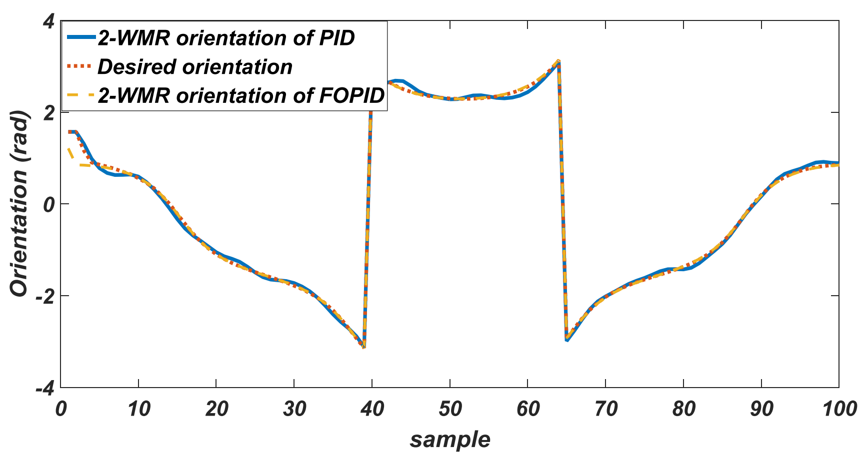


Figure (7) desired trajectory and actual 2-WMR trajectory





(a) Control action using PID controller. **(b) Control action using FOPID controller.**



(c) Desired and actual 2-WMR orientation using PID and FOPID controllers
Figure (8) Simulation results for infinity path.

The modified particle swarm optimization is used for tuning the parameters of the PID neural controller (K_x, K_y, K_θ) which is demonstrated in Table (1). The tuned parameters of the proposed FOPID controller (K_x, K_y, K_θ, α and λ) are shown in Table (2).

The Mean Square Error (MSE) has been used as the performance index in the control design methodology as equation and it is used as the objective function in the modified PSO:

$$e = \frac{1}{pop}\sum_{p=1}^{pop}((xr(k+1)_p - x(k+1)_p)^2 + (yr(k+1)_p - y(k+1)_p)^2 + (\theta_r(k+1)_p - \theta(k+1)_p)^2) \quad ...(35)$$

Where pop is a number of particles.

Table (1) Parameters of the PID controller

KP_x	KI_x	KD_x	KP_y	KI_y	KD_y	KP_θ	KI_θ	KD_θ
0.5376	1.1374	0.7944	1.8264	0.1743	6.4422	-1.538	-0.292	0.2756

Table (2) Parameters of the FOPID controller

KP_x	KI_x	KD_x	λ_x	α_x	KP_y	KI_y	KD_y	λ_y	α_y	KP_θ	KI_θ	KD_θ	λ_θ	α_θ
3.362	3.635	0.426	0.975	0.717	-1.594	1.918	1.845	1	0.408	-2.115	-1.081	-0.764	0.781	0.888

Case Study 2:

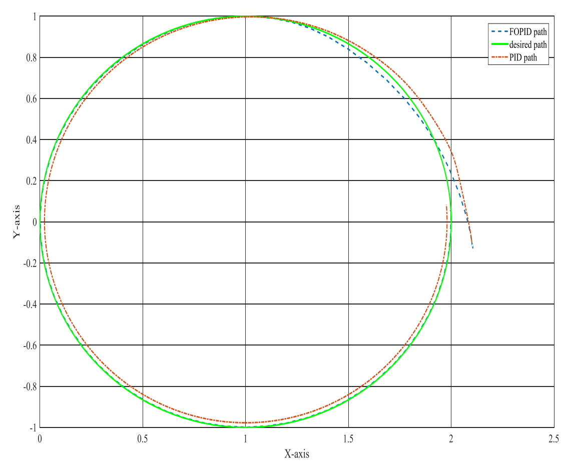
The circle path that the 2-WMR must follow can be describe by the following equations [16]:

$$x_r(t) = 1 + \cos(\frac{t}{10}) \quad ... (36)$$

$$y_r(t) = \sin\left(\frac{t}{10}\right) \quad .. (37)$$

$$\theta(t) = \frac{\pi}{2} + \frac{t}{10} \quad ... (38)$$

Equations (36 and 37) describe the path coordinates, while the orientation of the 2-WMR is expressed by equation (38). The initial situation of the 2-WMR model starts with p (0) = [1.1, -0.5, π/ 2]. The tracked path by 2-WMR is illustrated in Figure (9). This Figure explains the excellent performance obtained by the FOPID where position and orientation are mostly coincided on the desired track.



