Al-Khwarizmi Engineering Journal,Vol. 12, No. 1, P.P. 45- 60 (2016)

Al-Khwarizmi Engineering Journal

Design of a Kinematic Neural Controller for Mobile Robots based on Enhanced Hybrid Firefly-Artificial Bee Colony Algorithm

Nizar Hadi Abbas* **Basma Jumia saleh ****

,Department of Electrical Engineering / College of Engineering/ University of Baghdad*

* Email: drnizaralmsaodi@gmail.com

**Email: eng.basmaj@gmail.com

(Received 6 April 2015; accepted 21 October 2015)

Abstract

The paper present design of a control structure that enables integration of a Kinematic neural controller for trajectory tracking of a nonholonomic differential two wheeled mobile robot, then proposes a Kinematic neural controller to direct a National Instrument mobile robot (NI Mobile Robot). The controller is to make the actual velocity of the wheeled mobile robot close the required velocity by guarantees that the trajectory tracking mean squire error converges at minimum tracking error. The proposed tracking control system consists of two layers; The first layer is a multi-layer perceptron neural network system that controls the mobile robot to track the required path , The second layer is an optimization layer ,which is implemented based on hybrid Crossoved Firefly Algorithm with Artificial Bee Colony (CFA-ABC) to tune the controller's parameters to achieve the optimal path. The performance of the hybrid optimization algorithm is verified by various benchmark functions. The simulation results show that the utilizing of CFA and (CFA-ABC) are better than the original Firefly Algorithm. A simulation example is given to indicate the effectiveness of the proposed algorithm, the results have been done using MATLAB (R2013b), and all trajectory tracking results with two reference trajectories (circular and lemniscates) are presented.

Keywords: *Mobile Robot, Trajectory Tracking, Neural Networks, Kinematic Controller, National Instrument, Firefly Algorithm, Artificial Bee Colony Algorithm.*

1. Introduction

In recent years a plethora of research has been carried out on the control problem of the mobile robotic systems. This is mainly due to the growing application of these systems both in industrial and service environments [1].Intelligent wheeled mobile robots are the subject of the technical interest arising from possibility of practical application in: manufacturing, civil engineering, transportation, agriculture, space exploration, deep sea penetration, help for disable, medical surgery and in the other sectors of science and technology [2].

The pathing control problems of a nonholonomic wheeled mobile Robots have attracted considerable attention, there are two obvious approaches tracking control problems: Trajectory tracking and path planning. Kanayama proposed a method to design a trajectory tracking controller that meet a specific requirements. Therefore, for the trajectory tracking problem the kinematic model are suggested to close the target control objective [3].

Various control systems have been enhanced in order to serve as a robust control rule for mobile robot systems. The control algorithm of any mobile robot depends either on the dynamic model or the kinematic model of the mobile robot. This research depends on kinematic controller, the main purpose of the kinematic controller is to find out a near-optimum velocity for the mobile robot in order to reduce the error between the required path and the reference path of the mobile robot system [4].

This paper deal with the use of the Artificial Neural Networks (ANNs) because of their learning and adaptation capabilities. ANNs are one of the almost popular intelligent techniques generally applied in engineering for systems which are time variant. Additionally, their ability to learn complex input-output mapping, without detailed analytical model, approximation of nonlinear function and robustness for noise environment make them an ideal choice for real implementations [5]. Furthermore, its lateral structure makes the neural network achievement rapid than the achievement of other classical artificial intelligence techniques such as a Genetic algorithm.

In engineering problems, the optimization is to find out a solution that can maximize or minimize an objective function. Nowadays, stochastic method is more often used to deal with optimization problems [6]. Though there are several ways to organize them, a simple one is used to divide them into two groups conforming to their nature: deterministic and stochastic. Deterministic algorithms can bring the same solutions if the initial conditions are unaffected, because they always follow the accurate move. However, disregard of the initial values. Stochastic ones are based on convinced stochastic distribution; therefore they generally generate many solutions. In fact, both of them can find satisfactory solutions after some generations.

Recently, nature-inspired algorithms are prove its capability in solving numerical optimization problems more comfortably. These metaheuristic ways are developed to solve difficult problems, such permutation flow shop scheduling [7], reliability, high-dimensional function optimization , and other engineering problems .Nowadays, many other metaheuristic methods have occurred, like Ant Colony Optimization (ACO) , Harmony Search (HS), Particle Swarm Optimization (PSO), Firefly Algorithm (FA) , and Artificial Bee Colony (ABC) [8].

2. Kinematic Model of Differential Wheeled Mobile Robot

The aim of the wheeled mobile robot kinematic model is to find the mobile robot velocity in the inertial body as a function of the wheels velocities and the configuration coordinates of the robot. i.e. Authorized the robot speed $\dot{q} = [\dot{x}\ \dot{y}\ \dot{\theta}]^T$ as a function of the wheel translational velocities $\dot{\varphi}_R$ and $\dot{\varphi}_L$. The mobile robot kinematics commonly has two main analyses, Forward kinematics and feedback kinematics [10]:

- Forward mobile robot kinematics:

$$\dot{q}_l = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = f(\dot{\varphi}_R, \dot{\varphi}_L) \qquad \ldots(1)$$

- Feedback mobile robot kinematics:

$$\begin{bmatrix} \dot{\varphi}_R \\ \dot{\varphi}_L \end{bmatrix} = f(\dot{x}, \dot{y}, \dot{\theta}) \qquad \ldots(2)$$

The graphic of the differential wheeled mobile robot model shown in Fig. 1, consists of a cart with two standard wheels for motion on the same axis and castor wheel in the front of cart for stability. The castor wheel carries the mechanical format and keeps the robot more stable [8]. Suppose a differential wheeled mobile robot structure which has two wheels with the radius of R_a based with a length $L1/2$ from the mobile robot center:

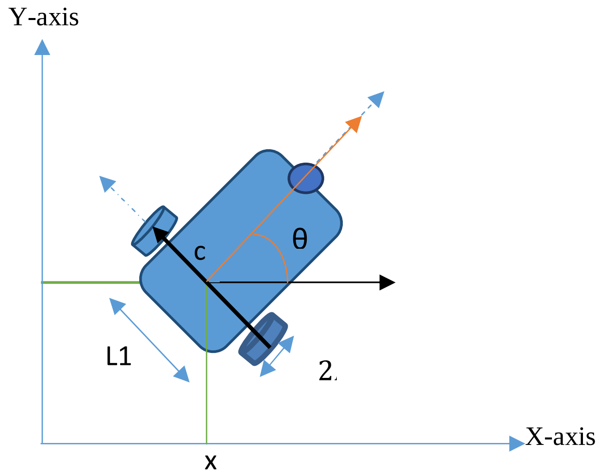

Fig.1. The differential drive mobile robot model.

At each instant in time, the left and right wheels follow a path as a show in Fig. 2.7 that moves around the instantaneous center of curvature mobile robot (ICCM) with the same angular rate [9].

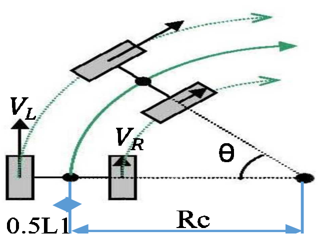

Fig. 2. The instantaneous center of curvature mobile robot.

$\Omega(\tau) = \frac{d\theta(\tau)}{d\tau}$ …(3)

Thus:

$V(\tau) = \Omega(\tau)Rc(\tau)$ …(4)

$V_L(\tau) = \left(\text{Rc}(\tau) + \frac{L1}{2}\right)\Omega(\tau)$ …(5)

$V_R(\tau) = (\text{Rc}(\tau) - \frac{L1}{2})\,\Omega(\tau)$ …(6)

After solving equations 2.3 and 2.4, then find the instantaneous center of curvature mobile robot (ICCM) trajectory relative to the mid-point axis (c) is given as equation 2.5 [9].

