Big Bang–Big Crunch optimization based interval type-2 fuzzy PID cascade controller design strategy

Article *in* Information Sciences · October 2014

DOI: 10.1016/j.ins.2014.06.005

Tufan Kumbasar
Istanbul Technical University

Hani Hagras
University of Essex

Information Sciences 282 (2014) 277–295

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Big Bang–Big Crunch optimization based interval type-2 fuzzy PID cascade controller design strategy

Tufan Kumbasar [a,*], Hani Hagras [b,c]

[a] The Control and Automation Engineering Department, Faculty of Electrical and Electronics Engineering, Istanbul Technical University, Maslak, TR-34469 Istanbul, Turkey
[b] The Computational Intelligence Centre, School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, United Kingdom
[c] Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia

ARTICLE INFO

Article history:
Received 23 November 2013
Received in revised form 26 March 2014
Accepted 1 June 2014
Available online 14 June 2014

Keywords:
Interval type-2 fuzzy logic system
Interval type-2 fuzzy set
Interval type-2 fuzzy PID controller

ABSTRACT

Fuzzy logic control is a recognized approach for handling the faced uncertainties within control applications. However, type-1 fuzzy controllers using crisp type-1 fuzzy sets might not be able to fully handle the high levels of uncertainties and nonlinear dynamics associated with real world control applications. On the other hand, interval type-2 fuzzy controllers using Interval Type-2 Fuzzy Sets (IT2-FSs) might be able to handle such uncertainties to produce a better control performance. However, the systematic design of interval type-2 fuzzy controllers is still a challenging problem due to the difficulty in determining the parameters of the IT2-FSs. In this paper, we will present the novel application of Big Bang–Big Crunch optimization (BB–BC) approach to optimize the antecedent membership parameters of Interval Type-2 Fuzzy PID (IT2-FPID) controllers in a cascade control structure. Since the IT2-FPID control structure involves more design parameters compared to its type-1 counterpart, it is beneficial to employ the BB–BC method which has a low computational cost and a high convergence speed. The presented BB–BC based optimized IT2-FPID cascade structure will be compared with its Type-1 Fuzzy PID (T1-FPID) and conventional PID controller counterparts which were also optimized with the BB–BC optimization. In addition, the proposed IT2-FPID structure will be compared with a self-tuning T1-FPID control structure. We will then present the novel application of the cascade control architecture to solve the path tracking control problem of mobile robots which inherits large amounts of uncertainties caused by the internal dynamics and/or feedback sensors of the controlled system. Several experiments were performed in simulation and in real world using the PIONEER 3-DX mobile robot which will act as a platform to evaluate the proposed control systems in this paper. The results illustrated that the IT2-FPID structure enhanced significantly the control performance even in the presence of uncertainties and disturbances when compared to the PID, T1-FPID and self-tuning T1-FPID structures. Moreover, it has been shown that the reason for the superior control performance of the IT2-FPID under high levels of uncertainty and noise is not merely for its use of extra parameters, but rather its different way of dealing with the uncertainties and noise present in real world environments by comparing with a self-tuning T1-FPID structure.

* Corresponding author. Tel.: +90 212 2856664; fax: +90 212 2852920.
E-mail address: kumbasart@itu.edu.tr (T. Kumbasar).

http://dx.doi.org/10.1016/j.ins.2014.06.005
0020-0255/© 2014 Elsevier Inc. All rights reserved.

1. Introduction

In the recent years, Interval Type-2 Fuzzy Logic Controllers (IT2-FLCs) have been widely used to control nonlinear and uncertain systems where the IT2-FLCs demonstrated significant performance improvements [8,10,23,27,40]. The internal structure of the IT2-FLC is similar to its type-1 counterpart. However, the major differences are that IT2-FLCs employ Interval Type-2 Fuzzy Sets (IT2-FSs) (rather than Type-1 Fuzzy Sets (T1-FSs)) and the IT2-FLCs process IT2-FSs, thus the IT2-FLC has the extra type-reduction process [16,24]. It has been shown in various works that IT2-FLCs achieve better performance than their type-1 counterparts because of the additional degree of freedom provided by the Footprint of Uncertainty (FOU) in their IT2-FSs [2,14]. In [21,39], it has been reported that the IT2-FLCs are generally more robust than their type-1 and conventional counterparts. Moreover, it has been shown in [41] that an IT2-FLC can implement a complex control surface that cannot be achieved by a Type-1 Fuzzy Logic Controller (T1-FLC) using the same rulebase. Nevertheless, the systematic design of IT2-FLCs is still a challenging problem due to the main difficulty in determining the parameters of the IT2-FSs. Recently; several studies have employed evolutionary algorithms for the design of IT2-FLCs [3,4,11,26,29,33]. However, due to the big number of parameters to be optimized, there is a need for optimization methods with relatively low computational cost and high convergence speed.

In this paper, we present the novel application of the evolutionary algorithm called Big Bang–Big Crunch (BB–BC) to optimize the antecedent Membership Functions (MFs) of Interval Type-2 Fuzzy PID (IT2-FPID) controllers in cascade control architecture. Since the number of the design parameters of the IT2-FPID is relatively big, the BB–BC optimization algorithm has been preferred as it has relatively a low computational cost and a high convergence speed. The control performance of the BB–BC based Optimized IT2-FPID (OIT2-FPID) controller is compared with a BB–BC Optimized Type-1 Fuzzy PID (OT1-FPID) and PID (OPID) controllers. In order to show the effect of the FOU on the system response clearly, the rule base is kept fixed for the fuzzy controller structures while only the parameters of the antecedent MFs of the IT2-FPID controllers are optimized for a reference trajectory. It will be illustrated that IT2-FLCs give the BB–BC an opportunity to find a more optimal solutions (and hence a better controller) than those obtained for the optimized PID and Type-1 Fuzzy PID (T1-FPID) controllers with respect to the Integral of the Absolute Error (IAE) performance index. The optimization results of the BB–BC method based cascade control system design approach are also compared with another high convergence evolutionary based algorithm which is the Particle Swarm Optimization (PSO). We will show that the BB–BC gave a better optimization result while converging to the optimal solution set with a fast convergence speed and lower computational time than PSO method.

We will also compare the proposed OIT2-FPID control structure with a self-tuning Error based Weight Adjustment based Type-1 Fuzzy PID (EBWT1-FPID) cascade control structure. In the EBWT1-FPID control structure, an additional degree of freedom is provided by defining the weights of the fuzzy rules as tuning parameters [15]. This will allow the type-1 fuzzy control structure to have equal degrees of freedom to the IT2-FPID structure and will enable us to establish if the power of the IT2-FPID structure lies in its ability to handle the high level of uncertainties rather than only having an extra degree of freedom provided by the FOUs in its IT2-FSs.

We presented also the novel application of the proposed conventional and fuzzy cascade control architecture to solve the path tracking control problem of mobile robots. In literature, various control structures were evaluated on the path tracking problem which inherit large amounts of nonlinearities and uncertainties caused by the internal dynamics and/or feedback sensors of the controlled system [9,27]. The path tracking control problem of mobile robots has attracted significant interest since it includes nonholonomic properties caused by nonintegrable differential constrains [9,36]. Path tracking is concerned with the ability to drive autonomously as close as possible to a desired reference trajectory. Several techniques have been proposed such as predictive control strategies [9,19,31], PID control strategies [30], adaptive control strategies [7], sliding mode control strategies [5] and fuzzy control strategies [1,12,27,28,35].

The paper presents several experiments to validate the proposed cascade control approaches, where we present first simulation studies comparing the OIT2-FPID structure to the OT1-FPID, the EBWT1-FPID and the OPID cascade structures. We will then present, real world experiments using the PIONEER 3-DX mobile robot which will act as a platform to evaluate the proposed systems in this paper. We have shown that the real-time tracking control performance of the OIT2-FPID cascade control structure is better in the presence of noise and unmodelled dynamics; and is more robust against disturbances when compared to the OT1-FPID, EBWT1-FPID and traditional PID counterparts with respect to defined performance measures. We have also addressed one of the open questions is that the reason for the superior control performance of IT2-FPID under high levels of uncertainty and noise is not merely for its use of extra parameters, but rather its different way of dealing with the uncertainties and noise present in real world environments.

Section 2 will briefly present the BB–BC optimization algorithm. Section 3 will present the control system design strategies for the OPID, OT1-FPID, EBWT1-FPID and OIT2-FPID cascade control structures. Section 4 will present the experiments and results while Section 5 will present the conclusions and future work.

2. A brief overview on BB–BC optimization

In [6], a new evolutionary algorithm named Big Bang Big Crunch was proposed. The working principle of this global search method can be explained as the transformation of a convergent solution to a chaotic state and then back to a single tentative solution point. The leading advantage of BB–BC is the high convergence speed and the low computation time. Erol and Eksin [6], made a comparison study where the performance of BB–BC was compared with Compact Genetic algorithm (C-GA) on benchmark problems. It has been demonstrated that the BB–BC outperforms the C-GA in the sense of computational time and convergence speed. Thus, the BB–BC algorithm has been implemented in various engineering applications where the computational time and convergence time are important factors. In [37], a study has been presented where BB–BC algorithm is implemented for a multi-modal optimization problem with high dimension and its performance was compared with the Particle Swarm Optimization (PSO) and Genetic Algorithms (GAs). The presented results of this study show that the BB–BC give better results than the PSO and GA. The efficiency of this evolutionary algorithm has been also demonstrated when the optimization problem must be solved in relatively small sampling times such as in inverse fuzzy model control and fuzzy model adaptation [22,23]. Moreover, in the optimization of highly nonlinear engineering problems such as in controller design [13], rescheduling power systems [20] and size reduction of space trusses [17], the BB–BC algorithm has been preferred because of its high convergence speed.

The BB–BC optimization method proposed in [6] consists of two main steps, namely the “Big Bang” and “Big Crunch” phases. The first step is the “Big Bang” phase where candidate solutions are randomly distributed over the search space. The initial “Big Bang” population is randomly generated over the entire search space just like the other evolutionary search algorithms. The next step is the “Big Crunch” phase where a contraction procedure calculates the center of mass for the population. All subsequent “Big Bang” phases are randomly distributed about the center of mass or the best fit individual in a similar fashion [6]. After the “Big Bang” phase, a contraction procedure is applied such as the “Big Crunch” phase to form a center or a representative point for further “Big Bang” operations. In this phase, the contraction operator takes the current positions of each candidate solution in the population and its associated cost function value and computes a center of mass. The center of mass can be computed as:

$$x_c = \frac{\sum_{k=1}^{T} \frac{1}{f^k} x^k}{\sum_{k=1}^{T} \frac{1}{f^k}} \tag{1}$$

where x_c is the position of the center of mass, x^k is the position of the candidate within a p-dimensional search space, f^k is the cost function value of the kth candidate, p is the number of parameters to be optimized and T is the population size. Instead of the center of mass, the best fit individual can also be chosen as the starting point in the “Big Bang” phase. The new generation for the next iteration “Big Bang” phase is normally distributed around x_c. The new candidates around the center of mass are calculated by adding or subtracting a normal random number whose value decreases as the iterations elapse. This can be formalized as:

$$x^{new} = x_c + (r\omega(x_{max} - x_{min}))/l \tag{2}$$

where r is random number generated in the interval $[-1,1]$; ω is a parameter limiting the size of the search space, x_{max} and x_{min} are the upper and lower limits defined within a p-dimensional search space; and l is the current iteration step. The pseudo-code of BB–BC optimization method is given in the Table 1. Note that, in this study, we will employ the maximum number of iterations as the stopping criteria. More detailed information about the BB–BC optimization method can be found in [6].

