

Design Interval Type-2 Fuzzy Like (PID) Controller for Trajectory Tracking of Mobile Robot

Mohamed J. Mohamed¹, Mustafa Y. Abbas²

^{1,2}Control and Systems Engineering Department, University of Technology, Baghdad, Iraq

Abstract— One of the major problems in the field of mobile robots is the trajectory tracking problem. There are a big number of investigations for different control strategies that have been used to control the motion of the mobile robot when the nonlinear kinematic model of mobile robots was considered. The trajectory tracking control of autonomous wheeled mobile robot in a changing unstructured environment needs to take into account different types of uncertainties. Type-1 fuzzy logic sets present limitations in handling those uncertainties while type-2 fuzzy logic sets can manage these uncertainties to give a superior performance. This paper focuses on the design of interval type-2 fuzzy like proportional-integral-derivative (PID) controller for the kinematic model of mobile robot. The firefly optimization algorithm has been used to find the best values of controller's parameters. The aim of this controller is trying to force the mobile robot tracking a pre-defined continuous path with minimum tracking error. The Matlab simulation results demonstrate the good performance and robustness of this controller. These were confirmed by the obtained values of the position tracking errors and a very smooth velocity, especially with regards to the presence of external disturbance or change in the initial position of mobile robot. Finally, in comparison with other proposed controllers, the results of nonlinear IT2FLC PID controller outperform the nonlinear PID neural controller in minimizing the MSE for all control variables and in the robustness measure.

Index Terms— Firefly algorithm, Interval type-2 fuzzy like (PID) controller, Mobile robot, Trajectory tracking.

I. INTRODUCTION

The investigations of different control strategies for differential wheeled mobile robots have been increased recently. The mobile robot is a mechanical device that is able to move in an environment with a specific degree of autonomy which is used in various services and industrial applications, such as transportation, factory automation, etc. The mobile robot with non-holonomic constraints means the motion of the robot is not completely free, where the robot can be moved in some directions (forward and backward) but not in others (sideways) [1]-[2].

The central problem of the mobile robot caused by the wheel's motion has freedom in three degrees (x , y , θ) whereas for the mobile robot control, it needs only two control signals (U_L, U_R) under the kinematic and non-holonomic constraints. There are several published studies to solve the control problems of mobile robot (navigation problems) which can be divided into three types: The first type is the position estimated control method for problems of navigation. The second type is path planning and implementation while the third type is trajectory tracking which is the goal of this work [3].

Received 6 June 2018; Accepted 4 December 2018

There are three prime causes that increase tracking error for mobile robots. The first one is due to the discontinuity of the path rotation radius of the differential driving mobile robot. The second is due to the small radius of rotation overlap with the exact drive of the mobile robot. The third is due to the radius of rotation, which is not uniform either randomly curvature or complicated curve [3].

There are various controllers that have been suggested for trajectory tracking of mobile robots. Some of these proposed controllers are conventional controllers that applied linear or nonlinear feedback control, while the rest used other techniques such as predictive control technique [3]. The adaptive trajectory tracking controller applied to the robot dynamics was discussed in [4]. Moreover, an intelligent control architecture using neural network and fuzzy logic for two wheeled mobile robots was developed in [5]. Fuzzy logic controllers (FLCs) type one (T1) and type two (T2) were applied to wheeled mobile robot in [6]-[7].

In this paper, an interval type-2 fuzzy logic controller (IT2FLC) is proposed to control the tracking of the mobile robot according to the predefined trajectory in the presence of disturbances with a minimum tracking error. The proposed structure of the IT2FLC is like PID which is a combination of proportional plus integral (PI) T2FLC and proportional plus derivative (PD) T2FLC with input and output gains as will be explained later. It is known that fuzzy PI control type is more feasible than fuzzy PD control type, since it is hard for the fuzzy PD to eliminate the steady state error. While the fuzzy PI control type is famous that it gives bad performance in the transient response due to the inner integration process. To combine the performance of fuzzy PI control and fuzzy PD control at the same time, an IT2FLC like (PI+PD) or (PID) is chosen to keep the accurate characteristics of the PID controller by employing the error and the rate of change of error as its inputs [8]. The gains of this IT2FLC like (PID) are tuned using the firefly algorithm (FA) as the optimization method to find the best values of controller gains.

The remainder of the paper is arranged as follows. In section-2, IT2FLC is introduced. In section-3, the kinematic model of differential wheel mobile robot is explained. In section-4, the suggested interval type-2 fuzzy logic system (IT2FLS) like (PID) trajectory tracking controller design is illustrated. In section-5 the proposed FA is explained in detailed steps. In section-6, the simulation and results are depicted and the conclusion is given in section-7.

II. INTERVAL TYPE-2 FUZZY LOGIC CONTROLLER

IT2FLC is selected in this work, because it is very convenient to apply to a highly nonlinear model and to process data and information affected by unprobabilistic uncertainty. These types of FLCs were designed to mathematically represent the ambiguity and uncertainty of linguistic problems [9]. The FLC for the tracking trajectory of a mobile robot in unstructured environments faces a lot of sources of uncertainty, some of these sources of uncertainty are listed as follows [10];

- The input uncertainties entered to the FLC which are translated to uncertainties in the antecedent's membership functions (MFs) as the sensor's measurements where their characteristics are changed by the environmental conditions such as (humidity, wind, rain, sunshine, etc.).
- The control action uncertainties which translate to uncertainties in the output MFs of the FLC. These uncertainties can result from the actuator's characteristics change due to the variations in the ground or changes in the environment. For example, the meaning of a

“fast speed” on a dry ground in sunny day may be different from “fast speed” on a muddy ground in a rainy day as wheels may slip, etc.

- The linguistic uncertainties as the exact meaning of the words utilized in the antecedents and consequents linguistic labels can be uncertain as the words that mean different things to different people. Moreover, experts do not always concur and they often prepare different consequents for the same antecedents.

Type-2 fuzzy logic systems (T2FLSs) were proposed by Zadeh [11]. FLS usually employs T1 fuzzy sets and symbolizes uncertainty by numbers in the range of [0,1] which are indicated as degrees of membership. T2 fuzzy sets are expansions of T1 fuzzy sets with an addition of the third dimension that coincides to the uncertainty about the degrees of membership. However, MFs of T2 are fuzzy themselves. Interval T2 fuzzy set is the simplest T2 fuzzy set whose element's degrees of membership are intervals with a secondary membership degree of (1.0) [9]. In T2 fuzzy set the grade of belonging of an element is expressed as T1 fuzzy number within (0, 1). This means the degree to which an element belongs to the set is uncertain at T2 fuzzy set \tilde{F} over domain X which is given by:

$$\mu_{\tilde{F}}(x, u): X * [0,1] \rightarrow [0,1] \quad \forall x \in X, \forall u \in J_x \subseteq [0,1] \quad (1)$$

$$\tilde{F} = \{(x, u), \mu_{\tilde{F}}(x, u)\} \quad (2)$$

$$\tilde{F} = \int_{x \in X} \int_{u \in J_x \subseteq [0,1]} \frac{\mu_{\tilde{F}}(x, u)}{(x, u)} \quad (3)$$

Where, X is the universe of discourse domain, x represents a primary domain element in the X, u is a secondary domain element of $J_x \subseteq [0,1]$ and \tilde{F} is T2 fuzzy set. J_x is called primary MF, the primary membership J_x is an interval between lower MF $\mu_{\underline{F}}(x)$ and upper MF $\mu_{\overline{F}}(x)$ as given in equation (4) [12].

$$J_x = [\mu_{\underline{F}}(x), \mu_{\overline{F}}(x)] \quad (4)$$

The uncertainty in the primary memberships of a T2 fuzzy set is shown in Fig.1. It consists of a bounded region that is called the Footprint of Uncertainty (FOU). The computations of an IT2FLC are presented in [9].

$$FOU(\tilde{F}) = \cup_{x \in X} J_x = \{(x, u) : u \in J_x \subseteq [0,1]\} \quad (5)$$

Many researchers debate to support IT2FLS because of its possibility to model and reduce the effects of dynamic uncertainties [9]. Usually, in various applications the performance of IT2FLSs is compared to their T1 counterparts, which often give a better performance, particularly when noise and uncertainty are considered in the system. The improved performance may belong to the FOU of the IT2FLSs, which provide additional dimension for designing the fuzzy MFs as shown in Fig.1.a. [13].

