IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.8, Issue.3, Mar 2020 E-ISSN: 2347-2693

Comparative Performance Study of Optimal Interval Type-2 Fuzzy PID Controllers with Practical System

Ritu Rani De (Maity)[1*], Rajani K. Mudi [2], Chanchal Dey[3]

[1] Department of Instrumentation and Electronics Engineering, Jadavpur University, Kolkata, India
[2] Department of Instrumentation and Electronics Engineering, Jadavpur University, Kolkata, India
[3] Instrumentation Engineering, Department of Applied Physics, University of Calcutta, Kolkata, India

DOI: https://doi.org/10.26438/ijcse/v8i3.16 | Available online at: www.ijcseonline.org

Received: 08/Mar/2020, Accepted: 12/Mar/2020, Published: 30/Mar/2020

Abstract—In this paper, the input and output scaling factors of the type-2 fuzzy PID Controller (IT2-FPID) are determined using three different optimization algorithms (Cuckoo search (CS), Particle swarm optimization (PSO), and Bee colony algorithm (BCA)) for a first-order integrating plus dead time (FOIPD) model. A comparative performance study is made for these three optimization algorithms in terms of various transient performance indices. The comparative analysis on the experimental results reveals that BCA based optimal IT2-FPID shows better performance on a simulation model whereas CS based optimal IT2-FPID is found to be superior for practical system over other algorithms.

Keywords— Particle swarm optimization(PSO), Cuckoo search algorithm (CS), Bee colony algorithm(BCA), Interval type-2 fuzzy controller.

I. INTRODUCTION

A conventional Proportional Integral and Derivative (PID) controller [1-3] with properly tuned coefficients may be suited for a fixed operating condition but for varying control environments like load disturbances, system non-linearity and change of plant parameters its performance may deteriorate. Literature survey shows that FLC yields superior results than that obtained by conventional controller [2, 5]. The type-1 FLC [4,5] follows expertise knowledge into an automatic control strategy and thus do not require any detailed mathematical model [6]. However, type-1 FLC shows poor performance compared to type-2 FLC when there is more uncertainties in measurement and structural uncertainties in the model [6-9]. Type-2 fuzzy sets [10] are characterized by membership structures that are 3-dimensional and include a foot-print of uncertainty (FOU). This third dimension and concept of FOU provides type-2 fuzzy to directly handle model, linguistic and numerical uncertainties [11-13, 6] associated with inputs and outputs of the FLC. In this study, we have focused on the most commonly used double input type FPID [14, 15] controller structure shown in Fig.1.
From the literature survey, it is found that many researchers have put their efforts towards designing enhanced type-2 FLCs for different applications [6-11]. Researchers have also explored various optimization techniques [16-18] for further fine tuning the controller parameters. The proposed controller is an optimal interval type-2 fuzzy PID (IT2-FPID) controller whose input-output scaling factors are determined by optimization algorithms to achieve a superior performance compared to a simple interval type-2 fuzzy controller. Here, we have used some of the bio-inspired algorithms like the particle swarm optimization (PSO) [19], cuckoo search (CS) [21] and bee colony algorithm (BCA) [20] for the optimization of the input and output scaling factors of IT2-FPID.
In section II, we will focus on the general structure of Interval Type-2 FPID controller. Section III provides brief discussions on various optimization algorithms. Section IV mentions the optimization process. Section V presents simulation and experimental results. Finally the article ends with a conclusion section.

Overview of an Interval Fuzzy Logic Controllers

Literature survey shows that most of the researches conducted are concentrated only on interval type-2 fuzzy sets and all points in the FOU having unity secondary membership grade [10]. In Type-2 fuzzy sets all the third dimensional values are equal to one. The use of interval type-2 FLC reduces computational complexity which is the prominent drawback of Type-2 FLC compared to Type-1 FLC. The internal information and design strategies are elaborated in the literature [11-13].
In literature there are different types of Fuzzy–PID Controllers [14, 15]. In this paper, we have considered the very effective structure using a two-dimensional linear rule base with simple triangular membership functions (MFs) [13] which combines Mamdani fuzzy proportional Integral (PI) and fuzzy proportional derivative (PD) controllers as shown in Fig.-1. For simplicity and ease of implementation, we use triangular MFs for error (e), change of error (Δe), and control action (u) as shown in Fig.2 (with 5 fuzzy sets). Observe that Fig.2 shows fuzzy partitions for the same universe of discourse of input (e, Δe) and output (u) linguistic variables. The controller

output u is determined by rules of the form: IF e is E and Δe is ΔE THEN u is U,

Where, $e(k) = r - y(k)$ (4)

