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 square 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 \( \dot{\phi}_R \) and \( \dot{\phi}_L \). The mobile robot kinematics commonly has two main analyses, Forward kinematics and feedback kinematics [10]:

- **Forward mobile robot kinematics:**

\[
\dot{q}_t = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = f(\dot{\phi}_R, \dot{\phi}_L) \quad \text{…(1)}
\]

- **Feedback mobile robot kinematics:**

\[
\begin{bmatrix} \dot{\phi}_R \\ \dot{\phi}_L \end{bmatrix} = f(x, y, \theta) \quad \text{…(2)}
\]

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.](image1)

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.](image2)
\[ \Omega(t) = \frac{d\hat{\theta}(t)}{dt} \]  
\( \text{(3)} \)

Thus:
\[ V(t) = \Omega(t) R_c(t) \]  
\( \text{(4)} \)

\[ V_L(t) = \left( R_c(t) + \frac{L_1}{2} \right) \Omega(t) \]  
\( \text{(5)} \)

\[ V_R(t) = \left( R_c(t) - \frac{L_1}{2} \right) \Omega(t) \]  
\( \text{(6)} \)

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].

\[ \Omega = \frac{\Omega}{R_c(t) + \frac{L_1}{2}} \]  
\( \text{(7)} \)

The angular velocity of the mobile robot is [11]:
\[ \Omega = \frac{\Omega}{R_c(t) - \frac{L_1}{2}} \]  
\( \text{(8)} \)

And the linear velocity of the mobile robot is [11]:
\[ V = \frac{\Omega}{R_c(t) + \frac{L_1}{2}} \]  
\( \text{(9)} \)

It is simulated that the wheeled mobile robot below the nonholonomic constraint, which is ideal rolling no skidding [11], as shown in (2):
\[ -\dot{x}(t) \sin \theta (t) + \dot{y}(t) \cos \theta (t) = 0 \]  
\( \text{(10)} \)

The kinematics equations of differential wheeled mobile robot is represented as follows [11]:
\[ \dot{x}(t) = V(t) \cos \theta(t) \]  
\( \text{(11)} \)

\[ \dot{y}(t) = V(t) \sin \theta(t) \]  
\( \text{(12)} \)

\[ \dot{\theta}(t) = \Omega(t) \]  
\( \text{(13)} \)

In order to obtain the position and orientation of the wheeled mobile robot, it is required to integrate equation as showed below [12]:
\[ x(t) = x_0 + \int_0^t V(t) \cos \theta(t) \, dt \]  
\( \text{(14)} \)

\[ y(t) = y_0 + \int_0^t V(t) \sin \theta(t) \, dt \]  
\( \text{(15)} \)

\[ \theta(t) = \theta_0 + \int_0^t \Omega(t) \, dt \]  
\( \text{(16)} \)

In the computer simulation, the form of the pose (position/orientation) equations as follows [11]:
\[ x(t) = x_0 + \int_0^t V(t) \cos \theta(t) \, dt \]  
\( \text{(17)} \)

\[ y(t) = y_0 + \int_0^t V(t) \sin \theta(t) \, dt \]  
\( \text{(18)} \)

\[ \theta(t) = \theta_0 + \int_0^t \Omega(t) \, dt \]  
\( \text{(19)} \)

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

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:

\[ \begin{bmatrix} x(t) \\ y(t) \\ \theta(t) \end{bmatrix} = \begin{bmatrix} \cos (\theta(t)) & 0 & 0 \\ \sin (\theta(t)) & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} V(t) \\ \Omega(t) \end{bmatrix} \]  
\( \text{(20)} \)

Let \( (x_r, y_r) \) is the desired position of the wheeled mobile robot, and \( \theta_r \) 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 \( \theta \), so that the real posture of the mobile robot is \( q_1 = [x, y, \theta]^T \).

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 \( \mathbf{R} \) [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 approximation of arbitrary nonlinear mappings, and ability of learning and adaptation [13, 14].

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

\[
F(dj) = \frac{2}{1+e^{-J}} \quad \ldots \ (26)
\]

The neural network weights (control parameter) \( V_j, W_k, B_j \) and \( B_k \) 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 \( F(rr) = \frac{F_0}{rr^2} \) [8].