$Rc(\tau) = \frac{L1\,(V_L(\tau)+V_R(\tau))}{2\,(V_L(\tau)-V_R(\tau))}$ …(7)

The angular velocity of the mobile robot is [11]:

$\Omega(\tau) = \frac{(V_L(\tau)-V_R(\tau))}{L1}$ …(8)

And the linear velocity of the mobile robot is [11]:

$V(\tau) = \frac{(V_L(\tau)+V_R(\tau))}{2}$ …(9)

It is simulated that the wheeled mobile robot below the nonholonomic constraint, which is ideal rolling no skidding [11], as shown in (2):

$-\,\dot{x}(\tau)\,sin\theta\,(\tau) + \dot{y}\,(\tau)\,cos\theta\,(\tau) = 0$ … (10)

The kinematics equations of differential wheeled mobile robot is represented as follows [11]:

$\dot{x}(\tau) = V(\tau)\cos\theta(\tau)$ ... (11)

$\dot{y}(\tau) = V(\tau)\sin\theta(\tau)$ ... (12)

$\dot{\theta}(\tau) = \Omega(\tau)$ ... (13)

In order to obtain the position and orientation of the wheeled mobile robot, it is required to integrate equation as showed below [12]:

$x(\tau) = x_{00} + \int_0^\tau V(\tau)\cos\theta(\tau)d\tau$ … (14)

$y(\tau) = y_{00} + \int_0^\tau V(\tau)\sin\theta(\tau)d\tau$ … (15)

$\theta(\tau) = \theta_{00} + \int_0^\tau \Omega(\tau)d\tau$ … (16)

In the computer simulation , the form of the pose (position/orientation) equations as follows[11]:

$xx(L) = 0.5[V_R(L) + V_L(L)]cos\theta(L)\Delta t + x(L-1)$ … (17)

$yy(L) = 0.5[V_R(L) + V_L(L)]sin\theta(L)\Delta t + y(L-1)$ … (18)

$$\theta\theta(L) = \frac{1}{R_a}[V_R(L) + V_L(L)]\Delta t + \theta(L-1)$$ … (19)

Where $xx(L), yy(L), \theta\theta(L)$ are the elements of the pose (position/orientation) at L and $\Delta\tau$.

3. The Trajectory Tracking Controller

The suggested building of the nonlinear kinematic neural controller may be addicted in the style of block diagram, as shown in Fig. 3. The access to control the wheeled mobile robot build upon the accessible information of the unknown nonlinear system and also be known by the input-output information and the control design. The hybrid Crossoved firefly with Artificial bee colony (CFA-ABC) optimization will produce the optimal Parameter for the kinematic neural controller to get best velocity control signal that will minimize the pathing error of the mobile robot.

The feedback kinematic controller is vital to keep the trajectory pathing error of the wheeled mobile robot system when the path of the robot obtained from the required trajectory event transient state. The exploration of the wheeled mobile robot starts with generating the desired trajectory where $q_r = [x_r\,, y_r\,, \theta_r]$ is the desired pose (position/ orientation) of the wheeled mobile robot [11].

This desired pose (position/ orientation) is used by the feedback kinematic of the wheeled mobile robot to produce the needed linear and angular velocities for the described trajectory with the needed orientation. The error vector is then mathematical calculation by examining the desired pose (position/ orientation) of the wheeled mobile with the real pose of robot structure by using the rotational matrix. This error signal is then used by the controller to generate a control action to obligate the wheeled mobile robot to follow the desired path [11,12].

The kinematic illustrative of the wheeled mobile robot can be written in the model:

$$\begin{bmatrix}\dot{x}(\tau)\\ \dot{y}\,(\tau)\\ \dot{\theta}\,(\tau)\end{bmatrix} = \begin{bmatrix}\cos\,(\theta(\tau)) & 0\\ \sin\,(\theta(\tau)) & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}V\\ \Omega\end{bmatrix}$$ ... (20)

Let $(x_r, y_r)$is the desired position of the wheeled mobile robot, and θ_r is the desired orientation of the wheeled mobile robot, so that the desired posture vector is $q_r = [x_r, y_r, \theta_r]^T$. The real position of the wheeled mobile robot is pretended to be (x, y) and the real orientation of the wheeled mobile robot is assumed to be θ, so that the real posture of the mobile robot is $q_l = [x, y, \theta]^T$.

Then the error $e = [ex, ey, e\theta]^T$can be calculated in the global modal:

$$\begin{bmatrix} ex \\ ey \\ e\theta \end{bmatrix} = \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} \quad \ldots (21)$$

The error vector e needs to be transformation into the wheeled mobile robot modal by using the rotation matrix Ŗ [11]:

The kinematic neural controller consists of two layer: Neural network topology and Optimization algorithms.

3.1 The First Layer: Neural Network Topology

The Neural Network is an arithmetical model inspired by biological neural networks. A neural network made up of an interconnected collection of neurons, and it progresses information using a connectionist way to computation. The Solution of problem of control wheeled mobile robot requires application of complex methods. Because of lack of a systematic approach to analysis and synthesis
of control of nonlinear systems so far , the artificial neural networks became an attractive tool used in theory of nonlinear systems. Multi-layer Perceptron (MLP) Neural networks owe their popularity to properties like: possibility of approxi

$$em = Ŗ * e$$

$$\begin{bmatrix} exm \\ eym \\ e\theta m \end{bmatrix} = \begin{bmatrix} \cos(\theta(\tau)) & \sin(\theta(\tau)) & 0 \\ -\sin(\theta(\tau)) & \cos(\theta(\tau)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} ex \\ ey \\ e\theta \end{bmatrix} \quad \ldots (22)$$

The error vector *em* is augment into the controller and at another time it is the accountability of the controller to generate the desired linear and angular velocities to minimize the error [12].

-mation of arbitrary nonlinear mappings, and ability of learning and adaptation [13, 14].

Let as consider Multi-layer Perceptron (MLP) Neural networks shown in Fig.4.

The number of the hidden layer nodes can be changed based on the required efficiency of the MLP neural network. The acceptable number of hidden layer nodes in the beginning can be taken equal to (*2m+1*) from [12].

The outturn of the MLP neural network is considering as follow:

$$V_R(L-1), V_L\,(L-1) = \sum_{J=1}^{mj} hntj * w_{kj} - Bk2 \quad \ldots (23)$$

Where *hntj* is the output of the hidden layer node *j* and is being Considerate as:

$$hntj = F(dj) \quad \ldots (24)$$

where:

$$dj = \sum_{i=1}^{mi} inputi * V_{ji} - Bj1 \quad \ldots (25)$$

The outputs of the MLP neural networks that can be reached from the activation function, which is bipolar sigmoid function is as follows:

$$F(dj) = \frac{2}{1+e^{d_j}} - 1 \quad \ldots (26)$$

The neural network weights (control parameter) Vj ,Wkj, $Bj1$ and $Bk2$ are adapted using optimization algorithms, describe as the next section [11].

3.2 The second layer: optimization algorithm

3.2.1 Firefly Algorithm

Now delineated the primary parts of the Firefly Algorithm developed by Xin-She Yang at Cambridge University [8] is briefly described. The FA is inspired from the flash model and features of fireflies. In order to describing the algorithm in the simple manner the following three idealised rules are used:

1) Whole fireflies are unisex so that each firefly will be appealed to other fireflies careless of their sex.
2) Appealingness is relative to their luminousness, thus for any two blinking fireflies, the less shinier one will move towards the shinier one. The appealingness is relative to the luminousness and they both reduction as their distance increments. If there is no shinier one than a appropriate firefly, it will be motion randomly.
3) The luminousness of a firefly is determined by the mural of the objective function.

