

A.S. Al-Araji 

University of Technology,
Control and Systems
Engineering Department.
Baghdad, Iraq.
60166@uotechnology.edu.iq

L.T. Rasheed

University of Technology,
Control and Systems
Engineering Department.
Baghdad, Iraq
60065@uotechnology.edu.iq

Received on: 27/06/2016

Accepted on: 20/10/2016

A Cognitive Nonlinear Fractional Order PID Neural Controller Design for Wheeled Mobile Robot based on Bacterial Foraging Optimization Algorithm

Abstract- The aim of this paper is to design a proposed non-linear fractional order proportional-integral-derivative neural (NFOPIDN) controller by modifying and improving the performance of fractional order PID (FOPID) controller through employing the theory of neural network with cognitive optimization techniques for the differential - drive wheeled mobile robot (WMR) multi-input multi-output (MIMO) system in order to follow a pre-defined trajectory. In this paper a cognitive bacterial foraging optimization algorithm (BFOA) has been utilized to find and tune the parameters of the proposed (NFOPIDN) controller and then find the optimal torque control signals for the differential - drive WMR. The simulation results show that the proposed controller can give excellent performance in terms of compared with other works (minimized tracking error for Ranunculoid-curve trajectory, smoothness of torque control signals obtained without saturation state and no sharp spikes action as well as minimum number of memory units needed for the structure of the proposed NFOPIDN controller).

Keywords- Non-linear Fractional Order proportional-integral-derivative Neural Controller; cognitive bacterial foraging optimization algorithm (BFOA); Wheeled Mobile Robot.

How to cite this article: A.S. Al-Araji and L.T. Rasheed "A Cognitive Nonlinear Fractional Order PID Neural Controller Design for Wheeled Mobile Robot based on Bacterial Foraging Optimization Algorithm," *Engineering and Technology Journal*, Vol. 35, Part A. No. 3, pp. 289-300, 2017.

1. Introduction

The dynamic WMR systems have drawn a lot of interest for the researchers recently because they are found in many applications in industry, transportations, military, security settings and other fields due to their ability of handling complex visual and information processing for artificial intelligence, loading capability and they can work in dangerous and hazardous environment. WMR suffer from non-holonomic constraints (pure rolling without side slipping motion) meaning that mobile robots can move only in the direction normal to the axis of the driving wheels. Trajectory tracking control is important topic of the WMR it means to apply control signals to the WMR in such a way that the WMR follow a curve that connects its actual position and orientation with the goal position and orientation of the predefined trajectory (desired or reference trajectory). In general, the dynamic trajectory tracking control is still active region of research because as we mentioned above that WMR systems have found in various industrial applications [1,2].

The Motivation for this work is the problems in the mapping and localization; cognition trajectory planning; path-tracking and motion control.

Therefore, different trajectory tracking control approaches for WMR have been proposed so as to achieve the best performance for the WMR including high speed, high tracking accuracy (minimized tracking error), low energy consumption and smoothness of velocity control signal obtained, such as fuzzy logic trajectory tracking controller [3,4]; sliding mode controller [5,6]; back-stepping technique [7, 8]; nonlinear PID neural trajectory tracking controller [9-11] and cognitive trajectory tracking neural controller [12,13].

The fundamental essence of the contribution novelty for this paper is described by the following points listed below:

- Modifying and improving the performance of FOPID controller by using the theory of neural network techniques.
- The analytically derived control law based NFOPIDN controller has considerably high computational accuracy to gain the optimal control signal and lead to minimum tracking error of the dynamic WMR with the minimum number of memory units (M) needed based cognitive BFOA.
- Validation of the controller adaptation performance through change the initial pose state.

The rest of the paper is arranged as follows: Section two is a description of the non-holonomic WMR. In section three, the proposed feedforward and feedback controllers are derived. The cognitive BFOA is explained in section four. In section five, the simulation results and discussion of the NFOPIDN controller are described finally the conclusions are presented in section six.

2. Non-Holonomic Wheeled Mobile Robot Model

The non-holonomic WMR shown in Figure 1 is a vehicle has two driving wheels placed on the same axis and two omni-directional castor wheels mounted in the front and rear of the vehicle. The two castor wheels carry the mechanical structure and keep the vehicle more stable. The WMR is driven by two independent DC analogue motors that are used as actuators of the two driving wheels for motion and orientation. The radius of these two wheels is the same and they are separated by the L distance. The center of gravity of the WMR is at point c, center of axis connected the two driven wheels [14]. The parameters of WMR can be described in Table 1. The location of the mobile robot in the world coordinate frame [O, X, Y] can be described by the vector as follow:

$$q = (x_c, y_c, \theta)^T \tag{1}$$

Where x_c and y_c are the coordinates of point c in the world coordinate frame, and θ is the angle of rotation of the robot measured from X-axis, These three generalized coordinates can be described as the configuration of the WMR.

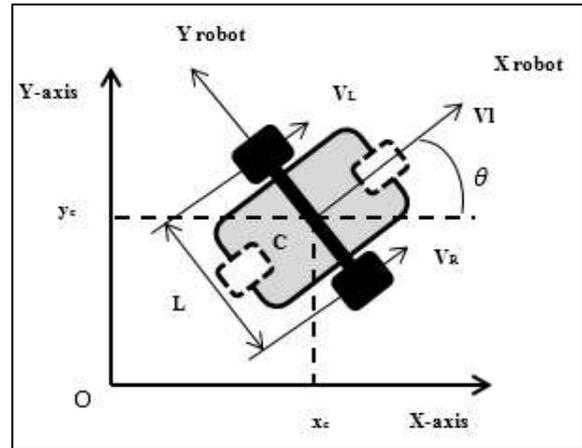


Figure 1: Model of the non-holonomic mobile robot

The WMR has a non-holonomic constraint that the driving wheels are purely roll without slipping [15]. This non-holonomic constraint can be written as in equation (2):

$$-\dot{x}(t)\sin \theta(t) + \dot{y} \cos \theta(t)=0 \tag{2}$$

The kinematics equations of the WMR can be written as in equation (3) [16]:

$$\dot{q} = \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} \cos \theta(t) & 0 \\ \sin \theta(t) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V_l \\ V_w \end{bmatrix} = S(q) \begin{bmatrix} V_l \\ V_w \end{bmatrix} \tag{3}$$

Where $S(q)$ is defining a full rank matrix and V_l and V_w , are the linear and angular velocities. The dynamic model of the WMR can be represented by the dynamic equations based on Euler Lagrange formulation as in equation (4) [17]:

Table 1: Parameters of the model's kinematics

r	Radius of each driving wheel (m).
L	Distance separating the two driving wheels (m).
θ	The angle of rotation of the robot measured from X-axis.
C	Centre of axis of rear wheels.
V_l	Linear velocity of the mobile robot (m/sec).
V_w	Angular velocity of the mobile robot (rad/sec).
V_L	Linear velocity of the left wheel (m/sec).
V_R	Linear velocity of the right wheel (m/sec).
$[O,X,Y]$	The world coordinate frame.
x_c and y_c	The coordinates of point c in the global coordinate frame.

$$\begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\theta} \end{bmatrix} + \tau_d = \frac{1}{r} \begin{bmatrix} \cos(\theta) & \cos(\theta) \\ \sin(\theta) & \sin(\theta) \\ \frac{L}{2} & -\frac{L}{2} \end{bmatrix} \begin{bmatrix} \tau_{TL} \\ \tau_{TR} \end{bmatrix} + \begin{bmatrix} -\sin(\theta) \\ \cos(\theta) \\ 0 \end{bmatrix} \lambda \quad (4)$$

where τ_{TL} : the torque of the left motor, τ_{TR} : the torque of the right motor, m : the mass of the robot, I : the inertia of the robot, λ : the vector of constraint forces and τ_d denotes the bounded unknown disturbances including unstructured and unmodelled dynamics.

