Online ISSN 1848-3380, Print ISSN 0005-1144
ATKAFF 55(1), 69–78(2014)

*Mohammad Hassan Khooban, Davood Nazari Maryam Abadi, Alireza Alfi, Mehdi Siahi*

# Optimal Type-2 Fuzzy Controller For HVAC Systems

DOI 10.7305/automatika.2014.01.219
UDK 681.515.8.017-53:[697+644]
IFAC 2.2; 5.5.1.5

**Original scientific paper**

In this paper a novel Optimal Type-2 Fuzzy Proportional-Integral-Derivative Controller (OT2FPIDC) is designed for controlling the air supply pressure of Heating, Ventilation and Air-Conditioning (HVAC) system. The parameters of input and output membership functions, and PID controller coefficients are optimized simultaneously by random inertia weight Particle Swarm Optimization (RNW-PSO). Simulation results show the superiority of the proposed controller than similar non-optimal fuzzy controller.

**Key words:** HVAC systems, Optimal Type-2 Fuzzy Proportional-Integral-Derivative controller (OT2FPIDC), Random inertia weight particle swarm optimization (RNW-PSO)

**Optimalni neizraziti reglutor tipa 2 za sustave za grijanje, ventilaciju i klimatizaciju.** U radu je predložena nova upravljačka shema optimalnog neizrazitog PID regulatora tipa 2 za upravljanje sustavima za grijajne, ventilaciju i klimatizaciju. Predložena je shema zasnovana na neizrazitom regulatoru (FLC) učestalo korištenom za upravljajne nelinearnim procesima. Kako bi se premostio problem neizrazitih regulatora, neodstatak metode dizajnirajna, parametri ulazno-izlaznih funkcija pripadanja, kao i parametri PID regulatora se optimiraju metodom roja čestica sa slučajnim parametrima inercije (RNW-PSO). Simulacijski rezultati pokazuju izvedivost predloženog pristupa.

**Ključne riječi:** HVAC sustavi, neizrazitog PID regulatora tipa 2 za upravljanje sustavima (OT2FPIDC), algoritam roja čestica sa slučajnim parametrima inercije (RNW-PSO)

## 1 INTRODUCTION

Heating, Ventilation and Air-Conditioning (HVAC) mechanisms are needed for setting environmental variables including, temperature, moisture, and pressure. As with other industrial usages, most of the processes associated with HVAC are controlled by PID controllers. The prevalent PID controllers are extensively applied because of their easy calculations, easy application, appropriate robustness, high dependability, stabilizing and zero persistent state error. However HVAC mechanism is a nonlinear and time variant mechanism. It is hard to access favorable tracking control efficiency, because tuning and self-adapting adjustment of parameters automatically are a perennial issue of PID controller. During the recent decades various methods for identifying PID controller parameters have been presented. In some techniques the open loop response information of system is used, for instance Cohen-Coon reaction curve procedure [1].

In recent years, researchers have extensively used the fuzzy logic for modeling, identification, and control of highly nonlinear dynamic systems [2,3]. In [4-8], different combination of control methods are suggested to improve the efficiency of fuzzy PI or PID controllers. Adjustment process of PID controller coefficients can take a long time, and can be hard and costly work [8,9]. Usually a proficient gainer attempts to control the process by adjusting the coefficients of controller according to error and change rate of error in order to achieve the optimal response. In this paper the optimal adjustment is obtained by random inertia weight Particle Swarm Optimization (RNW-PSO).

In the HVAC mechanism the supply air pressure is tuned by changing the speed of a supply air fan. The relationship between fan speed and pressure of air source can be expressed by a delayed second order transfer function as is described by Bi and Cai [11]. Since in various operating conditions both fans and dampers show non-linear behaviour from themselves, even a well-regulated controller is unable to meet design requirements due to the existing uncertainties in parameters of system.

Motivated by the aforementioned researches, the purpose of this paper is to present a novel Optimal Type-2 Fuzzy Proportional Integral Derivative Controller (OT2FPIDC) for regulating the air supply pressure of

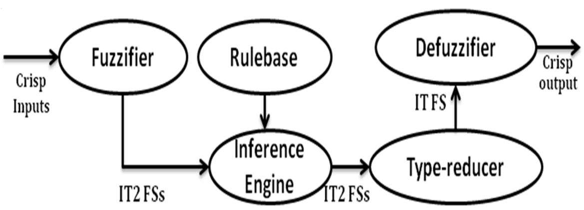


*Fig. 1. An IT2 FLS*

Heating, Ventilation and Air-Conditioning (HVAC) system. The parameters of input and output membership functions, and PID controller coefficients are optimized simultaneously by random inertia weight Particle Swarm Optimization (RNW-PSO). Simulation results indicate that the proposed controller has faster response, smaller overshoot and higher accuracy than Proportional Integral Derivative PID, Adaptive Neuro Fuzzy (ANF), and Self-Tuning Fuzzy PI Controlle (STFPIC) under normal condition and under existing uncertainties in parameters of model.

## 2 TYPE-2 FUZZY SETS AND SYSTEMS

Type-2 fuzzy sets and systems generalize (type-1) fuzzy sets and systems so that more uncertainty can be handled. From the very beginning of fuzzy sets, criticism was made about the fact that the membership function of a type-1 fuzzy set has no uncertainty associated with it, something that seems to contradict the word *fuzzy*, since that word has the connotation of lots of uncertainty.

### 2.1 Interval Type 2 Fuzzy Sets (IT2 FSs)

In spite of having a name which carries the concept of uncertainty, studies has demonstrated that there are restrictions in the ability of T1 FSs to model and minimize the effect of uncertainties [12-15]. This is because a T1 FS is fixed this means that its membership degrees are crisp amounts. Lately, type-2 FSs [16], specified by MFs that are themselves fuzzy, have been attracting interests. Interval type-2 (IT2) FSs [14], a special case of type-2 FSs, are currently the most widely used for their reduced computational cost.

### 2.2 Interval Type-2 Fuzzy Logic System (IT2 FLS)

Fig. 1 indicates the schematic diagram of an IT2 FLS. It is similar to its T1 equivalent, the main difference being that at least one of the FSs in the rule base is an IT2 FS. Hence, the outputs of the inference engine are IT2 FSs, and a type-reducer is required to convert them into a T1 FS before defuzzification can be performed.