In the firefly algorithm, there are two significant events: the fluctuation of flash intensity and creation of the luminousness. For easiness, the luminousness of a firefly was determined by its shininess which in movement is related with the objective function. In the clear form, the flash intensity $F(rr)$ changes consorting to the reverse square law($F(rr) = \frac{F_s}{rr^2}$) [8].

For a given modal with a fixated flash absorption coefficient γ, the flash intensity F changes with the distance rr [22], that is

$$F = F_0 e^{-\gamma rr} \quad \ldots (27)$$

As a firefly's luminousness is relative to the flash intensity seen by neighboring fireflies, the luminousness βL of a firefly was specified by [8]:

$$\beta L = \beta L_0 e^{-\gamma r r^2} \quad ...(28)$$

The distance between any two fireflies $i1$ and $j1$ at xl_{i1} and xl_{j1}, specifically, is the Cartesian form [22]:

$$ri1j1 = \|xl_{i1} - xl_{j1}\| = \sqrt{\sum_{M=1}^{d}\left(xl_{i1,M} - xl_{j1,M}\right)^2} \quad ...(29)$$

Where $xl_{i,M}$ is the Mth component of the relating coordinate xl_i of ith firefly, In the 2-D case as shown in eq. (30), from [22]:

$$rri1j1 = \sqrt{(xl_{i1} - xl_{j1})^2 + (yl_{i1} - yl_{j1})^2} \quad ...(30)$$

The motion of a firefly $i1$ is appealed to another more shinier (brighter) firefty j1 by [22]:

$$xl_{i1} = xl_{i1} + \beta L_0 e^{-\gamma r r_{ij}^2}\left(xl_{j1} - xl_{i1}\right) + \alpha 1\ (rand - 0.5) \quad ...(31)$$

Where the second term is due to the attractive feature. The third part is randomisation with a hold parameter $\alpha 1$, which makes the investigation of the search distance more effective [15], after few imitations, use $\beta_0 = 1, \alpha 11 \in [0,1], \gamma = 1$,,for most operation.

3.2.2 Crossover Firefly Algorithm (CFA)

In standard Firefly, if there no brighter one than a particular firefly, the fireflies will move randomly, and by using Flat crossover algorithms the updating mechanism at each iteration by the coefficient delta is a uniform distributed random variable within optional interval [0,1]. The flat crossover coefficient (alpha) is defined as follows:

$$Alpha = (delta) * \left(alpha^{iteration}\right) \quad ...(32)$$

Where alpha changing at each iteration. And the Flat crossover operate based on the following equation:

$$xl_{i1} = (alpha) * xl_{j1} + (1 - alpha) * xlj1 \quad ...(33)$$

Instead of fireflies move randomly using the possibility of crossover, then the candidates for the best solution are increasing. On the other hand, FA become needs less time to search for the best solution and its performance significantly developments with the increases the population size because of decreasing the randomness.

3.2.3 Artificial Bee Colony(ABC)Algorithm

Artificial Bee Colony (ABC) algorithm invented by D. Karaboga [20] is a relatively new population-based algorithm; it is a nature-inspired meta- heuristic algorithm, which inspire the foraging action of bees. ABC as a stochastic performance is clear to employment, and could easily be modified and hybridized with other meta- heuristic algorithms [16]. In real colonies, including bees, some tasks are performed by specified individuals. These specified bees try to expand the nectar quantity stored in the hive using efficient organization.

The ABC algorithm simulation represented by three kinds of bees: employed bees, onlooker bees and scout bees. Divided of the colony includes of employed bees, and the other half made up of onlooker bees. More bees should send to high-quality sources, and fewer bees should be sent to low-quality sources or even deserted. At the starting, the colony sends employed bees to productive flower patches. These bees catch flower's nectar and take them to the hive. When they return to the hive, empty their nectar and go to a desired area in the hive called ‘dance floor’ to share their data with other bees. The connections is executed by a especial dance. If the place visited by an employee is adjacent, it executes "round dances" in the hive, and if not, the bee executes a "waggle dance". Round dance include data about the nectar quality of a visited flower patches so the other bees can discovery its position by their smelling sense when they escape of the hive.

Waggle dance carries three parts of information about the flower patch: its guidance with regard to Sun, its length from the colony, and the nectar quantity [17]. Onlooker bees lookout the dances in the hive and select the good flower patches to go for. Indeed, flower patches which have greater qualities appeal more bees than smaller quality flower patches. Employed bees that their travelled to place is abandoned through low quality have two choices: start the dance floor and watch dances of other bees then go to a flower patch as an onlooker bee, or it search about the hive automatically as a scout bee for a new food source due to few inner inspiration or potential outer hint [18].

ABC algorithm can be summarized as in the following subsections [18]:

3.2.3.1 Population Initialization

A population of *FM1* individuals is produced randomly, which is equalise to the number of food sources. Each solution $xl_{i1}(i1 = 1, 2\ldots FM1)$ describing an individual is a *D1*-dimensional vector. hither. Each solution can be produced by Eq. (34) [19]:

$$xl_{i1j1} = xl_{j1}^{min} + (xl_{j1}^{max} - xl_{j1}^{min}).rand(0,1) \quad \ldots(34)$$

Where $i1^q = 1,2,\ldots\ldots,FM1,\ j1^q = 1,2,\ldots\ldots,D1$.

Next, the fitness of each food source is evaluated by

$$fitness\ i1 = \begin{Bmatrix} \frac{1}{1+fi1} & if\ fi1 \geq 0 \\ 1 + |fi1| & if\ fi1 < 0 \end{Bmatrix} \quad \ldots (35)$$

3.2.3.2 Employed Bee Stage

At this stage, a new nominee solution vl_{i1j1} is produced for the employed bee of food source xl_{i1} (only one parameter of the solution is updated by using Eq. (35) :

$$vl_{i1j1} = xl_{i1j1} + \emptyset_{i1j1}.\left(xl_{i1j1} - xl_{k1j1}\right) \quad \ldots(36)$$

where $k1 \in [1 \ldots.FN1]$ *and* $j1 \in [1 \ldots\ldots D1]$are arbitrary numbers, and $k1$ has to be other from $i1$, $\emptyset_{i1j1}$. After creating a new food source vl_{ij}, it will be evaluated and compared with xl_{i1j1} immediately. Then a greedy selection action is applied by comparing the fitness of both solutions produced via Eq. (35). If the fitness of solution vl_{i1j1} is more well than or equal to that of the solution $xl_{i1j1,}$, xl_{i1j1} will be replaced with vl_{i1j1} and the individual vl_{i1j1}will become a new member of the population and $(Countl_{i1)}$is reset. Otherwise, food source xl_{i1j1} is kept unchanged and $(Countl_{i1)}$is increased by 1.

3.2.3.3 Onlooker Bee Stage

Later all employed bees completed the looking for operation; each onlooker bee chooses an employed bee to improve its solution with a probability value, which is calculated by roulette wheel using Eq. (37).

$$pi1 = \frac{fitness\ i1}{\sum_{i1=1}^{FN1} fitness\ i1} \quad \ldots (37)$$

Where $fitness\ i1$, which is correlative to the nectar quantity of that food source. Apparently, the greater the $i1$, the extra the probability of selecting the $i1th$ food source is. In the onlooker bees' stage, an artificial onlooker bee chooses its food source determined by the above observed probability value. Thereafter, a food source for an onlooker bee is selected, a new region source is determined according to the Eq. (36) and its fitness value is integrated. Just as in the employed bees stage, a greedy selection is given.