3. Cascade control system design strategies

Cascade control is one of the widely used control strategies in system control because of its ability to enhance single-loop control performances [25]. Cascade control structures are usually implemented when the first control unit contains dominant nonlinearities that limit the overall control performance or in the presence of disturbances [18]. In this context, fuzzy cascade control structures have been designed and implemented in various engineering problems since fuzzy sets are powerful tools in presence of nonlinearities and uncertainties [32,38].

The cascade control system architecture is illustrated in Fig. 1a. Here, the controlled system is represented with two cascaded subsystems, i.e. System-1 and System-2. In this cascade control structure, there are two controllers of which one controller’s output drives the reference value of another controller. The controller generating the reference value ($u_o(t)$) is called the outer loop controller while the controller receiving the reference value is called the inner loop controller which generates the actual control signal of the control system ($u_i(t)$). Thus, an inner measurement point (System-1 output: ($y_i(t)$) and an inner loop controller in cascade to the outer controller can be used to improve the response of the system (System-2 output:

Table 1
The pseudo-code of the BB–BC optimization algorithm.

Step 1	An initial generation of N candidates is generated randomly in the search (Initial Big Bang phase)
Step 2	The cost function values of all the candidate solutions are computed
Step 3	Big Crunch Phase: The center of mass is calculated using Eq. (1). Either the best fit individual or the center of mass is chosen as the starting point for the Big Bang Phase
Step 4	Big Bang phase: New candidates are calculated around the new point calculated in Step 3 as described in Eq. (2)
Step 5	Return to Step 2 until stopping criteria has been met, i.e., maximum number of iterations

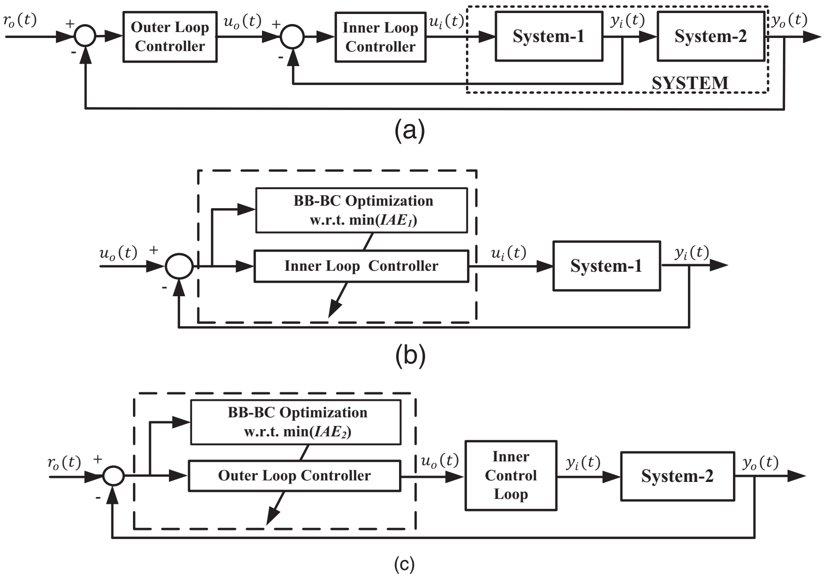

Fig. 1. Illustration of the (a) cascade control structure, (b) optimization procedure of the inner loop (c) optimization procedure of the outer loop.

$y_o(t)$) to disturbances as well as to improve the dynamic properties of the control loop [18]. The design method for cascade control structures is usually accomplished by tuning the parameters of the controllers in the loops individually. At first, the inner loop control is designed with respect to a defined performance criterion. Once the parameters of the inner loop controller are tuned, the parameters of the outer loop controller are designed.

In this study, we have implemented the OPID, OT1-FPID, EBWT1-FPID and OIT2-FPID as the inner and outer loop controllers of the cascade control structure, respectively. In this context, we present the design strategies employed for the PID, T1-FPID and IT2-FPID controller structures within this section. We start by presenting a BB–BC optimization based design strategy employed for the conventional PID cascade controller structure. Then, we present the BB–BC optimization and the online rule weighting adjustment design strategies employed for T1-FPID cascade structures. Furthermore, we present the BB–BC optimization based design strategy employed for the design of the IT2-FPID cascade control structure.

3.1. The design strategy of the PID cascade control system

In this subsection, we start by presenting the implemented PID controller and then we present the BB–BC based design strategy employed for the OPID cascade control structure.

3.1.1. The internal structure of the PID controller

In this paper, the control output of the PID structure in either the inner ($u_i(t)$) and outer loop ($u_o(t)$) controller is defined as follows:

$$u(t) = K_c\left(e(t) + \frac{1}{\tau_I}\int e(t) + \tau_D\frac{de(t)}{dt}\right) \tag{3}$$

where $e(t)$ is the feedback error, K_c is the proportional gain, τ_I is the integral gain and τ_D is the derivative gain of the controller structures.

3.1.2. The BB–BC optimization of the PID cascade control structure

In this study, the parameters of the conventional PID controllers employed within the inner and outer loop controllers are optimized to minimize the IAE performance index via the BB–BC optimization. Thus, the optimization variables for the BB–BC are defined as $x_{PID} = (K_c, \tau_I, \tau_D)$. The optimization procedure of the PID cascade control structure is accomplished in two stages.

In the first stage, the inner loop controller parameters are optimized via the BB–BC optimization algorithm for a defined reference trajectory. The defined inner loop performance measure is defined as:

$$IAE_1 = \int |r_i(t) - y_i(t)|dt \tag{4}$$

where $r_i(t)$ is the desired inner loop reference (generated by the outer controller, i.e. $u_o(t)$) and $y_i(t)$ is the inner loop feedback value. The schematic diagram of the inner loop optimization procedure is given in Fig. 1b to obtain the parameters of the inner loop controller (x_{PID_i})

In the second stage, the outer loop PID controller parameters are optimized via the BB–BC optimization algorithm to minimize outer performance measure which is defined as:

$$IAE_2 = \int |r_o(t) - y_o(t)|dt \tag{5}$$

where $r_o(t)$ is the desired outer loop reference and $y_o(t)$ is the outer loop feedback value. The outer PID controller parameters (x_{PID_o}) are optimized such that to minimize IAE_2 via the BB–BC algorithm while parameters of the inner loop controller (x_{PID_i}) are kept fixed for outer reference trajectory. The schematic diagram of the outer PID control optimization procedure is illustrated in Fig. 1c. The pseudo-code of the proposed BB–BC optimization based PID control system design strategy is given in Table 2 where the controller will be PID and the corresponding parameter set will be x_{PID_i} and x_{PID_o} for the inner and outer loop controllers, respectively.

3.2. The design strategy of the T1-PID cascade control system

In this subsection, we start by presenting the internal structure of the implemented T1-FPID controller and then we present the design strategies for the T1-FPID controller structure which are based on the BB–BC optimization and an online rule weighting adjustment.

3.2.1. The internal structure of the T1-FPID controller

In this subsection, we present the implemented standard two input T1-FPID controllers as illustrated in Fig. 2a [34]. The input to the T1-FPID controllers are the error signal ($e(t) = r(t) - y(t)$) and the change of error ($\Delta e(t)$), the output is the control signal ($u(t)$). The input scaling factors K_e for error and K_d for the change of error normalize the inputs to the range in which the membership functions of the inputs are defined (thus $e(t)$ and $\Delta e(t)$) are converted after normalization into E and ΔE). While K_0 and K_1 are the output scaling factors (thus the output of the T1-FLC U is converted into u via K_0 and K_1).

In the handled T1-FPID structures, a symmetrical 3 × 3 rule base is used and given in Table 3. The rule structure of the T1-FPID is as follows:

$$R_n : \text{If } E \text{ is } A_{1j} \text{ and } \Delta E \text{ is } A_{2j} \text{ Then } U \text{ is } G_n \text{ with } w_n \tag{6}$$

Here, the inputs of all T1-FLC structures are defined with three ($J = 3$) triangular type MFs and are denoted as N (Negative), Z (Zero) and P(Positive). The T1-FSs of the T1-FLC are defined with three parameters (l_{ij}, c_{ij}, r_{ij}; $i = 1, 2, j = 1, 2, 3$), as shown in Fig. 2b. The outputs of the T1-FLCs are defined with five crisp singleton consequents (Negative (N), Negative Medium (NM) Zero (Z), Positive Medium (PM), Positive (P)) as illustrated in Fig. 2c (the outputs will not be optimized in the fuzzy controllers presented in this paper and will be kept the same for the T1-FLC and IT2-FLC structures). Moreover, each rule is assigned with a weight ($w_n \in [0,1]$, $n = 1, \ldots, 9$) which shows the importance of that fuzzy rule. The implemented T1-FLCs use the product implication and the center of sets defuzzification method.

3.2.2. The BB–BC optimization of the T1-FPID cascade control structure

In the first T1-FPID design strategy, the parameters of the antecedent MFs and scaling factors of the T1-FPID controllers for the inner and outer loop controllers are optimized via the BB–BC method to minimize the IAE performance index. The antecedent MFs of the T1-FPID controller that are labeled as the "N" and "P" are defined for each input with two parameters each which are c_{i1}, r_{i1} (for N) and l_{i3}, c_{i3} (for P) ($i = 1, 2$), respectively while the linguistic label "Z" is defined with three parameters which are l_{i2}, c_{i2}, r_{i2}, ($i = 1, 2$). Hence for two inputs, the total number of the antecedent MF parameters to be optimized for the OT1-FPID design is then 2 × 7 = 14. Moreover, the input and output scaling factors (K_e,K_d, K_0 and K_1) of the T1-FPID controller are also optimized. Thus, there are 14 structural and 4 scaling parameters (14 + 4 = 18) to be optimized for the

Table 2
The pseudo-code of BB–BC based cascade design method.

```
Step 1   Define the parameters of the BB–BC optimization, i.e. the population size and the number of iterations
Step 2   Define the controller type (PID/T1-FPID/ IT2-FPID) and the corresponding parameter set
            (a) $x_{PID_i}/x_{T1-FPID_i}/x_{IT2-FPID_i}$ for the inner loop controller
            (b) $x_{PID_o}/x_{T1-FPID_o}/x_{IT2-FPID_o}$ for the outer loop controller
Step 3   Choose the performance measure to be minimized
            (a) the $IAE_1$ performance measure for the inner loop controller
            (b) the $IAE_2$ performance measure for the outer loop controller
Step 4   Generate an initial population randomly for the parameter set of the controller (Initial Big Bang Phase)
Step 5   Simulate the closed loop control system for each population member. Calculate the corresponding cost values for the generated population
         members
Step 6   Calculate the center of mass via Eq. (1). Choose the best fit individual or the center of mass as the point of Big Bang Phase (Big Crunch Phase)
Step 7   Calculate new population members around the new point calculated in Step 6 via Eq. (2) (Big Bang Phase)
Step 8   Return to Step 5 until stopping criteria has been met, i.e., maximum number of iterations
```

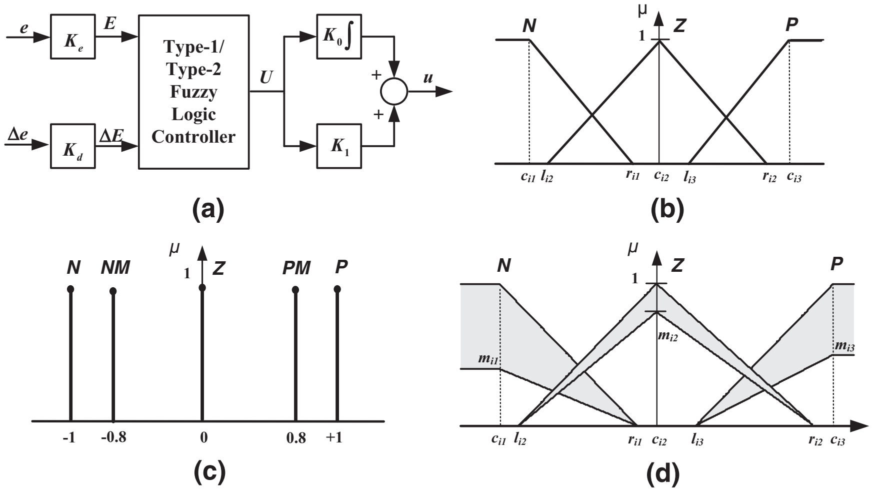

Fig. 2. Illustration of the (a) T1-FPID/IT2-FPID controller structure, (b) T1-FSs, (c) Consequent MFs and (d) IT2-FSs.