Actually the calculations in an IT2 FLS can be considerably simplified. Consider the rulebase of an IT2 FLS consisting of N rules, supposing the following form:

$$R^n : IF\ x_1\ is\ \tilde{X}_1^n ... and\ x_1\ is\ \tilde{X}_1^n.\ THENy\ is\ Y^n, n=1,2,...,N,$$

where $\tilde{X}_i^n$ ($i$ = 1, . . ., $I$) are IT2 FSs, and $Yn$ = [$\underline{y}n$, $\overline{y}n$] is an interval, which can be understood as the centroid [13, 16] of a consequent IT2 FS, or the simplest TSK model, for its simplicity. In many applications we use $\underline{y}n = \overline{y}n$ , i.e., each rule consequent is a crisp number. Suppose the input vector is $\mathbf{x}' = (x_1', x_2', \ldots, x_I')$. Typical calculations in an IT2 FLS include the following steps:

1. Calculate the membership of $x_i'$ on each $\mathbf{X}_i^n$,

$$[\mu\underline{X}_i^n(x_i'), \mu\overline{X}_i^n(x_i')], i = 1, 2, \ldots, I, n = 1, 2, \ldots, N. \quad (1)$$

2. Calculate the firing interval of the $n^{th}$ rule, $F^n(\mathbf{x})$:

$$F^n(\mathbf{x}') = [\mu\underline{X}_1^n(x_1') \times \ldots \times \mu\underline{X}_1^n(x_1'), \mu\bar{X}_1^n(x_1') \times \ldots \times \mu\bar{X}_1^n(x_1')] \equiv [\underline{f}^n, \bar{f}^n], \qquad n = 1, \ldots, N \quad (2)$$

3. Apply type-reduction to combine $F^n(\mathbf{x}')$ and the related rule consequents. There are many such methods. The most commonly used one is the center-of-sets type-reducer [13]:

$$Y_{\cos}(x') = \bigcup_{\substack{f^n \in F^n(x') \\ y^n \in Y^n}} \frac{\sum_{n=1}^N f^n y^n}{\sum_n^N f^n} = [y_l, y_r] \quad (3)$$

It has been demonstrated that [14,18,19]:

$$y_l = \min_{k\in[1,N-1]} \frac{\sum_{n=1}^k \bar{f}^n \underline{y}^n + \sum_{n=k+1}^N \underline{f}^n \underline{y}^n}{\sum_{n=1}^k \bar{f}^n + \sum_{n=k+1}^N \underline{f}^n} = \frac{\sum_{n=1}^L \bar{f}^n \underline{y}^n + \sum_{n=L+1}^N \underline{f}^n \underline{y}^n}{\sum_{n=1}^L \bar{f}^n + \sum_{n=L+1}^N \underline{f}^n}$$

$$y_l = \max_{k\in[1,N-1]} \frac{\sum_{n=1}^k \bar{f}^n \underline{y}^n + \sum_{n=k+1}^N \underline{f}^n \underline{y}^n}{\sum_{n=1}^k \bar{f}^n + \sum_{n=k+1}^N \underline{f}^n} \quad (4)$$

$$= \frac{\sum_{n=1}^R \bar{f}^n \underline{y}^n + \sum_{n=R+1}^N \underline{f}^n \underline{y}^n}{\sum_{n=1}^R \bar{f}^n + \sum_{n=R+1}^N \underline{f}^n} \quad (5)$$

, where the *switch points* L and R are specified by

$$\underline{y}^L \leq y_l \leq \underline{y}^{L+1} \quad (6)$$

$$\bar{y}^R \leq y_r \leq \bar{y}^{R+1} \tag{7}$$

and $\underline{y}^n$ and $\overline{y}^n$ have been sorted in ascending order, respectively. $y_l$ and $y_r$ can be calculated using the Karnik-Mendel (KM) algorithms [14].

**KM Algorithm for Computing $y_l$ [20]:**

1. Sort $\underline{y}_n$ $(n = 1, 2, \ldots, N)$ in increasing order and call the sorted $\underline{y}_n$ by the same name, but now $\underline{y}_1 = \underline{y}_2 \cdots = \underline{y}_N$. Match the weights $F_n(\mathbf{x}')$ with their respective $\underline{y}_n$ and renumber them so that their index corresponds to the renumbered $\underline{y}_n$.
2. Initialize $f_n$ by setting
$$f^n = \frac{\underline{f}^n + \bar{f}^n}{2} \qquad n = 1, 2, ..., N \tag{8}$$
and then compute
$$y = \frac{\sum_{n=1}^{N} f^n \underline{y}^n}{\sum_{n}^{N} f^n} \tag{9}$$
3. Find switch point $k$ (1 $k$ $N$ – 1) such that
$$\underline{y}^k \leq y \leq \underline{y}^{K+1} \tag{10}$$
4. Set
$$f^n = \{^{\bar{f}^n, \quad n \leq k}_{\underline{f}^n, \quad n \succ k} \tag{11}$$
And calculate
$$y' = \frac{\sum_{n=1}^{N} f^n \underline{y}^n}{\sum_{n}^{N} f^n} \tag{12}$$
5. Check if $y' = y$. If yes, stop and set $y_r = y$ and $L = k$. If no, go to Step 6.
6. Set $y' = y$ and go to Step 3.

**KM Algorithm for Computing $y_r$ [20]:**

1. Sort $\overline{y}^n (n = 1, 2, \ldots, N)$ in increasing order and call the sorted $\overline{y}^n$ by the same name, but now $\overline{y}_1 = \overline{y}_2 \ldots \overline{y}_N$. Match the weights $F_n(\mathbf{x}')$ with their respective $\overline{y}n$ and renumber them so that their index corresponds to the renumbered $\overline{y}_n$.
2. Initialize $f_n$ by setting
$$f^n = \frac{\underline{f}^n + \bar{f}^n}{2} \qquad n = 1, 2, ..., N$$
and then calculate
$$y = \frac{\sum_{n=1}^{N} f^n \overline{y}^n}{\sum_{n}^{N} f^n} \tag{13}$$
3. Find switch point $k$ (1 $k$ $N$ – 1) such that
$$\bar{y}^k \leq y \leq \bar{y}^{K+1} \tag{14}$$
4. Set
$$f^n = \{^{\underline{f}^n, \quad n \leq k}_{\bar{f}^n, \quad n \succ k} \tag{15}$$
and calculate
$$y' = \frac{\sum_{n=1}^{N} f^n \overline{y}^n}{\sum_{n}^{N} f^n} \tag{16}$$
5. Check if $y' = y$. If yes, stop and set $y_r = y$ and $R = k$. If no, go to Step 6.
6. Set $y' = y$ and go to Step 3.
   The main idea of the KM algorithm is to find the switch points for *yl* and *yr*.
7. Compute the defuzzified output as:
$$y = \frac{y_l + y_r}{2} \tag{17}$$

## 3 PARTICLE SWARM OPTIMIZATION

The PSO algorithm is a partly new population-based heuristic optimization method which is based on a metaphor of social interaction, specifically bird flocking. The main benefits of PSO are: 1) The cost function's gradient is not needed, 2) PSO is more compatible and robust compared with other classical optimization techniques, 3) PSO guarantees the convergence to the optimum solution, and 4) In comparison with GA, PSO lasts fewer time for each function evaluation as it does not apply many of GA operators such as mutation, crossover and selection operator.

