

A hybrid optimization method with PSO and GA to automatically design Type-1 and Type-2 fuzzy logic controllers

Ricardo Martínez-Soto · Oscar Castillo ·
Luis T. Aguilar · Antonio Rodriguez

Received: 9 October 2012 / Accepted: 8 April 2013 / Published online: 23 April 2013
© Springer-Verlag Berlin Heidelberg 2013

Abstract In this paper we propose the use of a hybrid PSO-GA optimization method for automatic design of fuzzy logic controllers (FLC) to minimize the steady state error of a plant's response. We test the optimal FLC obtained by the hybrid PSO-GA method using benchmark control plants. The bio-inspired and the evolutionary methods are used to find the parameters of the membership functions of the FLC to obtain the optimal controller. Simulation results are obtained to show the feasibility of the proposed approach. A comparison is also made among the proposed Hybrid PSO-GA, GA and PSO to determine if there is a significant difference in the results.

Keywords Fuzzy logic controllers · Genetic algorithms · Particle swarm optimization

1 Introduction

Optimization is a term used to refer to a branch of computational science concerned with finding the “best”

solution to a problem. Here, “best” refers to an acceptable (or satisfactory) solution to a problem, which may be the absolute best over a set of candidate solutions, or a good alternative of the candidate solutions. The characteristics and requirements of the problem determine whether the overall best solution can be found [10]. Bio-inspired optimization algorithms are search methods, where the goal is to find a solution to an optimization problem, such that a given objective function is optimized, possibly subject to a set of constraints [10, 23–25, 30]. Some optimization methods are based on populations of solutions [30]. Unlike the classic methods of optimization, in this case, each iteration of the algorithm maintains a set of solutions. These methods are based on generating, selecting, combining and replacing a set of solutions. Since they maintain and they manipulate a set, instead of a unique solution throughout the entire search process, they use more computer time than other meta-heuristic methods. This fact can be aggravated because the “convergence” of the population requires a great number of iterations. For this reason a concerted effort has been dedicated to obtaining methods that are more aggressive and manage to obtain solutions of quality in a nearer horizon.

This paper is concerned with bio-inspired optimization methods, and in particular the hybrid PSO-GA approach that is proposed. In this case, we combine each individual method in order to obtain the best features to design an optimal fuzzy logic controller (FLC) applied to the benchmark problems. Better results are expected of the hybrid PSO-GA method than those of traditional methods, like GA and PSO, because of the mixing of the best characteristics of both individual methods. We are also comparing the proposed method with genetic algorithms and particle swarm optimization used separately. The bio-inspired methods are used to find the parameters of the

R. Martínez-Soto · A. Rodriguez
School of Engineering UABC University, Tijuana, Mexico
e-mail: mc.ricardo.martinez@hotmail.com

A. Rodriguez
e-mail: ardiaz@uabc.mx

O. Castillo (✉)
Division of Graduate Studies and Research, Tijuana Institute
of Technology, Tijuana, Mexico
e-mail: ocastillo@tectijuana.mx; ocastillo@hafsamx.org

L. T. Aguilar
Centro de Investigación y Desarrollo de Tecnología Digital,
Instituto Politécnico Nacional, Tijuana, Mexico
e-mail: luis.aguilar@ieee.org

Table 1 Fuzzy rules of the FLC

	Negative	Zero	Positive
Negative	N	N	Z
Zero	N	Z	P
Positive	Z	P	P

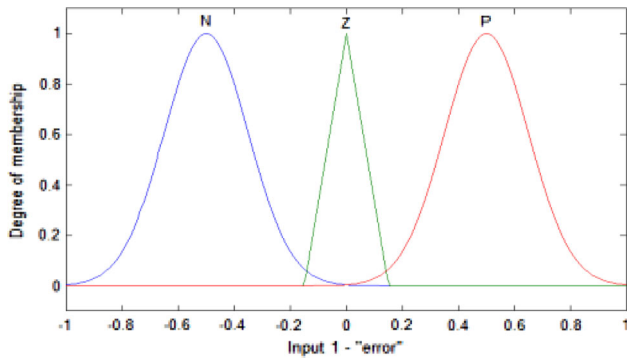


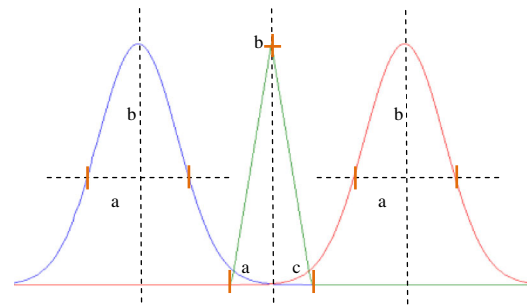
Fig. 1 Gaussian and triangular membership functions and Table 1 show the fuzzy rules