Table 3
The rule base of the type-1 and interval type-2 fuzzy controllers with rule weights.

$E/\Delta E$	N	Z	P
N	$N(w_1)$	$NM(w_2)$	$Z(w_3)$
Z	$NM(w_4)$	$Z(w_5)$	$PM(w_6)$
P	$Z(w_7)$	$PM(w_8)$	$P(w_9)$

OT1-FPID controller via the BB–BC method in order to minimize the *IAE* performance index. In this context, the optimization variables for the BB–BC algorithm are defined as $x_{T1-FPID} = (l_{11}, c_{11}, l_{12}, c_{12}, r_{12}, c_{13}, r_{13}, l_{21}, c_{21}, l_{22}, c_{22}, r_{22}, c_{23}, r_{23}, K_e, K_d, K_0, K_1)$. *It should be noted that the weights of the rules* (w_n) *are not optimized and set as "1".*

The optimization procedure of OT1-FPID structure is accomplished similar to the OPID cascade control structure. However, in order to obtain convex T1-FSs, the parameters of the antecedent MFs of the inner loop and outer loop are optimized under the following constraints

$$\begin{aligned} &c_{i1} < r_{i1} \\ &l_{i2} < c_{i2} < r_{i2} \\ &c_{i3} < r_{i3} \\ &c_{i1} < c_{i2} < c_{i3} \end{aligned} \tag{7}$$

In the design of the T1-FPID cascade structure, at first, the parameters of the inner loop OT1-FPID $(x_{T1-FPID_i})$ (18 parameters) are optimized via the BB–BC method such that to minimize the performance index IAE_1 given in Eq. (4) for a reference trajectory as illustrated in Fig. 1b. Then, the outer OT1-FPID parameters $(x_{T1-FPID_o})$ (18 parameters) are optimized to minimize Eq. (5) via the BB–BC algorithm while the parameters of the inner OT1-FPID $(x_{T1-FPID_i})$ are kept fixed for a reference as shown in Fig. 1c. The pseudo-code of the BB–BC method based T1-FPID cascade control design strategy is presented in Table 2 where the controller will be T1-FPID and the corresponding parameter set will be $x_{T1-FPID_i}$ and $x_{T1-FPID_o}$ for the inner and outer loop controller, respectively.

3.2.3. The design of the EBWT1-PID cascade structure

In the second T1-PID cascade control design strategy, an online Error based rule Weighting Adjustment (EBW) based tuning mechanism is implemented for both the inner and outer T1-FPID controllers as illustrated in Fig. 1. In this self-tuning structure, the weight (w_n) of each rule is used as tuning parameter and is adjusted online with the respect to the error signal [15]. In fact, an additional degree of freedom is obtained by defining the fuzzy rule weights as tuning parameters.

In this strategy, the step response of the closed loop system is divided into four main regions as illustrated in Fig. 3. The meta-rules derived to achieve the satisfactory control system performance are discussed in [15]. Since the importance of a fuzzy rule is related to its weight value, the change of the importance of a fuzzy rule at any region can be expressed as changing the weight value of that rule. In this context, the value of the system error can be used for tuning the fuzzy rule weights. Since the interval for rule weight values should lie within the range [0,1], i.e., $w_n \in [0,1]$, the interval of the normalized system error $E \in [-1,1]$ is mapped to the interval [0,1] by using the absolute value function. Then, the error function could be expressed as follows:

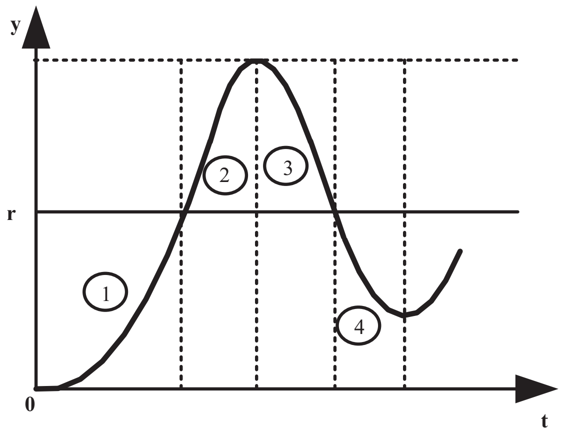

Fig. 3. The partitioning of the step response into regions.

$$\sigma_1(E) = a.Abs(E); \quad \sigma_2(E) = 1 - Abs(E) \tag{8}$$

$\sigma_1(E)$, $\sigma_2(E)$ can be directly assigned as the rule weight values. Consequently, the error value becomes the tuning variable of the rule weights. The online weight tuning strategy for each region can be accomplished as follows [15]:

- At region 1, the absolute value of error decreases from 1 to 0. When the value of error decreases from 1 to 0, the importance of the fuzzy rules R_8 and R_7 decreases while the importance of fuzzy rules R_5 and R_4 increases in order to obtain the required fastness of the system response. Thus, the weights can be tuned as $w_4 = \sigma_2(E)$, $w_5 = \sigma_2(E)$, $w_7 = \sigma_1(E)$, $w_8 = \sigma_1(E)$. Here, the value of "a" is taken to be "1" for all the regions except the first region. Since the error value possesses its extreme value at the region 1, the value of "φ" can take values like 0.4 or 0.5 so as to prevent the possible overshoots in the system response.
- At region 2, the absolute value of error increases from 0 to maximum overshoot. In order to prevent overshoot, the importance of the fuzzy rules R_1, R_2, R_4 and R_5 should be kept at their maximum. So the weights can be tuned as; $w_1 = \sigma_2(E)$, $w_2 = \sigma_2(E)$, $w_4 = \sigma_2(E)$, $w_5 = \sigma_2(E)$.
- At region 3, the absolute value of error decreases from maximum overshoot to zero. So the importance of the fuzzy rules R_2 and R_3 decreases while the importance of fuzzy rules R_5 and R_6 increases in order to obtain the required fastness of the system response sufficiently. So the weights can be tuned as; $w_2 = \sigma_1(E)$, $w_3 = \sigma_1(E)w_5 = \sigma_2(E)$, $w_6 = \sigma_2(E)$.
- Finally at region 4, the absolute value of error increases from 0 to 1. In order to prevent undershoot, the importance of the fuzzy rules R_5, R_6, R_8 and R_9 should be kept maximum. So the weights can be tuned as; $w_5 = \sigma_2(E)$, $w_6 = \sigma_2(E)$, $w_8 = \sigma_2(E)$, $w_9 = \sigma_2(E)$.

A comprehensive analysis on the fuzzy rule weights can be found in [15]. Note that, we will employ the same antecedent MFs, rule base and scaling factors presented for the OT1-FPID structure in this study.

3.3. The design strategy of the IT2-FPID cascade structure

In this subsection, we will firstly present the internal structure of the implemented IT2-FPID controller and then the proposed BB–BC based design strategy for the IT2-FPID cascade control structure.

3.3.1. Internal structure of the IT2-FPID controller

Similar to the T1-FPID structure, a two input IT2-FPID controller structure with two input and two output scaling factors (i.e. (K_e, K_d) and (K_0, K_1)) is used as shown in Fig. 2a. In the handled IT2-FPID structures, the same 3 × 3 rule base presented for the T1-FPID controller is employed (given in Table 3). The rule structure of the IT2-FPID controller is as follows:

$$R_n : \text{If } E \text{ is } \widetilde{A}_{1j} \text{ and } \Delta E \text{ is } \widetilde{A}_{2j} \text{ Then } U \text{ is } G_n \text{ with } w_n \tag{9}$$

The input MFs of the IT2-FLC structure are represented by three linguistic labels which are N (Negative), Z (Zero) and P (Positive) as illustrated in Fig. 2d. The outputs MFs of the IT2-FLC are defined with five crisp singleton consequents (Negative (N), Negative Medium (NM) Zero (Z), Positive Medium (PM), Positive (P)) (as shown in Fig. 2c as was the case of the T1-FLCs to allow for a fair comparison). The antecedent IT2-FSs are defined with four parameters (l_{ij}, c_{ij}, r_{ij}, m_{ij}; $i = 1, 2, j = 1, 2, 3$), as shown in Fig. 2d. The input IT2-FS ($\widetilde{A}$) can be described in terms of an upper MF ($\bar{\mu}_{\widetilde{A}}$) and a lower MF ($\underline{\mu}_{\widetilde{A}}$). Thus, the total firing strength for the n^{th} rule is

$$\tilde{f}_n = \left[\bar{f}_n \quad \underline{f}_n\right] \tag{10}$$

where $\tilde{f}_n$ is the total firing interval and is defined as

$$\underline{f}_n = \underline{\mu}_{\tilde{A}_{1j}} * \underline{\mu}_{\tilde{A}_{2j}} \tag{11}$$

$$\bar{f}_n = \bar{\mu}_{\tilde{A}_{1j}} * \bar{\mu}_{\tilde{A}_{2j}} \tag{12}$$

Here, "*" represents the product implication (the *t*-norm).

The implemented the IT2-FLC uses the center of sets type reduction method [16]. It has been demonstrated that the defuzzified output of an IT2-FLC can be calculated as:

$$u = \frac{u_l + u_r}{2} \tag{13}$$

where u_l and u_r are the end points of the type reduced set. To be able to calculate the two end points of the type reduced set, first $G_n w_n$ is re-ordered such that $G_1 w_1 \leqslant G_2 w_2 \leqslant \cdots G_N w_N$ and the corresponding firing interval sets are matched. Then, u_l and u_r can be computed as follows:

$$u_l = \frac{\sum_{n=1}^{L} \bar{f}_n G_n w_n + \sum_{L+1}^{N} \underline{f}_n G_n w_n}{\sum_{n=1}^{L} \bar{f}_n w_n + \sum_{L+1}^{N} \underline{f}_n w_n} \tag{14}$$

$$u_r = \frac{\sum_{n=1}^{R} \underline{f}_n G_n w_n + \sum_{R+1}^{N} \bar{f}_n G_n w_n}{\sum_{n=1}^{R} \underline{f}_n w_n + \sum_{R+1}^{N} \bar{f}_n w_n} \tag{15}$$

Here (L, R) are the switching points of the typed-reduced sets and are calculated via the Karnik–Mendel type-reduction method [41].

3.3.2. The BB–BC optimization of the IT2-FPID cascade structure

In the design strategy of the IT2-FPID cascade control systems, the parameters of the antecedent MFs of the IT2-FPIDs for the inner and outer controllers are optimized via the BB–BC algorithm in order to minimize the *IAE* performance index. The antecedent IT2-FSs of OIT2-FPID that are labeled as the "*P*" and "*N*" for each input are defined with three parameters each which are (c_{i1}, r_{i1}, m_{i1}) and $(l_{i3}, c_{i3}, m_{i3}, i = 1, 2)$, respectively while the IT2-FS labeled as "*Z*" is defined with four parameters $(l_{i2}, c_{i2}, r_{i2}, m_{i2}, i = 1, 2)$. Consequently, for the two inputs the total numbers to be optimized by the BB–BC for the OIT2-FPID design is $2 \times 10 = 20$. It is obvious that, the IT2-FPID has 6 more structural parameters, i.e. extra degrees of freedom, than the T1-FPID structure (14 parameters). The structural parameters of the OIT2-FPID are optimized via the BB–BC method such that to minimize the *IAE* performance index. Thus, the optimization parameter set for the BB–BC algorithm is defined as $x_{IT2-FPID} = (l_{11}, c_{11}, m_{11}, l_{12}, c_{12}, m_{12}, r_{12}, c_{13}, r_{13}, m_{13}, l_{21}, c_{21}, m_{21}, l_{22}, c_{22}, m_{22}, r_{12}, c_{23}, r_{23}, m_{23})$.