In PSO, any nominee solution is named "Particle". Each particle in the swarm demonstrates a nominee solution to the optimization problem, and if the solution is composed of a series of variables, the particle can be a vector of variables. In PSO, each particle is flown through the multidimensional search space, regulating its position in search space based on their momentum and both personal and global histories. Then the particle uses the best position faced by itself and that of its neighborhood to position itself toward an optimal solution. The appropriateness of each particle can be assessed based on the cost function of optimization problem. At each repetition, the speed of every particle will be computed as follows:

$$v_i(t+1) = \omega v_i(t) + c_1 r_q (P_{id} - x_i(t)) + c_2 r_2 \left(P_{gd} - x_i(t)\right), \tag{18}$$

where $x_i(t)$ is the present position of the particle, $p_{id}$ is one of the finest solutions this particle has achieved and $p_{gd}$ is one of the finest solutions all the particles have achieved. After computing the speed, the new position of each particle will be computed as follows

$$x_i(t+1) = x_i(t) + v_i(t+1). \tag{19}$$

The PSO algorithm is replicated using Eqs. 18 and 19 which are updated at each repetition, up to pre-defined number of generations is achieved.

### 3.1 Random inertia weight PSO

Although Standard PSO (SPSO) includes some significant improvements by providing high rate of convergence in particular problems, it does demonstrate some deficiencies. It is shown that SPSO has a weak capability to look for a fine particle due to the lack of speed control mechanism. Most of the procedures are tried to ameliorate the efficiency of SPSO by changeable inertia weight. The inertia weight is essential for the efficiency of PSO, which equilibrates global exploration and local exploitation capabilities of the swarm. A large inertia weight simplifies exploration, but it prolongs the convergence of particle. Unlike, a small inertia weight leads to rapid convergence, but it sometimes results local optimum. Therefore different inertia weight conformity algorithms have been recommended in the literatures [21]. In 2003 Zhang [22] studied the effect of random inertia weight in PSO (RNW-PSO), reporting empirical results that show its superior efficiency to LDW-PSO [23]. Eberhart and Shi [24] have recommended a random inertia weight factor for tracking dynamic systems. The new version of PSO namely RNW-PSO can be obtained by changing Eq. ((18)) as below

$$v_i(t+1) = r_0 v_i(t) + c_1 r_1 (P_{id} - x_i(t)) + c_2 r_2 \left(P_{gd} - x_i(t)\right), \tag{20}$$

where $r_0$ is a uniformly distributed random number inside the interval [0, 1], and other parameters are same as before. The RNW can overcome two bugs of LDW. First, decreasing the affiliation of inertial weight on the maximum repetition that is hardly predicted before tests. Second, abstaining from the lacks of local search capability in the beginning of run and global search capability at the end of run. Before starting the optimization procedure, a performance benchmark should be first presented.

### 3.2 Empirical Studies

In order to examine the effect of inertia weight on the PSO efficiency, three non-linear benchmark functions presented in literature [25, 26] were used because they are famous problems. The first function is the Rosenbrok function:

*Table 1. $V_{max}$ and $X_{max}$ values used for tests*

| Function | $X_{max}$ | $V_{max}$ |
|---|---|---|
| $f_1$ | 100 | 100 |
| $f_2$ | 10 | 10 |
| $f_3$ | 600 | 600 |

$$f_1(x) = \sum_{i=1}^{n} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2), \tag{21}$$

where $\mathbf{x} = [x_1, x_2, \ldots, x_n]$ is an n-dimensional real-valued vector.

The second is the generalized Rastrigrin function:

$$f_2(x) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10). \tag{22}$$

The third is the generalized Griewank function:

$$f_3(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1. \tag{23}$$

Three various amounts dimensions were tested: 10, 20 and 30. The maximum numbers of repetition were set as 1000, 1500 and 2000 in accordance with the dimentions 10, 20 and 30, respectively. For evaluation the scalability of PSO algorithm, three population sizes 20, 40 and 80 were used for each function according to various dimensions. Acceleration constants took the values $c_1 = c_2 = 2$. Constriction factor $C = 1$. To perform comparison, all the $V_{max}$ and $X_{max}$ were assigned by same parameter settings as in literature [26] and mentioned in Table 1. 500 trial runs were taken for each case

## 4 THE PROPOSED CONTROL METHOD

General scheme of proposed controller is shown in Fig. 2. The two inputs of the controller are the error $e$ and the change rate of error $\dot{e}$ , respectively and the output of controller is $\boldsymbol{U}$. The main shortage of the optimal Type-2 fuzzy-PID controller is the lack of systematic approaches to define fuzzy rules and fuzzy membership functions. As we know, most fuzzy rules are based on human knowledge and differ among persons despite the same system performance. Because of this, it is complex to assume that the given expert's knowledge captured in the form of the fuzzy controller leads to optimal control. Therefore, the efficient approaches for tuning the membership function and control rules without a trial and error method are significantly required. Because of this, the idea of employing RNW-PSO algorithm to achieve best rising time ($t_r$), settling time ($t_s$),

*Table 2. The used parameters of RNW-PSO*

| | |
|---|---|
| Size of the Swarm | 50 |
| Dimension of Problem | 20 |
| Maximum Number of iterations | 100 |
| Cognitive Parameter $C_1$ | 1 |
| Social Parameter $C_2$ | 1 |
| Construction Factor $C$ | 1 |

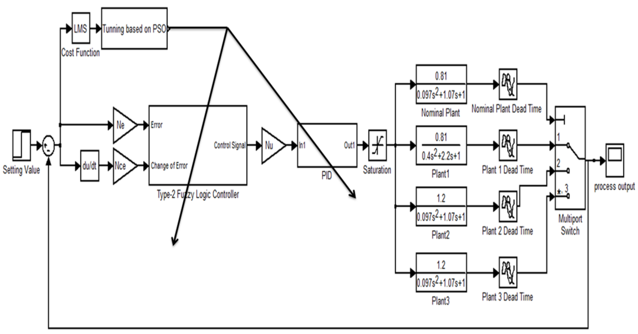


*Fig. 2. Optimal Type-2 Fuzzy-PID controller*

% peak overshoot ($M_p$), steady-state error ($E_{ss}$) is represented [28]. Generally, the heuristic algorithm like PSO only requires to check the cost function for guidance of its search and no longer requiring informations about the system. So, in this paper, the Least Mean Square (LMS) of error is considered. The parameters of RNW-PSO are also listed in Table 2.

