

# Comparative Study of Inspired Algorithms for Trajectory-Following Control in Mobile Robot

**Basma Jumaa Saleh**

Computer Engineering Dep., Al-Mustansiriyah University, Baghdad, Iraq  
Email: eng.basmaj@gmail.com

**Ali Talib Qasim al-Aqbi**

Computer Engineering Dep., Al-Mustansiriyah University, Baghdad, Iraq  
Email: alnser.Ali@gmail.com

**Ahmed Yousif Falih Saedi**

Computer Engineering Dep., Al-Mustansiriyah University, Baghdad, Iraq  
Email: mr.a7med86@gmail.com

**Lamees abdalhasan Salman**

Computer Engineering Dep., Al-Mustansiriyah University, Baghdad, Iraq  
Email: lameesiteng2013@gmail.com

Received: 10 August 2018; Accepted: 26 August 2018; Published: 08 September 2018

**Abstract**—This paper is devoted to the design of a trajectory-following control for a differentiation nonholonomic wheeled mobile robot. It suggests a kinematic nonlinear controller steer a National Instrument mobile robot. The suggested trajectory-following control structure includes two parts; the first part is a nonlinear feedback acceleration control equation based on back-stepping control that controls the mobile robot to follow the predetermined suitable path; the second part is an optimization algorithm, that is performed depending on the Crossed Firefly algorithm (CFA) to tune the parameters of the controller to obtain the optimum trajectory. The simulation is achieved based on MATLAB R2017b and the results present that the kinematic nonlinear controller with CFA is more effective and robust than the original firefly learning algorithm; this is shown by the minimized tracking-following error to equal or less than (0.8 cm) and getting smoothness of the linear velocity less than (0.1 m/sec), and all trajectory-following results with predetermined suitable are taken into account. Stability analysis of the suggested controller is proven using the Lyapunov method.

**Index Terms**—Trajectory-following Mobile Robot, Back-stepping Control, Kinematic Nonlinear Controller, National Instrument, Firefly Algorithm.

## I. INTRODUCTION

Over the last few years, the interest in the machine learning and how it has been employed to help mobile robots in the navigation has been increased. Mobile robots have a lot of potential in the industry and in the

house servant applications. Navigation control of wheeled mobile robots has been studied by many sources in the last decade ever after they are progressively used in an extensive range of applications. The motion of mobile robots must be modeled by mathematical calculation to estimate physical environment and run in delineated trajectories. At first, the research work was focused only on the kinematic model if there is optimum linear and angular velocity following [1]. Mobile robots are mostly used in manufacturing, healthcare and medicine [2].

The basic estimation problems in wheeled mobile robot trajectory-following are still waiting to be addressed; then the implicit essence of the motivation for this paper is to generate optimum control parameters for the mobile robot, to follow the required path with small trajectory-following error, to beat unmolded kinematic disturbances, and to maintain the battery energy of the robot system. Traditionally, the problems of wheeled mobile robot controller have been processes by stabilizing point or by predetermined suitable problems like a trajectory-following controller [3].

The navigation problem is the basic aim of a wheeled mobile robot in order that it requires a decision where to direct and that information is taken by an actual path. The problem of following may be classified into dynamic and kinematic trajectory-following control model. Kinematics controller targets are to reach angular and linear velocity as output to converge trajectory-following error to zero [4].

The modern survey of improvement nonholonomic control systems is illustrated in [5]. To the authors' knowledge, the trajectory-following control problem is one of research challenges for robot nonholonomic mobile systems in but has yet to be widely studied. From

past researches, an approximation for trajectory-following control, back-stepping control has been applied to the trajectory-following control of robots [6] and is lately receiving expansive interest of researches about nonholonomic control systems.

Jiang and Pomet [7], utilized back-stepping theory for robot nonholonomic mobile systems by adaptation. Guldner and Utkin [8], suggested a Lyapunov theory to define a set of given configuration for required trajectories, then given a trajectory-following feedback controller to minimizing error and reducing chattering.

In this work, we propose an optimum back-stepping controller for solving trajectory-following problems for a robot nonholonomic mobile system, then a required trajectory with asymptotic stability can be obtained using the suggested control.

In engineering problems, the optimization is to detect a solution that can minimize or maximize a cost function. These days, the stochastic method is more often used to solve the optimization problem [9]. Recently, the nature-inspired algorithm is proving its ability in solving numerical problems of optimization more suitably. These optimization ways are improved to solve complex problems, such as scheduling of flow shop [10], high-dimensional function optimization, reliability, and other engineering matters. Recently, many other algorithms have been used, like Artificial Bee Colony (ABC), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Harmony Search (HS) and Firefly Algorithm (FA).

Firefly Algorithm (FA) is a relatively new heuristic optimization algorithm, and it was first developed by Xin-She Yang. 2001 [11], which is based on the lighting of fireflies. Since then, it has been used to solve different problems of optimization, including function optimization, water apportionment network, groundwater modeling, structural design, energy-salvage, and others.

The remainder of this paper is organized as follows: Section 2 gives a Modeling of Differentiation Nonholonomic NI-Robot. Section 3 describes the Trajectory-Following Controller. Section 4 presents the Simulation. Conclusion is given in the final section.

## II. MODELING OF DIFFERENTIATION NONHOLONOMIC NI-ROBOT

In this research, we consider a differentiation two-wheeled drive mobile robot is collected of two standard drive wheels on the same axis and a free driving castor wheel for stability. In order that robot positioning, trajectory-following, the linear velocity, angular velocity and kinematic model, as shown in Fig. 1. The variables for this mathematical kinematic model of a differentiation two-wheeled drive mobile robot are an actual positional and angle parameterization  $(x, y, \theta)$ , desired positional and angle parameterization  $(x_r, y_r, \theta_r)$ , radius of each wheel ( $R_w$ ), centre of robot is (c) and length between two wheels ( $L_1$ ), together with linear and angular velocities ( $V, \Omega$ ), respectively [12].