3.2.3.4 Scout Bee Stage

The abandonment counter $(Countl_{i1)}$ of all employed bees are tested with a preset number of trials, called $limit1$. The employed bee, which cannot improve self-solution until the abandonment counter contacts to the limit1, becomes scout bee. Then a new solution is calculated for the scout bee by using Eq. (35) and the abandonment counter is reset. The scout bee, which a solution was produced for itself, becomes the employed bee. Therefore, scout bees in ABC prevent stagnation of employed bee population.

3.2.4 Crossoved Firefly -Artificial Bee Colony (CFA-ABC) Algorithm

The meta-heuristic methods are more powerful for the search of global solution for complicated problems and it is better than deterministic algorithm. Though their disadvantage, e.g., the time of convergence which is due the high number of the population size and repetition, and in order to deal with this problem two meta-heuristic methods are joined, the crossoved firefly and the artificial bee colony algorithm with a lesser symbol of bees and fireflies as possible.

This paper introduce a hybrid design which has two search steps. The first step is a search by Crossoved Firefly Algorithm (CFA) and second step is searching by Artificial Bee Colony algorithm (ABC) [22].

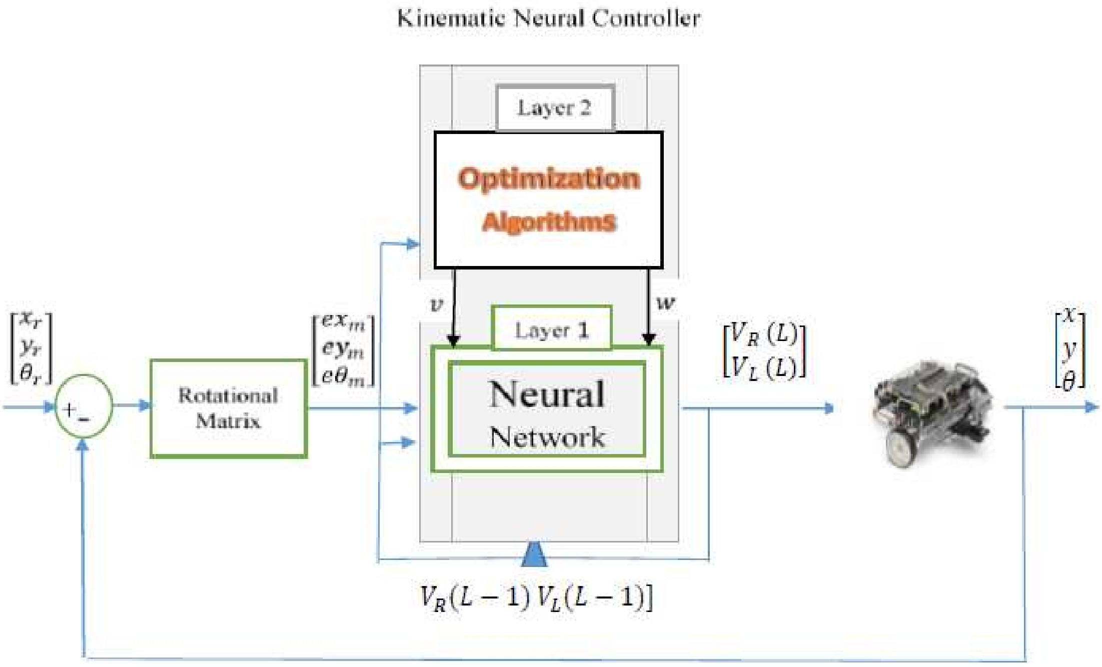

Fig. 3: Structure of the Kinematic Neural Controller.

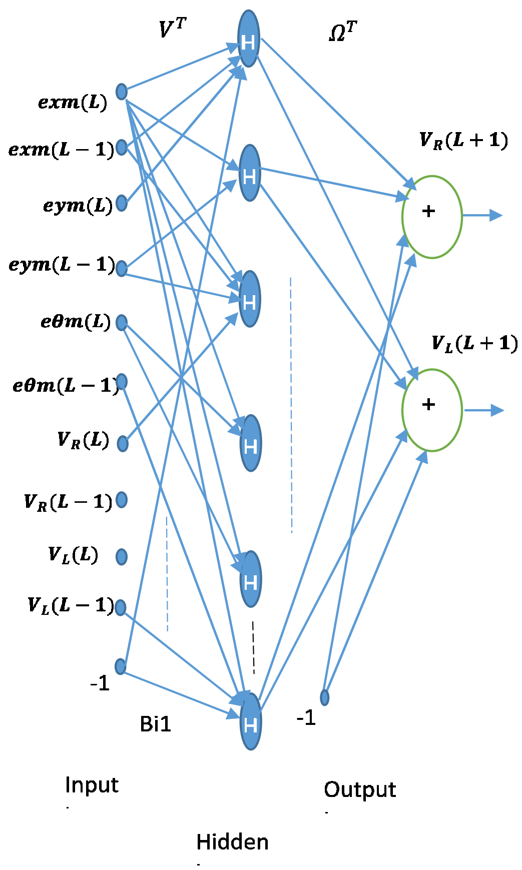

Fig. 4.The structure of Multi-layer Perceptron (MLP) Neural networks.

The summarization of the computation procedure of hybrid method is described by the pseudo code below[22]:

Step1: Initial pop of fireflies xl_{i1} , $i1 = 1,2, \dots \dots, S$.
Step 2: Evaluate Objective function $f(x_{i1})$ for each firefly.
Step 3: Set flash absorption coefficient γ , $\beta L_0, \alpha 1$.
Step 4: Flash intensity F_{i1} at xl_{i1} is calculated by $f(xl_{i1})$.
Step 5: while t < Max Iteration.
Step 6: calculate the value of alpha by Eq. (32)
Step 7: For i1=1: S for all S fireflies.
For j1=1: S for all S fireflies.
Step 8: if $(F_{j1} > F_{i1})$
Movement all firefly i1 towards j1 (d-Dimension) by eq. (31) Concording to the luminousness between i1 and j1 can be determined by eq. (30)
Step 9: else
Movement all fireflies by Eq. (33) .
Step 10: end if.
Step 11: valuate the new fireflies and update flash intensities.
Step 12: end for j ,end for i1
Step 13: Order the fireflies and find the best.
Step 14: end while.

The best solution found by CFA are regarded as initial point for ABC.

Step 15: Evaluate the fitness function for each employed bees by Eq. (35).

Step 16: While (termination criterion not satisfied).

Employed bees stage

Step 17: For each employed bee.

create new food source positions by Eq. (36).

use greedy selection mechanism.

Step 18: End for.

Calculate the probability pl_{i1} values for the solution xl_{i1j1} by the Eq. (37).

Onlooker bees stage.

Step 19: For each Onlooker bee.

Chooses a food source xl_{i1j1} depending on create new food source positions by Eq. (36).

use greedy selection mechanism.

Step 20: End for.

Scout bee stage.

If there is the employed bee becomes scout then change it with new random source solutions by Eq.(34).

3.2.5 Results of optimization algorithms

The Results of optimization algorithms are showing in the next section after Appling on the benchmarks function for minimization, when the benchmarks are the test function.

Table 1,
The Optimization Results.

Bench function	FA	CFA	CFA-ABC
F1	1.0086e+003	743.1403	**603.734**
F2	1.1321e+006	3.6879e+005	**936272**
F3	1.6745e+003	1.2057e+003	**2462.28**
F4	1.1776	1.0803	**8.7319e007**
F5	10.1603	8.85986	**0.0138406**

Table 2,
Standard Benchmark Functions.