In the use of Gaussian membership functions we will face with three different cases. 1) Gaussian membership functions with the same means and variances, 2) Gaussian membership functions with the same means and variable variances, and 3) Gaussian membership functions with variable means and the same variances. In [28] an optimal fuzzy-PI controller is designed for a nonlinear delay differential model of glucose-insulin regulation system, and it is shown that Gaussian membership functions with variable means and the same variances have better performance in controlling this system, therefore we applied this idea in design process with the difference that the variances are selected interval.

The specifications of the input and output variables are given in Tables 3 and 4, respectively.

The rulebase has the following nine rules:

- $R^1$ : IF *e* is *E*-$\tilde{N}$ and $\dot{e}$ is *CE*-$\tilde{N}$, THEN ***U*** is $\widetilde{NL}$.
- $R^2$ : IF *e* is *E*-$\tilde{N}$ and $\dot{e}$ is *CE*-$\tilde{Z}$, THEN ***U*** is $\widetilde{NS}$.
- $R^3$ : IF *e* is *E*-$\tilde{N}$ and $\dot{e}$ is *CE*-$\tilde{P}$, THEN ***U*** is $\tilde{Z}$.
- $R^4$ : IF *e* is *E*-$\tilde{Z}$ and $\dot{e}$ is *CE*-$\tilde{N}$, THEN ***U*** is $\widetilde{NS}$.

*Table 3. The Parameters of Input Gaussian Membership Functions*

| Input Variables | Membership Functions | Mean | Variance Interval |
|---|---|---|---|
| Eror (E) | Negative $(E-\tilde{N})$ | $-0.0751$ | [0.0791 0.1881] |
| | Zero $(E-\tilde{Z})$ | $0.0527$ | [0.0791 0.1881] |
| | Positive $(E-\tilde{P})$ | $7.7634\times10^{-4}$ | [0.0791 0.1881] |
| Change of Error (CE) | Negative $(CE-\tilde{N})$ | $-0.1612$ | [0.0070 0.0231] |
| | Zero $(CE-\tilde{Z})$ | $0.0311$ | [0.0070 0.0231] |
| | Positive $(CE-\tilde{P})$ | $0.0215$ | [0.0070 0.0231] |

*Table 4. The Parameters of Output Gaussian Membership Functions*

| **Output Variables** | **Membership Functions** | **Mean** | **Variance Interval** |
|---|---|---|---|
| Control Input (U) | Negative Large $(\widetilde{NL})$ | $-0.0141$ | [0.0122 0.0486] |
| | Negative Small $(\widetilde{NS})$ | $-0.1051$ | [0.0122 0.0486] |
| | Zero $(\tilde{Z})$ | $-0.1681$ | [0.0122 0.0486] |
| | Positive Small (PS) | 0.0549 | [0.0122 0.0486] |
| | Positive Large (PL) | 0.3496 | [0.0122 0.0486] |

- $R^5$ : IF *e* is *E*-$\tilde{Z}$ and $\dot{e}$ is *CE*-$\tilde{Z}$, THEN ***U*** is $\tilde{Z}$.
- $R^6$: IF *e* is *E*-$\tilde{Z}$ and $\dot{e}$ is *CE*-$\tilde{P}$, THEN ***U*** is $\widetilde{PS}$.
- $R^7$ : IF *e* is *E*-$\tilde{P}$ and $\dot{e}$ is *CE*-$\tilde{N}$, THEN ***U*** is $\tilde{Z}$.
- $R^8$ : IF *e* is *E*-$\tilde{P}$ and $\dot{e}$ is *CE*-$\tilde{Z}$, THEN ***U*** is $\widetilde{PS}$.
- $R^9$ : IF *e* is *E*-$\tilde{P}$ and $\dot{e}$ is *CE*-$\tilde{P}$, THEN ***U*** is $\widetilde{PL}$.

The firing intervals and consequents of the nine rules given in Table 5.

From the KM algorithms,$y_l$ and $y_r$ can be computed as follow:

$$y_l = \frac{\overline{f}^1\widetilde{NL}^1 + \underline{f}^2\widetilde{NS}^2 + \underline{f}^3\tilde{Z}^3 + \underline{f}^4\widetilde{NS}^4}{\overline{f}^1 + \underline{f}^2 + \underline{f}^3 + \underline{f}^4 + \underline{f}^5 + \underline{f}^6 + \underline{f}^7 + \underline{f}^8 + \underline{f}^9} + \frac{\underline{f}^5\tilde{Z}^5 + \underline{f}^6\widetilde{PS}^6 + \underline{f}^7\tilde{Z}^7 + \underline{f}^8\widetilde{PS}^8 + \underline{f}^9\widetilde{PL}^9}{\overline{f}^1 + \underline{f}^2 + \underline{f}^3 + \underline{f}^4 + \underline{f}^5 + \underline{f}^6 + \underline{f}^7 + \underline{f}^8 + \underline{f}^9}$$