Figurer(9)Desired trajectory and actual 2-WMR **trajectory.**

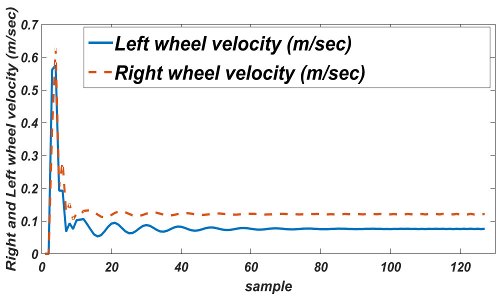
The simulation results demonstrate the ability of the proposed controller to generate small, smooth values of the control input speeds for right and left wheels without an acute shape as shown in Figure (10). Tables (3) and (4) list the tuned parameters of the nonlinear PID and the proposed FOPID controller for this case study respectively.

Table (3) Parameters of the PID controller

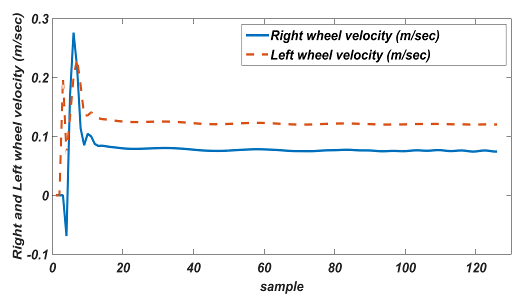
KP_x	KI_x	KD_x	KP_y	KI_y	KD_y	KP_θ	KI_θ	KD_θ
2.3523	0.2898	0.1588	-0.978	-0.519	0.8552	-4.332	-2.289	0.4336

Table (4) Parameters of the FOPID controller

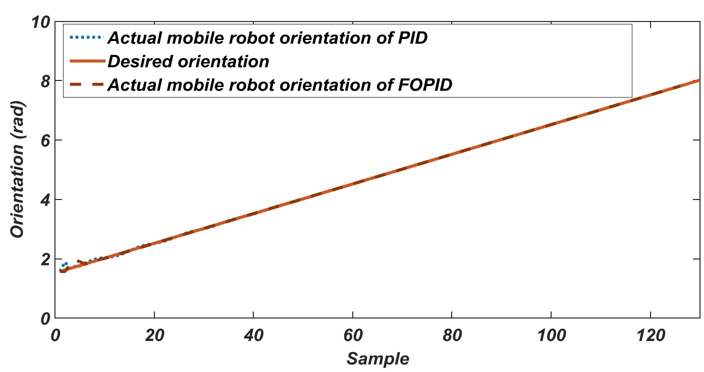
KP_x	KI_x	KD_x	λ_x	α_x	KP_y	KI_y	KD_y	λ_y	α_y	KP_θ	KI_θ	KD_θ	λ_θ	α_θ
0.7022	1.0137	-0.218	0.8005	0.4167	-0.308	-0.473	0.063	0.9393	0.976	-0.879	-1.061	0.0694	0.526	1.24



(a) Control action using PID controller.



(b) Control action using FOPID controller.



(c) Desired and actual 2-WMR orientation of PID and FOPID

Figure (10) Simulation results for circular path

Finally considers the trajectory errors for both PID and FOPID controllers, it can be seen from table (5) that the proposed nonlinear FOPID neural controller gives small trajectory errors than the conventional nonlinear PID neural controller. While the difference in the error is small for small L (distance between the two wheels), it is shown from simulations that with larger values of L the difference in the error becomes very apparent.