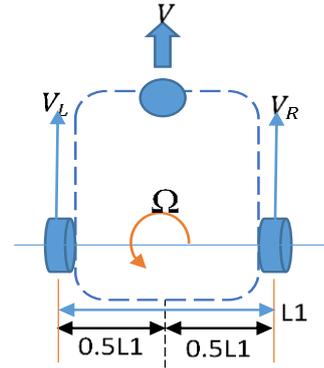


Fig.1. Model of a differentiation mobile robot.

At each instant in time, the left and right wheels follow a trajectory as shown in Fig. 2 that moves around the instantaneous centre of curvature mobile robot (ICCM) with the same angular rate [13].

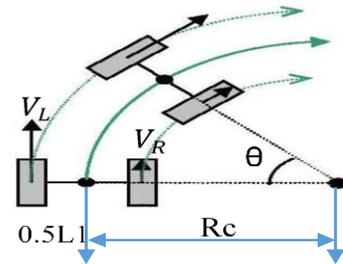


Fig.2. The instantaneous centre of curvature.

$$\Omega(\tau) = d\theta(\tau)/d\tau \quad (1)$$

Thus:

$$V(\tau) = \Omega(\tau)Rc(\tau) \quad (2)$$

$$V_L(\tau) = \left(Rc(\tau) + \frac{L_1}{2}\right)\Omega(\tau) \quad (3)$$

$$V_R(\tau) = \left(Rc(\tau) - \frac{L_1}{2}\right)\Omega(\tau) \quad (4)$$

After solving Eqs. (3, 4), we found the instantaneous centre of curvature mobile robot (ICCM) path close to the centre point axis (c) is given as Eq. (5) [13].

$$Rc(\tau) = \frac{L_1(V_L(\tau) + V_R(\tau))}{2(V_L(\tau) - V_R(\tau))} \quad (5)$$

The angular velocity of the mobile robot is [14]:

$$\Omega(\tau) = \frac{(V_L(\tau) - V_R(\tau))}{L_1} \quad (6)$$

And the linear velocity of the mobile robot is [14]:

$$V(\tau) = \frac{(V_L(\tau) + V_R(\tau))}{2} \quad (7)$$

Therefore, the prediction kinematic model shown in Eq. (8). The constraint is non-holonomic, shown in Eq. (9) should be respected at rolling no spilling [15].

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V \\ \Omega \end{bmatrix} \quad (8)$$

$$-\dot{x}(\tau)\sin\theta(\tau) + \dot{y}(\tau)\cos\theta(\tau) = 0 \quad (9)$$

The  $(x, y, \theta)$  parameterization can be explicit as

$$x(\tau) = x_{00} + \int_0^\tau V(\tau)\cos\theta(\tau)d\tau \quad (10)$$

$$y(\tau) = y_{00} + \int_0^\tau V(\tau)\sin\theta(\tau)d\tau \quad (11)$$

$$\theta(\tau) = \theta_{00} + \int_0^\tau \Omega(\tau)d\tau \quad (12)$$

The discrete kinematic model by using Euler's theory with the instant time  $(L)$  and the sampling time  $(\tau)$ , can be expressed by Eqs. (13, 14, 15):

$$x(L) = 0.5[V_R(L) + V_L(L)]\cos\theta(L)\Delta t + x(L-1) \quad (13)$$

$$y(L) = 0.5[V_R(L) + V_L(L)]\sin\theta(L)\Delta t + y(L-1) \quad (14)$$

$$\theta(L) = \frac{1}{R_a}[V_R(L) + V_L(L)]\Delta t + \theta(L-1) \quad (15)$$

Finally, Fig. 3 show the configuration trajectory-following error  $S_{ee} = [x_{ee}, y_{ee}, \theta_{ee}]^T$  can be presented by  $S_{ee} = R * \mathfrak{E}$ , where  $\mathfrak{E} = [x_e, y_e, \theta_e]^T$  and  $(R)$  rotational matrix, which can be shown by the Eqs. (16,17) [16]:

$$S_{ee} = \begin{bmatrix} x_{ee} \\ y_{ee} \\ \theta_{ee} \end{bmatrix} = \begin{bmatrix} \cos(\theta(\tau)) & \sin(\theta(\tau)) & 0 \\ -\sin(\theta(\tau)) & \cos(\theta(\tau)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} \quad (16)$$

$$\begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} \quad (17)$$

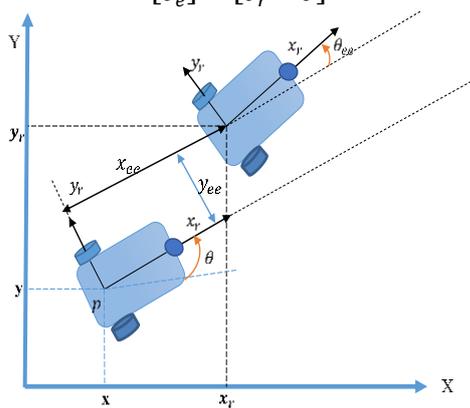


Fig.3. Configuration error of mobile robot [12].

### III. TRAJECTORY-FOLLOWING CONTROLLER

The proposed kinematic nonlinear controller is represented by the block diagram illustrated in Fig. 4. The Crossed firefly algorithm optimization algorithm which is used to optimum tuning parameters of adaptive

sliding mode kinematic controller to minimize the trajectory-following error, generates the perfect velocity control actions (linear and angular velocity) and tracks the desired trajectory-following coordinates.