As shown in Fig.1.b the third dimension everywhere for the mysterious IT2 group has the same value. In other words, the third dimension of the IT2 fuzzy set doesn't include new information. The IT2FLS block diagram is depicted in Fig.2.

If we take the Gaussian interval T2 MF as an example, Fig.3 shows two types of a Gaussian IT2FLC MF. The first is a Gaussian IT2FLC MF with variance width (σ), and the second is a Gaussian IT2FLC MF with variance center (c) presented in Fig.3.a and b, respectively [14].

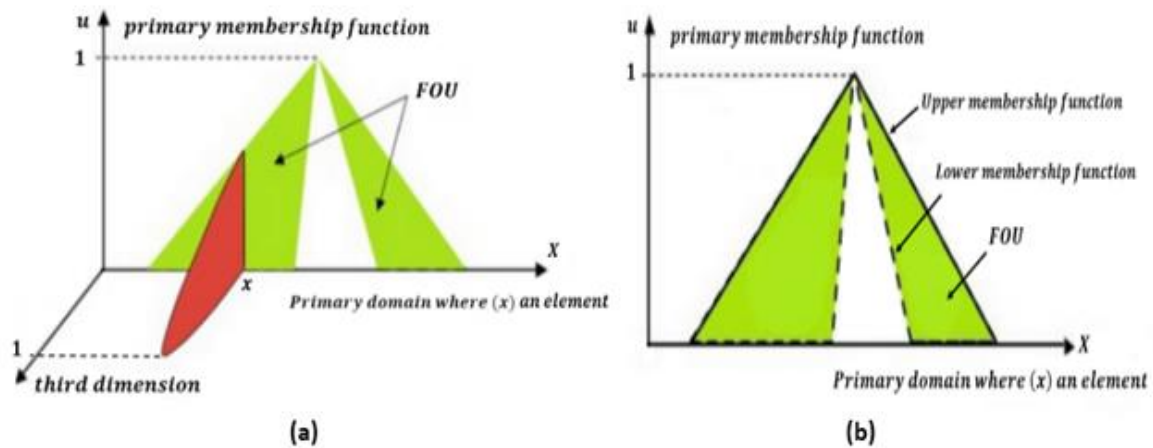


FIG. 1. T2 TRIANGULAR MF. [13]

(a) GENERAL T2 TRIANGULAR MF. (b) INTERVAL T2 TRIANGULAR MF

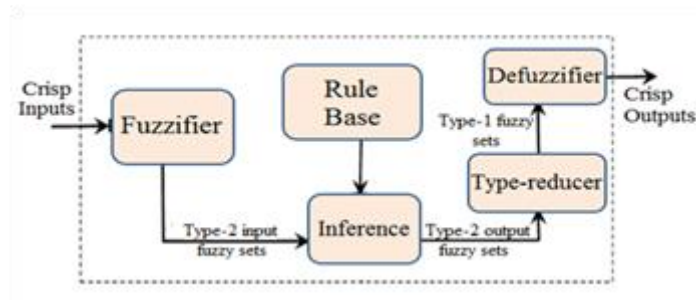


FIG. 2. BLOCK DIAGRAM OF A T2FLS. [13]

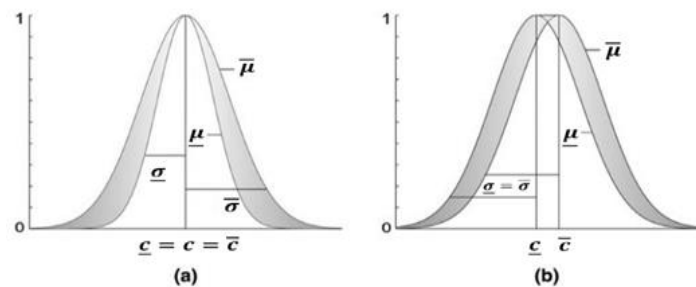


FIG. 3.a. GAUSSIAN IT2FLC MF WITH VARIANCE WIDTH (Σ).

FIG. 3.b. GAUSSIAN IT2FLC MF WITH VARIANCE CENTER (C). [14]

III. THE KINEMATIC MODEL OF DIFFERENTIAL WHEEL MOBILE ROBOT

The differential drive mobile robot scheme is shown in Fig.4. The structure of mobile robot consists of a Wagon body, two wheels for driving installed on the same shaft and established in the rear of the cart and an omnidirectional wheel in the front of the cart. The front wheel holds the mechanical construction and makes the platform of the mobile robot steadier. Two autonomous similar DC motors represent the actuators of right and left wheels move and orient the mobile robot [15] [16]. The two driving wheels have the same radius denoted by R (m), and $2L$ (m) is the space between the two wheels. The center of

gravity of the mobile robot is situated at point c, and there is a distance from the center of mass to the center of the axis of the wheels, which is denoted by a.

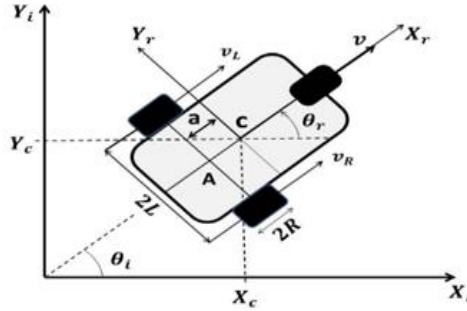


FIG. 4. THE MOBILE ROBOT STRUCTURE AND COORDINATES. [16]

Where, $\{X_i, Y_i, \theta_i\}$ is an inertial frame (global frame) that represents the static coordinate system in the robot plane, and $\{X_r, Y_r, \theta_r\}$ is the robot's frame (local frame) coordinate system that is attached to the robot [2].

θ : Angel of rotation for mobile robot (rad).

v : The linear or (average) velocity of the platform in the robot frame (m/sec).

ω : The rotational velocity of the cart in the global and local frames (rad/sec).