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

\[
F = F_0e^{-\gamma rr} \quad \ldots \ (27)
\]
As a firefly's luminousness is relative to the flash intensity seen by neighboring fireflies, the luminousness $g$ of a firefly was specified by [8]:

$$g = g_0 e^{-rr_2}$$  \hspace{1cm} (28)

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

$$r1j1 = \|xl_{i1} - xl_{j1}\| = \left(\sum_{M=1}^{d} (xl_{i1,M} - xl_{j1,M})^2\right)^{1/2}$$  \hspace{1cm} (29)

Where $xl_{i1,M}$ is the $M$th component of the relating coordinate $xl_i$ of ith firefly. In the 2-D case as shown in eq. (30), from [22]:

$$rr1j1 = \sqrt{(xl_{i1} - xl_{j1})^2 + (yl_{i1} - yl_{j1})^2}$$  \hspace{1cm} (30)

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

$$xl_{i1} = xl_{i1} + \beta g_0 e^{-rr_2} (xl_{j1} - xl_{i1}) + \alpha1 (rand - 0.5)$$  \hspace{1cm} (31)

Where the second term is due to the attractive feature. The third part is randomisation with a hold parameter $\alpha1$, which makes the investigation of the search distance more effective [15], after few imitations, use $\beta0 = 1$, $\alpha1 \in [0,1], y = 1$, 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:

$$Alpha = (delta) \ast (alpha^{iteration})$$  \hspace{1cm} (32)

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

$$xl_{i1} = (alpha) \ast xl_{j1} + (1 - alpha) \ast xl_{j1}$$  \hspace{1cm} (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 equal to the number of food sources. Each solution $xl_{i1}(i1 = 1, 2... FM1)$ describing an individual is a $D1$-dimensional vector. Each solution can be produced by Eq. (34) [19]:

$$xl_{i1} = x_{i1}^{min} + (x_{i1}^{max} - x_{i1}^{min}) \cdot rand(0, 1) \text{ ... (34)}$$

Where $i1 = 1,2,...,FM1$, $j1 = 1,2,...,D1$

Next, the fitness of each food source is evaluated by

$$fitnessi1 = \left\{ \begin{array}{ll} \frac{1}{1+|fi1|} & \text{if } fi1 \geq 0 \\ 1 + |fi1| & \text{if } fi1 < 0 \end{array} \right. \text{ ... (35)}$$

### 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)):

$$vl_{i1j1} = xl_{i1j1} + \theta_{i1j1} \cdot (xl_{i1j1} - xl_{k1j1}) \text{ ... (36)}$$

where $k1 \in [1...FN1] and j1 \in [1...D1]$ are arbitrary numbers, and $k1$ has to be other from $i1$, $\theta_{i1j1}$. After creating a new food source $vl_{i1j1}$, 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).

$$p1 = \frac{fitnessi1}{\sum_{i=1}^{FN1} fitnessi1} \text{ ... (37)}$$

Where $fitnessi1$, 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].
The summarization of the computation procedure of hybrid method is described by the pseudo code below\cite{22}:

**Step 1:** Initial pop of fireflies $x_{11}, i_1 = 1, 2, ..., S$.

**Step 2:** Evaluate Objective function $f(x_{11})$ for each firefly.

**Step 3:** Set flash absorption coefficient $\gamma, \beta L_0, \alpha 1$.

**Step 4:** Flash intensity $F_{11}$ at $x_{11}$ is calculated by $f(x_{11})$.

**Step 5:** while $t < \text{Max Iteration}$.

**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) According 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.

- Onlooker bees stage.
- Step 19: For each Onlooker bee.
  - chooses a food source \( x_{1j1} \) 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>
<thead>
<tr>
<th>Table 1, The Optimization Results.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bench function</td>
</tr>
<tr>
<td>-----------------</td>
</tr>
<tr>
<td>F1</td>
</tr>
<tr>
<td>F2</td>
</tr>
<tr>
<td>F3</td>
</tr>
<tr>
<td>F4</td>
</tr>
<tr>
<td>F5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2, Standard Benchmark Functions.</th>
</tr>
</thead>
<tbody>
<tr>
<td>No.</td>
</tr>
<tr>
<td>-----</td>
</tr>
<tr>
<td>F1</td>
</tr>
<tr>
<td>F2</td>
</tr>
<tr>
<td>F3</td>
</tr>
<tr>
<td>F4</td>
</tr>
<tr>
<td>F5</td>
</tr>
</tbody>
</table>

### 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 \( (V_{ji}, W_{kj}, B_{ji}, B_{kj}) \) 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]:
   \[
   (\text{MSE}) = \frac{1}{3} \sum_{p=1}^{3} (x_p(L) - x(L))^2 + (y_p(L) - y(L))^2 + (\theta_p(L) - \theta(L))^2
   \]
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 step 14 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.