The structure of trajectory-following control includes two parts; the first part is a nonlinear feedback acceleration control equation based on back-stepping control that controls the mobile robot to follow the predetermined suitable path; the second part is an optimization algorithm, that is performed depending on the Crossed Firefly algorithm to tune the parameters of the controller to obtain the optimum trajectory-following variables.

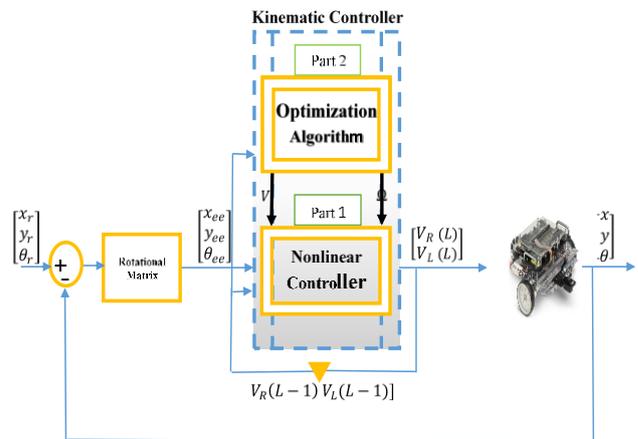


Fig.4. Structure of the nonlinear kinematic trajectory-following controller.

The following two subsections are dedicated to explaining the nonlinear back-stepping kinematic controller and the proposed optimization algorithm to achieve tuning parameters of controls.

#### 3.1 Back-stepping Kinematic Controller

Back-stepping is a method for nonlinear state feedback design by successive extensions of a Lyapunov function in order to guarantee stability. Two-step design procedure is known as back-stepping. Since a system starts at one desired output, and back-steps through the system selecting desirable values of the state components until the actual control input is reached. Since the integrator back-stepping procedure discovered by D.E. Koditschek (1987) [17] for the first time. It has been applied to resolve the path tracking problem. The nonlinear trajectory-tracking controller, which represented by the nonlinear feedback acceleration control approaches based on back-stepping execution and Lyapunov theory. The mobile robot nonholonomic kinematic model are transformed from cartesian actual parameterization to polar desired parameterization using that suggested control [18]. The suggested rule of controller that is responsible for finding the smooth and suitable actual right and left wheels velocity control action for nonholonomic mobile robot; thus, the trajectory-following error will be close to zero, and an asymptotically stable of system can be satisfied [19], proposed nonholonomic control rule is illustrated in Fig. 5.

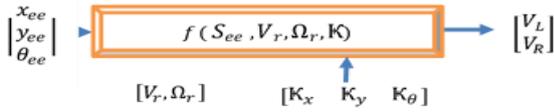


Fig.5. The kinematic nonlinear back-stepping structure.

The trajectory-following configuration transformation error Eq. (16) was used to facilitate the stability analysis and trajectory-following error closed loop system development. The derived configuration trajectory-following error with state frame is  $S_{ee} = [\dot{x}_{ee}, \dot{y}_{ee}, \dot{\theta}_{ee}]^T$  can be expressed as Eq. (18) by Derivative of Eq. (16) and rearranging with Eq. (8).

$$\dot{S}_{ee} = \begin{bmatrix} \dot{x}_{ee} \\ \dot{y}_{ee} \\ \dot{\theta}_{ee} \end{bmatrix} = \begin{bmatrix} V_r \cos(\theta_{ee}) + y_{ee}\Omega - V \\ V_r \sin(\theta_{ee}) - x_{ee}\Omega \\ \Omega_r - \Omega \end{bmatrix} \quad (18)$$

The desired virtual parameterization of the mobile robot shown as Eqs. (26, 20 and 21) which is based on the desired parameterization of the mobile robot  $S_r = [x_r, y_r, \theta_r]^T$  in [21].

$$\dot{x}_r = V_r \cos \theta_r \quad (19)$$

$$\dot{y}_r = V_r \sin \theta_r \quad (20)$$

$$\dot{\theta}_r = \Omega_r \quad (21)$$

The required linear velocity and the required angular velocity for the required trajectory are shown by Eqs. (29 and 23), respectively [13].

$$V_r = \sqrt{(\dot{x}_r)^2 + (\dot{y}_r)^2} \quad (22)$$

$$\Omega_r = \frac{\dot{y}_r \dot{x}_r - \dot{x}_r \dot{y}_r}{(\dot{x}_r)^2 + (\dot{y}_r)^2} \quad (23)$$

The cartesian actual velocity control rule  $V = f(S_{ee}, V_r, \Omega_r, K)$  such that  $\lim_{t \rightarrow \infty} (S_r - S) = 0$  is asymptotically stable can be determine with  $V_r > 0$  and  $\Omega_r > 0$ , for all  $\tau$ . Where  $S_{ee}, V_r, \Omega_r$  and  $K$  are the tracking-following model error, the reference linear and angular velocity and control gain, respectively, which must be the parameters of control are  $K_x, K_y, K_\theta > 0$ , so the control law as follows:

$$v = \begin{bmatrix} V \\ \Omega \end{bmatrix} = \begin{bmatrix} V_r \cos(\theta_{ee}) + K_x x_{ee} \\ \Omega_r + K_y V_r y_{ee} + K_\theta V_r \sin(\theta_{ee}) \end{bmatrix} \quad (24)$$

Finally, the suggested kinematic nonlinear trajectory-following control rule based on back-stepping theory and velocity (right and left wheels velocity) as a control action can be expressed by Eq. (25) [18].