No.	Function
F1	$Sphere = \sum_{i=1}^{n} x_i^2$
F2	$Rosenbroch = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$
F3	$Rastrigin = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_{i)} + 10)$
F4	$Griewank = \sum_{i=1}^{n} (\frac{x_i^2}{4000}) - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
F5	$Ackley = 20 + e - 20 * e^{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}} - e^{\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)}$

4. Kinematic Neural Controller Design based on Hybrid CFA-ABC Algorithm

This paper introduce a CFA-ABC for searching the optimal controller parameters .The searching procedures of proposed method shown as below:

1. Initial searching weights ($V_{ji}^0, W_{kj}^0, B_{j1}^0, B_{k2}^0$) of each firefly are created randomly with in the determined range. Note that the dimension of search space consists of all the control parameters needed in the Kinematic neural controller as shown in Figure (2).
2. The objective function is calculated for each firefly by the mean square error function as equation [1]:

$$(MSE)=\frac{1}{2}\sum_{p=1}^{pop}(x_r(L)^p - x(L)^p)^2 + (y_r(L)^p - y(L)^p)^2 + (\theta_r(L)^p - \theta(L)^p)^2 \quad ...(38)$$

3. Applied step3 to step 13 from the pseudo code of (CFA-ABC) to find the best weight with minimizing trajectory tracking error.
4. The best weight are regarded as initial point for ABC.
5. The objective function is calculated for each bee by the mean square error function as equation (38).

6. Applied step14 to step 19 from the pseudo code of (CFA-ABC) to find the optimal weight.
7. If the iteration number arrive at maximum, then exit, else ,go to (2).

5. Simulation Results

The aim of the simulations is to examine the effectiveness and performance of the proposed kinematic neural controller based on (CFA-ABC) optimization algorithm to the wheeled mobile robot by using MATLAB programming environment.

The resulting wheeled mobile robot trajectory tracking, get by the proposed Kinematic neural controller is containing trajectory tracking, tracking error, and linear and angular velocity of mobile robot, linear velocity of right and left wheel and the mean square error.
These simulation are performed with the parameters setting illustrated in Table 4.

Table 3,
Simulation Parameters Setting.

Parameter	Setting
The number of fireflies	20
The number of bees	20
The length of NI-robot	0.36 m
The Radius of NI-robot	Ra=0.05
The period time	0.5 sec
Iteration of CFA	100
Iteration of ABC	100
Required linear velocity	0.1 m/sec
Required angular velocity	0.1 rad/sec

Case study1:

The required circular trajectory, can be explained by the next equations:

$$x_r(\tau) = \cos\left(\frac{\tau}{10}\right)$$

$$y_r(\tau) = \sin\left(\frac{\tau}{10}\right)$$

$$\theta_r(\tau) = \frac{\pi}{2} + \frac{\tau}{10}$$

The wheeled mobile robot model starts from the initial pose (position/ orientation),

$$q_{I0} = [1.1, -0.5, \frac{pi}{2}]$$

While the real initial pose (position/ orientation) of the mobile robot $q_r =$ [0.9,0.05,1.62], circular trajectory tracking simulation of the NI mobile robot is shown in Fig. (1a). It's so clear that great tracking performance that achieved by the proposed controller based on optimization, because of eliminating the error with fast convergence.

Case study2:

The required lemniscates trajectory, can be explained by the next equations:

$$x_r(\tau) = 0.75 + 0.75 * \sin\left(\frac{2\pi\tau}{50}\right)$$

$$y_r(\tau) = \sin\left(\frac{4\pi\tau}{50}\right)$$

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

The wheeled mobile robot model begins from the initial pose (position/ orientation),

$$q_{I0} = [0.7, -0.02, \frac{pi}{2}]$$

While the real initial pose (position/ orientation) of the mobile robot $q_r =$ [0.79,0.12,1.21], lemniscates trajectory tracking simulation of the NI mobile robot is view in Fig. (1b). the initial values of learning rate and weights are tuned such a way that it provides better performance.

- The simulation results explained the effectualness of the suggested Kinematic neural controller based on (CFA-ABC) technique by displaying its ability to produce limited smooth values of the control velocities for right and left wheels outside sudden spikes .as a result showing in Fig. (2a), Fig. (2b).
- Fig. (3a), Fig. (3b) indicate the convergence of the pose path and orientation errors for the robot model motion MSE (ex,ey,eθ) equal (0.027,0.012,0.0031) for case study1 and MSE(ex,ey,eθ) equal (±0.005,0.0072, -0.02) for case study2.
- The performance index MSE for pose path and orientation errors for the mobile robot model motion for 100 iterations are shown in Fig. (4a) is (0.0063) and Fig. (4b) is (0.0066).
- Fig. (5a) demonstrates the mean linear velocity (0.02 m/s) and the maximum peak of the angular velocity (0.005 rad/s) of the NI mobile robot>
- Fig. (5b) demonstrates the mean linear velocity (0.055 m/s) and the maximum peak of the angular velocity (±0.027rad/s) of the NI mobile robot.
- The performance described in Fig. 6a, fig.6b, Fig. (7a), Fig. (7b), Fig. (8a) and Fig. (8b) verify the effectiveness of the proposed control algorithm based on (CFA-ABC) algorithm is

clear by viewing the convergence of the pose Error (position/orientation) for the wheeled mobile robot with small error.

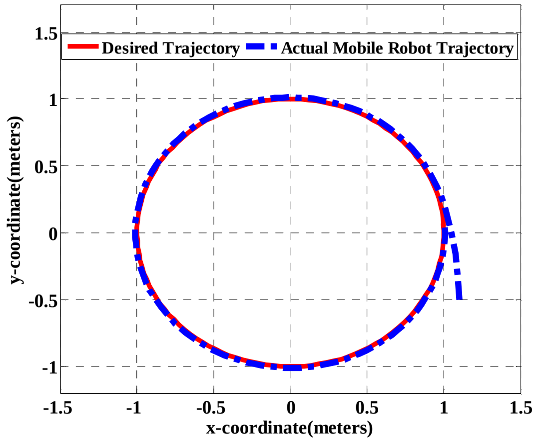

Fig. (1a). Actual trajectory of mobile robot and desired Circular trajectory tracking.

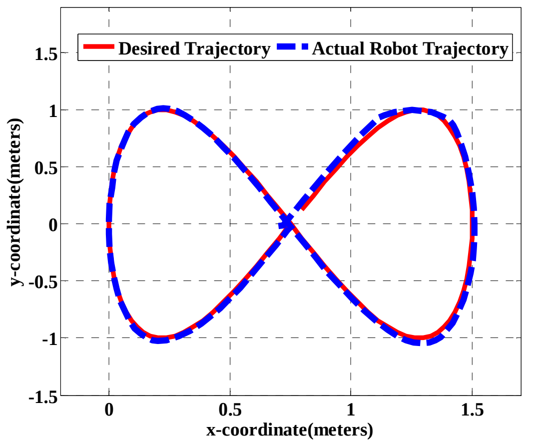

Fig. (1b). Actual trajectory of mobile robot and desired Lemniscates trajectory tracking.