And $\Delta e(k) = e(k) - e(k-1)$ (5)

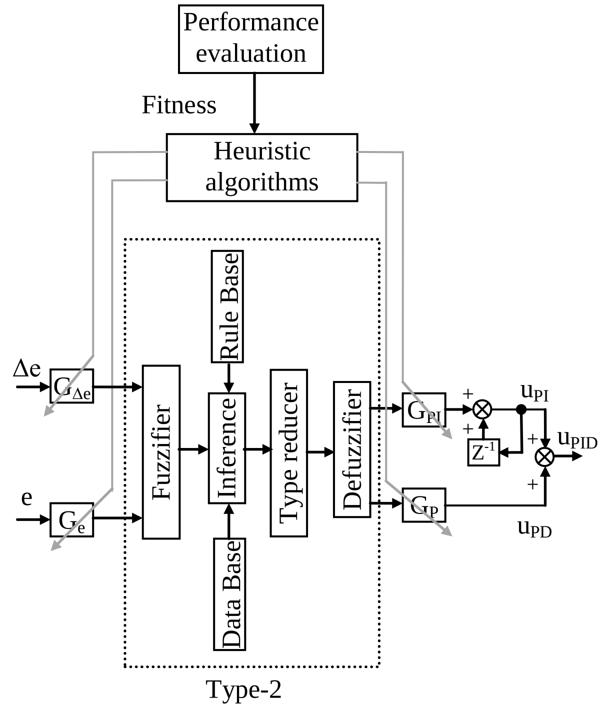

Figure 1 Block diagram of T2-FPID

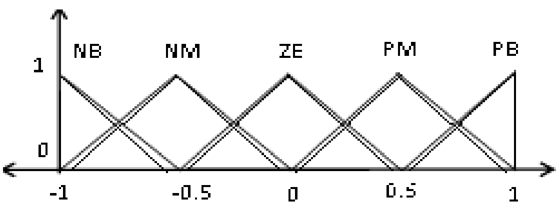

Figure 2. Membership functions of type-2 fuzzy sets for e, Δe, u, Δu

Table 1: Fuzzy control rule-base

Δe/e	**NB**	**NM**	**ZE**	**PM**	**PB**
NB	NB	NB	NB	NM	ZE
NM	NB	NB	NM	ZE	PM
ZE	NB	NM	ZE	PM	PB
PM	NM	ZE	PM	PB	PB
PB	ZE	PM	PB	PB	PB

Usually, the input variables $(\boldsymbol{e})$ and $\Delta(\boldsymbol{e})$ are partitioned by $\boldsymbol{N}$ number of membership functions, and assuming that for each possible combination of the N sets there is a rule for this two domain inputs $\breve{\boldsymbol{X}}_{\mathbf{1}}, \breve{\boldsymbol{X}}_{\mathbf{2}}$, therefore, the rule-base comprises of $\boldsymbol{N}^{\mathbf{2}}$ rules. The controller output u is determined by the n^{th} rule,

If e is $\breve{X}_1^i$ and Δe *is* $\breve{X}_2^j$ then u^n *is* U^n (i, j=1,2…….N), where U^n is a type-2 fuzzy set in the interval $[\underline{u}^n, \overline{u}^n]$. Next we compute the membership of input variable $\breve{X}_1^i, \breve{X}_2^j$ as [$\underline{\mu}_{\breve{X}_1^i}$ (e), $\bar{\mu}_{\breve{X}_1^i}(e)$] and $[\underline{\mu}_{\breve{X}_2^j}(\Delta e), \bar{\mu}_{\breve{X}_2^j}(\Delta e)]$ for (i, j=1,2…….N). $\underline{\mu}_{\breve{X}_1^i}$ (e), $\bar{\mu}_{\breve{X}_1^i}(e)$ are the upper and lower membership grades of the interval firing set $\mu_{\breve{X}_1^i}(e)$. Similarly, $\underline{\mu}_{\breve{X}_2^j}(\Delta e)$ and $\bar{\mu}_{\breve{X}_2^j}(\Delta e)$ are the lower and upper membership grades of $\mu_{\breve{X}_2^j}(\Delta e)$.