$$v = \begin{bmatrix} V_L \\ V_R \end{bmatrix} = \begin{bmatrix} V_r \cos(\theta_{ee}) + K_x x_{ee} + L1/2(\Omega_r + K_y V_r y_{ee} + K_\theta V_r \sin(\theta_{ee})) \\ V_r \cos(\theta_{ee}) + K_x x_{ee} - L1/2(\Omega_r + K_y V_r y_{ee} + K_\theta V_r \sin(\theta_{ee})) \end{bmatrix} \quad (25)$$

Lyapunov theory is used to demonstrate the stability of the control rule in order to become asymptotically stable at feedback system because this theory is access to a successful and simple method to getting the kinematic nonholonomic stability; thus, the constructional Lyapunov functions theory is described by Eq. (26) [13]:

$$V = \frac{1}{2}(x_{ee}^2 + y_{ee}^2) + \frac{1}{K_y}(1 - \cos(\theta_{ee})) \quad (26)$$

The derivative time of Equation (27) to be:

$$\dot{V} = \dot{x}_{ee} x_{ee} + \dot{y}_{ee} y_{ee} + \frac{1}{K_y} \dot{\theta}_{ee} \sin(\theta_{ee}) \quad (27)$$

Substituting Eq. (24) in Eq. (18), the derivative state vector error becomes as follows [18]:

$$\dot{S}_{ee} = \begin{bmatrix} \dot{x}_{ee} \\ \dot{y}_{ee} \\ \dot{\theta}_{ee} \end{bmatrix} = \begin{bmatrix} (\Omega_r + V_r(K_y y_{ee} + K_\theta \sin(\theta_{ee})))y_{ee} - K_x x_{ee} \\ -(\Omega_r + V_r(K_y y_{ee} + K_\theta \sin(\theta_{ee})))y_{ee} + V_r \sin(\theta_{ee}) \\ -(K_y V_r y_{ee} + K_\theta V_r \sin(\theta_{ee})) \end{bmatrix} \quad (28)$$

Then,

$$\begin{aligned} \dot{V} = & ((\Omega_r + V_r(K_y y_{ee} + K_\theta \sin(\theta_{ee})))y_{ee} - K_x x_{ee})x_{ee} \\ & + (-(\Omega_r + V_r(K_y y_{ee} \\ & + K_\theta \sin(\theta_{ee})))y_{ee} + V_r \sin(\theta_{ee})) \\ & y_{ee} + \frac{1}{K_y} (-(K_y V_r y_{ee} + K_\theta V_r \sin(\theta_{ee})) \sin(\theta_{ee})) \end{aligned} \quad (29)$$

$$\dot{V} = -K_x x_{ee}^2 - V_r \frac{K_\theta}{K_y} \sin^2 \theta_{ee} \leq 0 \quad (30)$$

Clearly,  $V \geq 0$ . If  $S_{ee} = 0$ ,  $V = 0$ . If  $S_{ee} \neq 0$ ,  $V > 0$  and  $\dot{V} \leq 0$ . if  $S_{ee} = 0$ ,  $\dot{V} = 0$ . if  $S_{ee} \neq 0$ ,  $\dot{V} < 0$ .

Then,  $V$  represents the Lyapunov function that will make the closed loop system globally asymptotically stable with three weighting parameters of control ( $K_x, K_y, K_\theta$ )  $> 0$ .

### 3.2 Swarm-Based Optimization Algorithms

The stochastic algorithm which uses multiple agents (solutions) to move through the search space in the process of solving an optimization problem is known as

the optimization algorithm. Some of these effective stochastic techniques that mimic the behaviours of certain animals or insects (birds, ants, bees, flies and even germs) are called Nature-Inspired Algorithm used for auto-tuning controllers. Two of these techniques are discussed here:

### 3.2.1 Firefly Algorithm

The firefly algorithm (FA), based on the behavior of fireflies flying towards an illumination source and their interaction with bioluminescent signals, is one of the algorithms relationship to the group of swarm algorithms (developed by Xin-She Yang [11]). The phenomenon of a firefly moving towards the brighter individual is the basis of the algorithm. One of the rules used in the firefly algorithm is that all fireflies are unisex. Furthermore, attractiveness of fireflies is proportional to the intensity of their sent-out light, where in the light intensity specified by the value of the objective function decreases with increasing distance between the fireflies. If there is no more attractive individual, a firefly moves randomly [20, 21].

#### 3.2.1.1 Attractiveness

Each firefly has a certain light intensity  $I$ , which varies according to the distance  $r$  between two individuals, and attractiveness  $A$ , which is proportional to the light intensity seen by the neighboring fireflies. Therefore, attractiveness ( $A$ ) is dependent on distance and the light absorption coefficient  $\gamma$  [21]:

$$A = A_0 e^{-\gamma r^2} \quad (31)$$

where  $A_0$  denotes the attractiveness at  $r = 0$ .

#### 3.2.1.2 Distance

The distance between any two fireflies  $m$  and  $n$  at  $X_m$  and  $X_n$ , respectively, is the Cartesian distance as follows:

$$r_{mn} = \|X_m - X_n\| = \sqrt{\sum_{L=1}^d (X_{m,L} - X_{n,L})^2} \quad (32)$$

Where  $X_{m,L}$  is the  $L$ th component of the spatial coordinate  $X_m$  of  $M$ th firefly and  $d$  is the number of dimensions.

#### 3.2.1.3 Movement

The movement, during which the firefly  $m$  being in the location  $X_m$  tries to get closer to the brighter individual  $n$  in the position  $X_n$  is determined by the following equation [21]:

$$X_m = X_m + A_0 e^{-\gamma r_{mn}^2} (X_n - X_m) \alpha (rand - 0) \quad (33)$$

where  $X_m$  is the current location of a firefly  $m$ , the second term denotes attractiveness and the third term is due to random movement ( $rand$  is a random number generator uniformly distributed in the range  $[0, 1]$ , and  $\alpha \in [0, 1]$ ).