In the design of the IT2-FPID controllers, the BB–BC will only tune the parameters set $x_{IT2-FPID}$, while the scaling factors (K_e, K_d, K_0, K_1) and the rule base are kept fixed in order to show the effect the extra degrees of freedom provided by the antecedent IT2-FSs on the closed loop system performance. Thus, the same scaling factors and rule base of the OT1-FPID controller are employed. *It should be noted that the weights of the rules* (w_n) *are not optimized and set as "1".*

The optimization procedure of OIT2-FPID cascade structure is accomplished similar to the conventional and type-1 fuzzy cascade control structures. Similar to the OT1-FPID cascade design strategy, in order to obtain convex MFs, the parameters of the antecedent MFs of the inner loop and outer loop are optimized under the constraints presented in Eq. (7). Furthermore, in order to obtain IT2-FSs, the heights of the lower MFs are restricted as follows:

$$0 < m_{ij} < 1 \tag{16}$$

At first, the inner loop OIT2-FPID controller parameters $(x_{IT2-FPID_i})$ (20 parameters) are optimized via the BB–BC method such that to minimize the performance index IAE_1 given in Eq. (4) for a defined inner loop reference trajectory as shown in Fig. 1a. Then, the outer loop OIT2-FPID parameters $(x_{IT2-FPID_o})$ (20 parameters) are optimized to minimize Eq. (5) via the BB–BC optimization algorithm while parameters of the inner loop IT2-FPID $(x_{IT2-FPID_i})$ are kept fixed for a defined outer loop reference trajectory as illustrated in Fig. 1c.

The pseudo-code of the proposed BB–BC optimization based IT2-FPID cascade control system design strategy is presented in Table 2 where the controller will be IT2-FPID and the corresponding parameter set will be $x_{IT2-FPID_i}$ and $x_{IT2-FPID_o}$ for the inner and outer loop controllers, respectively.

4. Experiments and results

In this section, we have compared the performance of the OPID, OT1-FPID, EBWT1-FPID and OIT2-FPID controllers in a novel cascade control structure to solve the path tracking control problem of a mobile robot. Thus, we start by presenting kinematic motion equations of the mobile robot which is used as a platform to compare the various controllers. We then present the novel application of the presented cascade control structure for the tracking control problem of mobile robots.

Moreover, the *IAE* performance evaluations of the optimized cascade control structures are examined for each control loop. Finally, the control performance (evaluated both in simulations and real world robots) of the presented OPID, OT1-FPID, EBWT1-FPID and OIT2-FPID structures are examined. In order to show the effect the extra degrees of freedom provided by the antecedent IT2-FSs on the closed loop system performance, the rule base and scaling factors were kept fixed while only the antecedent parameters of the OIT2-FPID controller were optimized for a defined reference trajectory.

The presented simulations were done on a personal computer with an Intel Core I5 1.61 GHz processor, 4 GB RAM and using MATLAB/Simulink 7.4.0. Throughout the simulation studies, the simulation solver option is chosen as ode5 (dormand-prince) and the step size is fixed at a value of 0.1s. In the simulation studies, the presented cascade control structures are implemented as discrete-time versions with the sampling time $T_s = 0.1$ s.

4.1. The kinematic motion equations of the employed mobile robot

The schematic diagram of the differential driving Pioneer 3-DX mobile robot's architecture is illustrated in Fig. 4a. The kinematic motion equations of the mobile robot have a nonintegrable constraint with the assumption that the robot mobile robot as $s(q)$ that span the null space of the matrix $A(q)$.The kinematics model then can be written as:

$$\dot{q}(t) = s(q)\begin{bmatrix} v(t) \\ w(t) \end{bmatrix} = \begin{bmatrix} \cos\theta(t) & 0 \\ \sin\theta(t) & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} v(t) \\ w(t) \end{bmatrix} \tag{17}$$

where $v(t)$ and $w(t)$ are the linear and angular velocities, $\theta(t)$ is the heading angle of the mobile robot. Moreover, the bounded velocity and acceleration constraints are considered [19].

To control the system given in Eq. (17), the reference trajectory should also be described by a reference state vector $q_r(t) = [x_r(t)y_r(t)\theta_r(t)]^T$ and control signal vector $u_r(t) = [v_r(t)w_r(t)]^T$.

$$\dot{q}_r(t) = \begin{bmatrix} \cos\theta_r(t) & 0 \\ \sin\theta_r(t) & 0 \\ 0 & 1 \end{bmatrix} u_r(t) \tag{18}$$

Then, to track the reference trajectory an error state can be defined as follows:

$$\dot{q}_e(t) = \begin{bmatrix} e_x(t) \\ e_y(t) \\ e_\theta(t) \end{bmatrix} = \begin{bmatrix} \cos\theta(t) & \sin\theta(t) & 0 \\ -\sin\theta(t) & \cos\theta(t) & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x_r(t) - x(t) \\ y_r(t) - y(t) \\ \theta_r(t) - \theta(t) \end{bmatrix} \tag{19}$$

Consequently, the tracking control problem is converted into a regulation problem [9].

4.2. The proposed cascade structure for the mobile robot tracking control

In this section, we present the novel application of the cascade control structure to solve the path tracking control problem of the mobile robot. In this study, we consider the linear velocity ($v(t)$) to be constant while the angular velocity of the mobile robot as the manipulated variable ($w(t)$). Consequently, the model is a multivariable one-input three output system where the angular velocity of the mobile robot has been chosen as the control variable and the mobile robot's heading angle ($\theta(t)$) and local position ($x(t)$, $y(t)$) are the controlled variables. Although the mobile robot model should have a single input and three controlled variables, it has been reported that the performance is also satisfactory when two controlled variables ($\theta(t)$ and $y(t)$) are considered since $x(t)$ can be defined as a function of $\theta(t)$ and $y(t)$) [26].

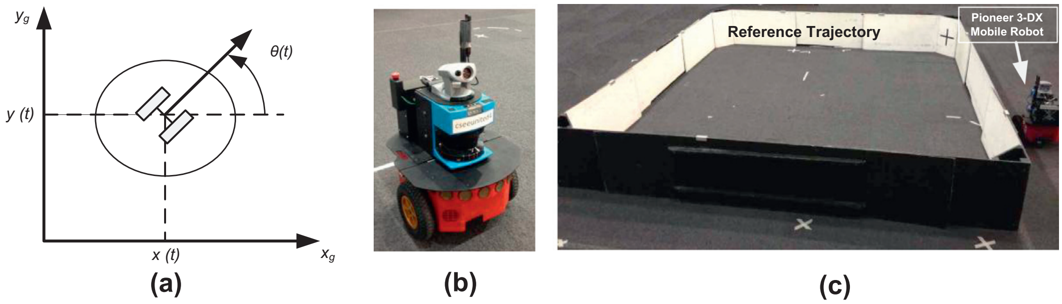

Fig. 4. Illustration of the (a) geometric description, (b) general view of the Pioneer 3-DX mobile robot, and (c) indoor environment of the Pioneer mobile robot.

The proposed overall cascade architecture is illustrated in Fig. 5a. The implemented cascade control structure consists of three main blocks which are the transformation block, the outer loop controller and the inner loop controller. In the transformation mechanism, the robot pose ($x(t)$, $y(t)$, $\theta(t)$) and the desired reference trajectory are used to compute the local error signal on y-axis ($e_y(t)$) via Eq. (22). This error signal is then fed as the input to the outer loop controller to generate the reference steering angle $\theta_{ref}(t)$. This reference value of the inner loop controller corresponds to the necessary heading angle of a mobile robot to track the desired trajectory. In other words, the aim of the inner loop controller is to force the heading angle of the mobile robot to a desired heading angle which is generated by the outer loop controller so that the mobile robot tracks the desired reference trajectory.

4.3. The IAE performance evaluation of the cascade control structures

In this subsection, the optimization results of the OPID, OT1-FPID and OIT2-FPID structures for the mobile robot are examined and compared with respect the *IAE* performance measure. Here, the linear speed of the mobile robot is fixed and set as $\nu(t) = 0.27$ m/s. Since the aim of this subsection is to make a fair comparison, the population size and the number of iterations for all three BB–BC based design strategies have been chosen as 100 and 1000, respectively. As it has been asserted in Section 3, the optimization procedure of the cascade control structures is accomplished in two stages.

In the first stage, the inner controller parameters are optimized via the BB–BC algorithm as illustrated in Fig. 4b. The defined inner loop performance measure is defined as:

$$IAE_1 = \int |\theta_{ref}(t) - \theta(t)|dt \tag{20}$$

where $\theta_{ref}(t)$ is the desired steering angle and $\theta\,(t)$ is the current steering angle. The inner PID, T1-FPID and IT2-FPID are optimized such that to minimize the IAE_1 given in Eq. (20) for the reference heading trajectory with the values of $\pi/2$, π, $3\pi/2$ and 2π radian. For the desired reference values, the best values of the OPID and the scaling factors of the T1-FPID are found as $K_c = 2.2$, $\tau_I = 1.83$, $\tau_D = 0.045$ and $K_e = 0.63$,$K_d = 0.05$, $K_0 = 3.01$, $K_I = 3.96$, respectively while the optimized antecedent MFs of the inner loop OT1-FPID and OIT2-FPID controllers are shown in Fig. 6. The obtained best IAE_1 values of the OPID, OT1-FPID and OIT2-FPID are 6.279, 6.229 and 6.180, respectively.

In the second stage, the outer controller parameters are optimized via the BB–BC algorithm while the parameters of the inner loop controller are kept constant as illustrated in Fig. 5c. The defined outer loop performance measure is defined as:

$$IAE_2 = \int |e_y(t)|dt \tag{21}$$

where $e_y(t)$ is error signal on y-axis and is calculated via

$$e_y(t) = \cos\theta(t)(y_r(t) - y(t)) - \sin\theta(t)(x_r(t) - x(t)) \tag{22}$$

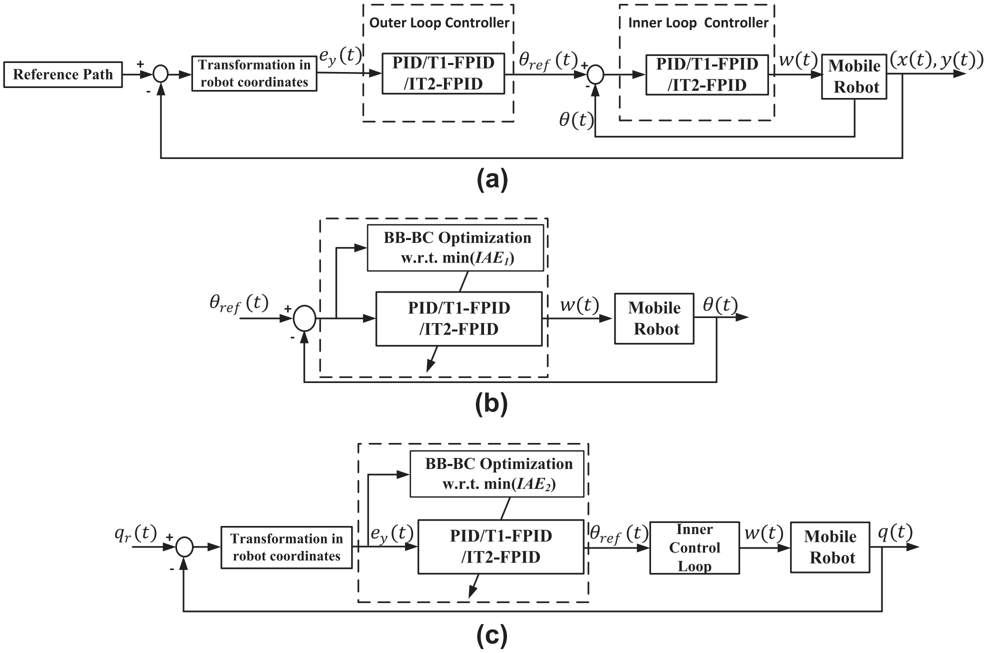

Fig. 5. Illustration of the (a) mobile robot cascade control structure, (b) inner loop controller design strategy, and (c) outer loop controller design strategy.