Next, we compute the firing interval of the n^{th} rule,

$$F^n(e, \Delta e) = \left[\underline{f}^n(e, \Delta e), \overline{f}^n(e, \Delta e)\right],$$

$$where\ n = 1, \dots \dots \dots N. \quad (6)$$

$$= [\underline{\mu}_{\breve{X}_1^i}(e) \times \underline{\mu}_{\breve{X}_2^j}(\Delta e), \bar{\mu}_{\breve{X}_1^i}(e) \times \bar{\mu}_{\breve{X}_2^j}(\Delta e)],$$

$$i = 1,2 \dots \dots \dots . N\ and\ j = 1,2 \dots \dots \dots \dots N \quad (7)$$

With the $F^n(e, \Delta e)$ and the corresponding rule consequents we perform the centre–of–sets type reduction (COS),

$$U_{COS(e,\Delta e)} = \bigcup_{\substack{f^n \in F^n(e,\Delta e) \\ u^n \in U^n}} \frac{\sum_{n=1}^{N} f^n u^n}{\sum_{n=1}^{N} f^n}$$

$$= [u_l, u_r] \quad (8)$$

$$u_l = \frac{\sum_{n=1}^{L} \underline{u}^n \overline{f}^n + \sum_{n=L+1}^{N} \underline{u}^n \underline{f}^n}{\sum_{n=1}^{L} \overline{f}^n + \sum_{n=L+1}^{N} \underline{f}^n}; \ u_r = \frac{\sum_{n=1}^{R} \overline{u}^n \underline{f}^n + \sum_{n=R+1}^{N} \overline{u}^n \overline{f}^n}{\sum_{n=1}^{R} \underline{f}^n + \sum_{n=R+1}^{N} \overline{f}^n} \quad (9)$$

here, the switch points L and R are determined by Karnik and Mendel (KM) Algorithm such that $\underline{u}^L \le u_l \le \underline{u}^{L+1}$ *and* $\overline{u}^R \le u_r \le \overline{u}^{R+1}$. Finally, computation of the defuzzified output,

$$U = \frac{u_l + u_r}{2}, \quad (10$$

The rule-base corresponding to the 5 fuzzy partitions of Fig.2 consisting of 25 rules for computing u is shown in Table-1, which is a commonly used rule-base designed in sliding mode principle [23]. From Fig.1, we find that the final controller output U is obtained by the relation:

$$U = G_{PD}.u + G_{PI}.\Delta u \quad (11)$$

In this IT2-FPID of Fuzzy PID [22] the values of the actual inputs e and Δe are mapped to the interval [-1, 1] by the input SFs, G_e and $G_{\Delta e}$ respectively. The defuzzified output u_N is translated into the actual output u by the output SF, G_{PI} and G_{PD}. Initial settings with suitable values of G_e, $G_{\Delta e}$, G_{PI} and G_{PD} are found based on the knowledge about the process to be controlled, and sometimes through trial. Here, we have used some heuristic optimization algorithms to find the optimal values of the two input scaling factors for error G_e, change of error $G_{\Delta e}$ and two output scaling factor G_{PI}, and G_{PD}.

III. INTRODUCTION TO PSO, CS, BCA

A. Particle Swarm Optimization (PSO)

PSO technique was inspired by fish and bird swarm intelligence and was originally reported by Kennedy et al. [19]. The solution to an optimization problem under investigation is presented here by the position of a particle vector. Each particle has a position as well as a current velocity, which represents the magnitude and direction towards achieving a better solution. In each iterative cycle the fitness of each particle is evaluated based on the objective function $f(x)$. At every iteration, velocity ($v(t)$) for each particle is updated based on its current velocity, and the local as well as global swarm information as given by Eqn. 12(a). Subsequently, new position for each particle is updated using Eqn. 12(b).

$$v(t+1) = w.v(t) + (c_1.r_1(p(t) - x(t))) + (c_2.r_2(g(t) - x(t))) \tag{12a}$$

$$x(t+1) = x(t) + v(t+1) \tag{12b}$$

In Eqn. 12(a), present velocity is obtained based on the weighted velocity of the previous sampling instant. The other parameters of Eqn. 12(a) are the local ($p(t)$) and global ($g(t)$) positional information of the particle. Here, c_1 and c_2 are the cognitive components related to the local and global weight respectively and both of them are considered to be two (c_1 = c_2 = 2). In Eqn. 12(a), r_1 and r_2 are the random variables and their values are in the range [0, 1] and w is the inertia weight whose value decreases from 0.9 to 0.4 with the number of iteration. Such technique of choosing inertia weight offers better performance compared to random choice of inertia weight. Based on some predefined criteria related to the convergence of fitness value defined as the reciprocal of the objective function ($f(x)$ optimal particle position vector is achieved after a number of iterations.