membership functions to obtain the optimal fuzzy controller to stabilize the plant, in particular for the Type-2 fuzzy controller, which is more difficult to design [21].

1.1 Related work

Up to date there are several research papers using genetic algorithms and particle swarm optimization for different optimization problems, like mathematical functions [1, 14, 32], fuzzy controllers [17, 33–35] control parameters [31], the vehicle trajectories [22], scheduling problems [28] and many more. However, in this paper we used the hybrid PSO-GA approach to generate an optimal fuzzy logic controller finding the best parameters of the membership functions to control stable and unstable linear systems and this is the difference between our method and the methods of the following authors. For example, Valdez et al. [32]

Table 2 Parameters of the membership functions



Plant 1				Plant 2			
MF type	Point	Min value	Max value	MF type	Point	Min value	Max value
Gauss	a	0.3	0.6	Gauss	a	1.8	2.8
	b	-1.2	-0.8		b	-6	-4
Triang	a	-0.8	-0.3	Triang	a	-3	-0.5
	b	0	0		b	0	0
	c	0.3	0.8		c	0.5	3
Gauss	a	0.3	0.6	Gauss	a	1.8	2.8
	b	0.8	1.2		b	4	6

uses fuzzy logic to integrate the results obtained by PSO and GA applied to benchmark functions. Muhammad et al. [27] proposed a hybrid method that optimizes the rule base of the k-means method that is used to detect an impostor in a call on the Smartphones. Marinakis et al. [22] proposed a hybrid Genetic-PSO to help in the offspring of the individuals to be transmitted to all the population to get the best solution that in this case was applied to vehicle routing. Niu et al. [28] proposed a hybrid PSO using genetic operators inside the PSO to help to optimize the fuzzy ranking numbers for job scheduling.