3. Controller Design

The general structure of the proposed control system consists of two parts as shown in Figure 2:

- Feedforward controller based on numerical equations.
- Feedback controller (NFOPIDN controller) based on cognitive optimization technique.

The feedforward controller is necessary in keeping the steady-state tracking error at zero. This means that the actions of the feedforward controller represent the reference left and right torques of the steady state outputs of the WMR. Reference input torques keep the robot on a reference path in the presence of any disturbances or initial state errors. The control actions of the feedforward controller can be calculated from the linear and angular reference velocities.

The linear reference velocity and angular reference velocity are calculated from the desired trajectory and can be described in equations (5-10) [18-20]:

$$\tau_{lin-ref}(t) = m \frac{dV_r(t)}{dt} \quad (5)$$

$$\tau_{ang-ref}(t) = I \frac{dW_r(t)}{dt} \quad (6)$$

$$V_r(t) = \sqrt{\dot{x}_r^2(t) + \dot{y}_r^2(t)} \quad (7)$$

$$W_r(t) = \frac{\dot{x}_r(t)\dot{y}_r^2(t) - \dot{y}_r^2(t)\dot{x}_r(t)}{\dot{x}_r^2(t) + \dot{y}_r^2(t)} \quad (8)$$

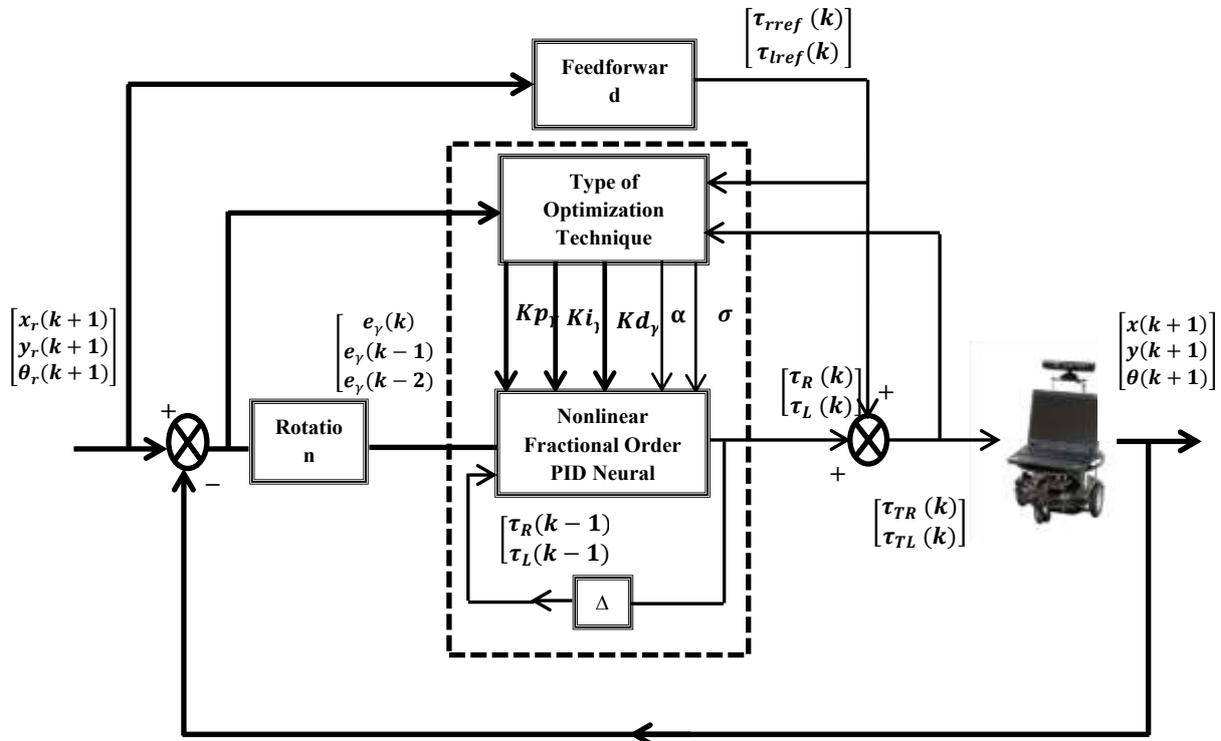


Figure 2: General structure of the proposed feedforward and a cognitive feedback (NFOPIDN) controller for mobile robot model

$$\tau_{lref}(t) = 0.5r\tau_{lin-ref}(t) + \frac{r\tau_{ang-ref}(t)}{L} \quad (9)$$

$$\tau_{rref}(t) = 0.5r\tau_{lin-ref}(t) - \frac{r\tau_{ang-ref}(t)}{L} \quad (10)$$

Where V_r and W_r are the linear and angular reference velocities respectively, $\tau_{lin-ref}$ and $\tau_{ang-ref}$ are the linear and angular reference torques of the wheeled mobile robot respectively and τ_{lref} and τ_{rref} are the left and right wheel reference torques respectively.

The feedback controller (NFOPIDN controller) is essential to stabilize the tracking error of the mobile robot system when the trajectory of the robot drifts from the reference trajectory during transient state. The control actions of this controller are $\tau_L(k)$ and $\tau_R(k)$ and represent the left and right torques respectively.

In the feedback controller the cognitive BFOA has been utilized to obtain the optimal gains for the NFOPIDN controller so as to get the left and right torques control actions. The sum of this control action with the feedforward control action will minimize the tracking error of differential-drive WMR.

The error and control action signals are denoted in FOPID controller as $e(t)$ and $u(t)$ respectively and they are associated as shown in the equation (11) [21]:

$$u(t) = Kp e(t) + Ki D_t^{-\sigma} e(t) + Kd D_t^{\alpha} e(t) \quad (11)$$

Where:

Kp is the proportional constant. Ki is the integration constant. Kd is the differentiation constant. $D_t^{-\sigma}$ is the fractional integral operator. D_t^{α} is the fractional differential operator.