B. Cuckoo Search Algorithm (CS)

Cuckoo Search (CS) [21] is one of the nature-inspired metaheuristic algorithms, developed by X in-She Yang and Suash Deb. CS algorithm includes Lévy flight rather than simple isotropic random walks. This makes CS potentially more efficient than GA, PSO algorithms. The general idealized rules taken are that each Cuckoo lays one egg at a time and dumps it in a random nest. Secondly, the best nests with high quality will be transferred to the next generation. Thirdly, the available host nest number remains fixed, and the eggs laid is discovered by the host bird with a probability (ρ_a) within 0 to 1. In such case, the host bird (η) can determine either to get rid of the egg, or simply abandon the nest and build a new nest (with new random solutions). For minimization problem, the fitness function is reciprocal of the objective function.

The Lévy flight method used while generating a new solution for i^{th} cuckoo is given by Eqn. 13.

$$x_i(t+1) = x_i(t) + \alpha \oplus L\acute{e}vy(\lambda) \tag{13}$$

Here, Lévy flight (1< λ ≤3) is a random walk phenomenon while the random step length is drawn from a Lévy's distribution, which has an infinite variance with an infinite mean. For this reason CS has the global convergence property more than compared to PSO. PSO may converge prematurely to a local optimum, and CS can converge to global optimum. In (13), x_i is the chosen nest by the i^{th} cuckoo and the value of step size (α) is considered to be unity.

C. Bee Colony algorithm (BCA)

The Artificial Bee Colony algorithm (BCA) was proposed by Karaboga in 2005 for real-parameter optimization. ABC is an optimization algorithm which simulates the foraging behaviour of a bee colony [20]. The optimization algorithm is based on the following characteristics observed in nature (von Frisch, 1976): (i) a bees' colony can extend itself over in multiple directions and over long distances (more than 10 km) to avail a large number of food sources, and (ii) have the capacity of memorization, learning and transmission of information in colony, and thus forming the swarm intelligence.

In ABC algorithm for the minimal model of swarm-intelligent forage selection in a honey bee colony consists of three kinds of bees: employed bees(n_e), onlooker bees (n_o) and scout bees (n_s). Half of the colony consists of employed bees, and the other half includes onlooker bees. The responsibility of the employed bees are to exploit the nectar sources explored before and convey information to the waiting onlooker bees in the hive about the quality of the food source sites which they are exploiting. Based on the shared information of the employed bee, the onlooker bees wait in the hive and decide on a food source to exploit. If the source is abandoned, she becomes a Scout and starts to randomly search the environment in order to find a new food source.

The above algorithms have been used for proper designing of the Mamdani-type Interval type-2 fuzzy logic controllers for FOIPD processes. In this paper, we determine the optimal values of the input-output scaling factors, the primary parameters of the type-2 fuzzy controllers which influence immensely in the system performance.

IV. OPTIMIZATION OF PARAMETERS

In order to visualize the effect of optimization of the scaling factors of the interval type-2 fuzzy PID controller both simulation and real-time experimentation have been done. The improvement of the IT2-FPID without optimization and with optimization are demonstrated in terms of different performance indices – %OS (percentage overshoot), t_r (rise time), t_s (settling time), IAE (integral absolute error), ITAE (integral time absolute error). Noise sensitivity of the IT2-FPID and optimal IT2-FPID is also evaluated in presence of measurement noise.

Here, for finding the optimal values of the four parameters G_e , $G_{\Delta e}$, G_{PI}, and G_{PD} of the interval type-2 fuzzy controller, the above mentioned optimization algorithms CS, PSO and BCA [17-20] have been used. To achieve an optimal solution for set point tracking and load rejection phases the following objective function $f(x)$ is considered involving absolute value of *overshoot* (%*OS*), *settling time* (t_s), and *integral absolute error* (*IAE*):

$$Objective\ fn. = abs(\%OS) + t_s + IAE \qquad (14)$$

Each optimization algorithm is carried for 100 iterations and executed for 20 runs by proper setting of the parameters (as given in Table-2).