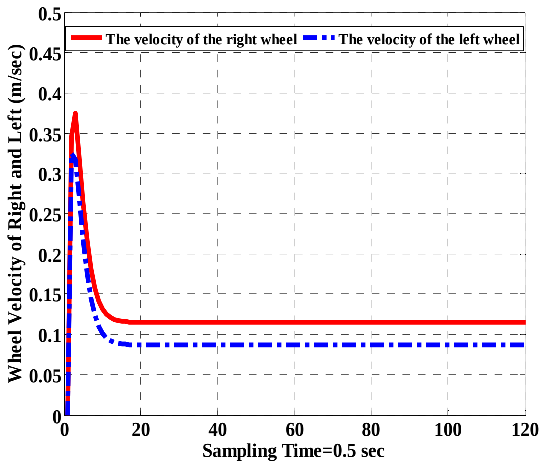

Fig. (2a). The right and left wheel velocity.

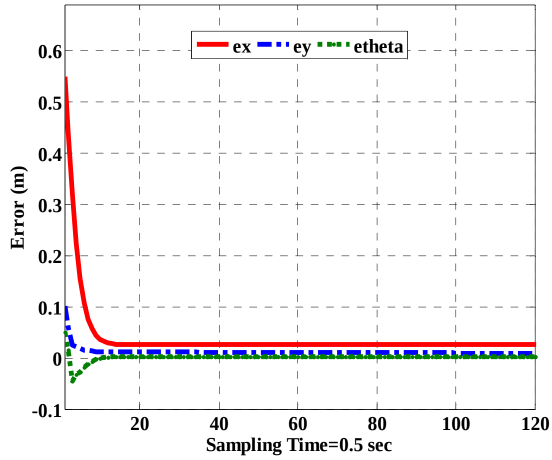

Fig. (3a). Trajectory tracking error.

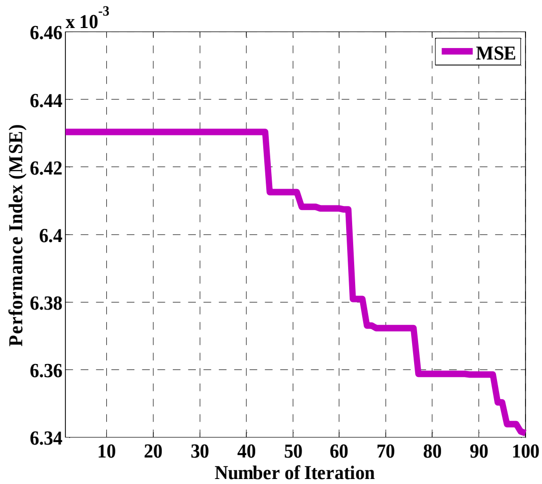

Fig. (4a). The performance index (MSE).

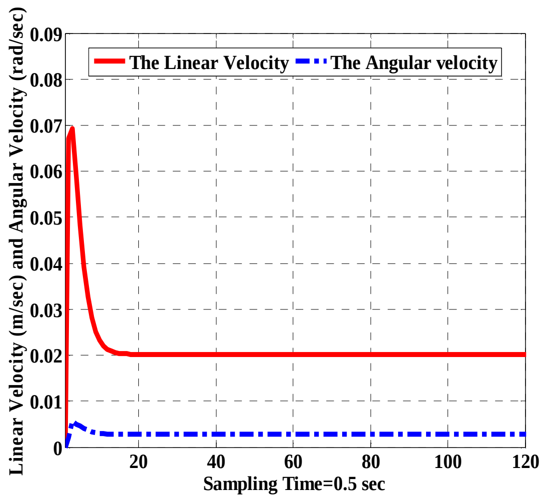

Fig. (5a). The linear and angular velocity.

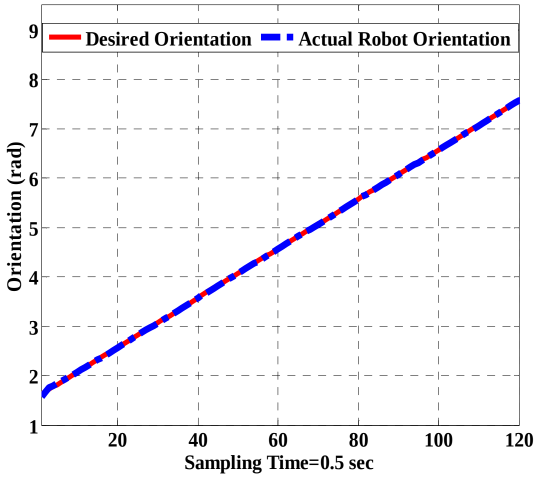

Fig. (6a). Desired orientation and actual mobile robot orientation.

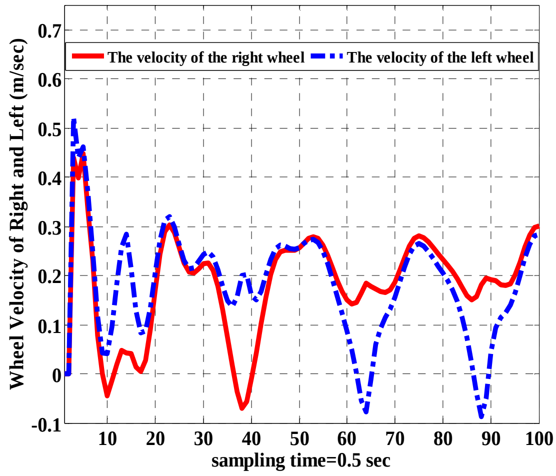

Fig. (2b): The right and left wheel velocity.

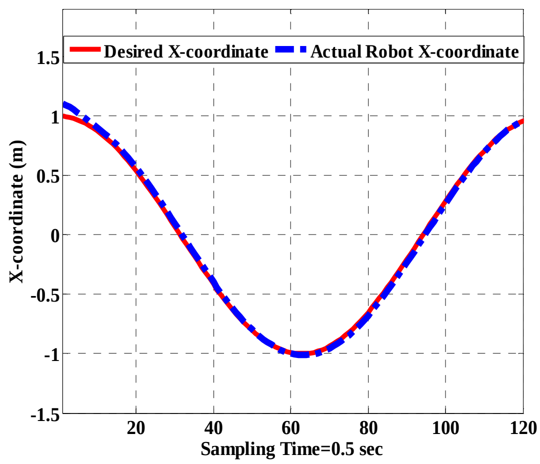

Fig. (7a). Position tracking in x- coordinate.

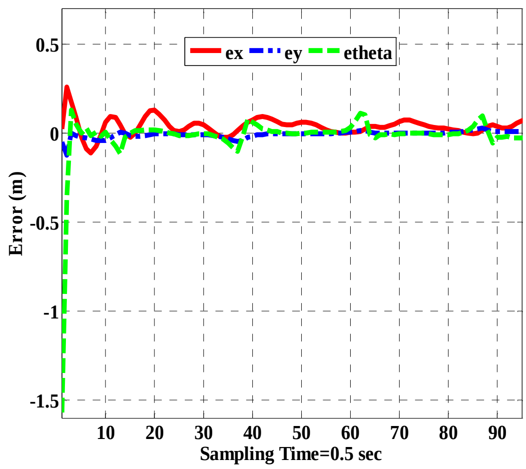

Fig. (3b). Trajectory tracking error.

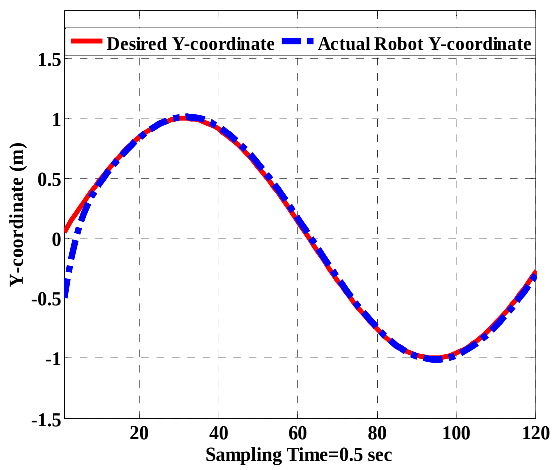

Fig. (8a). Position tracking in Y- coordinate.