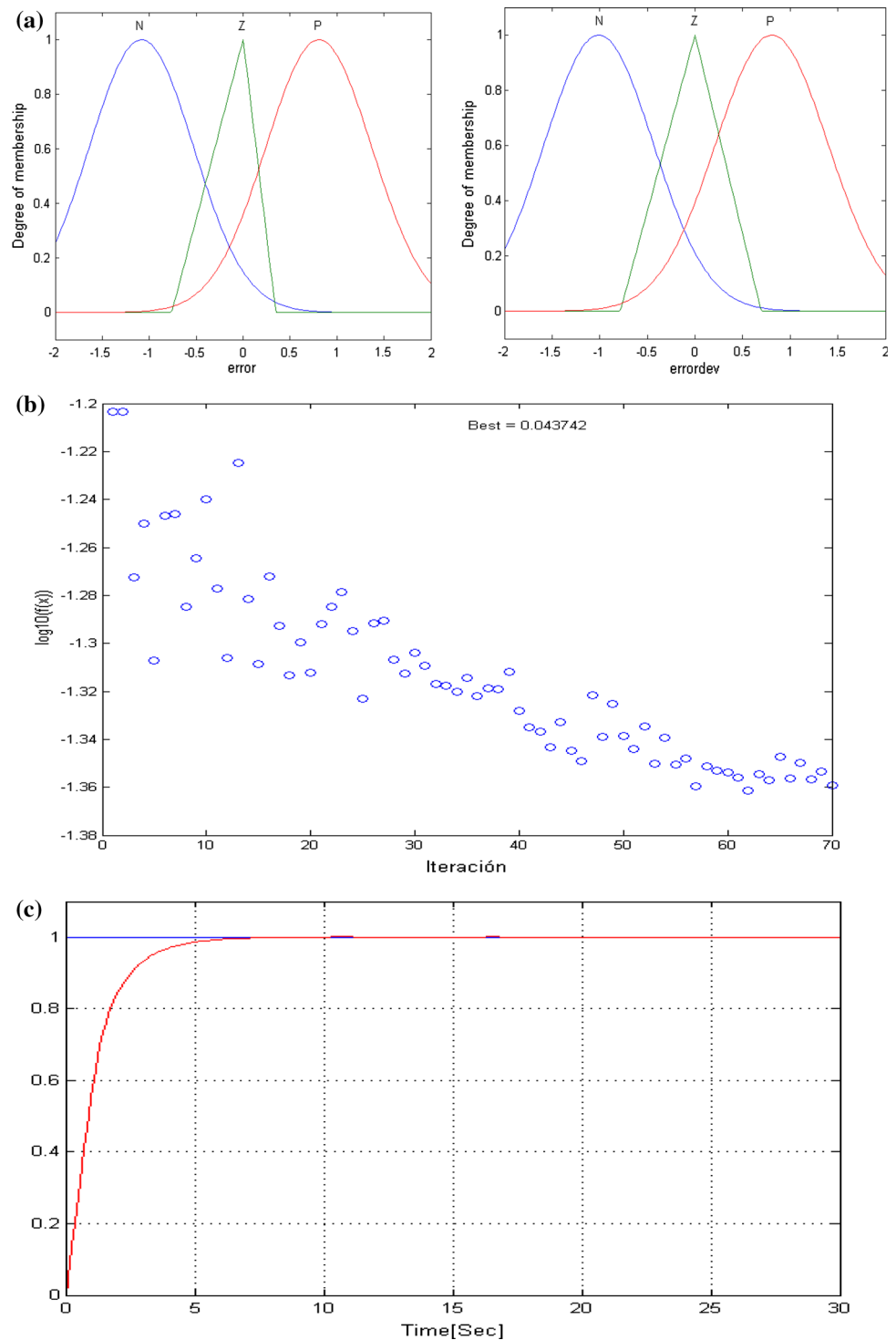
This paper is organized as follows: Section 2 presents the theoretical basis and problem statement. Section 3 introduces the controller design where a Hybrid PSO-GA is used to select the parameters of the membership functions of the FLC, and the GA and PSO are also applied for comparison purposes. Robustness properties of the

Table 3 Results of the Type-1 FLC obtained with PSO for Plant 1

No.	Particle (swarm)	Iterations	C1	C2	Inertia	PSO time exec	Average error
1	200	70	0.4542	0.5052	0.9038	0:26:26	0.081179
2	200	70	0.9667	0.3883	0.8501	0:44:46	0.086795
3	200	70	0.183	0.8722	0.9289	0:09:23	0.087324
4	200	70	0.6798	0.7371	0.6119	0:54:22	0.089289
5	200	70	0.0016	0.8994	0.6983	0:45:59	0.090177
6	200	70	0.484	0.704	0.7194	0:49:16	0.092952
7	200	70	0.7046	0.9151	0.9988	0:05:59	0.09528
8	200	70	0.1344	0.3959	0.2534	0:58:29	0.096134
9	200	70	0.1223	0.5346	0.0681	0:57:48	0.103938
10	200	70	0.1589	0.0366	0.6223	1:04:45	0.109765

Bold values indicate best result

Fig. 2 **a** Input 1 and input 2 membership functions of the Type-1 FLC, **b** particles behavior of the PSO method and **c** closed-loop response of Plant 1 with the optimized FLC



closed-loop system are achieved with a fuzzy logic control system using a Takagi–Sugeno model where the error and the change of error, are considered as the linguistic variables. Section 4 provides a simulation study of the plants using the controllers described in Sect. 3. Finally, Sect. 5 presents the conclusions.