*Table 5. Firing intervals of the nine rules*

| Rule No.: | Firing Interval | Consequent |
|---|---|---|
| $R_1$ | $[\underline{f}1, \overline{f}1] = [\mu\underline{E-\tilde{N}}(e) \times \mu\underline{CE-\tilde{N}}(\dot{e}), \mu\overline{E-\tilde{N}}(e) \times \mu\overline{E-\tilde{N}}(\dot{e})]$ | $[\underline{\widetilde{NL}}1, \overline{\widetilde{NL}}1]$ |
| $R_2$ | $[\underline{f}2, \overline{f}2] = [\mu\underline{E-\tilde{N}}(e) \times \mu\underline{CE-\tilde{Z}}(\dot{e}), \mu\overline{E-\tilde{N}}(e) \times \mu\overline{CE-\tilde{Z}}(\dot{e})]$ | $[\underline{\widetilde{NS}}2, \overline{\widetilde{NS}}2]$ |
| $R_3$ | $[\underline{f}3, \overline{f}3] = [\mu\underline{E-\tilde{N}}(e)\ \mu\underline{CE-\tilde{P}}(\dot{e}), \mu\overline{E-\tilde{N}}(e) \times \mu\overline{CE-\tilde{P}}(\dot{e})]$ | $[\underline{\tilde{Z}}3, \overline{\tilde{Z}}3]$ |
| $R_4$ | $[\underline{f}4, \overline{f}4] = [\mu\underline{E-\tilde{Z}}(e) \times \mu\underline{CE-\tilde{N}}(\dot{e}), \mu\overline{E-\tilde{Z}}(e) \times \mu\overline{CE-\tilde{N}}(\dot{e})]$ | $[\underline{\widetilde{NS}}4, \overline{\widetilde{NS}}4]$ |
| $R_5$ | $[\underline{f}5, \overline{f}5] = [\mu\underline{E-\tilde{Z}}(e) \times \mu\underline{CE-\tilde{Z}}(\dot{e}), \mu\overline{E-\tilde{Z}}(e) \times \mu\overline{CE-\tilde{Z}}(\dot{e})]$ | $[\underline{\tilde{Z}}5, \overline{\tilde{Z}}5]$ |
| $R_6$ | $[\underline{f}6, \overline{f}6] = [\mu\underline{E-\tilde{Z}}(e) \times \mu\underline{CE-\tilde{P}}(\dot{e}), \mu\overline{E-\tilde{Z}}(e) \times \mu\overline{CE-\tilde{P}}(\dot{e})]$ | $[\underline{\widetilde{PS}}6, \overline{\widetilde{PS}}6]$ |
| $R_7$ | $[\underline{f}7, \overline{f}7] = [\mu\underline{E-\tilde{P}}(e) \times \mu\underline{CE-\tilde{N}}(\dot{e}), \mu\overline{E-\tilde{P}}(e) \times \mu\overline{CE-\tilde{N}}(\dot{e})]$ | $[\underline{\tilde{Z}}7, \overline{\tilde{Z}}7]$ |
| $R_8$ | $[\underline{f}8, \overline{f}8] = [\mu\underline{E-\tilde{P}}(e) \times \mu\underline{CE-\tilde{Z}}(\dot{e}), \mu\overline{E-\tilde{P}}(e) \times \mu\overline{CE-\tilde{Z}}(\dot{e})]$ | $[\underline{\widetilde{PS}}8, \overline{\widetilde{PS}}8]$ |
| $R_9$ | $[\underline{f}9, \overline{f}9] = [\mu\underline{E-\tilde{P}}(e) \times \mu\underline{CE-\tilde{P}}(\dot{e}), \mu\overline{E-\tilde{P}}(e) \times \mu\overline{CE-\tilde{P}}(\dot{e})]$ | $[\underline{\widetilde{PL}}9, \overline{\widetilde{PL}}9]$ |

$$y_r = \frac{\underline{f}^1\overline{\widetilde{NL}}^1 + \underline{f}^2\overline{\widetilde{NS}}^2 + \underline{f}^3\overline{\tilde{Z}}^3 + \underline{f}^4\overline{\widetilde{NS}}^4}{\underline{f}^1 + \underline{f}^2 + \underline{f}^3 + \underline{f}^4 + \underline{f}^5 + \underline{f}^6 + \underline{f}^7 + \underline{f}^8 + \overline{f}^9} + \frac{\underline{f}^5\overline{\tilde{Z}}^5 + \underline{f}^6\overline{\widetilde{PS}}^6 + \underline{f}^7\overline{\tilde{Z}}^7 + \underline{f}^8\overline{\widetilde{PS}}^8 + \overline{f}^9\overline{\widetilde{PL}}^9}{\underline{f}^1 + \underline{f}^2 + \underline{f}^3 + \underline{f}^4 + \underline{f}^5 + \underline{f}^6 + \underline{f}^7 + \underline{f}^8 + \overline{f}^9}$$

Finally, the crisp output of the IT2 FLS, y, can be calculated as follow:

$$y = \frac{y_l + y_r}{2}. \tag{24}$$

## 5 SIMULATIONS AND RESULTS

In order to simulate the proposed controller, MATLAB software is applied. The simulation is run on a personal computer Core 2 Due, 2.8 GHz, 4 Gbytes RAM, under Windows 7. The RNW-PSO optimizes the controller's parameters dynamically. To minimize fitness function, in each iteration, the parameters are randomly chosen by RNW-PSO algorithm. These parameters consist of mean and variance of Gaussian membership functions and PID controller's coefficients. Then the program will be run. In the end of run, the fitness function's value is calculated and is compared with the value calculated in previous iterations. If the new value be better than previous values, the new estimated values for parameters are stored. After completion of iteration loop, RNW-PSO algorithm offers the best answer as an optimal answer. The optimal parameters of PID controller are given in Table 6. The transfer function of the supply air pressure loop under normal circumstances is as follows:

*Table 6. Optimal parameters of PID controller*

| | |
|---|---|
| Proportional Gain - Kp | 1.1814 |
| Derivative Gain - Kd | 0.0473 |
| Integral Gain - Ki | 1.5056 |

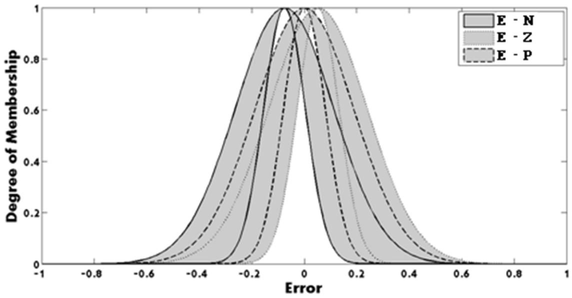


*Fig. 3. Obtained membership functions of input 1*

$$G(s) = \frac{0.81e^{-2s}}{(0.97s+1)(0.1s+1)}, \tag{25}$$

where gain $K = 0.81$, $\tau_1 = 0.97$, $\tau_2 = 0.1$ and dead time $\delta = 2$ sec. For this process weighting parameters are defined $N_e = 0.9$, $N_{\dot{e}} = 5$ and $N_u = 2.5$. Input and output membership functions of designed optimal type-2 fuzzy-PID controller namely error (Input 1), change of error (Input 2), and control input are shown in Figs. 3, 4, and 5 respectively. It can be observed from these Figs that the RNW-PSO has improved the logical sequence of membership functions. For instance, about input 2 the membership function CE-P comes before CE-Z.