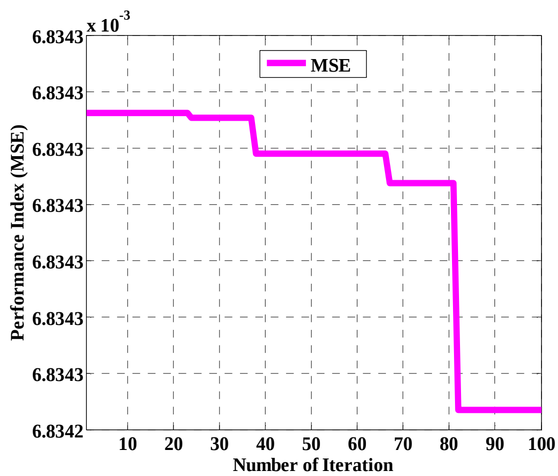

Fig. (4b): the Performance index (MSE).

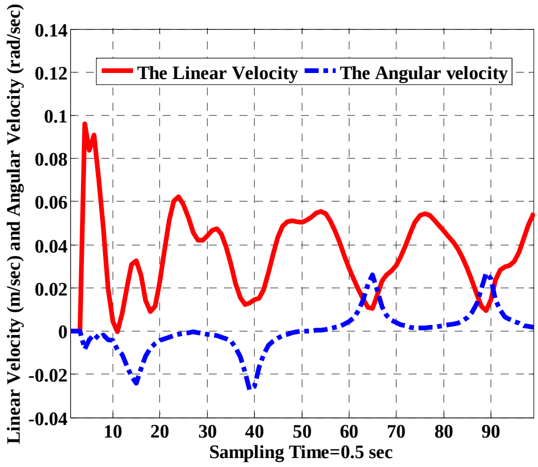

Fig. (5b): The linear and angular velocity.

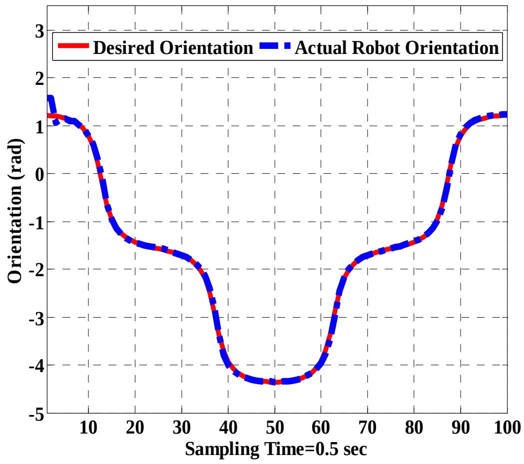

Fig. (6b). Desired orientation and actual mobile robot orientation.

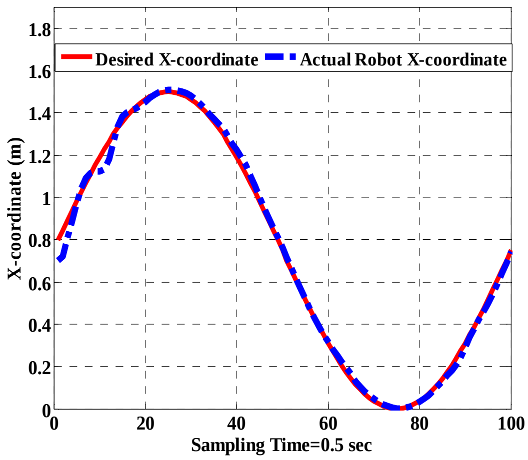

Fig. (7b): Position tracking in x- coordinate.

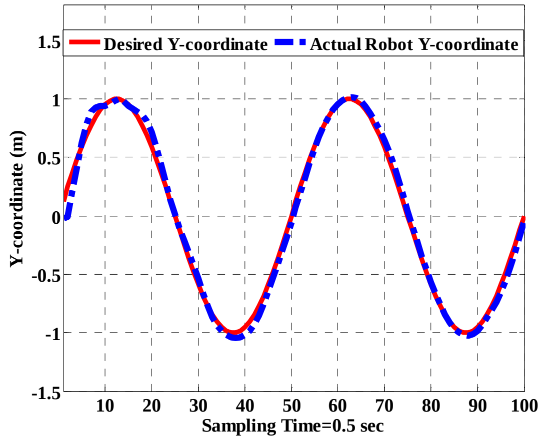

Fig. (8b). Position tracking in Y- coordinate.

6. Conclusion

In this paper, an enhanced Firefly algorithm called Crossover Firefly Algorithm (CFA) and hybrid with Artificial Bee Colony (ABC) are proposed. These algorithms are utilized to enhance the optimization solution value by shortening the randomness step by step. The simulation results for finding the global optima of diverse test functions are used to verify the functioning of the suggested optimization algorithms. Moreover, the research has presented an optimal kinematic neural controller based on (CFA-ABC) algorithm for two-wheeled differential mobile robots to achieve both trajectory tracking and stabilization based on the kinematic model. It has been built and tested using MATLAB (R2013b) environment applied on National Instrument mobile robot (NI Mobile Robot). The main result shows that the proposed control gets stability in the position errors and good follow of the trajectory tracking without perturbations, such that the mobile robot follows a desired path trajectory with minimize error. Additionally, the simulation results show clearly the ability of reworking and robustness of the suggested kinematic neural controller that based on (CFA-ABC) learning technique, which has wonderful position trajectory tracking achievement and it has the ability of producing smooth and acceptable linear and angular velocity. The suggested controller has explained the ability of tracking continuous gradients required trajectory and effectively reducing the tracking errors of the wheeled mobile robot model with *MSE* of a circular trajectory is (0.0063) and

MSE of the lemniscates trajectory is (0.0066) after 100 iterations, better then MSE in [21]and[11].

List of Symbols

Symbol	Definition
Alpha	The flat crossover coefficient
B	The weight between the bias and hidden (j).
B	The weight between the bias and output (k).
C	The mobile robot center (meter).
$Countl_{i1)}$	The abandonment counter of the employed bee.
$D1$	The dimensional vector of ABC .
d	Parameter to reduce the randomness.
e	The error vector.
F	The flash intensity.
F_0	The standard flash intensity.
$fi1$	The fitness value of the solution$j$.
$fitness\ i$	The fitness of each food source.
FM1	A population of ABC produce randomly.
F_s	The intensity at the beginning.
hntj	The output of the hidden layer node j.
ICCM	The instantaneous center of curvature mobile robot.
$inputi$	*(exm(L),exm(L-1),…VL (L-1)).*
L	Measure of the running step.
$L1$	The length between each wheel and the robot axis of symmetry in *y*- guidance (m).
m	The number of input of neural network.
mi	The number of input layer nodes.
mj	The number of hidden layer nodes.
n	The number of output layer nodes.
Pop	Number of individuals.
R	The radius of each wheel (rad).
rr	The distance between any two fireflies.
r	The random number generator uniformly distributed in [0, 1].
V	The translational velocity of the robot (meter/second).
vl_{i1j1}	A new nominee solution of ABC.
Vji	The weight between input (i) and output (j).
V	The linear left wheel speed (meter/second).
V	The linear right wheel speed (meter/second).
W	The weight between the hidden (j) and output (k).
$x_{00}, y_{00}, \theta_{00})$	The initial pose (position/orientation) of the robot.
(x, y, θ)	The actual pose (position/orientation) of the robot.
(x_r, y_r, θ_r)	The desired pose (position/orientation) of the robot.
xl_{i1j1}	Solution produced from ABC.
$xl_{j1}^{min}, xl_{j1}^{max}$	The lower boundary and upper boundary of the parameter $j1$.
FM1	A population of ABC produce randomly.
F_s	The intensity at the beginning.
hntj	The output of the hidden layer node j.
ICCM	The instantaneous center of curvature mobile robot.
$inputi$	*(exm(L),exm(L-1),…VL (L-1)).*
L	Measure of the running step.
$L1$	The length between each wheel and the robot axis of symmetry in *y*- guidance(m).
m	The number of input of neural network.
mi	The number of input layer nodes.
mj	The number of hidden layer nodes.
$\alpha 1$	The randomization parameter.
βL_0	The luminousness at rr=0.
βL	The luminousness of firefly.
$\dot{q}$	The angular left wheel speed (rad/second).