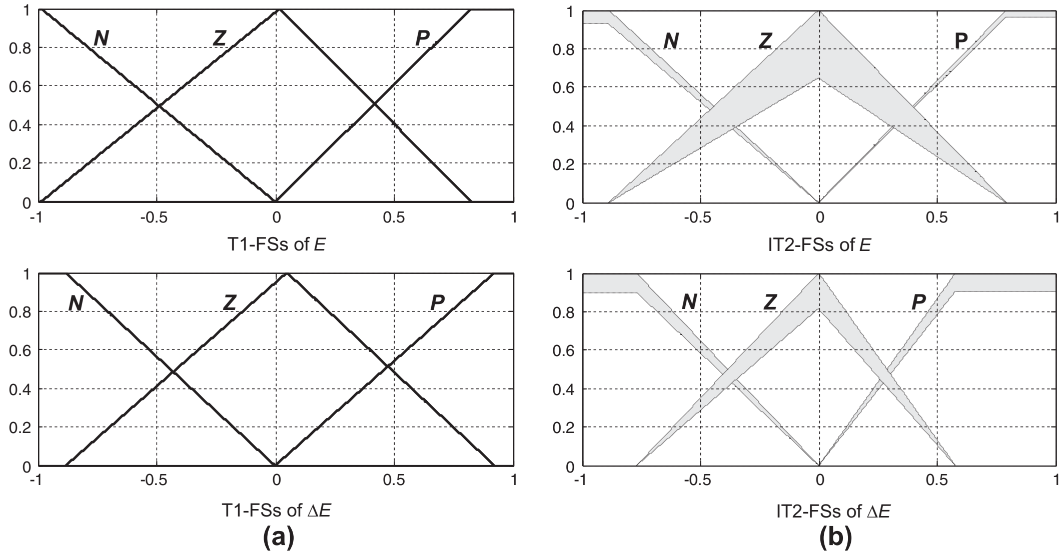

Fig. 6. Illustration of the optimized antecedent MFs of the inner (a) OT1-FPID controller and (b) OIT2-FPID controller.

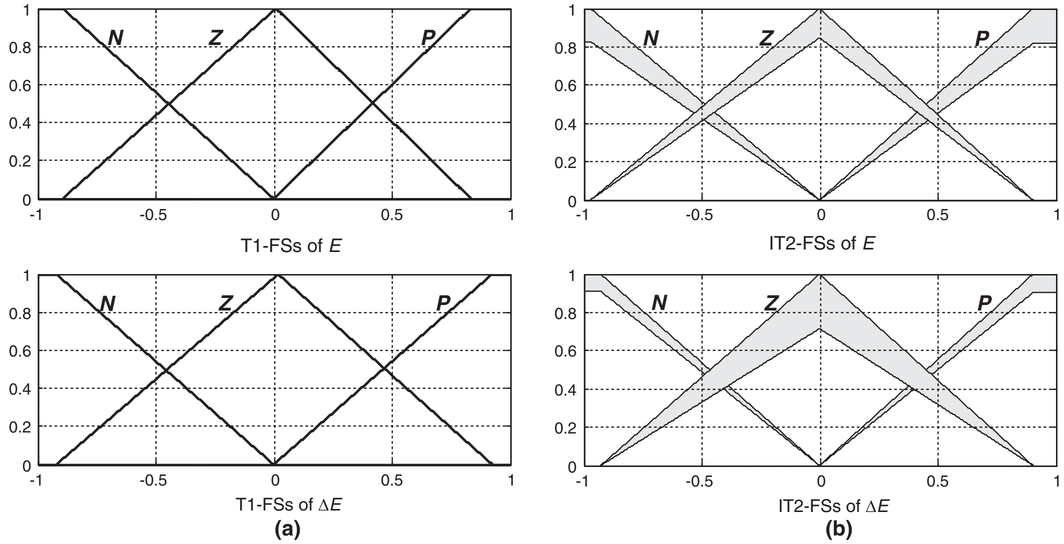

Fig. 7. Illustration of the optimized antecedent MFs of the outer (a) OT1-FPID controller (b) and OIT2-FPID controller.

The outer loop controller parameters are optimized to minimize IAE_2 via the BB–BC while parameters of the inner controller are kept fixed for a square like reference trajectory. For the trajectory, the best values of the OPID and T1-FPID are found as $K_c = 1.55$, $\tau_I = 15$, $\tau_D = 0.066$ and $K_e = 1.03$, $K_d = 0.496$, $K_0 = 0.103$, $K_I = 2.53$, respectively. The optimized antecedent MFs of the outer loop OT1-FPID and OIT2-FPID controllers are illustrated in Fig. 7. For the outer control performance, the obtained best IAE_2 values of the OPID, OT1-FPID and OIT2-FPID are 32.313, 23.417 and 17.995, respectively.

The values of the IAE_1 and IAE_2 plotted against the number of iterations for the obtained best solution sets are illustrated in Figs. 8 and 9, respectively. It can be clearly seen that the BB–BC optimization based OPID and OT1-FPID structures converge with a much higher IAE_1 and IAE_2 values than the OIT2-FPID cascade controller structure which ends up faster with better performance values. This confirms the results in [41] that the IT2-FLC can implement more sophisticated control models that cannot be achieved by OPID or OT1-FPID using the same rulebase and scaling factors.

Moreover, the optimization results of the BB–BC method are also compared with the PSO method which is another fast and efficient evolutionary based strategy. Similar to the BB–BC, the population size and the number of iterations of the PSO have been chosen as 100 and 1000, respectively. To make a fair comparison a statistical evaluation is performed since both the BB–BC and PSO method are stochastic global search methods. The presented two stage optimization based design routine for each of the cascade controller structure is repeated 20 times for both the BB–BC and PSO search methods.

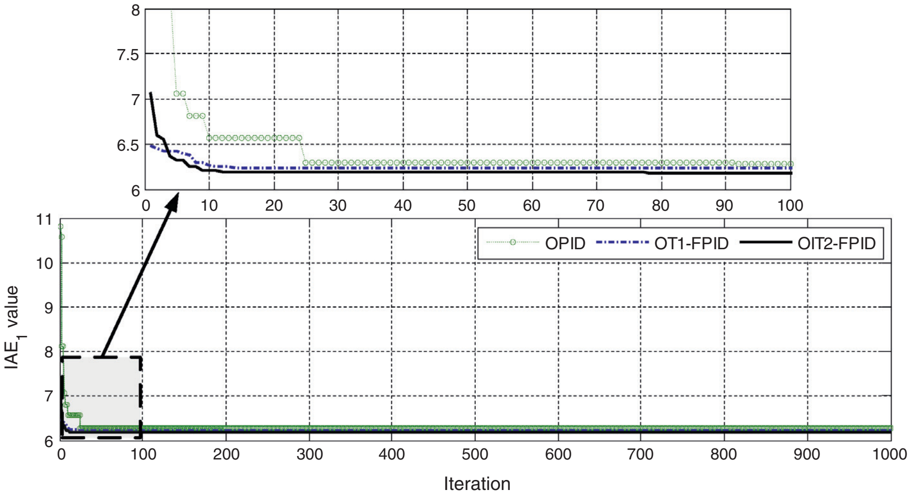

Fig. 8. The BB–BC performance evolution of the IAE_1 values for the obtained best solution set.

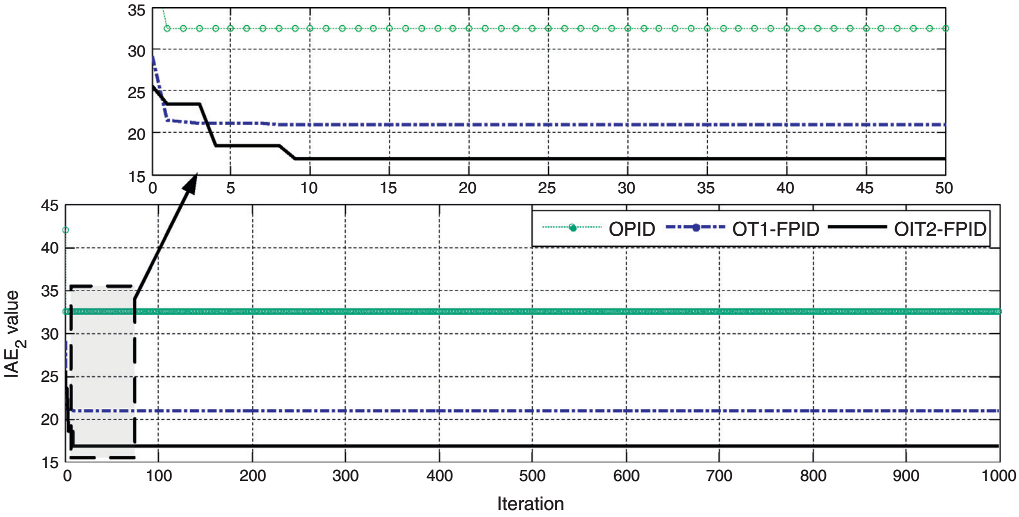

Fig. 9. The BB–BC performance evolution of the IAE_2 values for the obtained best solution set.

In Table 4, the best, the worst and the average performance values and the average computation time for the twenty runs are tabulated. It can be concluded that although IT2-FPID structure has the largest number of parameters to be optimized, it has lower best, worst and average *IAE* values when compared to the OPID and OT1-FPID counterparts (regardless the optimization method employed). Moreover, it is clear from the results that the BB–BC based design method can obtain higher quality solutions in comparison with PSO with less computational time. For instance, in comparison with the PSO method, the BB–BC method was able to find lower IAE_1 and IAE_2 values in the design of the OIT2-FPID structure while decreasing the computational time about 25% and 21% for the inner and outer loop optimization, respectively. Hence, the use of the BB–BC can help in the future to design more complex IT2-FLCs with more inputs in applications where there is a need to get a very good optimization result in relatively fast time frame.

4.4. Simulation results

In the simulation studies, a square like reference trajectory is applied to the proposed cascade structures at which the controllers were optimized. Moreover, the results of the EBWT1-FPID structure are presented for comparison. In order to make a fair comparison, three performance measures are considered. Two of these measures are selected as the IAE_1 and IAE_2 while the other performance measure is the maximum deviation from the corners where the sudden change of the reference occurs and is defined as:

Table 4
Performance comparison of BB–BC and PSO based cascade control system design procedures.