In order to evaluate controller performance against the existing uncertainties in parameters of nominal model three different transfer function has been introduced. To investigate this issue the applied transfer functions in [29] is used.

1. when gain $K = 0.81$, $\tau_1 = 0.2$, $\tau_2 = 2$ and dead time

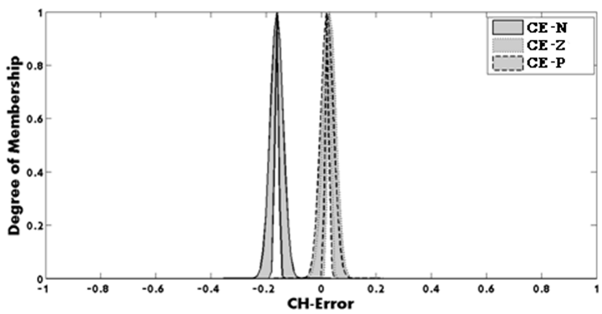


*Fig. 4. Obtained membership functions of input 2*

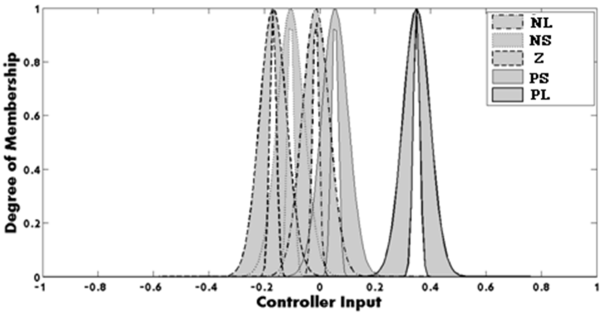


*Fig. 5. Obtained membership functions of Output*

$\delta = 2$ sec., then the transfer function of the supply air pressure loop is as follow

$$G(s) = \frac{0.81e^{-2s}}{(0.97s + 1)(0.1s + 1)} \qquad (26)$$

For this process weighting parameters are defined $N_e = 0.9$, $N_{\dot{e}} = 15$ and $N_u = 0.3$.

2. when gain $K = 1.2$, $\tau_1 = 0.97$, $\tau_2 = 0.1$ and dead time $\delta = 3$ sec., then the transfer function of the supply air pressure loop is as follow

$$G(s) = \frac{1.2e^{-3s}}{(0.97s + 1)(0.1s + 1)} \qquad (27)$$

For this process weighting parameters are defined $N_e = 0.9$, $N_{\dot{e}} = 3$ and $N_u = 1$.

3. when gain $K = 1.2$, $\tau_1 = 0.97$, $\tau_2 = 0.1$ and dead time $\delta = 4$ sec., then the transfer function of the supply air pressure loop is as follow

$$G(s) = \frac{1.2e^{-4s}}{(0.97s + 1)(0.1s + 1)} \qquad (28)$$

For this process weighting parameters are defined $N_e = 0.9$, $N_{\dot{e}} = 3$ and $N_u = 1$.

The Figs. 6-9 and Table 7 are indicated that the supply air pressure loop of HVAC acts satisfactorily both under nominal transfer function and also under existing uncertainties in parameters of model. Table 8 implies that both the rise time and settling time are highly appropriate. Peak overshoots are also demonstrated insignificant when Optimal Type-2 Fuzzy-PID Controller (OT2FPIDC) is applied. The proposed controller in this paper is much less complicated than the existing non-optimal fuzzy controller in [30]. The designed controller in this paper has only 9 rules whereas with these limited rules the design requirements are satisfied. But in [30] for achieving the satisfactory results 49 rules are defined. This fact shows the superiority of the controller in this paper than the controller proposed in [30].