By taking the Laplace transform of fractional derivative and fractional integral of $e(t)$ are given as equation (12 and 13) respectively [21]:

$$L\{D^{\alpha} e(t)\} = S^{\alpha} E(s) - [D^{\alpha-1} e(t)]_{t=0} \dots (12)$$

$$L\{D^{-\sigma} e(t)\} = S^{-\sigma} E(s) \quad (13)$$

After taking Laplace transform of equation (11) and submitting equations (12 and 13) in equation (11), the transfer function of (FOPID) controller will be as equation (14) [21]:

$$G_c(s) = Kp + Ki S^{-\sigma} + Kd S^{\alpha} \quad (14)$$

The first step in the design of the proposed (NFOPIDN) controller is converted the continuous-time (FOPID) control equation to

discrete-time (FOPID) control equation by using the generating function technique with three steps:

- Pre-warped Tustin transform is used as equation (15) that converted equation (14) from s-domain to z-domain.

$$S^{\alpha} = \left(\frac{w_c}{\tan\left(\frac{w_c T}{2}\right)} \times \frac{1-Z^{-1}}{1+Z^{-1}} \right)^{\alpha} = \Omega^{\alpha} \times \left(\frac{1-Z^{-1}}{1+Z^{-1}} \right)^{\alpha} \quad (15)$$

Where:

T : sampling period of the system (0.2 sec).

w_c : is gain crossover frequency of the open – loop transfer function and it is taken from [20] and equal to (15 rad/sec).

$$\Omega : \frac{w_c}{\tan\left(\frac{w_c T}{2}\right)} = 1.0637 \quad (16)$$

α : is order of derivative.

- Power series expansion (PSE) is used as the resulted expression, which is acquired in the term of (z) with a limited (minimum) memory, must necessarily use for any practical discrete-time controller.

- The (PSE) of the expression in the right hand side of equation (15) must be calculated in order to check the stability of the controller as shown in Figure 3 as follows:

1- If the poles and zeros are inside the unit circle of the (z) plane the system is stable.

2- If the poles and zeros are outside the unit circle of the (z) plane the system is unstable.

3- If the poles and zeros are on the border of the unit circle of the (z) plane the system is critically stable.

After applying the generating function technique with the three steps, the proposed control law can be driven as follows:

- The derivative term S^{α} of equation (14) can be expanded as follows:

Assuming $w = z^{-1}$ in equation (15) yield:

$$\Omega^{\alpha} \times \left(\frac{1-w}{1+w} \right)^{\alpha} = \Omega^{\alpha} \sum_{j=0}^{\infty} f_j(\alpha) w^j \quad (17)$$

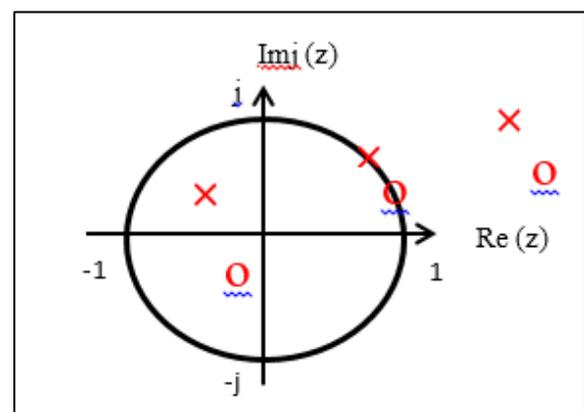


Figure 3: Z- plane pole and zero locations.

Where:

$$f_j(\alpha) = \frac{1}{j!} \times \frac{d^j}{dw^j} \left(\frac{1-w}{1+w} \right)^\alpha \Big|_{w=0} \quad (18)$$

By substitution of $w=z^{-1}$ in equation (17) yield:

$$S^\alpha = \Omega^\alpha \sum_{j=0}^{\infty} f_j(\alpha) z^{-j} \quad (19)$$

Where the coefficients $f_j(\alpha)$ are calculated from equation (18).

It can be shown (using Maple) that the first few coefficients in (18) are as follow:

$$f_0(\alpha) = 1, f_1(\alpha) = -2 \times \alpha, f_2(\alpha) = 2 \times \alpha^2, f_3(\alpha) = -\frac{4}{3} \times \alpha^3 - \frac{2}{3} \times \alpha$$

$$f_4(\alpha) = \frac{2}{3} \times \alpha^4 + \frac{4}{3} \times \alpha^2, f_5(\alpha) = -\frac{4}{15} \times \alpha^5 - \frac{4}{3} \times \alpha^3 - \frac{2}{3} \times \alpha$$

$$f_6(\alpha) = \frac{4}{45} \times \alpha^6 + \frac{8}{9} \times \alpha^4 + \frac{46}{45} \times \alpha^2, \dots$$

• The integral term $S^{-\sigma}$ of equation (14) can be expanded as follows:

$$S^{-\sigma} = \left(\frac{1}{S} \right) \times S^{1-\sigma} \quad (20)$$

$$S^{-\sigma} = S^{-1} \times S^{1-\sigma}$$

(21)

Where:

σ : is the order of integral.

By applying a Tustin method in equation (21), then:

$$S^{-\sigma} = \Omega^{-1} \times \left(\frac{1+z^{-1}}{1-z^{-1}} \right) \times \Omega^{1-\sigma} \times \left(\frac{1-z^{-1}}{1+z^{-1}} \right)^{1-\sigma} \quad (22)$$

$$S^{-\sigma} = \Omega^{-\sigma} \left(\frac{1+z^{-1}}{1-z^{-1}} \right) \sum_{j=0}^{\infty} f_j(1-\sigma) z^{-j} \quad (23)$$

Where the term $\left(\frac{1-z^{-1}}{1+z^{-1}} \right)^{1-\sigma}$ can be calculated in the same manner as $\left(\frac{1-z^{-1}}{1+z^{-1}} \right)^\alpha$ in equations (15 and 17).

The term $f_j(1-\sigma)$ is the again which can be calculated as in equation (18).

Then, substitution of equations (19) and (23) in equation (14) yield, the general control law becomes:

$$C_d(z) = kp + kd \sum_{j=0}^{\infty} f_j(\alpha) z^{-j} + ki \left(\frac{1+z^{-1}}{1-z^{-1}} \right) \sum_{j=0}^{\infty} f_j(1-\sigma) z^{-j} \quad (24)$$

Where:

$$\left. \begin{aligned} kp &= Kp \\ kd &= Kd \times \Omega^\alpha \\ ki &= Ki \times \Omega^{-\sigma} \end{aligned} \right\} \quad (25)$$

The infinite numbers of memory units are needed for its realization and consequently the computational cost is increased by increasing the

time. In other words, in practice the upper bound of ∞ as in (24) cannot be considered equal to infinity. By restricting the number of memory units to (M) the equation (24) will be:

$$C_d(z) = kp + kd \sum_{j=0}^M f_j(\alpha) z^{-j} + ki \left(\frac{1+z^{-1}}{1-z^{-1}} \right) \sum_{j=0}^M f_j(1-\sigma) z^{-j} \quad (26)$$

By multiplying each side of equation (26) by $(1-z^{-1})$, then the equation will be:

$$(1-z^{-1}) C_d(z) = (1-z^{-1}) kp + (1-z^{-1}) kd \sum_{j=0}^M f_j(\alpha) z^{-j} + ki(1+z^{-1}) \sum_{j=0}^M f_j(1-\sigma) z^{-j} \quad (27)$$

$$(1-z^{-1}) U(z) = (1-z^{-1}) kp E(z) + (1-z^{-1}) kd E(z) \sum_{j=0}^M f_j(\alpha) z^{-j} + ki(1+z^{-1}) E(z) \sum_{j=0}^M f_j(1-\sigma) z^{-j} \quad (28)$$

Finally, the discrete-time equation of FOPID controller will be:

$$U(k) = U(k-1) + kp(e(k) - e(k-1)) + kd \sum_{j=0}^M f_j(\alpha) (e(k-j) - e(k-j-1)) + ki \sum_{j=0}^M f_j(1-\sigma) (e(k-j) + e(k-j-1)) \quad (29)$$

Based on the discrete-time equation of FOPID controller, the proposed control law for the NFOPIDN controller is driven as follows:

$$\tau_{R,L}(k) = \tau_{R,L}(k-1) + kp_\gamma (e_\gamma(k) - e_\gamma(k-1)) + kd_\gamma \sum_{j=0}^M f_j(\alpha) (e_\gamma(k-j) - e_\gamma(k-j-1)) + ki_\gamma \sum_{j=0}^M f_j(1-\sigma) (e_\gamma(k-j) + e_\gamma(k-j-1)) \quad (30)$$

Where $\gamma = x, y, \theta$.