Table 2 Tuning values of the parameters for PSO, CS and BCA

Algorithm	Parameter	Value
PSO	c_1, c_2, population, Max. Iteration	2, 2, 50, 100
CS	ρ_a, nest, Max. Iteration	0.25, 50,100
BCA	Colony size n_e, n_o, n_s, Limit	50, 50% of the colony, 50% of the colony, 1, 2.

The optimization algorithms have been used to achieve the minimum value of the objective function which is based on the closed loop performance using the IT2-FPID for FOIPD system. With the optimal values of $G_e, G_{\Delta e}, G_{PI}$, and G_{PD}, (using the above mentioned optimization algorithms) performance of the type-2 fuzzy controller for FOIPD process is verified on i) a simulation model and ii) an experimental set up QUBE-Servo-2 by Quanser, Canada. The optimal results of PSO, BCA and CS are given in Table-3 and 4. The results in Table-3 and Table-4 show that the execution time CS algorithm for 100 iterations is approximately 60% less and also the fitness value is least compared to PSO and BCA. The detail experimental results are discussed in the following section.

Table 3:Optimization results of parameters $G_e, G_{\Delta e}, G_{PI}$ and G_{PD} for model in (16)

Algorithm used	G_e	$G_{\Delta e}$	G_{PI}	G_{PD}	Fitness Value	Execution time(s)
CS	0.517	27.92	0.022	0.129	18.903	70.84
PSO	0.500	20.00	0.030	0.010	11.709	185.7
BCA	1.277	31.71	0.083	0.485	14.374	173.9

Table 4: Optimization results of parameters $G_e, G_{\Delta e}, G_{PI}$ and G_{PD} for QUBE-Servo-2

Algorithm used	G_e	$G_{\Delta e}$	G_{PI}	G_{PD}	Fitness Value	Execution time(s)
CS	0.571	27.27	0.014	0.447	0.376	147.97
PSO	0.905	46.70	0.100	0.010	0.455	286.91
BCA	0.443	49.43	0.016	0.275	0.467	677.11

V. RESULTS AND DISCUSSION

A. Simulation study

For the simulation study, the following first-order integrating plus dead time (FOIPD) model is considered:

$$G_p(s) = \frac{K\, e^{-\theta_d s}}{s(\tau s+1)} \qquad (15)$$

Here, we consider time constant, τ=1 s, dead time $\theta_d = 0.1\ s,$ and open loop gain, K=1. Performance of IT2-FPID is compared with PSO-IT2-FPID, CS-IT2-FPID, BCA-IT2-FPID. The fitness value, execution time, the values of G_e , $G_{\Delta e}, G_{PI}$, and G_{PD} after repeated runs are given in Table-3. Robustness of the IT2-FPID is also evaluated for 50% increased value of dead time (i.e. $\theta_d = 0.15\ s$), along with the optimal IT2-FPID. Responses for the nominal model are given in Fig.3. Performance indices for the nominal and perturbed models (with 50% increased dead time) are given in Table-5. The performance indices are found to be improved for CS-IT2-FPID, PSO-IT2-FPID, BCA-IT2-FPID, compared to IT2-FPID. Similarly, the responses with noise power 0.02 are depicted in Fig.4 and performance indices in Table-6. Here, BCA-IT2-FPID shows lower overshoot, settling time and IAE, ITAE without noise and with noise compared to all other optimized controller. The simulation results show that BCA-IT2-FPID gives better performance than other algorithms. Next, the experimental results are taken in QUBE Servo-2.

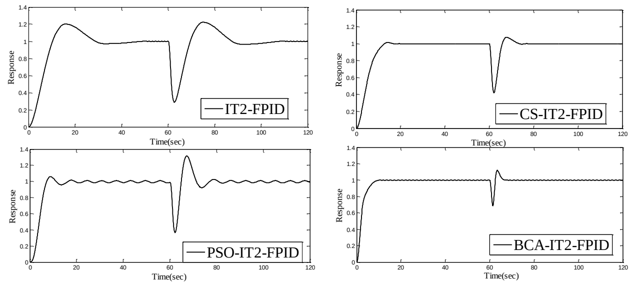

Figure 3. Responses of IT2-FPID using different optimizations for Model in (16) with dead time 0.15sec.