The aim of kinematic modeling of the robot is to make the velocity of the robot in the global frame as a function of the robot geometrical parameters and the two rear wheel velocities.

$$\dot{q} = [\dot{x} \ \dot{y} \ \dot{\theta}]^T \tag{6}$$

Where should establish the robot speed as a function of the wheel rotation velocities $\dot{\phi}_R$ and $\dot{\phi}_L$ (rad/sec) which means the rotational velocity of the right and left wheels, respectively. The velocity of any wheel in the local frame is $R\dot{\phi}$, where R (m) is radius of the wheels. Therefore, the average velocity in the local (robot) frame is the translational velocity or linear velocity $v(t)$ (m/sec) that combines the linear velocities of left wheel $v_L(t)$ (m/sec) and right wheel $v_R(t)$ (m/sec) [16]:

$$v_L(t) = R \dot{\phi}_L(t) \tag{7}$$

$$v_R(t) = R \dot{\phi}_R(t) \tag{8}$$

$$v(t) = R \frac{\dot{\phi}_L(t) + \dot{\phi}_R(t)}{2} = \frac{v_L(t) + v_R(t)}{2} \tag{9}$$

and the rotational velocity or angular velocity is:

$$\omega(t) = \frac{d\theta(t)}{dt} = \frac{R}{2L} (\dot{\phi}_L(t) - \dot{\phi}_R(t)) \tag{10}$$

The velocities also can be defined by immediate curvature radius of the robot trajectory that is denoted by $r(t)$ (m), which is the distance from the instantaneous center of curvature or rotation (ICC or ICR) to the midpoint (A) in the middle of two wheels [2], and L (m) is the distance between each driving wheel and point (A), as shown in Fig.5.

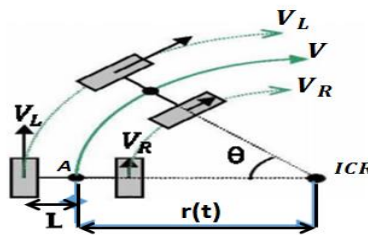


FIG. 5. INSTANTANEOUS CENTER OF ROTATION. [17]

$$v(t) = \omega(t) \cdot r(t) \tag{11}$$

$$v_L(t) = (r(t) + L) \omega(t) \tag{12}$$

Received 6 June 2018; Accepted 4 December 2018

$$v_R(t) = (r(t) - L) \omega(t) \quad (13)$$

After solving the above equations, the rotational velocity and the instantaneous curvature radius of the robot trajectory according to the midpoint axis (c) are found as given in the following equations: [2].

$$\omega(t) = \frac{v_L(t) - v_R(t)}{2L} \quad (14)$$

$$r(t) = \frac{L(v_L(t) + v_R(t))}{v_L(t) - v_R(t)} \quad (15)$$

It is simulating the non-holonomic constraint of a wheeled mobile robot, which represents the ideal rolling with no lateral slip [2]:

$$\dot{y} \cos\theta(t) - \dot{x} \sin\theta(t) = 0 \quad (16)$$

The kinematic equations of the mobile robot that represent the speed in the inertial or global frame are [2]:

$$\dot{x}(t) = v(t) \cos\theta(t) \quad (17)$$

$$\dot{y}(t) = v(t) \sin\theta(t) \quad (18)$$

$$\dot{\theta}(t) = \omega(t) \quad (19)$$

In order to take out the position and orientation of the wheeled mobile robot, the above equations are integrated as illustrated below.:

$$x(t) = x_o + \int_0^t v(t) \cos\theta(t) dt \quad (20)$$

$$y(t) = y_o + \int_0^t v(t) \sin\theta(t) dt \quad (21)$$

$$\theta(t) = \theta_o + \int_0^t \omega(t) dt \quad (22)$$

In the program simulation, they are expressed as the current format of the equations:

$$x(k) = x(k-1) + \frac{1}{2} [v_L(k) + v_R(k)] \cos\theta(k) \Delta t \quad (23)$$

$$y(k) = y(k-1) + \frac{1}{2} [v_L(k) + v_R(k)] \sin\theta(k) \Delta t \quad (24)$$

$$\theta(k) = \theta(k-1) + \frac{1}{2L} [v_L(k) - v_R(k)] \Delta t \quad (25)$$

Where, Δt is the sampling time. The above equations are utilized to construct the model of the mobile robot and applied to simulate the mobile robot in the Matlab program.

IV. IT2FLS LIKE (PID) TRAJECTORY TRACKING CONTROLLER DESIGN

Commonly, FLC principle is used to control systems or processes which are complicated or difficult to model and/or nonlinear and complex. The complete modified FLS consists of a single-loop feedback control system. The requirements to design an FLC so as to track a mobile robot desired path are, the data of desired trajectory, a structure of FLC, a mobile robot model, and an optimization algorithm which is employed to tune the gains of the controller. In the final stage of the controller design, the constructed control scheme should be able to solve the control problem of the mobile robot. This problem belongs to the motion of the wheels that have freedom in three degrees (x, y, θ). The three control signals of (x, y, θ) will be transformed into two control signals (U_L, U_R) as shown in Fig.6. The feedback of fuzzy PID controller is very essential to stabilize the error of the path tracking system when the mobile robot output is deviated from the desirable point. The general proposed structure of the nonlinear IT2FLC like (PID) for mobile robot model is shown in Fig.7.

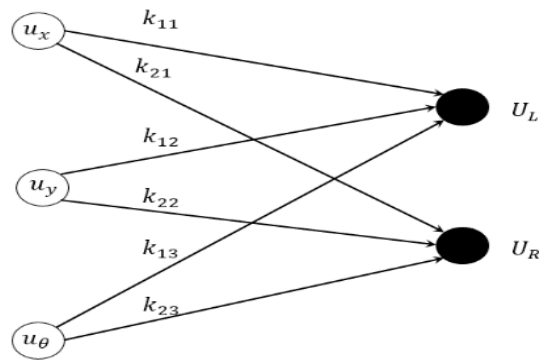


FIG. 6. THE INTERNAL STRUCTURE OF THE NETWORK

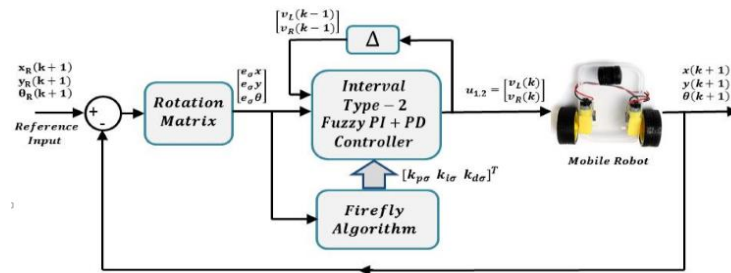


FIG.7. THE GENERAL PROPOSED STRUCTURE OF IT2FLC LIKE PID FOR MOBILE ROBOT [2].

In fact, the proposed nonlinear IT2FLC like PID cannot be amounted to state of classical PID controller due to two proportional gains that exist in the structure of the fuzzy PID controller, as shown in Fig.8. The physical connection between derivative and integral time constants remains the same. FLC can be modified using the same method as used for classical controllers, where the increase of proportional gain or decrease of integral time constant leads the system to higher oscillations in the time response. In the other side, the oscillations of the time response can be compensated to some range by the increase of derivative time constant. In slow time response case, it can increase the proportional gain or decrease the integral time constant to make the response faster [12].

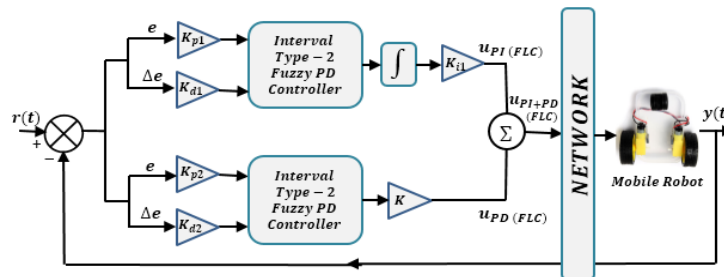


FIG. 8. THE INTERNAL STRUCTURE OF IT2FLC LIKE PID.