Case study1:
The required circular trajectory, can be explained by the next equations:

\begin{align*}
    x_r(t) &= 0.75 + 0.75 \times \sin \left( \frac{2\pi t}{50} \right) \\
    y_r(t) &= \sin \left( \frac{4\pi t}{50} \right) \\
    \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)
\end{align*}

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

\[ q_{10} = [0.7, -0.02, \frac{\pi}{2}] \]

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 optimization, because of eliminating the error with fast convergence.

Case study2:
The required lemniscates trajectory, can be explained by the next equations:

\begin{align*}
    x_r(t) &= 0.75 + 0.75 \times \sin \left( \frac{2\pi t}{50} \right) \\
    y_r(t) &= \sin \left( \frac{4\pi t}{50} \right) \\
    \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)
\end{align*}

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

\[ q_{10} = [0.7, -0.02, \frac{\pi}{2}] \]

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 optimization, because of eliminating the error with fast convergence.
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. (7a). Position tracking in x-coordinate.

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

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

Fig. (3b). Trajectory tracking error.

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

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alpha</td>
<td>The flat crossover coefficient</td>
</tr>
<tr>
<td>B</td>
<td>The weight between the bias and hidden (j).</td>
</tr>
<tr>
<td>B</td>
<td>The weight between the bias and output (k).</td>
</tr>
<tr>
<td>C</td>
<td>The mobile robot center (meter).</td>
</tr>
<tr>
<td>Countl(i)</td>
<td>The abandonment counter of the employed bee.</td>
</tr>
<tr>
<td>Dl</td>
<td>The dimensional vector of ABC.</td>
</tr>
<tr>
<td>e</td>
<td>Parameter to reduce the randomness.</td>
</tr>
<tr>
<td>f1l</td>
<td>The error vector.</td>
</tr>
<tr>
<td>F</td>
<td>The flash intensity.</td>
</tr>
<tr>
<td>F0</td>
<td>The standard flash intensity.</td>
</tr>
<tr>
<td>fitness i</td>
<td>The fitness of each food source.</td>
</tr>
<tr>
<td>FM1</td>
<td>A population of ABC produce randomly.</td>
</tr>
<tr>
<td>Fs</td>
<td>The intensity at the beginning.</td>
</tr>
<tr>
<td>hntj</td>
<td>The output of the hidden layer node j.</td>
</tr>
<tr>
<td>ICCM</td>
<td>The instantaneous center of curvature mobile robot.</td>
</tr>
<tr>
<td>inputi</td>
<td>The fitness value of the solution j.</td>
</tr>
<tr>
<td>L</td>
<td>Measure of the running step.</td>
</tr>
<tr>
<td>L1</td>
<td>The length between each wheel and the robot axis of symmetry in y- guidance (m).</td>
</tr>
<tr>
<td>m</td>
<td>The number of input of neural network.</td>
</tr>
<tr>
<td>mi</td>
<td>The number of input layer nodes.</td>
</tr>
<tr>
<td>mj</td>
<td>The number of hidden layer nodes.</td>
</tr>
<tr>
<td>m</td>
<td>The number of output layer nodes.</td>
</tr>
<tr>
<td>Pop</td>
<td>Number of individuals.</td>
</tr>
<tr>
<td>R</td>
<td>The radius of each wheel (rad).</td>
</tr>
<tr>
<td>rr</td>
<td>The distance between any two fireflies.</td>
</tr>
<tr>
<td>r1</td>
<td>The random number generator uniformly distributed in [0, 1].</td>
</tr>
<tr>
<td>V</td>
<td>The translational velocity of the robot (meter/second).</td>
</tr>
<tr>
<td>vl1i,j1</td>
<td>A new nominee solution of ABC.</td>
</tr>
<tr>
<td>Vji</td>
<td>The weight between input (i) and output (j).</td>
</tr>
<tr>
<td>Vr</td>
<td>The linear left wheel speed (meter/second).</td>
</tr>
<tr>
<td>Vr</td>
<td>The linear right wheel speed (meter/second).</td>
</tr>
<tr>
<td>W</td>
<td>The weight between the hidden (j) and output (k).</td>
</tr>
<tr>
<td>x00,y00,θ00</td>
<td>The initial pose (position/orientation) of the robot.</td>
</tr>
<tr>
<td>(x,y,θ)</td>
<td>The actual pose (position/orientation) of the robot.</td>
</tr>
<tr>
<td>(xr,yr,θr)</td>
<td>The desired pose (position/orientation) of the robot.</td>
</tr>
<tr>
<td>xl1i,j1</td>
<td>Solution produced from ABC.</td>
</tr>
<tr>
<td>xlj1min, xlj1max</td>
<td>The lower boundary and upper boundary of the parameter j1.</td>
</tr>
<tr>
<td>FM1</td>
<td>A population of ABC produce randomly.</td>
</tr>
<tr>
<td>Fs</td>
<td>The intensity at the beginning.</td>
</tr>
<tr>
<td>hntj</td>
<td>The output of the hidden layer node j.</td>
</tr>
<tr>
<td>ICCM</td>
<td>The instantaneous center of curvature mobile robot.</td>
</tr>
<tr>
<td>inputi</td>
<td>The actual pose (position/orientation) of the robot.</td>
</tr>
<tr>
<td>L</td>
<td>Measure of the running step.</td>
</tr>
<tr>
<td>L1</td>
<td>The length between each wheel and the robot axis of symmetry in y- guidance (m).</td>
</tr>
<tr>
<td>m</td>
<td>The number of input of neural network.</td>
</tr>
<tr>
<td>mi</td>
<td>The number of input layer nodes.</td>
</tr>
<tr>
<td>mj</td>
<td>The number of hidden layer nodes.</td>
</tr>
<tr>
<td>a1</td>
<td>The randomization parameter.</td>
</tr>
<tr>
<td>βL0</td>
<td>The luminousness at rr=0.</td>
</tr>
<tr>
<td>βL</td>
<td>The luminousness of firefly.</td>
</tr>
<tr>
<td>̇θl</td>
<td>The angular left wheel speed (rad/second).</td>
</tr>
</tbody>
</table>
The angular right wheel speed (rad/second).