2 Theoretical basis and problem statement

2.1 Particle swarm optimization (PSO)

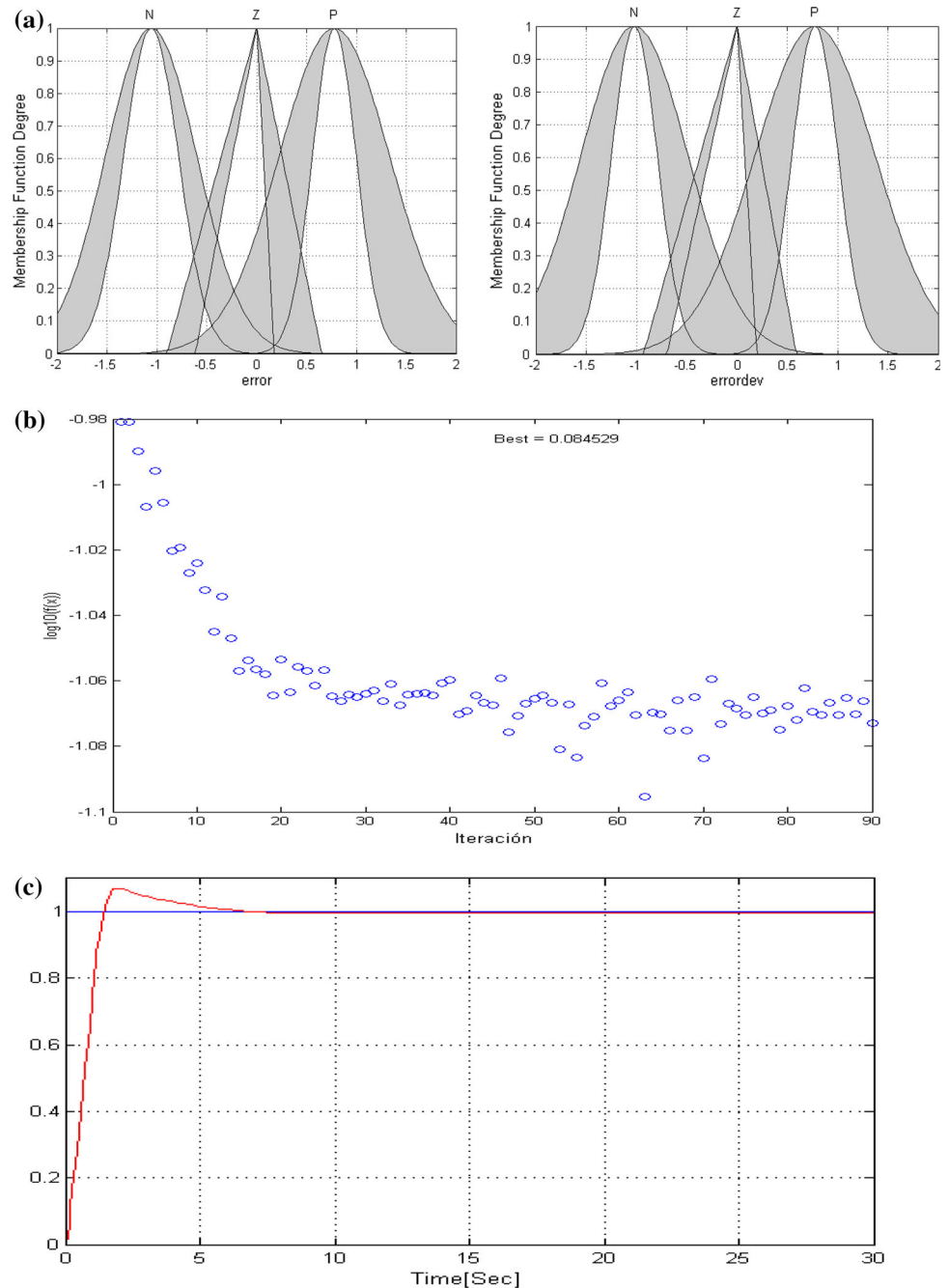
PSO is a population based stochastic optimization technique developed by Eberhart and Kennedy [11], inspired

Table 4 Results of the Type-2 FLC obtained by PSO for Plant 1

No	Particle (swarm)	Iterations	C1	C2	Inertia	PSO time exec	Average error
1	200	90	0.5149	0.3317	0.6808	7:05:20	0.08028
2	200	70	0.8149	0.9059	0.1706	12:44:19	0.08794
3	200	70	0.8129	0.8159	0.1906	10:52:09	0.08894
4	200	70	0.7646	0.9229	0.1096	9:50:45	0.12521
5	200	70	0.8168	0.9359	0.4806	11:05:43	0.1263