Table (5) The values of MSE for PID & FOPID controller

	MSE of PID neural controller	MSE of FOPID neural controller
Infinity path	0.006760	0.005092
Circular path	0.001691	0.001372

CONCLUSIONS

In this research the nonlinear PID neural network controller and nonlinear FOPID neural controller with modified PSO method for MIMO non-holonomic 2-WMR have been presented. Simulations and results show that the proposed nonlinear FOPID neural controller gives better results than the conventional nonlinear PID neural controller by following the desired path without oscillations and minimum mean squared error as seen from the results. The designed FOPID controller can track any path with minimum control action and perfect orientation. It is worthy to mention that the best result for this problem solved using FOPID tuned with modified PSO can be obtained by running the algorithm more than one time due the heuristic nature of the PSO algorithm so that best set of the tuned parameters can be found.

REFERENCES

[1].Kocaturk B., "Motion Control of Wheeled Mobile Robots," Interdisciplinary Description of Complex Systems 13(1), 41-47, 2015

[2].Tzafestas S. G.," Introduction to Mobile Robot Control," Elsevier, First edition 2014

[3].Rastogi A. and Tiwari P., "Optimal Tuning of Fractional Order PID Controller for DC Motor Speed Control Using Particle Swarm Optimization", International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Vol.3, pp.150-157, May 2013.

[4].Ramezanian H. and Balochian S., "Optimal Design a Fractional-Order PID Controller using Particle Swarm Optimization Algorithm", International Journal of Control and Automation Vol. 6, No. 4,pp. 55-68, August, 2013.

[5].Kumar D. N. Samalla H., Rao Ch. J., Naidu Y. S., Jose K. A. and Kumar B. M. ," Position and Orientation Control of a Mobile Robot Using Neural Networks, "Springer, Vol.2, pp.123-131, 2015.

[6].Fierro R. and Lewis F. L., "Control of a Nonholonomic Mobile Robot: Back stepping Kinematics into Dynamics, "Journal of Robotic Systems 14(3), (1997) John Wiley & Sons, Inc., pp.149–163, 1997.

[7].El-Teleity S. A., Abdel-Kader H. M., Nossair Z. B. and TagElDein A., " Fuzzy Logic Control of an Autonomous Mobile Robot," IEEE, (MMAR), 16th International Conference, pp. 188-193, 2011.

[8].Lee J. H., Lin C., Lim H. and Lee J. M., "Sliding Mode Control for Trajectory Tracking of Mobile Robot in the RFID Sensor Space," Springer, International Journal of Control, Automation, and Systems (2009) 7(3), pp. 429-435, 2009.

[9].Abdalla T. Y. and Hamzah M. I.," Trajectory Tracking Control for Mobile Robot using Wavelet Network," International Journal of Computer Applications, Vol. 74, No. 3, pp. 32-37,July 2013.

[10].Petras I., "Fractional-Order Nonlinear Systems," Higher Education Press, Beijing and Springer-Verlag Berlin Heidelberg, ISBN 978-7-04-031534-9, 2011.

[11].Maiti D., Acharya A., Mithun, Chakraborty A., Janarthanan R., "Tuning PID and $PI^{\lambda}D^{\delta}$ Controllers using the Integral Time Absolute Error Criterion," IEEE, 2008.

[12].Rao S. S., "Engineering Optimization: Theory and Practice," Fourth Edition, John Wiley & Sons, 2009.

[13].Dagher K. E. and Al-Araji A., "Design of a Nonlinear PID Neural Trajectory Tracking Controller for Mobile Robot based on Optimization Algorithm", Eng. & Tech. Journal , Vol.32,Part (A), No.4, pp. 973-985, 2014

[14].Abdoon Z. Y. and Jassar N. A.," Design and Implementation of a Real-Time Intelligent Controller for a Differential Drive Mobile Robot," International Journal of Scientific & Engineering Research, Vol.4, pp. 1556-1562, December 2013.

[15].Internet website http://www.parallax.com. Robotics with the Boe-Bot text manual v3.0. Accessed April 2013.

[16].Tawfik M. A. , Abdulwahb E. N. and Swadi S. M.," Trajectory Tracking Control for a Wheeled Mobile Robot using Fractional Order $PI^{a}D^{b}$ Controller," Al-Khawarizmi Engineering journal, Vol. 10, no. 3, pp. 39-52, 2014.