$\dot{\varphi}_R$ The angular right wheel speed (rad/second).

γ The fixed flash absorption coefficient.

$\Delta\tau$ The interval between two sampling times.

Ω The rotational velocity of the robot (rad/second).

$\emptyset$ A random real number in the interval of $[-1, 1]$.

$\underset{\cdot}{R}$ The rotation matrix.

7. References

[1] G. Artus, P. Morin, and C. Samson, Tracking of an omnidirectional target with a nonholonomic mobile robot, in Proc. IEEE Conf. Adv. Robot. (ICAR), 2003, pp. 1468–1473.

[2] K. Seo, and J. S. Lee, Kinematic path-following control of Mobile Robot under bounded angular velocity error, Advanced Robotics vol. 20, no. 1, pp. 1-23, 2006.

[3] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, A stable tracking control method for an autonomous mobile robot, In Proc. IEEE Int. Conf. Robot. Autom. Cincinnati, OH, vol. 1, pp.384-389, 1990 .

[4] O. Mohareri, R. Dhaouadi, and A. B. Rad, Indirect adaptive tracking control of a nonholonomic mobile robot via neural networks, Neurocomputing, vol. 88, pp. 54 – 66, 2012.

[5] I. Kav, Intelligent control schemes for nonlinear system, Ph.D Thesis, Indian Institute of Technology, Kanpur, India, 2008.

[6] G. Wang and L. Guo, A novel hybrid bat algorithm with harmony search for global numerical optimization, Journal of Applied Mathematics, vol. 2013, 21 pages, 2013.

[7] X. Li and M. Yin, An opposition-based differential evolution algorithm for permutation flow shop scheduling based on diversity measure, Advances in Engineering Software, vol. 55, pp. 10–31, 2013.

[8] X.-S. Yang, Z. Cui, R. Xiao, A. H. Gandomi, and M. Karamanoglu, Swarm Intelligence and Bio-Inspired Computation , Elsevier, Waltham, Mass, USA, 2013.

[9] A. Al-Araji, Design of a Cognitive Neural Predictive Controller for Mobile Robot,Brunel University, United Kingdom, 2012.

[10] J. Ye, Adaptive control of nonlinear PID-based analogue neural network for a nonholonomic mobile robot , Neurocompting, vol. 71, pp. 1561-1565, 2008.

[11] A. 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 ,May 2014.

[12] Z. Yousif, J. Hedley, and R. Bicker, Design of an Adaptive Neural Kinematic Controller for a National Instrument Mobile Robot System, IEEE International Conference on Control System, Computing and Engineering, 23 - 25 Nov. 2012, Penang, Malaysia.

[13] R. Fierro, and F. L. Lewis, Control of a nonholonomic mobile robot using neural networks, IEEE Trans. Neural Networks, pp. 589-600, 1998.

[14] F. L. Lewis, K. Liu, and A. Yesildirek, Neural net robot controller with guaranteed tracking performance , IEEE Transaction on Neural Networks, vol. 6, no. 3, pp. 703-715, 1995.

[15] X. S. Yang, Firefly algorithms for multimodal optimization , In Watanabe, O.,Zeug-mann,T.(eds.) SAGA 2009. LNCS, vol.5792, pp.169-178. Springer, Heidelberg, 2009.

[16] A. L. Bolaji, A. T. Khader, M. A. Al-betar, and M. A. Awadallah, Artificial Bee Colony Algorithm, Its Variants and Applications: A Survey, Journal of Theoretical and Applied Information Technology, vol. 47, no.2, pp. 434-459, 2013.

[17] M. H. Saffari, and M. J. Mahjoob, Bee Colony Algorithm for Real-Time Optimal Path Planning of Mobile Robots, Fifth International Conference on Soft Computing, Computing with Words and Perceptions in System Analysis, Decision and Control (ICSCCW), Eng., Univ. of Tehran, Tehran, Iran, 2-4 Sept. 2009.

[18] W. Xiang, S. Ma and M. An, An Improved Artificial Bee Colony Algorithm with Multiple Search Operators, Journal of Computational Information Systems, pp. 3129–3139, August 2013.

[19] D. Karaboga and B. Akay, A modified Artificial Bee Colony Algorithm for Real-Parameter Optimization, Swarm Intelligence

and Its Applications, Elsevier, vol. 192, pp.120-142, June 2010.

[20] D. Karaboga, An Idea based on Honey Bee Swarm for Numerical Optimization, Technical Report, Erciyes University, Engineering Faculty, Computer Engineering Department, pp. 1-10, 2005.

[21] A. Almeshal , K. Goher , M.R. Alenezi , A. Almazeed , J. Almatawah and M. Moaz, BFA optimized intelligent controller for path following unicycle robot over irregular terrains, International Journal of Current Engineering and Technology, Vol.5, No.2 (April 2015).

[22] M. Younes, R. L. Kherfane and F. Khodja , A hybrid Approach for economic power dispatch, Djillali Liabes University, vol. 8, July 2013.

تصميم وحدة سيطرة حركية عصبية لتتبع مسار الروبوتات المتنقلة بعجلات على أساس المحسن الهجين بين خوارزمية اليرعات المضيئة وخوارزمية خلية النحل

نزار هادي عباس*　　　　بسمة جمعة صالح**

،قسم الهندسة الكهربائية / كلية الهندسة/ جامعة بغداد*

*البريد الألكتروني: drnizaralmsaodi@gmail.com

**البريد الألكتروني:eng.basmaj@gmail.com

الخلاصة

يقدم هذا البحث ، تصميما مسيطرا لتتبع مسار عجلة الانسان الالي المتحرك المبني على أساس مسيطر تكاملي حركي عصبي لتوجيه الروبوت المحمول المصنع من قبل شركة National Instrument ، الغاية من المسيطر هي تقريب السرعة الحقيقية للأنسان الالي اقرب ما يكون من السرعة المطلوبة بجعل مقدار معدل الخطا اقرب الى الصفر. المسيطر المقترح في هذا البحث يتكون من طبقتين: الطبقة الأولى هي نظام شبكة عصبية متعدد الطبقات لتتبع المسار المطلوب و الطبقة الثانية هي طبقة الخوارزمية الأمثلية والتي تشمل خوارزمية هجينة محسنة وهي مكونة من خوارزمية اليرعات المضيئة وخوارزمية خلية النحل وتستخدم لأيجاد أفضل قيم للمسيطر والوصول للمسار الامثل. من خلال نتائج المحاكاة، اثبت المسيطر المقترح وخوارزمية الامثلية كفاءتهما بالوصول الى مقدار خطا وبأقل قيمة وتمت المحاكاة بأستخدام برنامج الماتلاب وتم تطبيقها على نوعين من المسارات (الدائري والانفنتي).