### 3.2.2 Crossed Firefly Algorithm

In standard firefly algorithm, if there no brighter one than a 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:

$$\text{Alpha} = (\delta) * (\text{alpha}^{\text{iteration}}) \quad (34)$$

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

$$X_m = (\alpha) * X_n + (1 - \alpha) * X_m \quad (35)$$

The possibilities of crossover are using to select fireflies instead of selecting randomly, then the candidates for the best solution are increasing. On the other hand, FA became needs less time to search for the best solution and its performance significantly development with the increases the population size because of decreasing the randomness, the pseudocode at algorithm 1.

Algorithm 1. The summarization of the computation procedure of CF algorithm

**Step 1:** Initialize the pop of fireflies  $X_m$ .  
**Step 2:** Evaluate the fitness function  $f(X_m)$ .  
**Step 3:** Set Flash absorption coefficient  $\gamma$ ,  $A_0$ ,  $\alpha$ .  
**Step 4:** Light intensity  $I_m$  at  $X_m$  is calculated.  
**Step 5:** While the halting criteria is not satisfied do  
     For  $m=1$ : S (for all S fireflies).  
     For  $n=1$ : S (for all S fireflies).  
     if ( $I_n > I_m$ ) Movement all firefly  $m$  towards  $n$  ( $d$ -dimension) by Eq. (33) conording to the luminousness between  $m$  and  $n$  can be determined by Eq. (31).  
     else  
     Movement all fireflies by Eq. (35).  
     End if.  
     Evaluate the new fireflies and update flash intensities.  
     End for  $n$   
     End for  $m$   
**Step 6:** Order the fireflies and find the best.  
**Step 7:** End.

### 3.2.3 Design of Kinematic Nonlinear Controller based on CF Algorithm

This work introduced a Crossed firefly algorithm to find the optimum parameter of control. The searching procedure of the suggested algorithm is described as follows:

1. Initial searching elements of each firefly are created randomly within the determined range. Note that the

dimension of search space consists of all the control parameters needed in the back-stepping nonlinear control as described in Fig. 4.

2. The fitness function for each firefly can be calculated by the mean square error theory as Eq. (36) from [18]:

$$(MSE) = \frac{1}{2} \sum_{p=1}^{pop} (x_r(L)^p - x(L)^p)^2 + (y_r(L)^p - y(L)^p)^2 + (\theta_r(L)^p - \theta(L)^p)^2 \quad (36)$$

3. Set step 3 to step 7 of the CF algorithm pseudo code to get the perfect value of controller's gain by minimizing trajectory-following error.

4. If the epochs number reach at max, then exit, else, go to step 2.

#### IV. SIMULATION

The aim of the simulations is to examine the effectiveness and performance of the proposed back-stepping kinematic nonlinear control depending on original firefly algorithm and CF optimization algorithms to a differentiation two-wheeled drive mobile robot by programming using MATLAB environment.

The resulting of a differentiation two-wheeled drive mobile robot trajectory-following, obtained by the suggested kinematic nonlinear control is involving trajectory-following, tracking-following error, the linear velocity of right and left wheel, linear and angular velocity of the mobile robot and (MSE). The execution of work can be simulated by the following table of the setting parameter (Firefly elements is equal to 3, NI-robot length is equal to 0.36 m and NI-robot radius is equal to 0.05 m).

##### Case study : (Lemniscates trajectory)

The desired trajectory-following lemniscates, can be expressed by the following Equations:

$$x_r(\tau) = 0.75 + 0.75 * \sin(2\pi\tau/50) \quad (37)$$

$$y_r(\tau) = \sin(4\pi\tau/50) \quad (38)$$

$$\theta_r(\tau) = 2 \tan^{-1} \left( \frac{\Delta y_r(\tau)}{\sqrt{(\Delta x_r(\tau))^2 - (\Delta y_r(\tau))^2} + \Delta x_r(\tau)} \right) \quad (39)$$

- The initial desired parameterization of robot at  $q_r = [0.75, 0, 0]$  and the initial actual parameterization of robot at  $q = [0, 0, 0]$ , then the simulation of trajectory-following lemniscates of the NI mobile robot using back-stepping kinematic control based on original Firefly learning algorithm is viewed in Fig. 6. And the simulation of trajectory-following lemniscates of the NI mobile robot using back-stepping kinematic controller based on CF learning algorithm is viewed in Fig. 7.

- The left and right velocity of back-stepping kinematic control based on original firefly algorithm is shown in Fig. 8, and for back-stepping kinematic controller based on CF learning is shown in Fig. 9.
- The performance index MSE for parameterization trajectory-tracking during 10 epochs are described in Fig. 10 which is (0.866) for the proposed control based on original firefly learning algorithm and Fig. 11 which is (0.7957) for the proposed control based on CF in the simulation case study, which has perfect parameterization trajectory-following performance and it had the ability of generating suitable and smooth velocity without spike.
- Figure 12 proves the average of linear velocity (0.3 m/s) and the top peak of the angular velocity ( $\pm 1$  rad/s) of the NI robot for the proposed control based on original firefly learning algorithm and Fig. 13 proves the average of linear velocity (0.1m/s m/s) and the top peak of the angular velocity ( $\pm 1$ rad/s) of the NI robot for proposed control based on CFA.
- Instantaneous suggested control parameters are shown in Fig. 14 which is tuned by using Crossed firefly algorithm.
- The performance described in (Fig. 15, Fig. 16 and Fig. 17) for the proposed control based CF algorithm is obvious by showing the convergence of parameterization trajectory-following errors with small value.