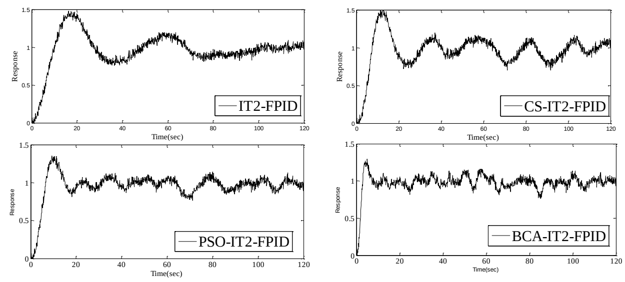

Figure 4. Responses of IT2-FPID using different optimization for Model in (16) with noise

Table 5 Comparison performances for Model in (16)

θ_d	Controller	$\%OS$	$t_s(s)$	IAE	ITAE
0.1	IT2-FPID	20.1	46.8	14.6	531.9
	CS-IT2-FPID	1.4	17.1	7.26	151.5
	PSO-IT2-FPID	19.3	5.5	8.09	260.0

	BCA-IT2-FPID	0.2	10	3.30	48.48
0.15	IT2-FPID	20.2	46.3	14.7	535.2
	CS-IT2-FPID	1.5	19.2	7.27	152
	PSO-IT2-FPID	1.4	23.4	8.39	279.6
	BCA-IT2-FPID	0.02	14.0	3.41	55.92

Table 6 Performance analysis for Model in (16) with noise

θ_d	Controller	IAE	ITAE
0.1	IT2-FPID	18.82	625.6
	CS-IT2-FPID	17.92	688.9
	PSO-IT2-FPID	11.89	424.1
	BCA-IT2-FPID	8.404	374.4
0.15	IT2-FPID	16.99	534
	CS-IT2-FPID	13.59	536.8
	PSO-IT2-FPID	12.07	434.1
	BCA-IT2-FPID	8.503	378.8

B. Experimental results

Efficiency of the IT2-FPID and the optimized IT2-FPID is experimentally verified on Quanser QUBE Servo-2 (Quanser, Canada) position control system. The snap shot of the actual experimental set up of QUBE-Servo-2 is given in Fig.5. For this system the desired angular position (set value) is given through PC. QUBE-Servo-2 position control are identified as first-order integrating plus time delay (FOIPD) model by bump test and derived as

$$G_{QUBE}(s) = \frac{22.7}{s(0.158s+1)} \quad (16)$$

Figure 5. Experimental set-up of QUBE-Servo 2

Using QUARC and MATLAB-Simulink, the reported controllers are implemented and their performances are evaluated. Performance study for set point tracking and load regulation with IT2-FPID, CS-IT2-FPID, PSO-IT2-FPID and BCA-IT2-FPID are shown in Fig.6 for QUBE Servo-2. The comparative study for the controllers is depicted in Table- 7.

Performance of QUBE-Servo 2 is also studied in presence of measurement noise for IT2-FPID and with optimal IT2-FPID. Here, noise power applied for QUBE Servo-2 with sample time 0.01 is 0.02. The related responses with measurement noise are given in Fig.7 for QUBE Servo-2 and analysis of performance indices in Table-8.

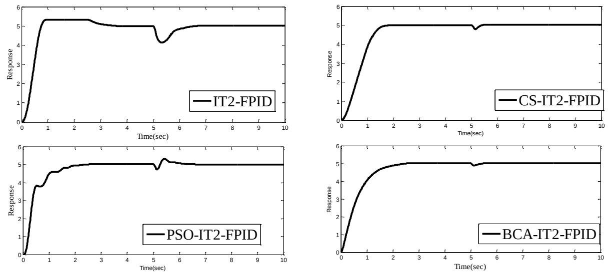

Figure 6. Responses of IT2-FPID using different optimization on QUBE-Servo-2.

Based on the response related performance indices it is clear that applying optimization algorithms in determining the value of $G_e, G_{\Delta e}, G_{PI}$, and G_{PD} of the IT2-FPID there is an immense improvement in close-loop response under model uncertainty and in presence of measurement noise. Moreover, simulation result's comparative study shows BCA optimization algorithm is more superior to other two heuristic algorithms, PSO and CS applied for the optimization of the scaling factors of IT2-FPID. Whereas the experimental results show a better performance of CS-IT2-FPID in terms of maximum overshoot, settling time, ITAE than other algorithms. The performance analysis makes it clear that BCA-IT2-FPID gives better simulation performance whereas, CS-IT2-FPID shows superior performance for practical system where there may be model uncertainties, nonlinearity, load disturbances and measurement errors.