The tuning NFOPIDN input vector includes $e_\gamma(k), e_\gamma(k-1), e_\gamma(k-j), e_\gamma(k-j-1)$ and $\tau_{R,L}(k-1)$ where $e_\gamma(k)$ is the input error signal and $\tau_{R,L}(k-1)$ is the controller output signal.

The proposed feedback control law can be represented by equations (30 and 31):

$$\tau_R(k) = \tau_R(k-1) + O_x + O_y \quad (31)$$

$$\tau_L(k) = \tau_L(k-1) + O_\theta + O_y \quad (32)$$

Where O_x, O_y and O_θ are the outputs of neural networks and can be obtained as follows:

$$O_y = \frac{2}{1+e^{-net_\gamma}} - 1 \quad (33)$$

net_γ is calculated from equation (34):

$$net_\gamma = kp_\gamma (e_\gamma(k) - e_\gamma(k-1)) + kd_\gamma \sum_{j=0}^M f_j(\alpha) (e_\gamma(k-j) - e_\gamma(k-j-1)) + ki_\gamma \sum_{j=0}^M f_j(1-\sigma) (e_\gamma(k-j) + e_\gamma(k-j-1)) \quad (34)$$

The control parameters of the NFOPIDN controller $kp_\gamma, ki_\gamma, kd_\gamma, \alpha$ and σ can be adjusted using bacterial foraging optimization technique with minimum number of memory (M) needed. The NFOPIDN controller structure for the WMR system can be depicted in the Figure 4.

The sum of the feedforward and feedback control actions represent the total left and right torques of the mobile robot system and denoted by $\tau_{TL}(k)$ and $\tau_{TR}(k)$ respectively as in equations (35 and 36):

$$\tau_{TL}(k) = \tau_{lref}(k) + \tau_L(k) \quad (35)$$

$$\tau_{TR}(k) = \tau_{rref}(k) + \tau_R(k) \quad (36)$$

4. Bacterial Foraging Optimization Algorithm

BFOA is first introduced and it is a swarm intelligence technique which models the individual and group foraging policies of the (E. Coli) bacteria in human Intestine. E. Coli bacteria always try to locate the place which has a high nutrient value and avoid noxious places using certain pattern of motion called “chemotaxis”. When E. Coli bacteria arrive at a place with a higher nutrient value than previous place each bacterium releases chemical substances of attractant to attract other bacteria and it will move forward (swim or run) continuously. However, if E. Coli bacteria arrive at a place with a lower nutrient value than previous place each bacterium releases chemical substances of repellent to repel other bacteria and it will tumble. Bacterial Foraging optimization theory is explained by following steps: Chemotaxis, Swarming, Reproduction and Elimination and dispersal [22-24].

Chemotaxis: The movement of an E. coli cell through swimming and tumbling via flagella may be described by equation (37):

$$\theta(i, j + 1, kk, l) = \theta(i, j, kk, l) + C(i)\phi(j) \quad (37)$$

Where $\theta(i, j, kk, l)$ is position of the i^{th} bacterium at the j^{th} chemotaxis kk^{th} reproduction, and l^{th} elimination and dispersal steps, $C(i)$ represents the step size taken in direction of the tumble and $\phi(j)$ is the unit length random direction taken at each step.

Swarming: Every bacterium that has searched the optimum path of food should try to attract other bacteria in order to reach the desired place quickly. Swarming makes the bacteria meet into groups and can be described by equation (38):

$$J_{cc}(\theta, p(j, kk, l)) = \sum_{i=1}^S J_{cc}^i(\theta, \theta(i, j, kk, l)) = \sum_{i=1}^S [-d_{attract} \exp(-w_{attract} \sum_{mp=1}^p (\theta_{mp} - \theta_{mp}(i))^2)] +$$

$$\sum_{i=1}^S [h_{repellent} \exp(-w_{repellent} \sum_{mp=1}^p (\theta_{mp} - \theta_{mp}(i))^2)] \quad (38)$$

Where: $J_{cc}(\theta, p(j, kk, l))$ represents the additional cost function to be added to the actual cost function, S represents the total number of bacteria, p represents the dimension of the search space, $\theta = [\theta_1, \theta_2, \dots, \theta_p]^T$ represents a point on the optimization domain, $\theta_{mp}(i)$ represents position of the mp -th component of the i -th bacterium and $d_{attract}, w_{attract}, h_{repellent}, w_{repellent}$ are different parameters that should be chosen properly.

Reproduction: The lower healthy bacteria ultimately die, while the healthier bacteria each will split into two bacteria, which are placed in the same locations of the Dead bacteria. This keeps the swarm size constant.

Elimination and Dispersal: probably, in the local environment the life of a population of bacteria changes either gradually because of the consumption of nutrients or suddenly due to some other impacts.

Events can kill or disperse all the bacteria in a region. In contrast, dispersal may place bacteria near good food sources.

The (BFOA) Algorithm

Step 1: Initialize the parameters $p, S, N_c, N_s, N_{re}, N_{ed}, P_{ed}, C(i) (i = 1, 2, \dots, S), \theta^i$ where:

p : the number of parameters to be optimized.

S : the total number of bacterium.

N_c : the total number of chemotactic steps.

N_s : the total number of swim steps.

N_{re} : the total number of reproductive steps.

N_{ed} : the total number of elimination and dispersal steps.

P_{ed} : the probability value of elimination and dispersal.

$C(i)$: the chemotactic step size during each (run or tumble).

Step 2: Elimination-dispersal loop: $l = l + 1$.

Step 3: Reproduction loop: $kk = kk + 1$.

Step 4: Chemotaxis loop: $j = j + 1$.

(a) For each bacteria ($i = 1, 2, \dots, S$), take a chemotactic step as follows.

(b) Evaluate the function, $J(i, j, kk, l)$.

(c) Let $J_{last} = J(i, j, kk, l)$ to save this value since a better value may be found via a run.

(d) Tumble: generate a random vector $\Delta(i) \in R^n$ with each element $\Delta_m(i), m = 1, 2, \dots, S$, a random number on $[-1, 1]$.