The output of the PI fuzzy controller in its standard form is the integral of output of the PD fuzzy controller as shown:

$$u_{PI}(t) = \int_0^t \left[K_p e(t) + K_d \frac{d}{dt} e(t) \right] dt \tag{26}$$

$$u_{PI}(t) = K_i \int_0^t e(\tau) d\tau + K_p e(t) \tag{27}$$

Where $u(t)$ is the control output, $K_p, K_d,$ and K_i are the proportional, derivative and integral gains, respectively, The PI FLC which is called the incremental control output and denoted by $\Delta u_{PI}(t)$ can obtain the control signal $u_{PI}(t)$, as represented by the following equation [18]:

$$u_{PI}(t) = u_{PI}(t-1) + \Delta u_{PI}(t) \quad (28)$$

The typical PID control law in its standard form is:

$$u_{PID}(t) = K_p \left[e(t) + T_d \frac{d}{dt} e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau \right] \quad (29)$$

Where T_d is the derivative time constant and T_i is the integral time constant. The equation can also be written as [18]:

$$u_{PID}(t) = K_p e(t) + K_d \frac{d}{dt} e(t) + K_i \int_0^t e(\tau) d\tau \quad (30)$$

Where: $K_d = K_p T_d$ and $K_i = \frac{K_p}{T_i}$, so the fuzzy controller, which is constructed from individual PI and PD controllers, has a combined output to perform as a fuzzy like PID controller :

$$u_{PID}(t) = u_{PI}(t) + u_{PD}(t) \quad (31)$$

The design of FLS includes the determination of input scale factors, a rule base, a proper choice of the MFs, and output scale factors. The input scale factors convert the real inputs of fuzzy controller into normalized values to be in the range (-1,1), and the normalized outputs of fuzzy controller are converted to real values using output scale factors [2]. Several researches have indicated that the performance of FLS depends mainly on the design of MF more than the design of rule base [19].

T2 Gaussian MF is a common technique for designing fuzzy sets where it is used in this work because it has the advantage of being smooth and nonzero at all points [2]. The Gaussian MF is specified by two parameters $\{c, \sigma\}$. The general computations of upper and lower MFs for both types are given by the equations below: [14]

(a) The general computation formulas of Gaussian IT2 MF with variance width σ that varies between lower MF width $\underline{\sigma}$ and upper MF width $\bar{\sigma}$, where $\sigma \in [\underline{\sigma}, \bar{\sigma}]$ are as follows.

$$\mu_{upper}(x, \bar{\sigma}, c) = \exp \left[-\frac{1}{2} \left(\frac{x-c}{\bar{\sigma}} \right)^2 \right] \quad (32)$$

$$\mu_{lower}(x, \bar{\sigma}, c) = \exp \left[-\frac{1}{2} \left(\frac{x-c}{\underline{\sigma}} \right)^2 \right] \quad (33)$$

(b) The general computation formulas of Gaussian IT2 MF with variance center c that varies between lower MF center \underline{c} and upper MF center \bar{c} are given by equations (34) and (35), respectively, where $c \in [\underline{c}, \bar{c}]$ [14]:

$$\mu_{upper}(x, \sigma, \underline{c}, \bar{c}) = \begin{cases} \exp \left[-\frac{1}{2} \left(\frac{x-\underline{c}}{\sigma} \right)^2 \right] & \text{if } x < \underline{c} \\ 1 & \text{if } \underline{c} \leq x \leq \bar{c} \\ \exp \left[-\frac{1}{2} \left(\frac{x-\bar{c}}{\sigma} \right)^2 \right] & \text{if } x > \bar{c} \end{cases} \quad (34)$$

$$\mu_{lower}(x, \sigma, \underline{c}, \bar{c}) = \begin{cases} \exp \left[-\frac{1}{2} \left(\frac{x-\bar{c}}{\sigma} \right)^2 \right] & \text{if } x \leq \frac{\underline{c}+\bar{c}}{2} \\ \exp \left[-\frac{1}{2} \left(\frac{x-\underline{c}}{\sigma} \right)^2 \right] & \text{if } x > \frac{\underline{c}+\bar{c}}{2} \end{cases} \quad (35)$$

The standard T2 Gaussian MF (type of variance center) is used in the proposed IT2FLC for error (e), a derivative of the error (e') and FLC output (u_{FLC}), as shown in Fig.9. The difference between upper and lower center is equal to 0.2, and constant width σ is equal to 0.1415. The inputs and output relationship is depicted by a two dimensional linear rule base as shown in Table I and the Mamdani type inference engine. There are seven fuzzy linguistic variables that were used NB, NM, NS, Z, PS, PM, PB represent Negative Big, Negative Medium, Negative Small, Zero, Positive Small, Positive Medium and Positive Big, respectively. So, there are 49 rules in the rule base. The FLC output (u_{FLC}) is determined by using the centroid method of defuzzification.

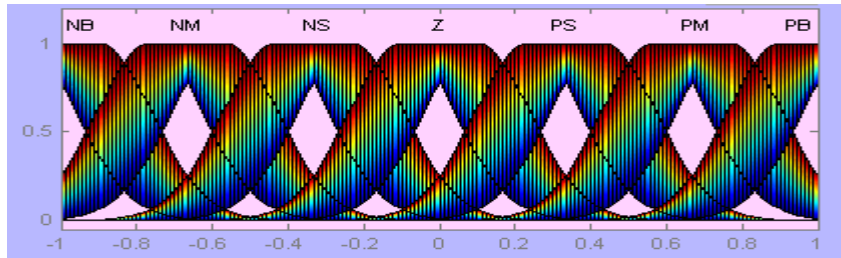


FIG. 9. IT2FLS MFS FOR ERROR, ERROR DERIVATIVE AND OUTPUT OF FLC.

Many of recent researches [9] proposed to use only the diagonal rules. So, the developed controller uses only the 7 diagonal fuzzy rules (highlighted with yellow color in Table I) to simplify the controller and reduce the complex computations. The complete proposed controller has six IT2FLC blocks, each block has seven IT2 Gaussian MFs and the defuzzification operation is implemented using the centroid method and output scale factors.

TABLE I. RULE BASE FOR ERROR, ERROR DERIVATIVE AND FLC OUTPUT.

e \ e'	NB	NM	NS	Z	PS	PM	PB
NB	PB	PB	PB	PB	PM	PS	Z
NM	PB	PM	PB	PM	PS	Z	NS
NS	PB	PM	PS	PS	Z	NS	NS
Z	PM	PM	PS	Z	NS	NM	NM
PS	PS	PS	Z	NS	NS	NM	NB
PM	PS	Z	NS	NM	NB	NM	NB
PB	Z	NS	NM	NB	NB	NB	NB

There are many different optimization algorithms available to use for optimizing the parameters of fuzzy like PID controller. Some of these algorithms, which are often employed, are genetic algorithms (GA), FA, particle swarm optimizations (PSO), etc.

In this work FA is used to find the optimal parameters of the IT2FLC like PID. The objective of optimization for the controller's parameters is to minimize the tracking error of the actual trajectory relative to the desired trajectory of the mobile robot.