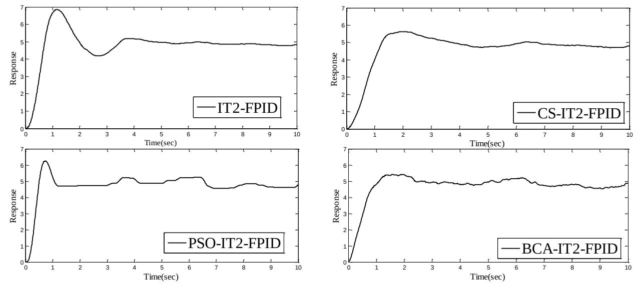

Figure 7. Responses of IT2-FPID using different optimizations on QUBE Servo-2 with noise (noise factor 0.02)

Table- 7: Comparison performances for QUBE-Servo-2

Controller	$\%OS$	$t_s(s)$	IAE	ITAE
IT2-FPID	6.7	3.62	0.6847	1.017
CS-IT2-FPID	0.2	1.84	0.7553	0.4804
PSO-IT2-FPID	1.4	2.52	0.5413	0.4703
BCA-IT2-FPID	0.8	2.73	0.6419	0.3827

Table-8: Performance analysis for QUBE-Servo-2 with noise

Controller	IAE	ITAE
IT2-FPID	1.074	2.139
CS-IT2-FPID	1.101	2.290
PSO-IT2-FPID	0.869	2.742
BCA-IT2-FPID	0.862	2.509

VI. CONCLUSION

This paper presents a comparative study of three types of nature inspired optimization algorithms based interval type-2 fuzzy PID controller for FOIPD systems. The analysis of results demonstrate a high performance of the IT2-FPID applying Cuckoo search(CS) algorithm optimization for experimental system, whereas the bee colony algorithm (BCA) shows better performance in simulation study. The computation time of the optimal parameter values for CS is also very low compared to other algorithms studied in this paper.

REFERENCES

[1] Thana Radpukdee " Sliding Mode Control with PID Tuning Technique: An Application to a DC Servo Motor Position Tracking Control", Energy Research Journal 1 (2), pp. 55-61, 2010

[2] Mohd Fua'ad Rahamat & Mariam md Ghazaly, "Performance Comparison between PID And Fuzzy Logic Controller in position Control System of DC Servomotor", Jurnal Teknologi Malayshia, 45(D), pp. 1-17, 2006.

[3] S. Bandyopadhyay, A. Das, "Emphasis on Genetic Agorithm (GA) over Different PID Tuning Methods of Controlling Servo System Using MATLAB", International Journal of Scientific Research in Computer Sciences and Engineering, Vol. 1, Issue-3, pp. 8-13, 2013.

[4] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning—1", Information Science, Vol. 8, 199–249, 1975

[5] A. Yadav, V.K. Harit, "Fault identification in Sub-station by Using Neuro-Fuzzy Technique", International Journal of Scientific Research in Computer Science and Engineering ,Vol-4, Issue-6, pp.-1-7,2016.

[6] E. Ontiveros-Robles, P. Melin, O. Castillo, "Comparative Analysis of Noise Robustness of Type-2 Fuzzy Logic Controller", *Kybernetika,*Vol. 54, No. 1,pp. 175-201, 2018.

[7] B. Sakalli, T. Kumbasar, "On the design and gain analysis of IT2-FPID with a case study on an electric vehicle", *IEEE International Conference on Fuzzy Systems, Vol.25, No.6*, pp. 1752-1764, 2017.

[8] H. A. Hagras, "A hierarchical Type-2 fuzzy logic control architecture for autonomous mobile robots", IEEE Trans. Fuzzy Syst., Vol. 12, No. 4, pp. 524–539,2004.

[9] D. Türkay, A. Baykasoglu, K. Altun, A. Durmusoglu, B. Türksen, "Industrial applications of type-2 fuzzy sets and systems: A concise review", *Computers in Industry; 62(1)*, pp.125-137, 2011.

[10] N. N. Karnik, and J. M. Mendel, "Introduction to type-2 fuzzy logic systems", Proceedings of IEEE International Conference on Fuzzy Systems, Vol. 2 pp. 915-920, 1998.

[11] J. M. Mendel, "Uncertain rule-based fuzzy logic systems: introduction and new directions", Prentice-Hall, New Jersey, 2001

[12] J. Mendel and R. John, "Type-2 Fuzzy Sets Made Simple," IEEE Transactions on Fuzzy Systems, vol. 10, pp.1 17-127, April 2002.