(e): Move using equation (39) these results in a step of size $C(i)$ in the direction of the tumble for bacteria i .

$$\theta(i, j + 1, kk, l) = \theta(i, j, kk, l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \quad (39)$$

(f) Evaluate $J(i, j + 1, kk, l)$ with $\theta(i, j + 1, kk, l)$.

(g) Swim (swimming loop):

(g.1) let $mm = 0$ (counter for swimming loop)

(g.2) while $mm < N_s$ (if not go to (g.2.3)).

(g.2.1) let $mm = mm + 1$.

(g.2.2) if $J(i, j + 1, kk, l) < J_{last}$, let $J_{last} = J(i, j + 1, kk, l)$ Then, another step of size $C(i)$ in this same direction will be taken as (39) and use the new generated $\theta(i, j + 1, kk, l)$ to evaluate the new $J(i, j + 1, kk, l)$.

(g.2.3) else, $mm = N_s$.

J_{health} values will split into two bacteria, and the copies that are made are placed at the same location as their parent.

Step 7: If $kk < N_{re}$ go to Step 3, otherwise, go to Step 8.

Step 8: Elimination-dispersal: for each bacterium ($i = 1, 2, \dots, S$), eliminate and disperse each bacterium, which has a probability value, less than (P_{ed}), which results in keeping the number of bacteria in the population constant. To do this, if a bacterium is eliminated, simply disperse one to a random location on the optimization domain.

(h) Go to next bacterium ($i + 1$), if $i \neq S$ go to (b) to process the next bacterium.

Step 5: If $j < N_c$, go to Step 4, otherwise, go to Step 6.

Step 6: Reproduction.

(a) For the given kk and l , and for each bacterium ($i = 1, 2, \dots, S$) calculate the health of the bacteria as follows:

$$J_{health}^i = \sum_{j=1}^{N_c+1} J(i, j, kk, l) \quad (40)$$

After that sort bacteria in order of ascending values of J_{health} .

(b) The S_r bacteria with the highest J_{health} values will die, and the other S_r bacteria with the lowest

Step 9: If $l < N_{ed}$, then go to Step 2, otherwise, end.

The proposed Mean Square Error (MSE) function is chosen as criterion for evaluating the WMR performance as in equation (41):

$$E = \frac{1}{N} * \sum_{k=1}^N (x_r(k) - x(k))^2 + (y_r(k) - y(k))^2 + (\theta_r(k) - \theta(k))^2 + ((\tau_{rref}(k) - \tau_{TR}(k))^2 + ((\tau_{lref}(k) - \tau_{TL}(k))^2) \quad (41)$$

N : the max no. of samples. k : the current sample. τ_{rref} : the right reference torque. τ_{lref} : the left reference torque. τ_{TR} : the total right wheel torque. τ_{TL} : the total left wheel torque.

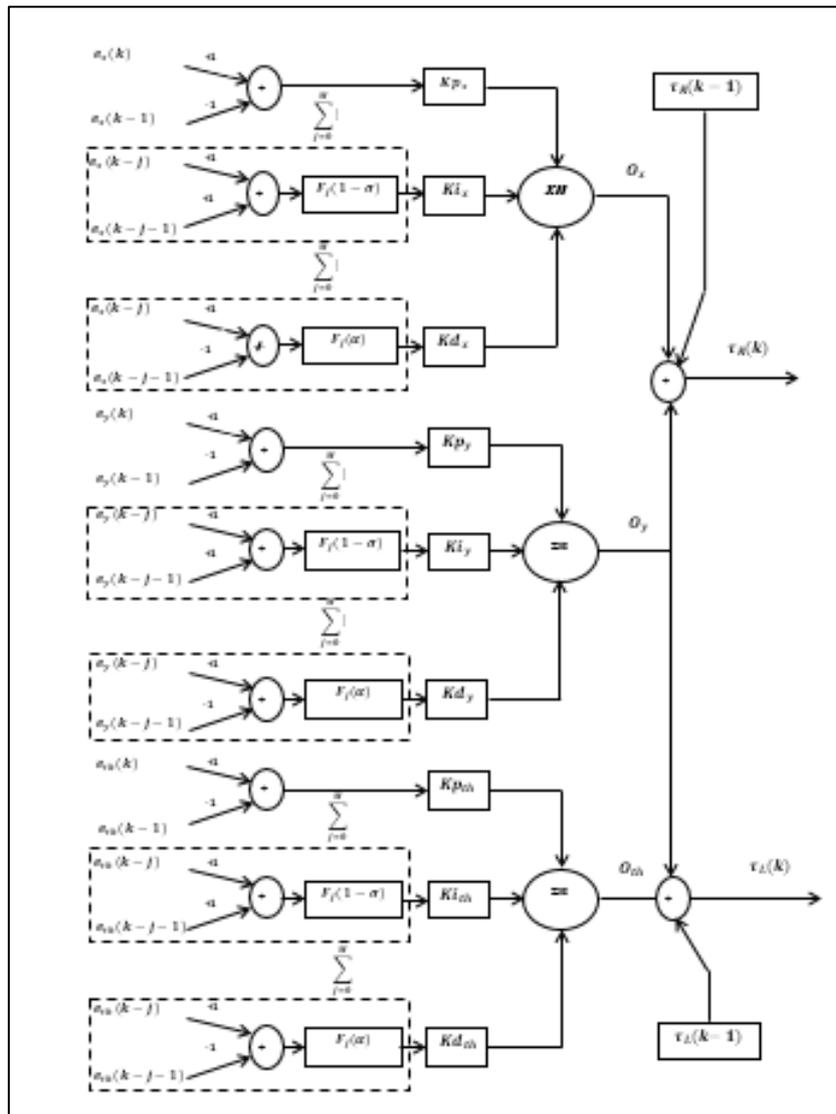


Figure 4: NFOPIDN controller structure

5. Simulation Results and Discussion

The simulation of the proposed controller is proved by using MATLAB program. The dynamic model of the differential-drive WMR explained in Section 2 is used. The simulation is executed off – line by tracking a desired position (x, y) and orientation (θ) with the (Ranunculoid-curve) path in the tracking control of the WMR. The parameter values of the mobile robot (Eddie WMR) are taken from [25]: m=11.5 kg, L=0.452 m, r=0.076 m, I=0.5873 kg.m² and sampling time equals to (0.2) sec. The NFOPIDN controller scheme in Figure (2) is applied to the WMR model and it is used the cognitive BFOA for finding and tuning the gains of the proposed feedback controller. The first step of operation is to set the following parameters of the BFOA: Population size S is equals to (10), maximum number of iteration equals to (12), number of chemotactic steps N_c equals to (14), swimming length N_s equals to (5), number of reproduction steps N_{re} equals to (4), number of elimination-dispersal events equals to (3) and the number of

elimination-dispersal probability equals to (0.25). The number of weight p in each bacterium equals to (11) because there are eleven parameters of NFOPIDN controller.