V. FIREFLY ALGORITHM

The major mission of a firefly's flash is to work as a signal system in order to attract other fireflies. The FA works with some assumptions, as shown below [20] [21]:

1. The fireflies are unisexual, where any single firefly can be tempted to each other fireflies.
2. The attraction of any firefly is proportionate to its lightening, where for any two fireflies, the one that has less light will be tempted or attracted by (and consequently move towards) the brighter one, but the (visible lightening) intensity is reduced when the mutual distance (r) increases.
3. If there are no fireflies, which are brighter than a specified firefly, it will move at random. The lightening must be connected with the objective function.

The procedure of FA is illustrated by the following steps:

Step-1: Define the objective function of the given optimization problem $f(x_i^k)$ where, $k=1, 2, \dots, d$ is number of parameters and $i=1, 2, \dots, n$ is number of fireflies.

Step-2: Generate a random initial population of fireflies x_i ($i=1,2,..n$) using the following equation (36).

$$x_i^k = (LB)^k + rand. \cdot ((UB)^k - (LB)^k) \quad (36)$$

Where, LB and UB denote the lower and the upper bounds of k -th parameter and $rand$ is a uniform distribution random number between (0,1).

Step-3: Specify the firefly algorithm's coefficients: the light absorption coefficient γ , where γ varies from 0.1 to 10, the maximum attractiveness β_0 (when $r = 0$), in most cases $\beta_0 = 1$, the constant $\alpha \in (0, 1)$, and maximum number of iterations.

Step-4: Calculate the value of the objective function (fitness value) $f(x_i)$ for $i=1, 2, \dots, n$.

Step-5: While ($G <$ maximum number of iterations) do the following steps.

Step-6: For $i=1$ to n (all n fireflies) do the following steps.

Step-7: For $j=1$ to n (all n fireflies) do the following steps.

Step-8: Calculate the light intensity I_i and I_j for the two fireflies i and j . where $I \propto f(x)$, or simplified by $I = f(x)$. The light intensity can be defined by:

$$I = I_0 e^{-\gamma r} \quad (37)$$

Where, I_0 is the original light intensity, (r) is the distance between a firefly i and a firefly j . The Cartesian distance between two fireflies i and j is given by:

$$r_{ij} = \|x_i - x_j\|_2 \quad (38)$$

$$r_{ij} = \sqrt{\sum_{k=1}^d (x_i^k - x_j^k)^2} \quad (39)$$

where, x_i^k is the k -th parameter (component) of the i -th firefly.

Step-9: If ($I_j > I_i$) move firefly i toward firefly j as shown in following operations;

The attractiveness is proportional to the light intensity seen by adjacent fireflies and thus the attraction β of a firefly j is determined by the following equation.

$$\beta = \beta_0 e^{-\gamma r_{ij}^2} \quad (40)$$

The movement of attracted firefly i to another more attractive (brighter) firefly j is determined by;

$$x_i^{t+1} = x_i^t + \beta (x_j^t - x_i^t) + \alpha \epsilon_i \quad (\text{movement is updated in all parameters}). \quad (41)$$

where the i th firefly's current position is represented by the first term, the firefly's attractiveness is represented by the second term and if there are no any brighter firefly, the last term is used for the random movement, where ϵ_i is a vector of random numbers which may be replaced by $(rand_i - \frac{1}{2})$.

Step-10: Else (if not ($I_j > I_i$)) do the following operations.

$$x_i^{t+1} = x_i^t + \alpha \epsilon_i \quad (\text{random movement to update each parameter}). \quad (42)$$

If the new position produces higher attractiveness value, the firefly is moved to the new position; otherwise the firefly will remain in the current position.

Step-11: End of if statement.

Step-12: Calculate the $f(x_i)$ for the new solution.

Step-13: End for j

Step-14: End for i

Step-15: Rank the fireflies and find the current best

Step-16: End of while (return to step-5 if $G <$ maximum number of iterations)

Step-17: Post process results and visualization;

Step-18: End of algorithm [20].

VI. SIMULATION AND RESULT

The modified IT2FLC like PID controller is applied on the kinematic model of differential mobile robot and also is verified with the computer simulation using Matlab 2014. In simulation study an infinity path and star path are used as reference trajectories to measure the difference between the actual mobile robot path and reference path. The two wheeled mobile robot physical characterization was clarified, where the mobile robot consists of two motors for the two rear wheels and one front wheel for

balance. The model dimensions of the mobile robot are taken as follows: Length=0.4 m, width = 0.2 m, tire width =0.1m, tire length=0.2 and the sampling time (T) is equal to 0.1 second. The FA was used to adjust the parameters of IT2FLC like PID controllers depending on the tracking error between mobile robot actual path and reference path. The FA setting is as follows; the population size $n=20$, number of parameters $d=24$, the lower and upper bounds for all parameters ($LB=-1$ and $UB=1$), $\gamma=1$, $\alpha=0.5$, $\beta_o=0.2$, and maximum number of iterations=700. The values of firefly parameters ($n, \beta_o, \alpha, \gamma$) are chosen by trial and error. The best solution present in the last iteration is considered as the result of FA. The system is tested for two cases, the first one is infinity trajectory which is defined in equations (43, 44, and 45) [2].

$$x_R(n) = 0.75 + 0.75 * \sin\left(2 * \pi * \frac{t}{100}\right) \tag{43}$$

$$y_R(n) = \sin\left(4 * \pi * \frac{t}{100}\right) \tag{44}$$

$$\theta_R(n) = atan2\left[\left(\frac{y_R(n)-y_R(n-1)}{t+\epsilon}\right), \left(\frac{x_R(n)-x_R(n-1)}{t+\epsilon}\right)\right] \tag{45}$$

The second case is star trajectory which is defined in equations (46, 47, and 48) [2].

$$x_R(n) = -2.5 * \sin\left(2 * \pi * \frac{t}{30}\right) \tag{46}$$

$$y_R(n) = 2.5 * \sin\left(2 * \pi * \frac{t}{20}\right) \tag{47}$$

$$\theta_R(n) = atan2\left[\left(\frac{y_R(n)-y_R(n-1)}{t+\epsilon}\right), \left(\frac{x_R(n)-x_R(n-1)}{t+\epsilon}\right)\right] \tag{48}$$

The best values of controller’s gains resulting from the last iteration of FA are appearing in Table II. These values are applied in the mobile robot simulation.

The reference and actual trajectories of the infinity path are shown in Fig.10, both with same initial position [0.75, 0, 0] which denoted to $[x, y, \theta]$ of the reference and actual trajectories, where they were represented by blue and red lines, respectively. Fig.s 11.a, b and c. show the actual and desired path for x, y and θ , respectively. Fig.s 12.a and b show the left and right velocities (v_R, v_L), respectively. Fig.s 13.a, b and c. show error for x, y, and θ , respectively. Finally, Fig. 14 represents the MSE of the path.