		Performance index variations					
		IAE$_1$ Performance			IAE$_2$ Performance		
		Best	Average	Worst	Best	Average	Worst
OPID	BB–BC	6.279	6.651	7.110	32.313	34.534	36.838
	PSO	6.406	6.752	7.339	34.144	36.142	37.107
OT1-FPID	BB–BC	6.229	6.448	6.713	23.417	25.263	28.183
	PSO	6.314	6.616	7.046	24.111	26.194	28.716
OIT2-FPID	BB–BC	6.180	6.402	6.654	17.995	20.882	22.476
	PSO	6.203	6.460	6.685	18.954	21.149	23.866

		Average total computational time	
		Inner loop optimization (min)	Outer loop optimization (min)
OPID	BB–BC	136.3408	163.6055
	PSO	213.4824	252.7801
OT1-FPID	BB–BC	244.3835	304.1666
	PSO	323.1563	389.5417
OIT2-FPID	BB–BC	307.3126	355.9521
	PSO	412.0418	453.6265

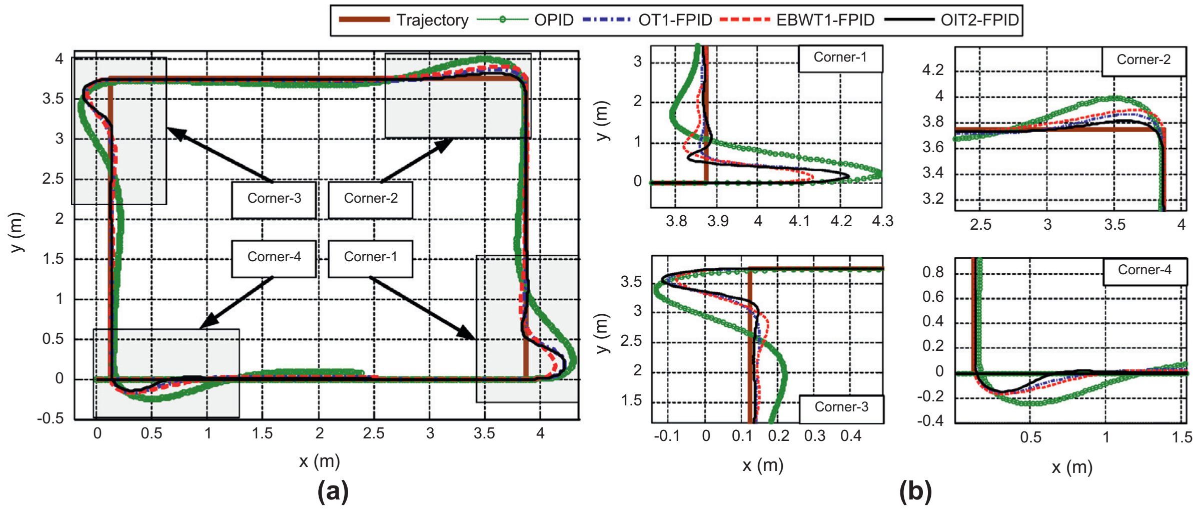

Fig. 10. Illustration of the simulation results: (a) tracking results and (b) corner variations.

$$Dev = \begin{cases} Max(|y_r(t) - y(t)|) \\ Max(|x_r(t) - x(t)|) \end{cases} \tag{23}$$

The tracking results on the *x*–*y* axis of the implemented cascade structures are illustrated in Fig. 10a for the constant linear speed of $v = 0.27$ m/s. As it can be clearly seen, the mobile robot converges faster to the reference trajectory when the OIT2-FPID structure is implemented with a low IAE_1 and IAE_2 value as tabulated in Table 5. Moreover, in order to examine the transient performance of the structures, the corners where the sudden change of the reference trajectory occurs are investigated and presented in Fig. 10b. The maximum deviation values are presented in Table 5. For instance, the OIT2-FPID structure has lower deviation values in Corner-2 and Corner-4 while in Corner-1 and Corner-3 has higher values in comparison with the EBWT1-FPID structure. However, the overall performance of the OIT2-FPID is superior in comparison with OPID, OT1-FPID and EBWT1-FPID with respect to IAE_1 and IAE_2 values.

It should be noted that, the presented IAE_1 performance values presented in Table 5 are not the same as the ones in Table 4. The IAE_1 performance values presented in Table 4 are calculated with respect to the reference trajectory with the values of $\pi/2,\pi,3\pi/2$ and 2π radian while the IAE_2 performance values, presented in Table 5, are calculated with respect to reference heading angle ($\theta_{ref}(t)$) generated by the outer loop controller.

Moreover, to establish if the proposed OIT2-FPID more complex cascade structure will cause drastic impact on the controllers real-time response, we have investigated the computation time needed by the OIT2-FPID, OT1-FPID and EBWT1-FPID

Table 5
Performance comparisons of the control structures for the simulation study.

	Maximum deviations from the corners				Total	Total
	Corner-1 (m)	Corner-2 (m)	Corner-3 (m)	Corner-4 (m)	IAE$_1$	IAE$_2$
OPID	0.425	0.250	0.250	0.250	70.794	32.313
OT1-FPID	0.345	0.120	0.220	0.170	64.495	23.417
EBWT1-FPID	0.255	0.150	0.190	0.170	74.519	18.787
OIT2-FPID	0.345	0.060	0.220	0.150	59.082	17.995

cascade structures (which are composed of two controllers, i.e. inner and outer controller) to map an input to an output for all possible combinations of the input values in the universe of discourses of $E \in [-1,+1]$ and $\Delta E \in [-1,+1]$ with a discretization magnitude of 0.1. It was found that the average computation times of the OT1-FPID, EBWT1-FPID and OIT2-FPI cascade control structures are 0.36 ms, 0.52 ms and 0.58 ms respectively, where the obtained maximum computation time values are 3.87 ms, 4.55 ms and 4.92 ms, respectively. Hence, the average and maximum computation times of IT2-FPID are close to the T1-FPID and EBWT1-FPID cascade control structures. Hence, the real-time control application of the OIT2-FPID cascade structure is feasible for systems with relatively small sampling periods and it will be shown later that the proposed OIT2-FPID structure produces a superior transient state and disturbance rejection performances in comparison to the OT1-FPID and EBWT1-FPID controllers' counterparts.

4.5. Real-time closed loop control performance analysis

In this section, we present real world experiments using the PIONEER 3-DX mobile robot to evaluate the proposed interval type-2 fuzzy controller structure. The robot is a typical nonholonomic mobile robot which has two rear driving wheels and a front caster. The robot measures 45 × 38 × 25 cm, includes two 19-cm wheels, one caster, and eight ultrasonic sensors front [9,36]. The pose of the mobile robots ($x(t)$, $y(t)$, $\theta(t)$) has been computed at each sampling time ($T_s = 0.1$ s) using the odometry system which is based on internal sensors. It is well known that odometry is a technique which has an accumulative error which implies the need to update the estimation from data provided from an external sensor system at a predetermined sampling period [30,31]. This issue is separate from the control problem, and has not been addressed in the paper. The experimental environment of the PIONEER 3-DX mobile robot is shown in Fig. 3c. It should be noted that, the presented reference trajectory in Fig. 3c is just for visualization. Thus, the ultrasonic sensors of mobile robot are not used to define the presented trajectory; the reference trajectory is predefined and implemented in the reference path mechanism which is illustrated in Fig. 5a.

In the real-time control studies, different analyses are presented to investigate the control performances of the OPID, OT1-FPID, EBWT1-FPID and OIT2-FPID cascade control structures. At first, we analyze the real-time control performances for the reference trajectory at which the controllers are designed to investigate their robustness against noise and unmodelled dynamics of the PIONEER 3-DX mobile robot. Next, we investigate the ability of the controllers to cope with the change of the mobile robot's initial pose to a different value at which they were not optimized. Finally, the input and output disturbance rejection performances are presented and examined.

4.5.1. Trajectory control performance analysis

This subsection presents an experimental comparison of the presented cascade control structures for a square like reference trajectory. The OIT2-FPID structure is compared with the OPID, the OT1-FPID and EBWT1-FPID cascade structures. The starting point of the mobile robot was the center on the origin of the coordinate system ($x_0 = 0$ m, $y_0 = 0$ m) with the initial heading of $\theta_0 = 0°$. The tracking results on the x–y axis are shown in Fig. 11a. The presented experimental results of the OIT2-FPID structure show that the mobile robot converges to the reference trajectory and the error converges to zero in short of time when compared to the OPID, OT1-FPID and EBWT1-FPID counterparts. As tabulated in Table 6, the mobile robot tracked the reference trajectory where the OIT2-FPID cascade control structure resulted in the lowest IAE_1 and IAE_2 performance values when compared to the employed counterparts.

Moreover, to examine the transient performances, the corners where the sudden reference variations are examined and presented in Fig. 12. As it can be clearly seen in Table 6, the OIT2-FPID cascade structure improved the overall performance of the system much better and reduced the variations of the Corner-1, Corner-2, Corner-3 and Corner-4 to 0.025 m, 0.185 m, 0.425 m and 0.225 m, respectively in comparison with the other employed controllers. The results confirm that OT1-FPID and OPID controllers can handle limited amounts of uncertainty as it has been widely shown in the literature while the OIT2-FPID structure has the ability to cope better with higher levels of noise and uncertainties compared to the other controllers. It should be noted that the performance values of the corner variations of the EBWT1-FPID structure is also satisfactory. However, the control signal of the EBWT1-FPID structure inherits high frequency chattering while the OIT2-FPID structure has smoother control signal as shown in Fig. 11b. Although the online rule weighing mechanism provides the EBWT1-FPID more extra degree of freedom in comparison with the OT1-FPID structure, this high frequency chattering behavior was expected in the presence of noise since the tuning mechanism is based on the value of the error. It should be also noted that, the OT1-FPID and OPID control signals do not inherit high frequency chattering as the EBWT1-FPID;

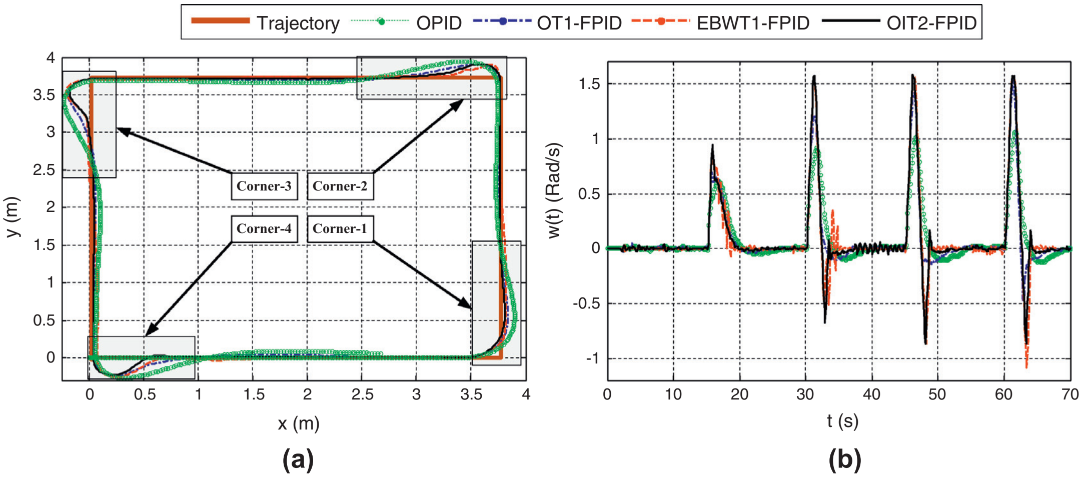

Fig. 11. Illustration of the real-time tracking performance: (a) tracking results and (b) control signals.

Table 6
Tracking Control Performance Comparisons of the Control Structures for the real-time control study.