Case Study

The desired (Ranunculoid-curve) path is represented by the equations (42 - 44):

$$x_r(t) = 4\cos\left(\frac{2\pi t}{25}\right) - \cos\left(\frac{8\pi t}{25}\right) \tag{42}$$

$$y_r(t) = 4\sin\left(\frac{2\pi t}{25}\right) - \sin\left(\frac{8\pi t}{25}\right) \tag{43}$$

$$\theta_r(t) = 2 \tan^{-1}\left(\frac{\Delta y_r(t)}{\sqrt{(\Delta x_r(t))^2 - (\Delta y_r(t))^2 + \Delta x_r(t)}}\right) \tag{44}$$

For the simulation objective, the desired path is represented by equations (42 and 43) while the desired orientation angle is described by equation (44). The differential-drive WMR starts from initial posture $q(0) = [4,0,1.0472]$.

The Figures 5 and 6 are exhibited the excellent position and orientation tracking performance as compared with other works such as [7,8].

The simulation results showed the efficiency of the controllers by exhibiting their ability to

produce small and smooth values of the torques control actions and without spikes as depicted in Figures 7 and 8.

The sum of the feedforward and feedback control actions represent the total left and right torques of the mobile robot system and denoted by $\tau_{TL}(k)$ and $\tau_{TR}(k)$, respectively which are also smooth and did not exceed ± 3.078 (N.m) as saturation state and without spikes action as depicted in Figure 9.

The actions described in Figure (10) show that the DC motors of the WMR model are driven by a low power. The linear velocity of the WMR did not exceed 0.335 m/sec, and the angular velocity is equal to 0.463 rad/sec as depicted in Figure 11. Optimized-off-line-tuning based on BFOA is used for finding and tuning the gains of the NFOPIDN controller as demonstrated in Table 2. Finally, according to equation (25) the gains of the NFOPIDN controller are described in Table 3.

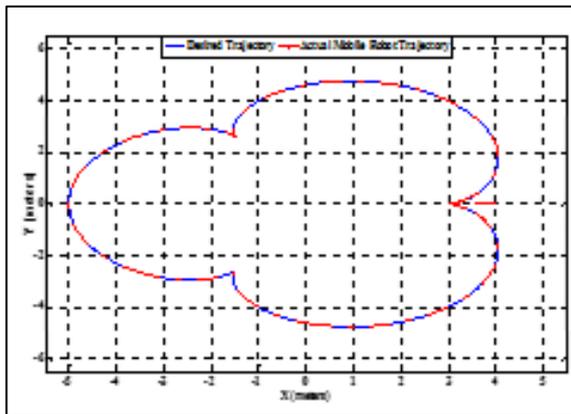


Table 3: The parameters of the nonlinear (FOPID) neural controller

Kp_x	0.2797
Ki_x	0.7614
Kd_x	0.8477
Kp_y	0.7177
Ki_y	0.2782
Kd_y	0.3294
Kp_θ	1.0707
Ki_θ	1.3720
Kd_θ	0.1827
α	0.1078
σ	0.8161

Figure 6: Desired trajectory and actual mobile robot trajectory

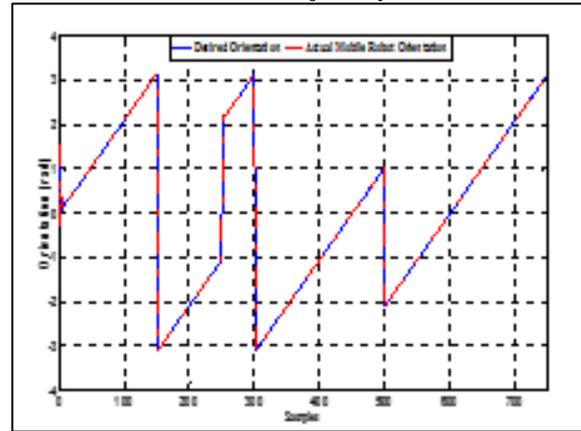


Figure 6: Desired orientation and actual mobile robot orientation

Table 2: Parameters of the proposed feedback controller using (BFOA)

Kp_x	0.2797
Ki_x	0.8008
Kd_x	0.8422
Kp_y	0.7177
Ki_y	0.2926
Kd_y	0.3273
Kp_θ	1.0707
Ki_θ	1.4437
Kd_θ	0.1816
α	0.1078
σ	0.8161