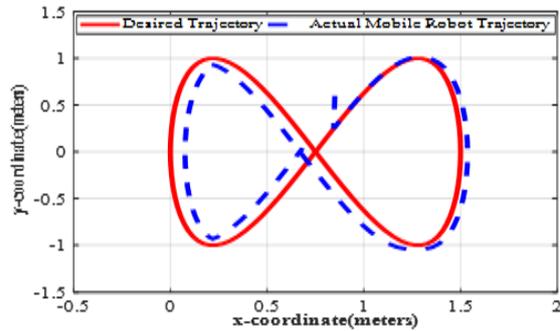


Fig.6. Actual and desired Lemniscates trajectory-following.

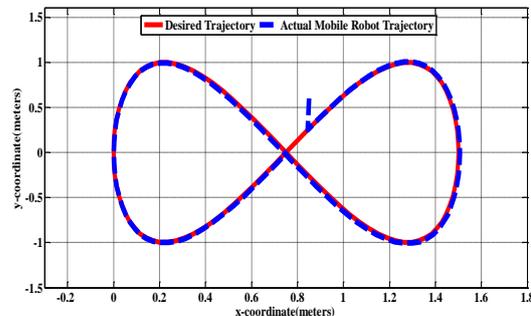


Fig.7. Actual and desired Lemniscates trajectory-following.

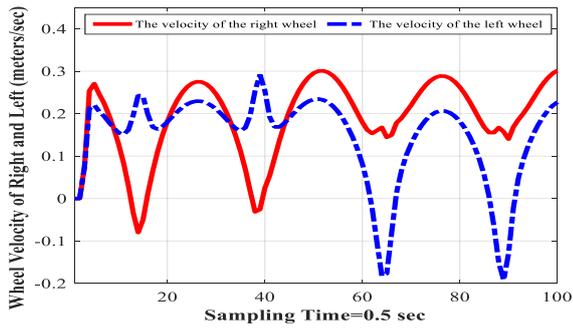


Fig.8. The right and left wheel velocity.

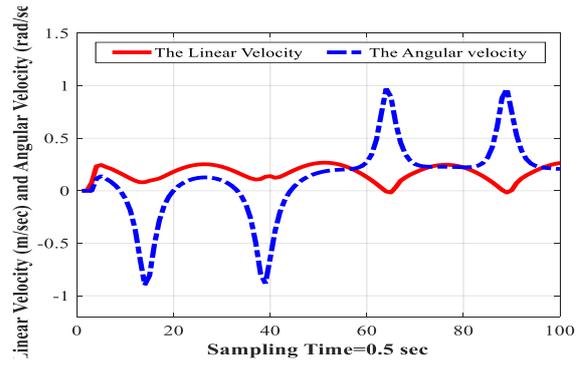


Fig.12. The linear and angular velocity.

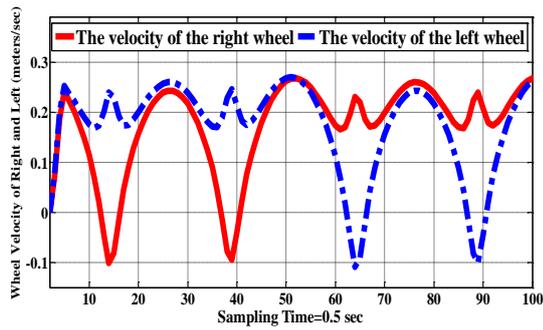


Fig.9. The right and left wheel velocity.

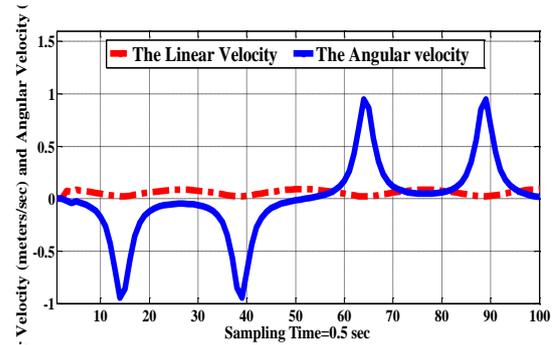


Fig.13. The linear and angular velocity.

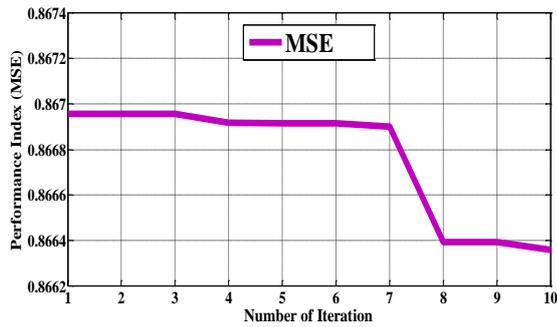


Fig.10. The performance index (MSE).

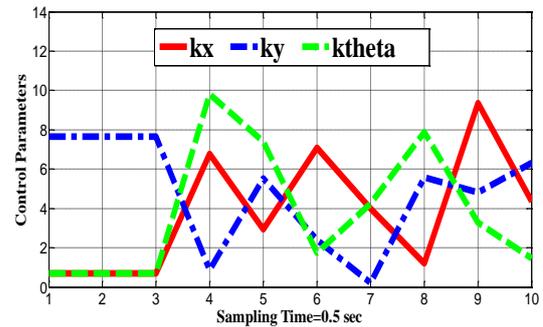


Fig.14. Control Parameters.