	Maximum deviations in the corners				Total	Total
	Corner-1 (m)	Corner-2 (m)	Corner-3 (m)	Corner-4 (m)	IAE$_1$	IAE$_2$
OPID	0.125	0.230	0.480	0.260	61.640	38.275
OT1-FPID	0.063	0.185	0.445	0.240	57.266	27.525
EBWT1-FPID	0.045	0.188	0.460	0.250	86.859	27.046
OIT2-FPID	0.025	0.185	0.425	0.225	42.168	22.334

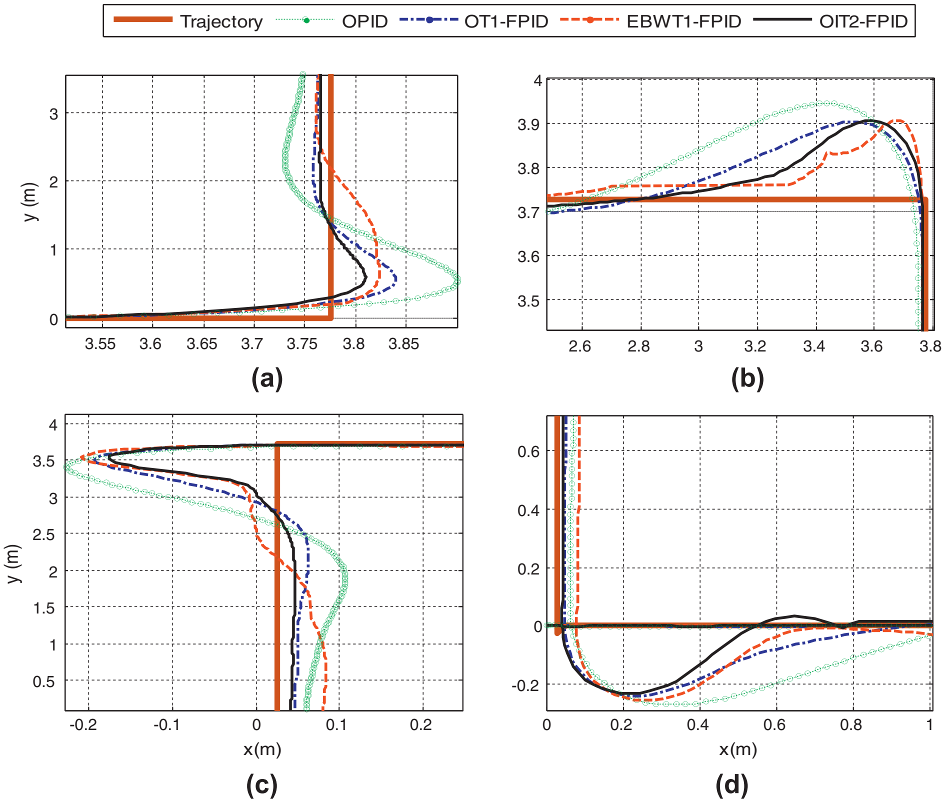

Fig. 12. Illustration of the real-time tracking results for the (a) Corner-1, (b) Corner-2, (c) Corner-3 and (d) Corner-4.

nevertheless the tracking performance of the OIT2-FPID structure is better than the OT1-FPID and OPID structures as shown in Table 6.

4.5.2. Effect of changing the initial starting pose

The performances of the designed cascade control structures have been also tested for a different initial pose at which they are not optimized. Thus, the same square like reference trajectory has been applied to the mobile robot with the initial pose of $(x_0, y_0, \theta_0) = (0\text{ m}, -0.2\text{ m}, 60°)$ to investigate how the controllers perform. In Fig.13a, the transient convergence of the mobile robot to the square like reference trajectory is presented. The OIT2-FPID structure provides a better control performance than the OPID and OT1-FPID cascade structures with the respect to the performance measures as given in Table 7. As illustrated in Fig. 13b, the OIT2-FPID cascade structure generates a smoother control signal while the EBWT1-FPID control signal inherits chattering. The results confirm that the OIT2-FPID structure has the ability to cope better with the change of the robot's initial pose when compared to the other controllers.

4.5.3. Disturbance rejection performance analysis

This subsection examines the input and output disturbance rejection performances of the implemented cascade control structures. Thus, it is assumed that the mobile robot is in steady state at the operating point $(y(t) = 0,\ \theta(t) = 0,\ w(t) = 0,\ \upsilon(t) = 0.27\text{ m/s})$. At first, a step input disturbance with a magnitude of "-0.5" has been applied the mobile robot. The input disturbance rejection performances of the control systems are presented in Fig. 14a. As it can be observed, OIT2-FPID cascade structure showed noticeably a superior performance compared with type-1, self-tuning and conventional PID controllers' counterparts while producing a smother control signal in comparison to the EBWT1-FPID cascade structure as shown in

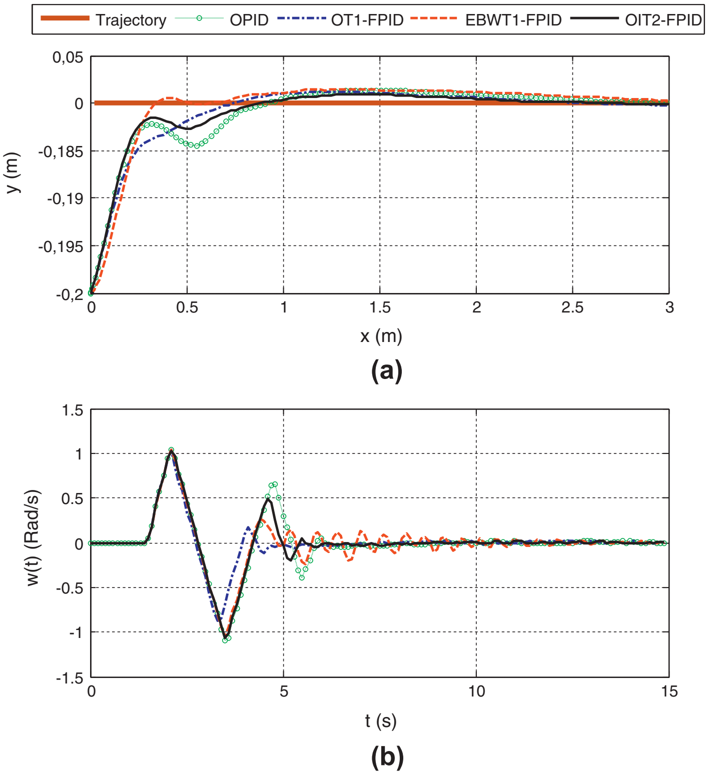

Fig. 13. Illustration of (a) the real-time tracking results and (b) the control signals for different initial pose.

Table 7
Robustness performance comparisons of the control structures for the real-time control study.

	Initial starting pose		Input disturbance		Output disturbance	
	IAE_1	IAE_2	IAE_1	IAE_2	IAE_1	IAE_2
OPID	9.831	6.530	19.900	8.339	12.130	5.339
OT1-FPID	13.407	5.986	13.115	8.072	9.518	4.616
EBWT1-FPID	11.841	6.207	9.958	6.616	13.315	5.972
OIT2-FPID	9.210	5.999	4.941	5.060	6.431	4.339

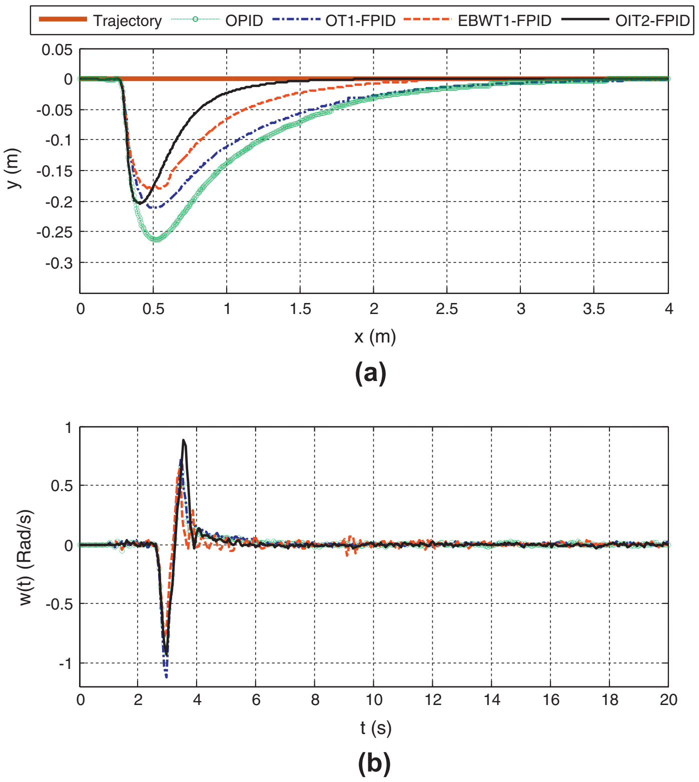

Fig. 14. Illustration of the input disturbance rejection performance (a) the tracking results and (b) the control signals.

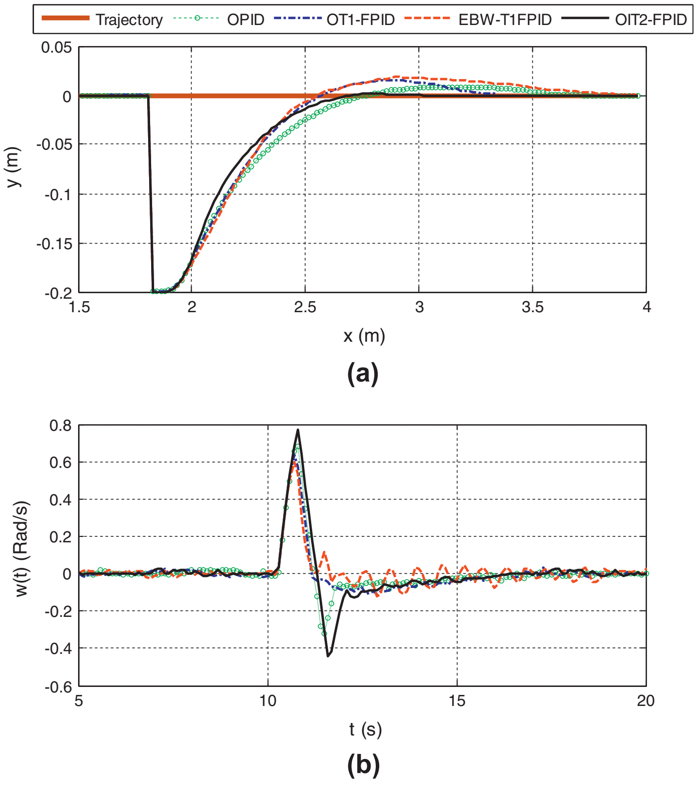

Fig. 15. Illustration of the output disturbance rejection performance (a) the tracking results and (b) the control signals.

Fig. 14b. It should be noted that, the OT1-FPID and OPID control structures have also generated smooth control signals; nevertheless the disturbance rejection performance of the OIT2-FPID cascade structure is much better in every sense as tabulated in Table 7.

Secondly, a step output disturbance with a magnitude of "−0.2" is applied to the control system. The control signals and tracking results are given in Fig. 15a. As it can be seen, the proposed OIT2-FPID structure compensated very effectively the disturbance in a short period of time compared to the other implemented controller structures. It should be noted that, the output disturbance rejection performance of the EBWT1-FPID structure is the poorest. As the EBWT1-FPID structure is sensitive against noise, it was not able to generate an appropriate control signal that will achieve a satisfactory system response as shown in Fig. 15b. In other words, the noise directly affected the output disturbance rejection performance of the EBWT1-FPID while OIT2-FPID cascade structure produced superior control performance even in the presence of noise and disturbances. Note that the OPID and OT1-FPID structures also generated smooth control signals but the disturbance rejection performance of the OIT2-FPID cascade structure is much better in every sense as tabulated in Table 7.

The results presented in Table 7 confirm that the OIT2-FPID cascade structure is more capable to handle disturbances in the presence of noise and uncertainties when compared to its conventional and fuzzy controller counterparts.