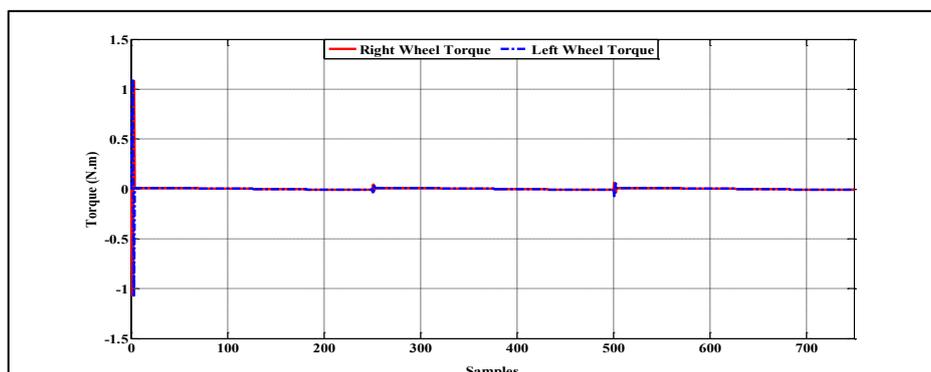


Figure 7: The feedforward torque control signal

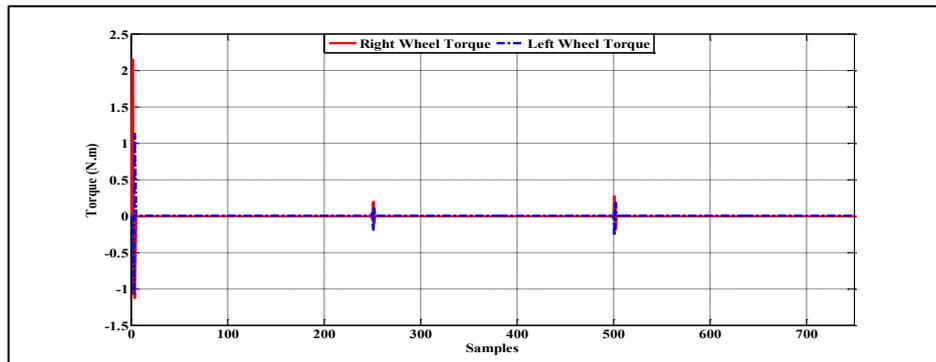


Figure 8: The feedback torque control signal

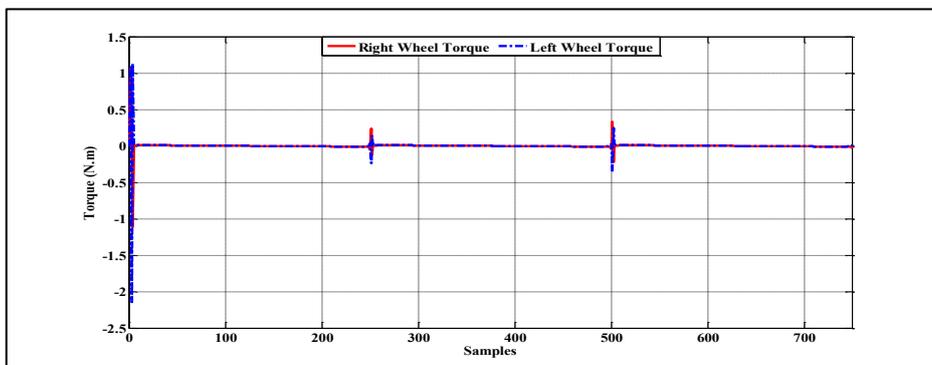


Figure 9: The sum of the feedforward and feedback control actions

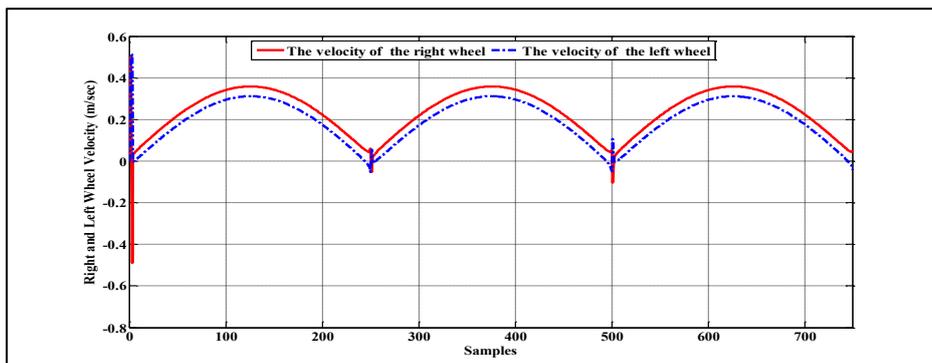


Figure 10: The right and left wheel action

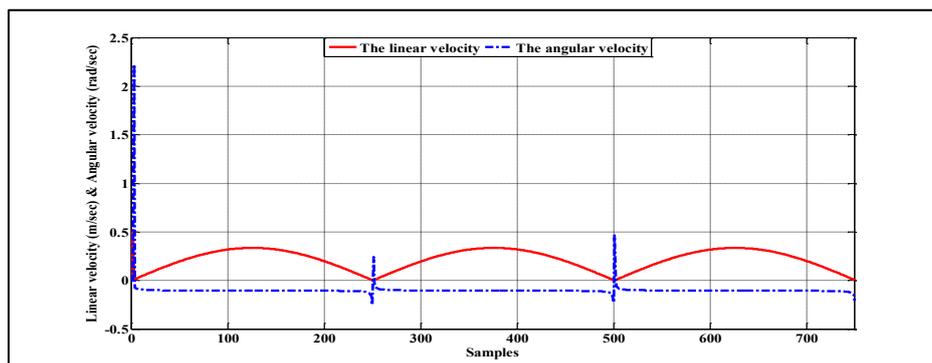


Figure 11: The linear and angular velocity

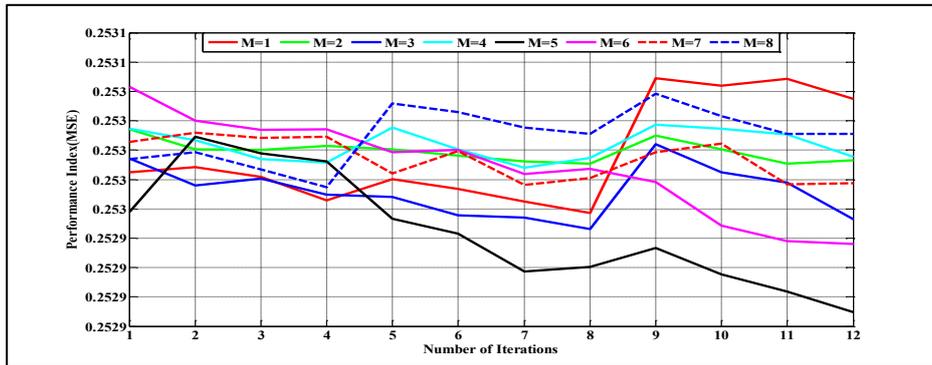


Figure 12): The performance indexes for different values of M (1, 2, 3, 4, 5, 6, 7 and 8)

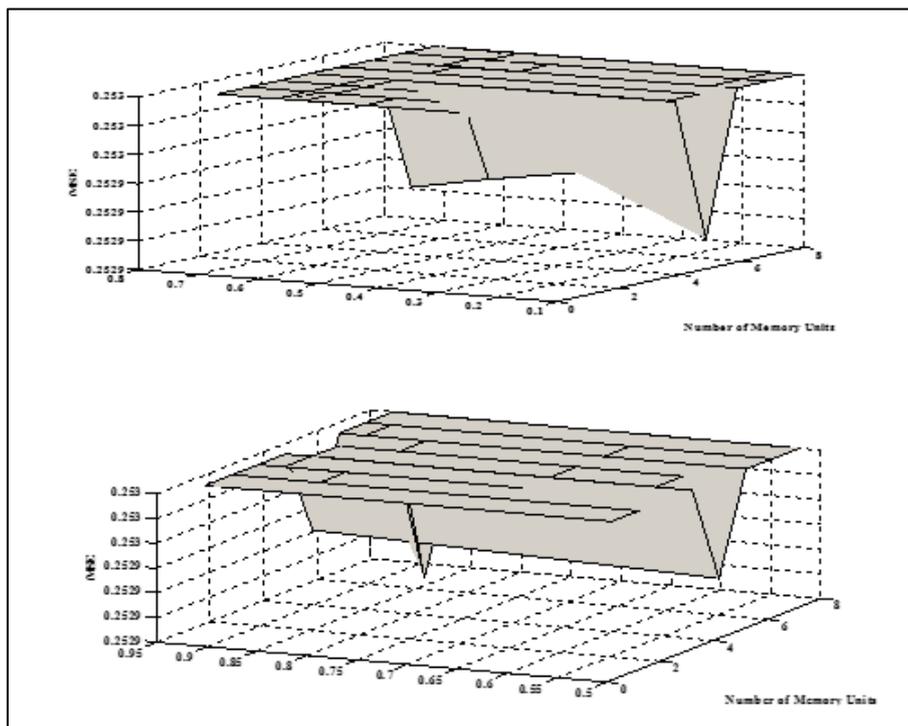


Figure 13: Memory units against minimum (MSE) (a) with the value of fractional derivative (α); (b) with the value of fractional integration orders (σ)

The performance index for different values of memory $M=1, 2, 3, 4, 5, 6, 7$ and 8 can be shown in the Figure (12). This Figure shows that the best number of memory units for the (Ranunculoid-curve) trajectory is equal to 5 . Figures (13-a and b) show that the best number of memory units needed against minimum value of (MSE) and the values of the fractional derivative (α) and fractional integration (σ) orders.

6. Conclusions

The Matlab simulation results on the off-line tuning BFOA for the nonlinear fractional order PID neural controller and feedforward controller are presented in this paper for the Eddie mobile robot model which shows preciously that the proposed controller's algorithm has the following:

- Fast and stable off-line tuning control parameters with a minimum number of memory units which is needed ($M=5$ for Ranunculoid-curve trajectory).
- Effective minimization capability of mean square tracking errors to follow a desired continuous gradients path.
- Efficiency of generating smooth and optimum suitable linear torque commands between $(-3.078$ to $3.078)$ N.m, without sharp spikes.