TABLE II. THE IT2FLC LIKE PID GAINS

Parameters	Value	Parameters	Value	Parameters	Value
k_{px1}	-4.9816	k_{iy}	1.6649	$k_{d\theta2}$	0.067018
k_{dx1}	-0.9764	k_{py2}	0.75326	k_{θ}	-0.37705
k_{ix}	5.6971	k_{dy2}	0.9504	k_{11}	4.483
k_{px2}	-0.7258	k_y	0.44458	k_{12}	2.6002
k_{dx2}	-0.2177	$k_{p\theta1}$	-3.2346	k_{13}	-0.70379
k_x	1.1452	$k_{d\theta1}$	-0.35453	k_{21}	4.4896
k_{py1}	0.77734	$k_{i\theta}$	2.7733	k_{22}	-0.10925
k_{dy1}	1.253	$k_{p\theta2}$	0.34293	k_{23}	2.7368

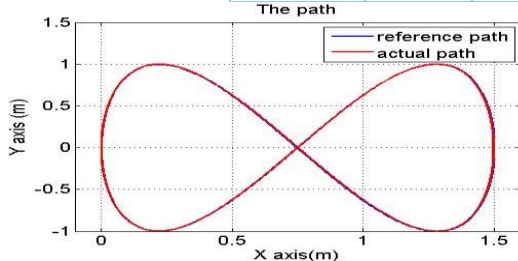


FIG. 10. REFERENCE AND ACTUAL PATHS.

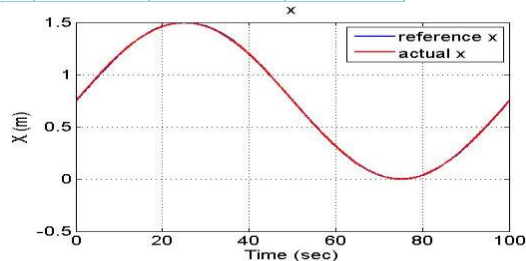


FIG.11.a. THE RESPONSE OF X VARIABLE.

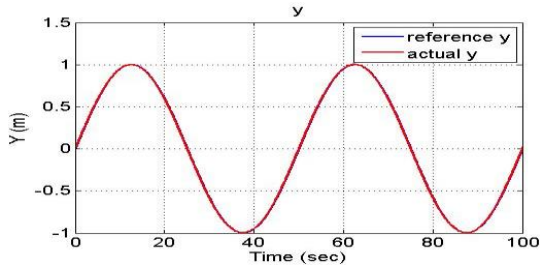


FIG. 11.b. THE RESPONSE OF Y VARIABLE.

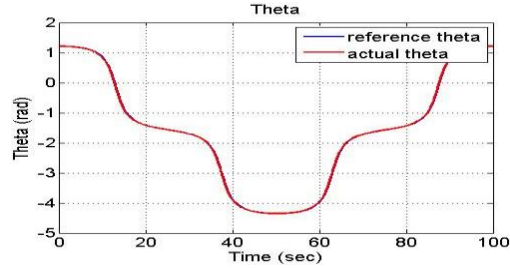


FIG. 11.c. THE RESPONSE OF θ VARIABLE.

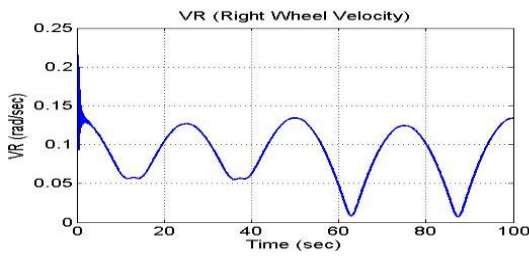


FIG. 12.a. THE RIGHT WHEEL VELOCITY.

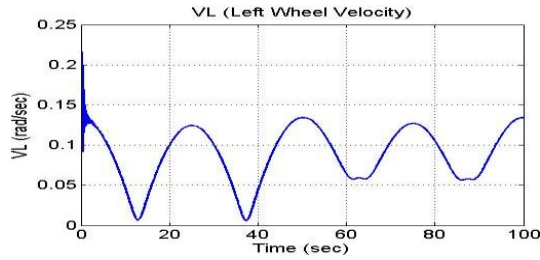


FIG. 12.b. THE LEFT WHEEL VELOCITY.

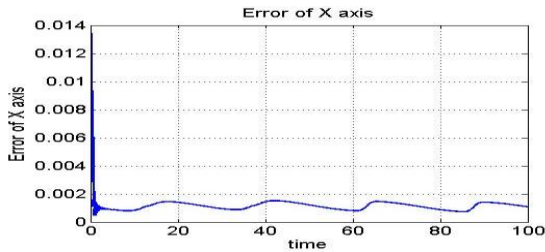


FIG. 13.a. THE ERROR OF X VARIABLE.

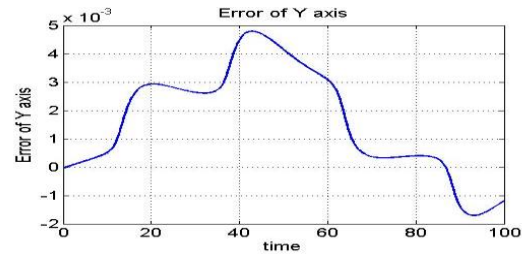


FIG. 13.b. THE ERROR OF Y VARIABLE.

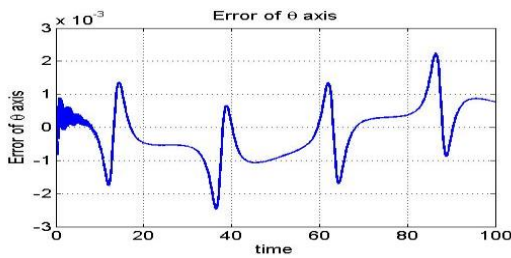


FIG. 13.c. THE ERROR OF θ VARIABLE.

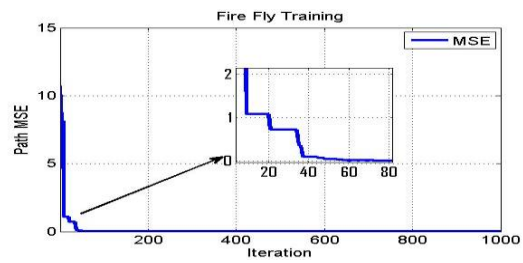


FIG. 14. THE MSE OF PATH TRACKING.

The reference and actual trajectories of the star path are shown in *Fig.15*, both with same initial position $[0, 0, -4.1243]$ which denoted to $[x, y, \theta]$, where they were represented by blue and red lines, respectively. *Figs 16.a, b and c.* show the reference path and actual path for x, y and θ , respectively. The right wheel and left wheel velocities (v_R, v_L) are shown in *Figs 17.a and b.*, respectively. *Figs. 18.a, b and c.* show error for x, y , and θ , respectively. Finally, MSE of the path is depicted in *Fig.19*.

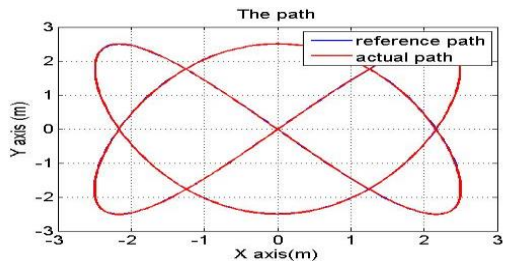


FIG. 15. REFERENCE AND ACTUAL PATHS.

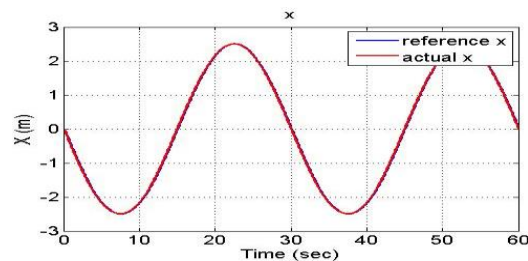


FIG. 16.a. THE RESPONSE OF X VARIABLE.