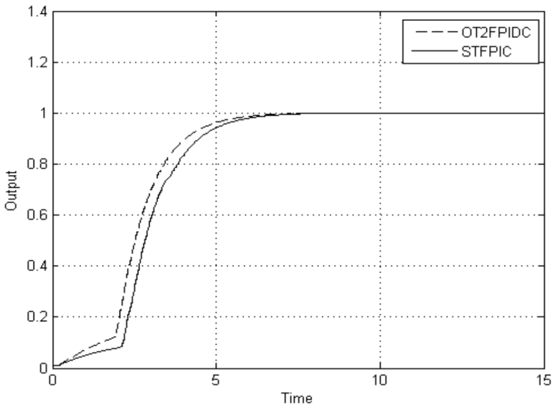


*Fig. 6. Performance of the transfer function given by Eq. (25)*

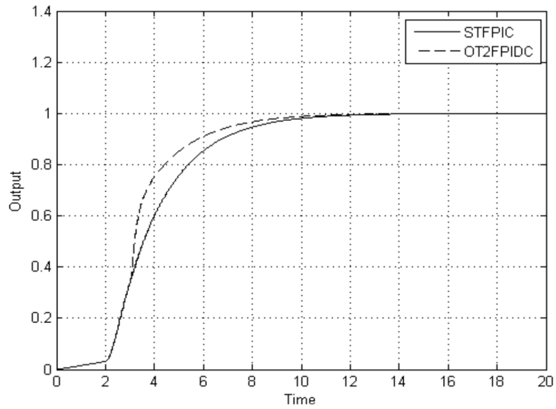


*Fig. 7. Performance of the transfer function given by Eq. (26)*

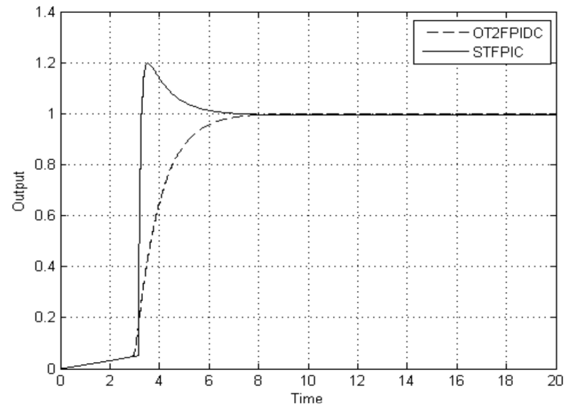


*Fig. 8. Performance of the transfer function given by Eq. (27)*

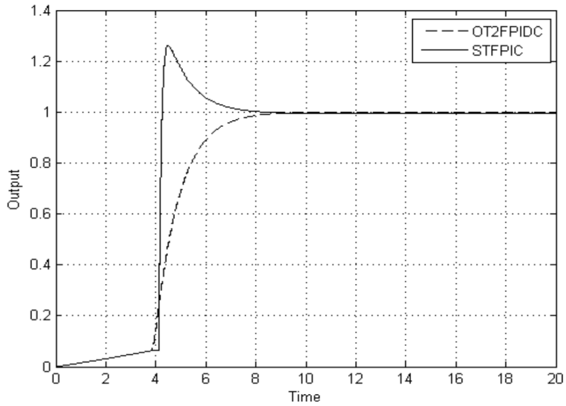


*Fig. 9. Performance of the transfer function given by Eq. (28)*

*Table 7. Performance analysis of OT2FPIDC for different HVAC-Supply Air Pressure Loop*

| Transfer function | $t_r$ sec | $t_s$ sec | $Mp\%$ | $Ess\%$ |
|---|---|---|---|---|
| $G(s)=\frac{0.81e^{-2s}}{(0.97s+1)(0.1s+1)}$ | 2.58 | 4.74 | 0.00 | 0.12 |
| $G(s)=\frac{0.81e^{-2s}}{(0.2s+1)(2s+1)}$ | 4.44 | 8.17 | 0.00 | 0.01 |
| $G(s)=\frac{1.2e^{-3s}}{(0.97s+1)(0.1s+1)}$ | 2.16 | 5.88 | 0.00 | 0.08 |
| $G(s)=\frac{1.2e^{-4s}}{(0.97s+1)(0.1s+1)}$ | 2.26 | 6.75 | 0.00 | 0.06 |

*Table 8. Comparison between performance of PID, ANF, STFPIC, and OT2FPIDC under normal condition and under existing uncertainties in parameters of model*

| Transfer Function | Controller Type | Mp % | $t_s$ sec |
|---|---|---|---|
| $G(s)=\frac{0.81e^{-2s}}{(0.97s+1)(0.1s+1)}$ | PID | 3.9 | 6.7 |
| | ANF | 3.5 | 7.5 |
| | STFPIC | 0.00 | 3.6 |
| | OT2FPIDC | 0.00 | 4.74 |
| $G(s)=\frac{0.81e^{-2s}}{(0.2s+1)(2s+1)}$ | PID | 17.9 | 16.2 |
| | ANF | 0.9 | 10.6 |
| | STFPIC | 0.088 | 8.9 |
| | OT2FPIDC | 0.00 | 8.17 |
| $G(s)=\frac{1.2e^{-3s}}{(0.97s+1)(0.1s+1)}$ | PID | 63 | 37 |
| | ANF | 56 | 19 |
| | STFPIC | 17.6 | 6 |
| | OT2FPIDC | 0.00 | 5.88 |
| $G(s)=\frac{1.2e^{-4s}}{(0.97s+1)(0.1s+1)}$ | PID | 100 | $\geq 120$ |
| | ANF | 59 | 32 |
| | STFPIC | 25 | 6.9 |
| | OT2FPIDC | 0.00 | 6.75 |

## 6 CONCLUSION

A novel optimal type-2 fuzzy-PID controller has been suggested for temperature regulation of HCAC system. Simulation results indicate that the new optimal fuzzy-PID controller has faster response, smaller overshoot and higher accuracy than PID, ANF, and STFPIC under normal condition and under existing uncertainties in parameters of model. The new optimal type-2 fuzzy-PID controller can be extensively applied in the HVAC industry.

## REFERENCES


[1] Radakovic, Z. R., Milosevic, V. M., and Radakovic, S. B., Application of temperature fuzzy controller in an indirect resistance furnace, *Applied Energy*, 2002, vol. 73, pp. 167-182

[2] Nguyen, H. T., Prasad, N. R., Walker, C. L., and Walker, E. A., *A First Course in Fuzzy and Neural Control*, USA: Chapman & Hall/ CRC, 2003.

[3] Z. Kovačić, Bogdan, S., Crnošija, P., An automated fuzzy controller design for high-order systems, Automatika-Ltd., vol. 2, no. 3, pp. 37-40, 1996.

[4] Qiao, W., Masaharu Mizumoto, *Fuzzy sets and systems*, 1996, vol. 78, pp. 23-35.

[5] Pedrycz, W., and de Oliveira, J. V., Optimization of fuzzy models, *IEEE Trans. Syst., Man, Cyber.*, 1996, vol. 26, no.4, pp. 627 – 636.

[6] Hyun-Joon, C., Kwang-Bo, C., Bo-Hyeun, W., Fuzzy-PID hybrid control: automatic rule generation using genetic algorithm, *Fuzzy sets and systems*, 1997, vol. 92, no. 3, pp. 305-316.

[7] Alcala, R., Casillas, J., Cordon, O., Gonzalez, A., and Herrera, F., A genetic rule weighting and selection process for fuzzy control of heating, ventilation and air conditioning systems, *Engineering application of Artificial Intelligence*, 2005, vol. 28, pp. 279 – 296.

[8] Qiang, X., Wen-Jian, C., and Ming, H., A practical decentralized PID auto-tuning method for TITO systems under closed –loop control, *International Journal of Innovative Computing, Information and Control*, 2006, vol. 2, no. 2, pp. 305-322

[9] Qing-Gao, W., Chang-Chieh, H., Yong, Z., and Qiang, B., Multivariable Controller Auto-Tuning with its Application in HVAC Systems, *Proc. of the American Control Conf.*, California, 1999, vol. 6, pp. 4353 – 4357

[10] Mudi, R.K., and Pal, N.R., A robust self-tuning scheme for PI and PD type fuzzy controllers, *IEEE trans. on fuzzy sys.*, 1997, vol. 7, no. 1, pp. 2-16.

[11] Q. Bi, Cai, W.J., Wang, Q.G., Hang, C.C., Lee, E.L., Sun, Y., Liu, K.D., Zhang, Y., and Zou, B., Advanced controller auto-tuning and its application in HVAC systems, Control Engineering Practice, vol. 8, no. 6, pp. 633-644. 2000.

[12] H. Hagras, “A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots,” *IEEE Trans. on Fuzzy Systems*, vol. 12, pp. 524–539, 2004.