References

- [1] X. Yang, "Adaptive Trajectory Tracking Control for WMR Based on Cascade-design Method," *International Journal of u- and e- Service, Science and Technology*, Vol. 8, No. 12, pp. 265 – 276, 2015.
- [2] L. Gracia and J. Tornero, "Kinematic Control of Wheeled Mobile Robots. *Latin American Applied Research*," Vol. 38, pp. 7-16, 2008.
- [3] D.N. Abadi and M.H. Khooban, "Design of Optimal Mamdani-type Fuzzy Controller for Non-holonomic Wheeled Mobile Robots," *Journal of King Saud University, Engineering Sciences*, Vol. 27, 2015, pp. 92-100.
- [4] Y-C. Yeh, T-H. Li and C-Y. Chen, "Adaptive Fuzzy-Sliding Mode Control of Dynamic Model Based Car-Like Mobile Robot," *International Journal of Fuzzy Systems*, Vol. 11, No. 4, pp. 272-286, 2009.
- [5] M. Yue, W. Sun and P.Hu, "Sliding Mode Robust Controller for Two Wheeled Mobile with Lower Center of Gravity," *International Journal of Innovative Computing, Information and Control*, Vol. 7, No. 2, pp. 637-646, 2011.
- [6] F.G. Rossomando, C. Soria and R. Carelli, "Sliding Mode Neuro Adaptive Control in Trajectory Tracking for Mobile Robot," *Journal of Intelligent & Robotic Systems*, Vol. 74, Issue, 3-4, pp.931-944, 2013.
- [7] A.S. Al-Araji, "Design of On-Line Nonlinear Kinematic Trajectory Tracking Controller for mobile robot based on Optimal Back-Stepping Technique," *Iraqi Journal of Computers, Communications, Control and systems Engineering (IJCCCE)*, Vol. 14, No. 2, pp. 25-36, 2014.
- [8] A.S. Al-Araji, "Development of Kinematic Path-Tracking Controller Design for Real Mobile Robot via Back-Stepping Slice Genetic Robust Algorithm Technique," *Arabian Journal for Science and Engineering*, Vol. 39, No. 12, pp. 8825-8835, 2013.
- [9] A.S. Al-Araji, M.F. Abbod and H.S. Al-Raweshidy, "Design of an Adaptive Nonlinear PID Controller for Nonholonomic Mobile Robot Based on Posture Identifier," *Proceedings of the 2011 IEEE International Conference on Control System, Computing and Engineering (ICCSCE 2011)*. Penang, Malaysia, Nov. 25-27, pp. 337-342, 2011.
- [10] K.E. Dagher and A.S. Al-Araji, "Design of a Nonlinear PID Neural Trajectory Tracking Controller for Mobile Robot based on Optimization Algorithm," *Engineering and Technology Journal*, Vol. 32, No. 4, pp. 973-986, 2014.
- [11] A.S. Al-Araji, "A Comparative Study of Various Intelligent Algorithms Based Nonlinear PID Neural Trajectory Tracking Controller for the Differential Wheeled Mobile Robot Model," *Journal of Engineering*, Vol. 20, No. 5, pp. 44-60, 2014.
- [12] A.S. Al-Araji and K. E. Dagher, "Cognitive Neural Controller for Mobile Robot," *Iraqi Journal of Computers, Communication and Control & Systems Engineering*. Vol.15, No.1, 2015, pp. 46-60.
- [13] A. S. Al-Araji, "A Cognitive PID Neural Controller Design for Mobile Robot Based on Slice Genetic Algorithm", *Engineering and Technology Journal*, Vol. 33, No. 1, pp. 208-222, 2015.
- [14] F. Arvin, K. Samsudin and M. A. Nasser, "Design of Differential-Drive Wheeled Robot Controller with Pulse-Width Modulation", *Conference on Innovative Technologies in Intelligent Systems and Industrial Applications*, pp. 143-147, 2009.
- [15] G. Oriolo, A. De Luca and M. Vendittelli, "WMR Control via Dynamic Feedback Linearization: Design, Implementation, and Experimental Validation", *IEEE Transactions on Control Systems Technology*, Vol. 10, No. pp. 835-852, 2002.
- [16] T.T. Mac, C. Copot, R. D. Keyser and T. Vu, "MIMO Fuzzy Control for Autonomous Mobile Robot," *Journal of Automation and Control Engineering*, Vol. 4, No. 1, Feb. 2016.

- [17] Spyros G. Tzafestas, "Introduction to Mobile Robot Control. London, Waltham, MA: Elsevier, 2014.
- [18] M. Brezak, I. Petrović and N. Perić, "Experimental Comparison of Trajectory Tracking Algorithm for Nonholonomic Mobile Robots," Industrial Electronics, 2009, IECON '09, 35th Annual Conference of IEEE, pp. 2229 – 2234, 2009.
- [19] M. Asif, M. Safwan, M. J. Khan and M. Rehan, "Feedforward and Feedback Kinematic Controllers for Wheeled Mobile Robot Trajectory Tracking," *Journal of Automation and Control Engineering*, Vol. 3, No. 3, pp. 178-182, 2015.
- [20] A.S. Al-Araji, "Design of a Cognitive Neural Predictive Controller for Mobile Robot," PhD thesis, Brunel University, United Kingdom, 2012.
- [21] I. Podlubny, "Fractional Differential Equations", Academic Press, San Diego, 1999.
- [22] S. Das, A. Biswas, S. Dasgupta and A. Abraham, "Bacterial Foraging Optimization Algorithm: Theoretical Foundations, Analysis, and Applications," Foundations of Computational Intelligence Vol. 203 of the series Studies in Computational Intelligence, pp. 23-55.
- [23] M.A. Muñoz, S. K. Halgamuge, W. Alfonso and E.F. Caicedo, "Simplifying the Bacteria Foraging Optimization Algorithm," WCCI 2010 IEEE World Congress on Computational Intelligence Jul. 18-23, 2010- CCIB, Barcelona, Spain, pp. 4095-4101, 2010.



- [24] A.J. Humaidi, "Tuning of PID Controller Based on Foraging Strategy for Pneumatic Position Control System," *Iraqi Journal of Computers, Communication and Control & Systems Engineering*, Vol.10, No.1, pp. 107-120, 2010.
- [25] Internet website <http://parallax.com/>. Robotics with the Eddie Mobile Robot text manual. Accessed Date Nov. 2015.

Author(s) biography

I have finished my B.Sc. degree in Control and Systems Eng. from the University of Technology in 1997 and M.Sc. degree in Mechatronics Engineering for the University of Technology in 2001. I have finished my PhD degree in Electronics and Computers in the School of Engineering and Design, Brunel University, London, UK in 2012. Currently I am a head of Mechatronics Eng. Dept. at University of Technology, Baghdad - Iraq. My current project is Cognitive Methodologies.

Luay Thamir Rasheed.

I finished my B.Sc. degree in Control and Systems Eng. from the University of Technology in 2007 and M.Sc. degree in Mechatronics Engineering for the University of Technology in 2016. Currently I am assistant lecturer in Mechatronics Eng. Dept. at University of Technology, Baghdad - Iraq. My current project is Cognitive Methodologies.