5. Conclusions and future work

It has been shown in various works that IT2-FLCs might be able to handle high levels of nonlinearities and uncertainties to produce a better control performance when compared to T1-FLCs. However, the design of IT2-FLCs is a challenging problem due to the difficulty in determining the parameters of the IT2-FSs. In this paper, we presented a BB–BC optimization based approach for the design of IT2-FPID controllers. Since the number of optimization parameters of IT2-FLC is relatively big, we employed the BB–BC method which has a relatively low computational cost and a high convergence speed. The optimization performances of the IT2-FPID cascade control structure was compared with an OT1-FPID and OPID controller structures based on the *IAE* performance index. In the design of the IT2-FPID structure, the rule base and scaling factors were kept fixed while only the antecedent parameters were optimized for a reference trajectory to clearly show the effect of the FOU on the system response.

We also compared the optimization results of the BB–BC method with the PSO method which is another fast and efficient evolutionary based strategy based on a statistical evaluation since both methods are stochastic global search methods. The presented results showed that the BB–BC based design method can obtain higher quality solutions in comparison with PSO with less computational time. Hence, the use of the BB–BC can help in the future to design more complex IT2FLCs with more inputs in applications where there is a need to get a very good optimization result in relatively fast time frame. Moreover, the results demonstrated that the OIT2-FPID controller outperforms the OT1-FPID and OPID structures with respect to *IAE* performance index (regardless the optimization method employed). This is because the IT2-FLC can implement more sophisticated control models that cannot be achieved by OPID or OT1-FPID using the same rulebase [41]. In other words, the extra degree of freedom of the IT2-FPID (provided by the FOUs of the IT2 FSs) gives the BB–BC method an opportunity to find a more optimal solution than those obtained for the OPID and OT1-FPID. As part of this study, we also considered a self-tuning EBWT1-FPID control structure which has the same degrees of freedom like the IT2-FPID structure since its fuzzy rule weights are handled as online tuning parameters. Thus, we also analyzed the control performances of the EBWT1-FPID and the OIT2-FPID cascade structures to illustrate the reason for the superior control performance of IT2-FPID under high levels of uncertainty and noise is not merely for its use of extra parameters, but rather its different way of dealing with the uncertainties and noise present in real world environments.

We evaluated the proposed conventional and fuzzy cascade control structures on the path tracking control performance of a mobile robot which inherit large amounts of nonlinearities and uncertainties. The presented controllers were embedded in simple and effective cascade architecture, which has been also developed in this paper, for the tracking control of mobile robots. We presented simulations and real world experiments using the PIONEER 3-DX mobile robot to evaluate the proposed interval type-2 fuzzy controller structure in this paper. The simulation and experimental results support the effectiveness of the proposed IT2-FPID cascade control design approach. It has been illustrated that especially the real-time tracking performance of the OIT2-FPID structure is better in presence of high levels of noise and is more robust against disturbances and uncertainties when compared to OT1-FPID, EBWT1-FPID and OPID cascade control structures. Moreover, since we also compared the OIT2-FPID with the EBWT1-FPID structure (which has extra degrees of freedom related to the rules weights), we came to the conclusion that the reason for the superior control performance of IT2-FPID structure under high levels of uncertainty and noise is not merely for its use of extra parameters, but rather its different way of dealing with the uncertainties and noise present in real world environments.

For our future work, we aim to focus on extending the optimization based cascade design strategy and the EBW tuning mechanism for generalized type-2 fuzzy controller structures.

Acknowledgement

This research is supported by the Project (1059B191200447) of Scientific and Technological Research Council of Turkey (TUBITAK). All of these supports are appreciated.

References

[1] G. Antonelli, G. Chiaverini, G. Fusco, A fuzzy-logic-based approach for mobile robot path tracking, IEEE Trans. Fuzzy Syst. 15 (2) (2007) 211–221.
[2] A.B. Cara, C. Wagner, H. Hagras, H. Pomares, I. Rojas, Multi-objective optimization and comparison of non-singleton type-1 and singleton interval type-2 fuzzy logic systems, IEEE Trans. Fuzzy Syst. 21 (3) (2013) 459–476.
[3] O. Castillo, P. Melin, A review on the design and optimization of interval type-2 fuzzy controllers, Appl. Soft Comput. 12 (2012) 1267–1278.
[4] O. Castillo, P. Melin, Optimization of type-2 fuzzy systems based on bio-inspired methods: a concise review, Inf. Sci. 205 (2012) 1–19.
[5] D. Chwa, Sliding-mode tracking control of nonholonomic wheeled mobile robots in polar coordinates, IEEE Trans. Control Syst. Technol. 12 (4) (2004) 633–644.
[6] O.K. Erol, I. Eksin, A new optimization method: Big Bang Big Crunch, Adv. Eng. Softw. 37 (2006) 106–111.
[7] T. Fukao, H. Nakagawa, N. Adachi, Adaptive tracking control of a nonholonomic mobile robot, IEEE Trans. Robot. Autom. 16 (5) (2000) 609–615.
[8] M. Galluzzo, B. Cosenza, A. Matharu, Control of a nonlinear continuous bioreactor with bifurcation by a type-2 fuzzy logic controller, Comput. Chem. Eng. 32 (12) (2008) 2986–2993.
[9] D. Gu, H. Huosheng, Receding horizon tracking control of wheeled mobile robots, IEEE Trans. Control Syst. Technol. 14 (4) (2006) 743–749.
[10] H. Hagras, A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots, IEEE Trans. Fuzzy Syst. 12 (4) (2004) 524–539.
[11] D. Hidalgo, O. Castillo, P. Melin, An optimization method for designing type-2 fuzzy inference systems based on the footprint of uncertainty using genetic algorithms, Expert Syst. Appl. 39 (4) (2012) 4590–4598.
[12] C.L. Hwang, L.J. Chang, Y.S. Yu, Network-based fuzzy decentralized sliding-mode control for car-like mobile robots, IEEE Trans. Ind. Electron. 54 (1) (2007) 574–585.
[13] S. Iplikci, A support vector machine based control application to the experimental three-tank system, ISA Trans. 49 (3) (2010) 376–386.
[14] E.A. Jammeh, M. Fleury, C. Wagner, H. Hagras, M. Ghanbari, Interval type-2 fuzzy logic congestion control for video streaming across IP networks, IEEE Trans. Fuzzy Syst. 17 (5) (2009) 1123–1142.
[15] O. Karasakal, M. Guzelkaya, I. Eksin, E. Yesil, An error-based on-line rule weight adjustment method for fuzzy PID controllers, Expert. Syst. Appl. 38 (8) (2011) 10124–10132.
[16] N.N. Karnik, J.M. Mendel, Q. Liang, Type-2 fuzzy logic systems, IEEE Trans. Fuzzy Syst. 7 (1999) 643–658.
[17] A. Kaveh, S. Talatahari, Size optimization of space trusses using Big Bang–Big Crunch algorithm, Comput. Struct. 87 (2009) 1129–1140.
[18] I. Kaya, Improving performance using cascade control and a Smith predictor, ISA Trans. 40 (3) (2001) 223–234.
[19] G. Klančar, I. Škrjanc, Tracking-error model-based predictive control for mobile robots in real time, Robot. Auton. Syst. 55 (6) (2007) 460–469.
[20] C.F. Kucuktezcan, V.M.I. Genc, Big Bang–Big Crunch based optimal preventive control action on power systems, in: 3rd IEEE PES International Conference and Exhibition on, 2012, pp. 1–4.
[21] T. Kumbasar, A simple design method for interval type-2 fuzzy PID controllers, Soft Comput. 18 (7) (2014) 1293–1304.
[22] T. Kumbasar, I. Eksin, M. Guzelkaya, E. Yesil, Adaptive fuzzy model based inverse controller design using BB–BC optimization algorithm, Expert. Syst. Appl. 38 (10) (2011) 12356–12364.
[23] T. Kumbasar, I. Eksin, M. Guzelkaya, E. Yesil, Type-2 fuzzy model based controller design for neutralization processes, ISA Trans. 51 (2) (2012) 277–287.
[24] Q. Liang, J.M. Mendel, Interval type-2 fuzzy logic systems: theory and design, IEEE Trans. Fuzzy Syst. 8 (5) (2000) 535–550.
[25] W.L. Luyben, Parallel cascade control, Ind. Eng. Chem. Fundam. 12 (4) (1973) 463–467.
[26] Y. Maldonado, O. Castillo, P. Melin, Particle swarm optimization of interval type-2 fuzzy systems for FPGA applications, Appl. Soft Comput. 13 (1) (2013) 496–508.
[27] J.S. Martínez, J. Mulot, F. Harel, D. Hissel, M.C. Péra, R.I. John, M. Amiet, Experimental validation of a type-2 fuzzy logic controller for energy management in hybrid electrical vehicles, Eng. Appl. Artif. Intel. 26 (7) (2013) 1772–1779.
[28] R. Martínez, O. Castillo, L.T. Aguilar, Optimization of interval type-2 fuzzy logic controllers for a perturbed autonomous wheeled mobile robot using genetic algorithms, Inf. Sci. 179 (13) (2009) 2158–2174.
[29] P. Melin, L. Astudillo, O. Castillo, F. Valdez, M. Garcia, Optimal design of type-2 and type-1 fuzzy tracking controllers for autonomous mobile robots under perturbed torques using a new chemical optimization paradigm, Expert Syst. Appl. 40 (8) (2013) 3185–3195.
[30] J.E. Normey-Rico, I. Alcalá, J. Gómez-Ortega, E.F. Camacho, Mobile robot path tracking using a robust PID controller, Control Eng. Pract. 9 (11) (2001) 1209–1214.
[31] J.E. Normey-Rico, J. Gomez-Ortega, E.F. Camacho, A Smith predictor based generalized predictive controller for mobile robot path tracking, Control Eng. Pract. 7 (1999) 729–740.
[32] S.-K. Oh, H.-J. Jang, W. Pedrycz, A comparative experimental study of type-1/type-2 fuzzy cascade controller based on genetic algorithms and particle swarm optimization, Expert. Syst. Appl. 38 (9) (2011) 11217–11229.
[33] W. Pedrycz, Granular Computing: Analysis and Design of Intelligent Systems, CRC Press, Boca Raton, 2013.
[34] W.Z. Qiao, M. Mizumoto, PID type fuzzy controller and parameters adaptive method, Fuzzy Sets Syst. 78 (1996) 23–35.
[35] F.M. Raimondi, M. Melluso, A new fuzzy robust dynamic controller for autonomous vehicles with nonholonomic constraints, Rob. Auton. Syst. 52 (2) (2005) 115–131.
[36] G. Scaglia1, A. Rosales, L. Quintero, V. Mut, R. Agarwal, A linear-interpolation-based controller design for trajectory tracking of mobile robots, Control Eng. Pract. 18 (3) (2010) 318–329.
[37] H. Tang, J. Zhou, S. Xue, L. Xie, Big Bang–Big Crunch optimization for parameter estimation in structural systems, Mech. Syst. Signal Process. 24 (8) (2010) 2888–2897.
[38] R.-J. Wai, M.-A. Kuo, J.-D. Lee, Cascade direct adaptive fuzzy control design for a nonlinear two-axis inverted-pendulum servomechanism, IEEE Trans. Syst. Man Cybern. 38 (2) (2008) 439–454.
[39] Wu D, Tan WW, Interval type-2 fuzzy PI controllers: why they are more robust, in: IEEE International Conference on Granular Computing, Silicon Valley, August 2010.
[40] D. Wu, W.W. Tan, Genetic learning and performance evaluation of internal type-2 fuzzy logic controllers, Eng. Appl. Artif. Intel. 19 (2006) 829–841.
[41] D. Wu, On the fundamental differences between type-1 and interval type-2 fuzzy logic controllers, IEEE Trans. Fuzzy Syst. 10 (5) (2012) 832–848.