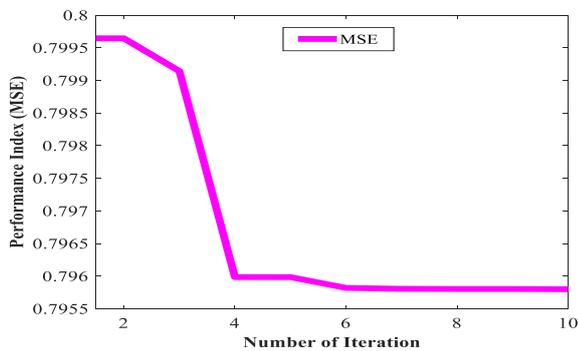


Fig.11. The performance index (MSE).

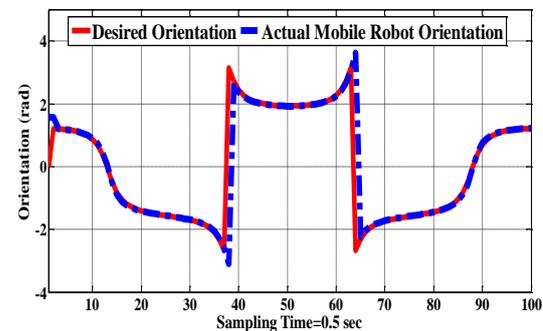


Fig.15. Orientation tracking error

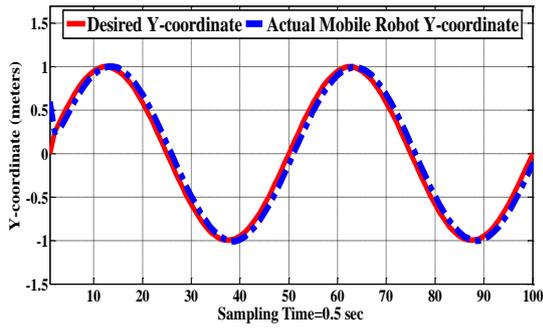


Fig.16. Position tracking error in Y- coordinate.

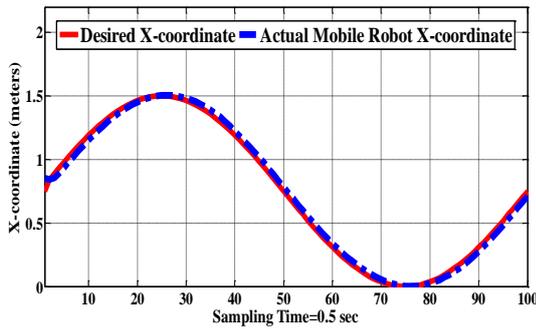


Fig.17. Position tracking error in X- coordinate.

V. CONCLUSION

The trajectory-following nonlinear kinematic control based on back-stepping theory with two optimization algorithms for the differentiation nonholonomic two wheeled drive mobile robot have been presented in this work. The suggested controller examined by MATLAB (R2017b) set on National Instrument mobile robot (NI Mobile Robot). The simulation results show evidently the capability of robustness and adaptation of the suggested back-stepping kinematic controller based on Crossed Firefly algorithm which optimum trajectory-following performance and it has the capability of generating more suitable and smooth velocities than the controller based on firefly algorithm because the CF algorithm has capability to get optimum controller parameters.

ACKNOWLEDGMENT

All thanks and praise be to Allah, the Compassionate, and the Merciful for all His Blessings. I would like to express my deep sense of gratitude and respect for Asst. Prof. Dr. Ahmed Al-Araji, Control & System Dept. University of Technology, for the excellent guidance.

List of Symbols

Symbol                      Definition

Alpha	The flat crossover coefficient.
C	The mobile robot center (meter).
L	The dimensional vector of FA. Parameter to reduce the randomness.
<i>delta</i>	The flash intensity.
A	The intensity at the beginning.
A0	The back-stepping control's gains
( $\kappa_x, \kappa_y, \kappa_\theta$ )	
L	The number of samples for mobile robot
L1	The length between each wheel and the robot axis of symmetry in y- guidance (m).
$R_a$	The radius of each wheel (rad). distance from the ICCM to the midpoint between the two wheels (c)
Rc	The distance between any two fireflies
r	The random number generator uniformly distributed in [0, 1].
<i>rand</i>	
R	Real number
rnn	The Cartesian distance between fireflies.
S	The pose vector in the surface
$S_{ee}$	The error vector
V	The linear velocity of the robot (meter/second).
$V_d$	The desired linear velocity of the robot (meter/second)
$V_L$	The linear speed of left mobile robot wheel (meter/second).
$V_R$	The linear speed of right mobile robot wheel (meter/second).
( $x, y, \theta$ )	The actual pose (position/orientation) of the robot.
( $x_r, y_r, \theta_r$ )	The desired pose (position/orientation) of the robot.
( $x_{ee}, y_{ee}, \theta_{ee}$ )	The configuration errors
$\alpha 1$	The randomization parameter
$\Delta\tau$	The sampling period between two sampling times.
$\Omega$	The angular velocity of the robot (rad/second).
$\Omega_d$	The desired angular velocity of the robot (rad/second)
$R$	The rotation matrix
$\mathbb{R}$	Real time.
$\tau$	Lyapunov functions
$\dot{V}$	Time derivative of Lyapunov functions
$\dot{V}$	