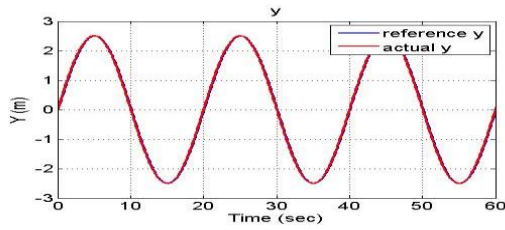


FIG. 16.b. THE RESPONSE OF Y VARIABLE.

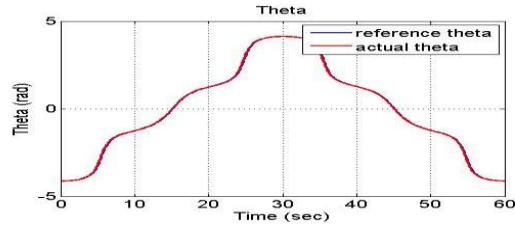


FIG. 16.c. THE RESPONSE OF θ VARIABLE.

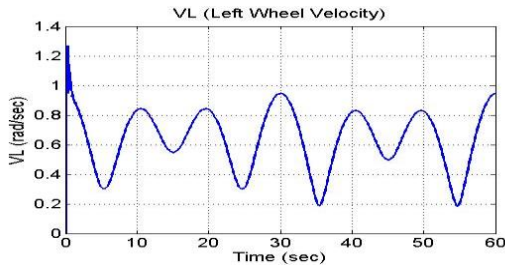


FIG. 17.a. THE RIGHT WHEEL VELOCITY.

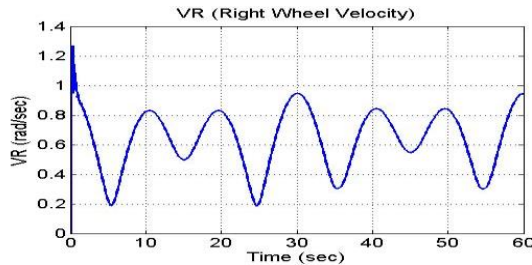


FIG. 17.b. THE LEFT WHEEL VELOCITY.

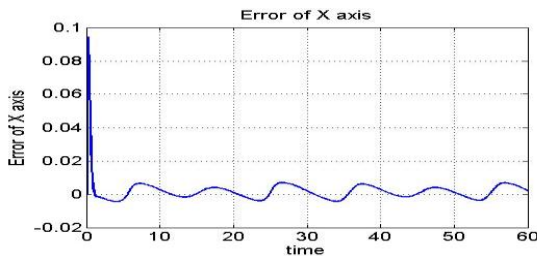


FIG. 18.a. THE ERROR OF X VARIABLE.

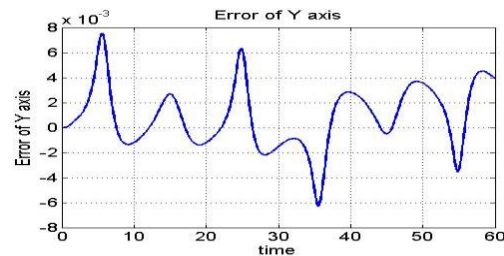


FIG. 18.b. THE ERROR OF Y VARIABLE.

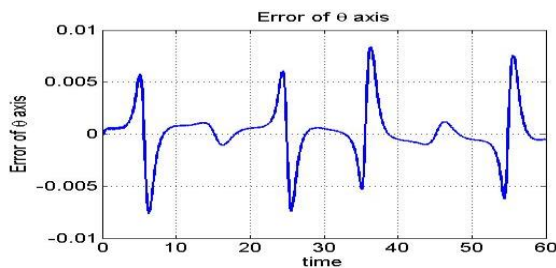


FIG. 18.c. THE ERROR OF θ VARIABLE.

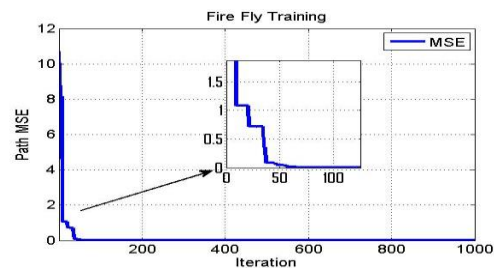


FIG. 19. THE MSE OF PATH TRACKING

TABLE III. THE MSE VALUES OF TWO CASES.

Trajectory	MSE of x-axis	MSE of y-axis	MSE of θ -axis	MSE of Path
Path 1	2.3603×10^{-7}	2.8634×10^{-8}	2.3867×10^{-7}	5.0333×10^{-7}
Path 2	2.1666×10^{-5}	5.2435×10^{-6}	3.4916×10^{-6}	3.0401×10^{-5}

To check the robustness of the IT2FLC like PID controller, another initial posture is taken to test the ability of the proposed controller to track the mobile robot on the desired path.

To test the infinity path, if we change the initial position of $[x, y, \theta]$ from $[0.75, 0, 0]$ to $[0.8, 0.03, 0]$ and add an unmodeled disturbance term $[0.1\sin(2t), 0.1\sin(2t)]$ to control actions $[v_R, v_L]$ for proving the robustness of IT2FLC like PID controller.

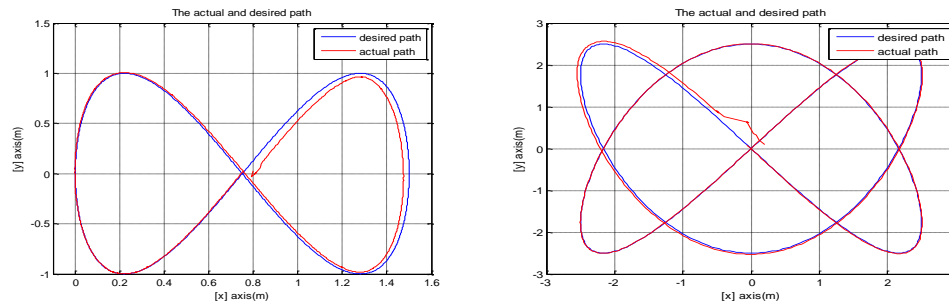


FIG.20.a and b. REFERENCE AND ACTUAL PATHS WITH PRESENCE OF DISTURBANCE AND INITIAL POSITION CHANGE IN (INFINITY, STAR) TRAJECTORY RESPECTIVELY

Therefore, without retraining the parameters (gains) of the IT2FLC like PID controller, the obtained result is shown in *Fig.20.a*. The values of MSEs of (e_x, e_y, e_θ) are $(1.3 \times 10^{-5}, 0.00041, 0.00137)$, respectively. This result is much better than the result of using a neural controller based on position and orientation predictor [21] although disturbance was not added to the model. Another test for star path is considered, if we change the initial position of $[x, y, \theta]$ from $[0, 0, -4.1243]$ to $[0.2, 0.1, -4.1243]$ and add an unmodeled disturbance term $[0.1\sin(2t), 0.1\sin(2t)]$ to the control actions $[v_R, v_L]$. Then, without retraining the parameters (gains) of IT2FLC like PID controller so that to confirm the robustness and the ability of the controller. The obtained result is shown in *Fig.20.b*. which is much better than the result of nonlinear PID neural trajectory tracking controller used in [3]. Table IV depicts the values of MSEs of both controllers. The results show that the proposed controller has a good ability to reduce the difference between actual and desired paths quickly and then follow the desired path with a good accuracy.