[13] Hagras, H.; “Type-2 FLCs: A new generation of fuzzy controllers,” *IEEE Computational Intelligence Magazine*, vol. 2, no. 1, pp. 30–43, 2007 *Intelligence Magazine*, vol. 2, no. 1, pp. 30–43, 2007.

[14] J. M. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*. Upper Saddle River, NJ: Prentice-Hall, 2001.

[15] Dongrui Wu, Woei Wan Tan, “Genetic learning and performance evaluation of type-2 fuzzy logic controllers,” *Engineering Applications of Artificial Intelligence*, vol. 19, no. 8, pp. 829–841, 2006.

[16] L. A. Zadeh , The concept of a linguistic variable and its application to approximate reasoning-1, *Information Sciences*, vol. 8, pp. 199–249, 1975.

[17] N. N. Karnik and J. M. Mendel, “Centroid of a type-2 fuzzy set,” *Information Sciences*, vol. 132, pp. 195–220, 2001.

[18] J. M. Mendel and D. Wu, *Perceptual Computing: Aiding People in Making Subjective Judgments*. Hoboken, NJ: Wiley-IEEE Press, 2010.

[19] D. Wu and J. M. Mendel, “Enhanced Karnik-Mendel Algorithms,” *IEEE Trans. on Fuzzy Systems*, vol. 17, no. 4, pp. 923–934, 2009.

[20] Dongrui, U.; Maowen Nie; Comparison and practical implementation of type-reduction algorithms for type-2 fuzzy sets and systems, 2011 IEEE International Conference on Fuzzy Systems June 27-30, 2011, Taipei, Taiwan.

[21] Modares, H., Alfi, A., and Naghibi Sistani, M. B., Parameter Estimation of Bilinear Systems Based on an Adaptive Particle Swarm Optimization, *Eng. Appl. Artifi. Intell.*, 2010, vol. 23, pp.1105-1111.

[22] Zhang, L., Yu, H., Hu, S., A New Approach to Improve Particle Swarm Optimization, *Proc. of the international conf. on Genetic and evolutionary computation*, 2003, 134-139.

[23] Shi, Y., Eberhart, R., A Modified Particle Swarm Optimizer, in *Proc. of the IEEE Conf. On Evolutionary Computation*, Singapore, 1998, pp. 69-73.

[24] Eberhart, R.C., and Shi, Y., Tracking and optimizing dynamic systems with particle swarms, in *Proc. IEEE Congr. Evolutionary Computation* , Seoul, Korea, 2001, pp. 94–97.

[25] J. Kennedy. The particle swarm: social adaptation of knowledge. Proc. IEEE International Conference on Evolutionary Computation (Indianapolis, Indiana), IEEE Service Center, Piscataway, NJ 303-308.

[26] P. J. Angeline. Using selection to improve particle swarm optimization. IEEE International Conference on Evolutionary Computation, Anchor age, Alaska, May4-9.

[27] Y. Shi and R. Eberhart. Fuzzy adaptive particle swarm optimization. 2001. Proc. Of the 2001 Congress on Evolutionary Computation, vol. 1 101-106

[28] Jian, W., and Wenjian, C., Development of an adaptive neuro-fuzzy method for supply air pressure control in HVAC system, *Syst., Man, Cybern., IEEE*, 2000.

[29] Al-Fandi, M., Jaradat, M.A.K., and Sardahi, Y., Optimal PI-fuzzy logic controller of glucose concentration using genetic algoritm, *International Journal of Knowledge-based and Intelligent Engineering Systems*, 2011, vol. 15, pp. 99-117.

[30] A.K., Pal, and Mudi, R.K., Self-Tuning Fuzzy PI Controller and its Application to HVAC Systems, *IJCC*, 2008, vol. 6, no. 1, pp. 25-30.


**Mohammad Hassan Khooban** received the B.Sc. degree in Control Engineering from the Fars Science and Research Branch, in 2010 and M.Sc. degrees in Control Engineering from Islamic Azad University, Iran, in 2012. His employment experience included working at Sarvestan Branch-Islamic Azad University of Iran, Advisor to Iranian Space Agency, Iranian Space Center, Mechanic Institute, Shiraz, Iran, since now. His research interests include Heuristic Optimization, Nonlinar Robust Control, Fuzzy Logic and Power Systems.

**Davood Nazari Maryam Abadi** received the B.Sc, in Elec. Eng. from the Sadjad Institute of Higher Education Mashhad, Iran. He received the M.Sc, in Elec. Eng. from the Azad University of Iran, Garmsar branch, Currently. His research interests are in Nonlinear control, Power system stability studies, Fuzzy systems and Artificial intelligence.

**Alireza Alfi** has received his B.Sc. degree from Ferdowsi University of Mashhad, Mashhad, Iran, in 2000, and his M.Sc. and Ph.D. degrees from Iran University of Technology, Tehran, Iran, in 2002 and 2007, all in Electrical Engineering. He joined Shahrood University of Technology in 2008, where he is currently an Assistant Professor of Electrical Engineering. His research interests include heuristic optimization, control theory, time delay systems, fuzzy logic and chaotic systems.

**Mehdi Siahi** recevied the B.Sc. degree in Elecrtical Engineering from the Yazd University, Yazd IRAN in 2001 and M.Sc. degree in Control Engineering from the Shahrood University of Thechnology, Shahrood, Iran, in 2003. He obtained the Ph.D. degree in cotrol engineering from shahrood Universtity of Thechnology, Shahrood, Iran, in 2008. He is now an assistant professor and has been with Faculty of Electrical Engineering, Islamic Azad University of garmsar Branch, Iran from 2004. His current research is on fault Tolerant Control systems, Robust Control and Nonlinear systems

**AUTHORS' ADDRESSES**

**Mohammad Hassan khooban, M.Sc.**
**Davood Nazari Maryam Abadi, M.Sc.**
**Asst. Prof. Mehdi Siahi. Ph.D.**
**Department of Electrical and Robotic Engineering,**
**Islamic Azad University of Garmsar Branch, Garmsart, Iran,**
**email: mhkhoban@gmail.com, d.nazari@ieee.org, mahdi_siahi@yahoo.com**

**Asst. Prof. Alireza Alfi, Ph.D.**
**Faculty of Electrical and Robotic Engineering,**
**Shahrood University of Technology, Shahrood, Iran,**
**email: a_alfi@shahroodut.ac.ir**

Received: 2012-03-18
Accepted: 2012-06-21