[13] N. N. Karnik and J. M. Mendel, "Centroid of a type-2 fuzzy set", In-form. Sci., vol. 132, pp. 195–220, 2001

[14] W.Z. Qiao, M. Mizumoto, "PID type fuzzy controller and parameters adaptive method.Fuzzy Sets and Systems", 78(1),pp. 23–35, 1996.

[15] E Yesil, T Kumbasar, F Dodurka, and A Sakalli, "Peak observer based self-tuning type-2 fuzzy PID controllers", In Proc. International Conference on Artificial Intelligence Applications and Innovations AIAI, pp. 487-497, 2014.

[16] Ali Al-Waily, R.S., "Design of Robust Mixed H2/H∞ PID Controller Using Particle Swarm Optimization", International Journal of Advancements in Computing Technology 2(5), pp.53–60, 2010.

[17] O. Castillo, L. Amador-Angulo, "A generalized type-2 fuzzy logic approach for dynamic parameter adaptation in bee colony optimization applied to fuzzy controller design", Information Sciences, Volumes 460-461, pp. 476-496, 2017.

[18] M. Konar, A. Bagis, "Performance Comparison of Particle Swarm Optimization, Differential Evolution and Artificial Bee Colony Algorithms for Fuzzy Modelling of Nonlinear Systems", Elektron. Elektrotech, Vol. 22, pp. 8-13, 2016.

[19] J. Kennedy, and R.C. Eberhart, "Particle swarm optimization", in: *Proc. of IEEE International Conference on Neural Networks*, Piscataway, NJ, pp. 1942–1948, 1995.

[20] D. Karaboga, B. Basturk, "On the performance of artificial bee colony (ABC) algorithm", Applied Soft Computing, Vol. 8, pp. 689-697, 2008.

[21] X.S. Yang , S. Deb, " Cuckoo Search ViaLevy flights", In Nature & Biologically Inspired Computing, 2009, World Congress on (IEEE 2009), pp. 210–214,2009.

[22] Documentation for the USER MANUAL Quanser QUBE-Servo-2, Ontario, Canada, 2016

[23] R. Palm, "Sliding mode fuzzy control", Proc. IEEE Int. Conf. on Fuzzy Systems – FUZZ-IEEE, pp. 519-526,1992.

Authors Profile

Mrs. Ritu Rani De(Maity) pursued Bachelor of Engineering in Electronics and Instrumentation Engineering from Vidyasagar University in 2003 and M.E. from Jadavpur University in 2008. She is currently pursuing her Ph. D in Department of Instrumentation and Electronics Engg. Jadavpur University and currently working as Assistant Professor of DR. B. Roy Engineering College, Durgapur. Her research interests are soft computing, intelligent control and fuzzy systems. He has 14 years of teaching experience and 5 years of Research Experience.

Mr. Rajani K. Mudi received B.Tech and M. Tech in Applied Physics in 1990 and 1992, respectively, from the University of Calcutta, India, and Ph.D from Jadavpur University in 1999. He is a Professor in the Department of Instrumentation and Electronics Engg, Jadavpur University, India. His research interests are in intelligent control, neurofuzzy systems, qualitative modeling and bioinformatics. He visited National Chiao Tung University and National Taiwan University, Taiwan, during October 2005 to May 2007. He was the Coordinator of AFSS-2002 and the Secretary of ICONIP-2004. He was the Student activities Chair of Fuzz-IEEE 2013. He co-edited a volume '*Neural Information Processing*' by Springer-Verlag, Germany, 2004 and served as a guest co-editor for a special issue of *International Journal of Intelligent Systems* (2003). He is an Associate Editor of *Electronics Letters*.

Mr. Chanchal Dey received B.Sc. in Physics and B.Tech in Instrumentation and Electronics Engineering in 1993 and 1996 respectively from Jadavpur University, M. Tech. in Instrumentation and Control Engineering in 1999 from University of Calcutta and Ph.D. from Jadavpur University in 2010. is an Associate Professor in the Instrumentation Engineering section of the University of Calcutta, India. From 1996 to 1997 he worked as Process Engineer in Gas Authority of India Ltd. He was a guest faculty in the Department of Instrumentation and Electronics Engineering at Jadavpur University from 1999 to 2005. His research interest involves designing of intelligent process control techniques using conventional and soft-computing tools.