TABLE IV. THE COMPARISON OF MSE VALUES WITH [3] FOR STAR PATH.

MSE of Nonlinear Interval Type-2 Fuzzy PID Controller with Firefly Algorithm			MSE of Nonlinear PID Neural Trajectory Tracking Controller with PSO Algorithm		
$e(x)$	$e(y)$	$e(\theta)$	$e(x)$	$e(y)$	$e(\theta)$
0.000355	0.00136	0.00242	0.0013	0.0038	1.18

From the figures of MSEs in Table IV, it is clear that the IT2FLS like PID controller gives better performance and robustness than that of the nonlinear PID neural controller used in [3]. The results confirm that the IT2FLC like PID controller is feasible, robust and able to follow the desired trajectory without re-training its parameters.

VII. CONCLUSIONS

The nonlinear IT2FLC like PID tuned by FA technique was applied on the kinematic model of nonholonomic wheeled mobile robot to track predefined continuous trajectories. In fact, the FA was used off-line to tune 24 parameters (controller parameters plus network connection weights) of IT2FLC like PID controller. The objective of the tuning is to minimize the gathering of the mean square errors between the desired and actual values of x , y and θ through the mobile trip.

The resultant *Fig.10* and *Fig.15* show that the proposed controller has a good ability to reduce the difference between actual and desired paths quickly and then follow the desired path with a good accuracy. In addition, the results show clearly that the suggested controller has an ability to produce suitable and smooth left and right velocities without sharp heels as shown in *Fig.12 a, b* and *Fig.17 a, b*.

It is clear that the IT2FLC like PID controller gives better performance and robustness compared to that of the nonlinear PID neural controller used in [3] where the values MSE_x , MSE_y and MSE_θ belong to the neural controller (0.0013, 0.0038, and 1.18) were reduced to (0.000355, 0.00136, and 0.00242), respectively in the IT2FLC like PID controller. Moreover, without re-training the proposed controller parameters, it demonstrated the ability of tracking the continuous desirable trajectory, especially when the robot has started from arbitrary initial position and a disturbance term is added to

its model as shown in *Fig.20 a and b*. Finally, the results confirm that the IT2FLC like PID controller is feasible, robust and able to follow the desired trajectory without re-training its parameters.

REFERENCES

- [1] A. Al-Araji, "Design of On-Line Nonlinear Kinematic Trajectory Tracking Controller for Mobile Robot based on Optimal Back Stepping Technique", Iraqi Journal of Computers, Communications, Control and Systems Engineering Vol.14, No.2, 2014.
- [2] Mohammed K. Hamza "Optimized intelligent kinematic Controller to the Wheeled Mobile Robot for Trajectory Tracking", MSc. Thesis, the Control and Systems Engineering Department, University of Technology, Iraq, 2014.
- [3] A. Al-Araji, K.E. Dagher, " Design of a Nonlinear PID Neural Trajectory Tracking Controller for Mobile Robot based on Optimization Algorithm", Eng. &Tech. Journal, Vol.32, Part (A), No.4, 2014.
- [4] Park B. S., S. J. Yoo, J. B. Park and Y. H. Choi, "A simple adaptive control approach for trajectory tracking of electrically driven nonholonomic mobile robots", IEEE Transactions on Control Systems Technology. Vol.18, No. 5, pp.1199-1206, 2010.
- [5] K-H. Su, Y-Y. Chen and S-F. Su, "Design of neural-fuzzy-based controller for two autonomously driven wheeled robot", Neuro computing. Vol. 73, pp. 2478-2488, 2010.
- [6] Bakir L., Jasmin V. , " Evolutionary Design of Fuzzy Logic Based Position Controller for Mobile Robot", Journal of Intelligent Robot System, Vol.63, pp.595-614, 2011.
- [7] Mohammad H. K., Alireza A., Davood N., Maryam A., " Teaching-learning based optimal interval type-2 fuzzy PID Controller Design: a nonholonomic Wheeled Mobile Robots" Robotica, pp.1 of 13 Cambridge University, 2013.
- [8] Zhi-Wei W., Hung Y. Chung, Jin J. Lin "A PID type fuzzy controller with self-tuning scaling factors" Fuzzy Sets and Systems 115, pp. 321-326, 2000.
- [9] Mohammed Y. Hassan, Saba T. Salim, Yamama A. Shafeek, "Interval Type-2 Fuzzy Control for Mean Arterial Pressure by Isoflurane Infusion During Anesthesia", Emirates Journal for Engineering Research, Vol.17 (1), pp.63-71, 2012.
- [10] Hani A. Hagra, "A Hierarchical Type-2 Fuzzy Logic Control Architecture for Autonomous Mobile Robots", IEEE Transactions on Fuzzy Systems, Vol. 12, No. 4, August, 2004.
- [11] Oscar C., Patricia M., " Type-2 Fuzzy Logic: Theory and Applications", Springer-Verlag Berlin, Heidelberg, 2008.
- [12] Nesrine B., Robert J., Adel M. A., "Interval Type-2 Fuzzy Logic Control of Mobile Robots" Journal of Intelligent Learning Systems and Applications, Vol. 4, pp. 291-302, 2012.
- [13] Dongrui W., "A Brief Tutorial on Interval Type-2 Fuzzy Sets and Systems", Ming Hsieh Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089-2564, USA.
- [14] Qilian L. and Jerry M. Mendel, "Interval Type-2 Fuzzy Logic Systems Theory and Design", IEEE Transactions on Fuzzy Systems, VOL. 8, No. 5, October, 2000.
- [15] Al-Araji A. S., M. F. Abbod and H. S. Al-Raweshidy, " Design of a neural predictive controller for nonholonomic mobile robot based on posture identifier", Proceedings of the IASTED International Conference Intelligent Systems and Control (ISC), pp. 198-207, July, 2011.
- [16] Omid M., "Mobile Robot Trajectory Tracking Using Neural Networks", M.Sc. Thesis, College of Engineering, American University of Sharjah, U.A.E, December, 2009.
- [17] Nizar H. Abbas, Basma J. saleh "Design of a Kinematic Neural Controller for Mobile Robots based on Enhanced Hybrid Firefly-Artificial Bee Colony Algorithm" Al.Khwarizmi Engineering Journal, Vol. 12, No. 1, pp. 45- 60, 2016.
- [18] Petr P., "Comparative Analysis of Fuzzy PI/PD/PID Controller Based on Classical PID Controller Approach", IEEE International Conference on Fuzzy Systems, Honolulu, Volume: I, January, 2002.
- [19] Indranil P., Saptarshi D., Amitava G. "Tuning of an optimal fuzzy PID controller with stochastic algorithms for networked control systems with random time delay" ISA Transactions, Vol 50, pp.28-36, 2011.
- [20] Nur F. Johari, Azlan M. Zain, Noorfa H. Mustaffa , and Amirmudin U. "Firefly Algorithm for Optimization Problem" Applied Mechanics and Materials Vol.421 (2013) pp 512-517 @ (2013) Trans Tech Publications, Swizerland.
- [21] Abeer F. Shimal, " Neural Controller for Nonholonomic Mobile Robot System Based on Position and Orientation Predictor", Iraqi Journal of Computers, Communications, Control and Systems Engineering, Vol.11, NO.1, 2011.