## REFERENCES

- [1] Kumar D N, Samalla H, Rao Ch J, Naidu S, Jose K A, Kumar B M. Position and Orientation Control of a Mobile Robot Using Neural Networks[J]. Computational Intelligence in Data Mining, Smart Innovation, Systems and Technologies, 2015, 2(13): 123-131.
- [2] Seo K, Lee J S. Kinematic path-following control of mobile robot under bounded angular velocity error[J]. Advanced Robotics, 2006, 20(1): 1-23.
- [3] Kanayama Y, Kimura Y, Miyazaki F, Noguchi T. A stable tracking control method for an autonomous mobile robot[J]. IEEE International Conference on Robotics and Automation, 1990: 384-389.
- [4] Zain A A, Daobo W, Muhammad S, Wanyue J, Muhammad Sh. Trajectory Tracking of a Nonholonomic Wheeled Mobile Robot Using Hybrid Controller[J]. International Journal of Modeling and Optimization, 2016, 6(3): 136-141.
- [5] Kolmanovsky I, McClamroch N H. Developments in nonholonomic control problems[J]. IEEE control systems, 1995: 20-26.
- [6] Blazic S. A novel trajectory-tracking control law for wheeled mobile robots[J]. Robot. Auton. Syst. 59, 2011: 1001-1007.
- [7] Jiang Z P, Pomet J B. Combining backstepping and time-varying techniques for a new set of adaptive controllers[J]. IEEE Int. Conf. Decision Contr., 1994: 2207-2212.
- [8] Guldner J, Utkin V I. Stabilization of nonholonomic mobile robots using Lyapunov functions for navigation and sliding mode control[J]. IEEE Int. Conf. Decision Contr., 1994: 2967-2972.
- [9] Geem Z W, Kim H J, Loganathan G V. A new heuristic optimization algorithm: Harmony search[J]. SAGE Journals, 2001, 76(2): 60-68.
- [10] Adithyan T, Vasudha Sh, Gururaj B, Chandrasegar Th. Nature inspired algorithm[C]. International Conference on Trends in Electronics and Informatics (ICEI), pp. 1131 - 1134.
- [11] Yang X S. Firefly Algorithms for Multimodal Optimization[J]. SAGE Journals, vol. 5792, pp. 169-178, 2009.
- [12] Nizar H, Basma J. Trajectory Tracking Controllers for Mobile Robot: Modeling, Design and Optimization[M]. Lambert Academic Publishing, 2016.
- [13] Al-Araji A. Design of a cognitive neural predictive controller for mobile robot[M]. Ph.D. thesis, Brunel University, United Kingdom, 2012.
- [14] Nizar H, Basma J. Design of a Kinematic Neural Controller for Mobile Robots based on Enhanced Hybrid Firefly-Artificial Bee Colony Algorithm[J]. AL - Khwarizmi Engineering Journal, vol. 12, no. 1, pp. 45-60, 2016.
- [15] Ye J. Adaptive control of nonlinear PID-based analogue neural network for a nonholonomic mobile robot[J]. Neurocomputing, vol. 71, no. 7, pp. 1561-1565, 2008.
- [16] Yousif Z, Hedley J, Bicker R. Design of an adaptive neural kinematic controller for a national instrument mobile robot system[J]. IEEE International Conference on Control System, Computing and Engineering, pp. 623-628, 2012.
- [17] Yuan G. Tracking Control of a Mobile Robot using Neural Dynamics based Approaches[M]. Master Thesis, University of Guelph, 2001.
- [18] Al-Araji A. Development of kinematic path-tracking controller design for real mobile robot via back-stepping slice genetic robust algorithm technique[J]. Arab J. Sci. Eng. vol. 39, no. 4, pp. 8825-8835, 2014.
- [19] Al-Araji A. Design of on-line nonlinear kinematic trajectory tracking controller for mobile robot based on Optimal Back-Stepping Technique[J]. Iraqi Journal of Computers, Communications, Control and Systems Engineering. vol. 2, no. 14, pp. 25-36, 2014.
- [20] Xing B, Gao W J. Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms[J]. Computational Intelligence and Complexity, 2014.
- [21] Yang X S. Nature-Inspired Metaheuristic Algorithms[M]. Second Edition, Luniver Press, 2010.

## Authors' Profiles



**Basma J. Saleh** was born on Feb. 13, 1988. M.Sc., Electrical Engineering dept. @ Baghdad Uni. 2015, B. Sc., Computer Engineering dept. @ AL Mustansiriyah Uni. 2010. Academic staff member in Computer Engineering department @Al-Mustansiriyah University. Interested area: Artificial Neural Networks, intelligent algorithms, Optimization Methods,

Robotic controller, and image processing.



**Ali T. al-Aqbi** was born on Feb. 29, 1988. M.Sc., Wright State Uni., United States of America 2017, B. Sc. Uni. of Technology, Iraq 2010. Academic staff member in Computer Engineering department @Al-Mustansiriyah University. Interested area: Artificial Neural Networks, intelligent algorithms, Optimization Methods, Robotic controller,

and image processing.



**Ahmed Y. Saedi** was born on Feb. 18, 1986. M.Sc., Ulyanovsk State Technical Uni., Russia 2017, B. Sc., Computer Engineering dept. @ AL Mustansiriyah Uni. 2007. Academic staff member in Computer Engineering department @Al-Mustansiriyah University. Interested area: Artificial Neural Networks, intelligent algorithms, Optimization Methods,

Robotic controller, and image processing.



**Lamees A. Salman** was born on Des. 25, 1985. B. Sc. Uni. of Technology, Iraq 2007. Academic staff member in Computer Engineering department @Al-Mustansiriyah University. Interested area: Artificial Neural Networks, intelligent Optimization algorithms.

**How to cite this paper:** Basma Jumaa Saleh, Ali Talib Qasim al-Aqbi, Ahmed Yousif Falih Saedi, Lamees abdalhasan Salman, "Comparative Study of Inspired Algorithms for Trajectory-Following Control in Mobile Robot", *International Journal of Modern Education and Computer Science(IJMECS)*, Vol.10, No.9, pp. 1-10, 2018.DOI: 10.5815/ijmecs.2018.09.01