Bold values indicate best result

Fig. 3 **a** Input 1 and input 2 membership functions of the Type-2 FLC, **b** particles behavior of the PSO method and **c** closed-loop response of Plant 1 with the optimized FLC



by social behavior of bird flocking or fish schooling. PSO shares many similarities with evolutionary computation techniques, such as genetic algorithms (GA) [12]. The

system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike the GA, the PSO has no evolution operators such as

Table 5 Results of the Type-1 FLC obtained by PSO for Plant 2

No	Particle (swarm)	Iterations	C1	C2	Inertia	PSO Time exec	Average error
1	200	70	0.0429	0.6746	0.8477	0:34:32	0.131933
2	200	70	0.7580	0.1716	0.5573	0:49:17	0.136293
3	200	70	0.7488	0.0884	0.9051	0:39:05	0.141524
4	200	70	0.8149	0.9059	0.1706	0:49:38	0.148655
5	200	70	0.0660	0.7696	0.5558	0:44:32	0.150427
6	200	70	0.8660	0.1036	0.5554	1:05:25	0.151731
7	200	70	0.5543	0.6067	0.4974	0:50:25	0.152046
8	200	70	0.7154	0.2479	0.5130	0:55:45	0.154302
9	200	70	0.0158	0.0882	0.2155	1:02:43	0.160239
10	200	70	0.4458	0.9566	0.3233	0:51:22	0.164745

Bold values indicate best result

crossover and mutation. In the PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles [5]. Each particle keeps track of its coordinates in the problem space, which are associated with the best solution (fitness) it has achieved so far (the fitness value is also stored). This value is called *pbest*. Another “best” value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the neighbors of the particle. This location is called *lbest*. When a particle takes all the population as its topological neighbors, the best value is a global best and is called *gbest* [15].

The particle swarm optimization concept consists of, at each time step, changing the velocity of (accelerating) each particle toward its *pbest* and *lbest* locations (local version of PSO). Acceleration is weighted by a random term, with separate random numbers being generated for acceleration toward *pbest* and *lbest* locations [5]. In the past several years, PSO has been successfully applied in many research and application areas. It has been demonstrated that PSO gets better results in a faster, cheaper way compared with other methods [16, 4, 36–38]. Another reason that PSO is attractive is that there are few parameters to adjust. One version, with slight variations, works well in a wide variety of applications. Particle swarm optimization has been used for problems across a wide range of applications, as well as for the particular case of designing fuzzy controllers [16].

2.2 Genetic algorithms (GAs)

Genetic algorithms (GAs) are adaptive heuristic search algorithms based on the evolutionary ideas of natural selection and genetic processes [2]. The basic principles of GAs were first proposed by John Holland in 1975, inspired by the mechanism of natural selection where stronger individuals are likely the winners in a competing environment [3, 6, 8]. The GA assumes that the potential solution to any problem is an individual and can be represented by a set of parameters. These parameters are

regarded as the genes of a chromosome and can be structured by a string of values in binary form. A positive value, generally known as a fitness value, is used to reflect the degree of “goodness” of the chromosome for solving the problem, which is highly related with its objective value. The GA works as follows:

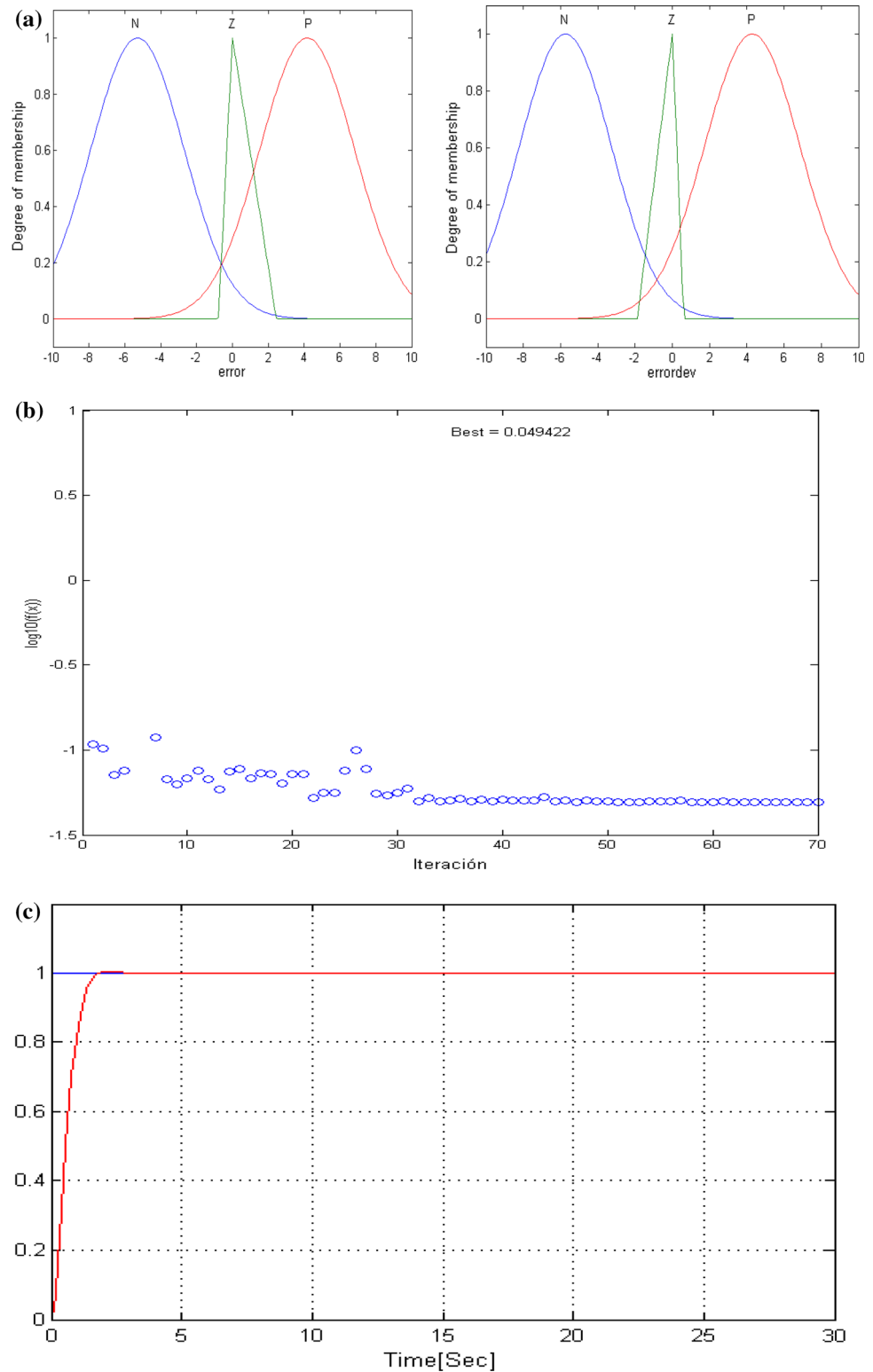
1. Start with a randomly generated population of n chromosomes (candidate solutions to a problem).
2. Calculate the fitness of each chromosome in the population.
3. Repeat the following steps until n offspring have been created:
 - a. Select a pair of parent chromosomes from the current population, the probability of selection being an increasing function of fitness. Selection is done with replacement, meaning that the same chromosome can be selected more than once to become a parent.
 - b. With a particular probability (crossover rate), perform crossover to the pair at a randomly chosen point to form two offspring.
 - c. Mutate the two offspring at each locus with probability (mutation rate), and place the resulting chromosomes in the new population.
4. Replace the current population with the new population.
5. Go to step 2.

The simple procedure just described above is the basis for most applications of GAs.

2.3 Hybrid PSO-GA

We propose a hybrid method using particle swarm optimization and genetic algorithms to find an optimal FLC to control the benchmark plants. The hybrid method uses the same population/swarm in the two methods in the same iteration obtaining the best individual/particle. PSO and

Fig. 4 **a** Input 1 and input 2 membership functions of the Type-1 FLC, **b** particles behavior of the PSO method and **c** closed-loop response of Plant 2 with the optimized FLC



GA communicate with each other; every four iterations the best individual/particle is inserted into the population of the worst method and vice versa. We used the maximum

number of iteration/generations to stop the method, keeping only the best FLC obtained [27]. The proposed method works as follows:

Table 6 Results of the Type-2 FLC obtained by PSO for Plant 2

No	Particle (swarm)	Iterations	C1	C2	Inertia	PSO time exec	Average error
1	200	70	0.3179	0.5411	0.7012	8:51:28	0.083388
2	200	70	0.7690	0.2082	0.8755	8:09:51	0.090310
3	200	70	0.3889	0.6394	0.7304	8:50:16	0.096099
4	200	70	0.9512	0.8014	0.6141	9:15:29	0.101980
5	200	70	0.7870	0.6322	0.5724	10:00:28	0.102010

Bold values indicate best result

Fig. 5 **a** Input 1 and input 2 membership functions of the Type-2 FLC, **b** particles behavior of the PSO method and **c** closed-loop response of Plant 2 with the optimized FLC

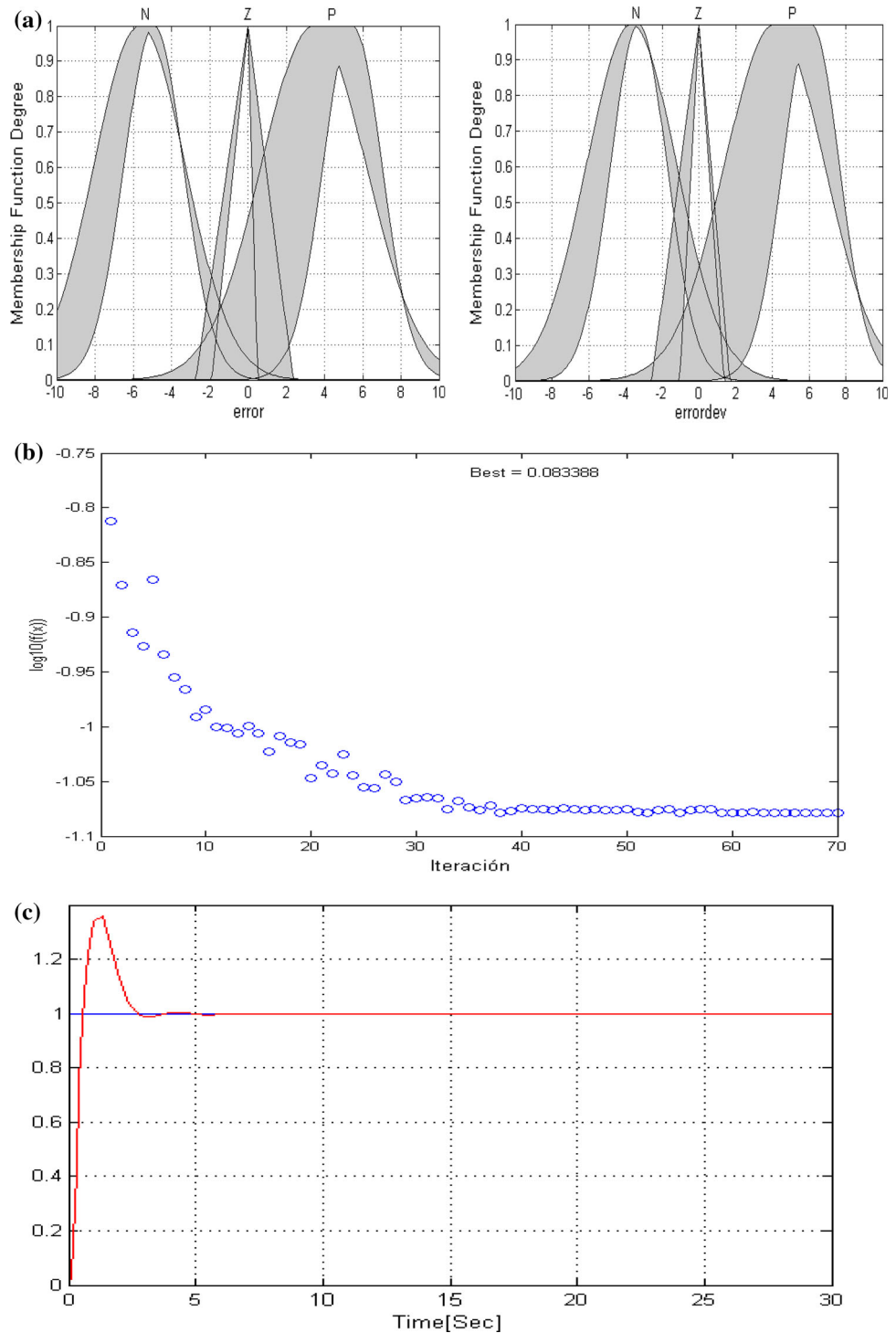


Table 7 Results of the Type-1 FLC obtained by GA for Plant 1

No	Individuals	Generations	Percentage replace	Cross	Mut	Selection method	GA time exec	Average error
1	90	35	0.7	0.6	0.3	Ruleta	0:18:06	0.05087
2	150	80	0.7	0.5	0.2	Ruleta	1:19:13	0.04431
3	80	50	0.7	0.5	0.2	Ruleta	0:23:01	0.071366
4	45	60	0.7	0.6	0.3	Ruleta	0:16:03	0.068477
5	75	50	0.7	0.6	0.1	Ruleta	0:21:37	0.068158
6	100	40	0.7	0.6	0.1	Ruleta	0:24:08	0.067052
7	65	35	0.7	0.7	0.2	Ruleta	0:15:35	0.069994
8	200	70	0.7	0.4	0.1	Ruleta	1:30:02	0.072356
9	25	15	0.7	0.8	0.3	Ruleta	0:02:58	0.129872
10	50	45	0.7	0.5	0.2	Ruleta	0:13:25	0.068855

Bold values indicate best result

1. Start with a randomly generated particles/population (candidate solutions to a problem).
 2. Assign to the variables *S* and *Chrom* the particles/population generated.
 3. Repeat the following steps until max number of iteration/generations
 - 3.1 Start PSO method.
 - 3.1.1 For each particle do.
 - 3.1.1.1 Calculate fitness value.
 - 3.1.1.2 If the fitness value is better than the best fitness value (*pBest*) in history then, set the current value as the new *pBest*.
 - 3.1.1.3 Choose the particle with the best fitness value of all particles as the *gBest*.
 - 3.1.1.4 Update all particle velocities.
 - 3.1.1.5 Update all particle positions.
 - 3.1.1.6 Until “*n*” particles.
 - 3.1.1.7 Save the best particle information in variable *BM*.
 - 3.2 Terminate PSO.