\(\gamma\) The fixed flash absorption coefficient.

\(\Delta\tau\) The interval between two sampling times.

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

\(\Phi\) A random real number in the interval of \([-1, 1]\).

\(\mathcal{R}\) The rotation matrix.

7. References


تصميم وحدة سيطرة حركية عصبية لتنبئ مسار الروبوتات المتنقلة بعجلات على أساس المحسن الهجين بين خوارزمية اليراعات المضينة وخوارزمية خليفة النحل

نزار هادي عباس*

*قسم الهندسة الكهربائية / كلية الهندسة / جامعة بغداد
البريد الإلكتروني: drnizaralmsaodi@gmail.com
البريد الإلكتروني: eng.basmaj@gmail.com

الخلاصة

يقدم هذا البحث، تصميمًا مسبطاً لتنبؤ مسار عجلة الإنسان الآلي المتحرك العيني على أساس مسيطر تكامل حركي عصبي. تتوجيه الروبوت المحمل المصنوع من قبل شركة National Instrument، الرؤية المطلوبة بجعل مسار معدل الخطأ أقرب إلى الصغر. المسيطر المقترح في هذا البحث يتكون من طبقتين: الطبقتين الأولى هي نظام شبكة عصبية متعددة الطبقات لتنبؤ المسار المطلوب، والطبقات الثانية هي طبقة الخوارزمية الأمثلة التي تشمل خوارزمية هجينة محسنة، وهي مكونة من خوارزمية اليراعات المضينة وخوارزمية خليفة النحل. وتستخدم لأجل أفضل قيم للمسيطر والوصول للمسار الأمثل. من خلال تجاوز المحاكي، أثبت المسيطر المقترح وخوارزمية الأمثلة كفاءتهما بالوصول إلى مدار خطأ وبأقل قيمة وتم المحاكاة باستخدام برنامج المطلوب وتم تطبيقها على نوعين من المسارات (الدائرية والخطية).