 - 3.3 Start GA method.
 - 3.3.1 Calculate the fitness of each chromosome in the population.
 - 3.3.2 Repeat the following steps until *n* offspring have been created:
 - 3.3.2.1 Select a pair of parent chromosomes from the current population, the probability of selection being an increasing function of fitness. Selection is done with replacement, meaning that the same chromosome can be selected more than once to become a parent.
 - 3.3.2.2 With probability (crossover rate), perform crossover to the pair at a randomly chosen point to form two offspring.
 - 3.3.2.3 Mutate the two offspring at each locus with probability (mutation rate), and place the resulting chromosomes in the new population.
 - 3.3.3 Replace the current population with the new population.
 - 3.3.4 Go to step 5.
 - 3.3.5 Save the best individual information in variable *BM*.
 - 3.4 Terminate GA.
 4. if counter = 4
 - 4.1 Obtain the minimum result of the *BM* variable
 - 4.2 if minimum value was obtained by PSO method then
 - 4.3 reinsert the information of the particle in the GA population (*Chrom*)
 - 4.3 else
 - 4.4 reinsert the information of the best individual of GA in the Swarm of the PSO
 5. Terminate the validation.
 6. Terminate the hybrid method until maximum number of iterations/generations.
- The procedure described above is used to find an optimal fuzzy logic controller combining the PSO and GA to exploit more completely the space of solutions.
- ## 2.4 Problem statement
- To test the optimized FLCs obtained with the bio-inspired methods; we used two different benchmark plants. We consider two benchmark problems called Plant 1 and Plant 2 with different levels of complexity. We used this kind of applications to observe the behavior of the proposed method with respect to the GA and PSO methods [21].
- Plant 1 is given by the following stable second order transfer function:

Fig. 6 **a** Input 1 and input 2 membership functions of the Type-1 FLC, **b** evolution of the GA method and **c** closed-loop response of Plant 1 with the optimized FLC

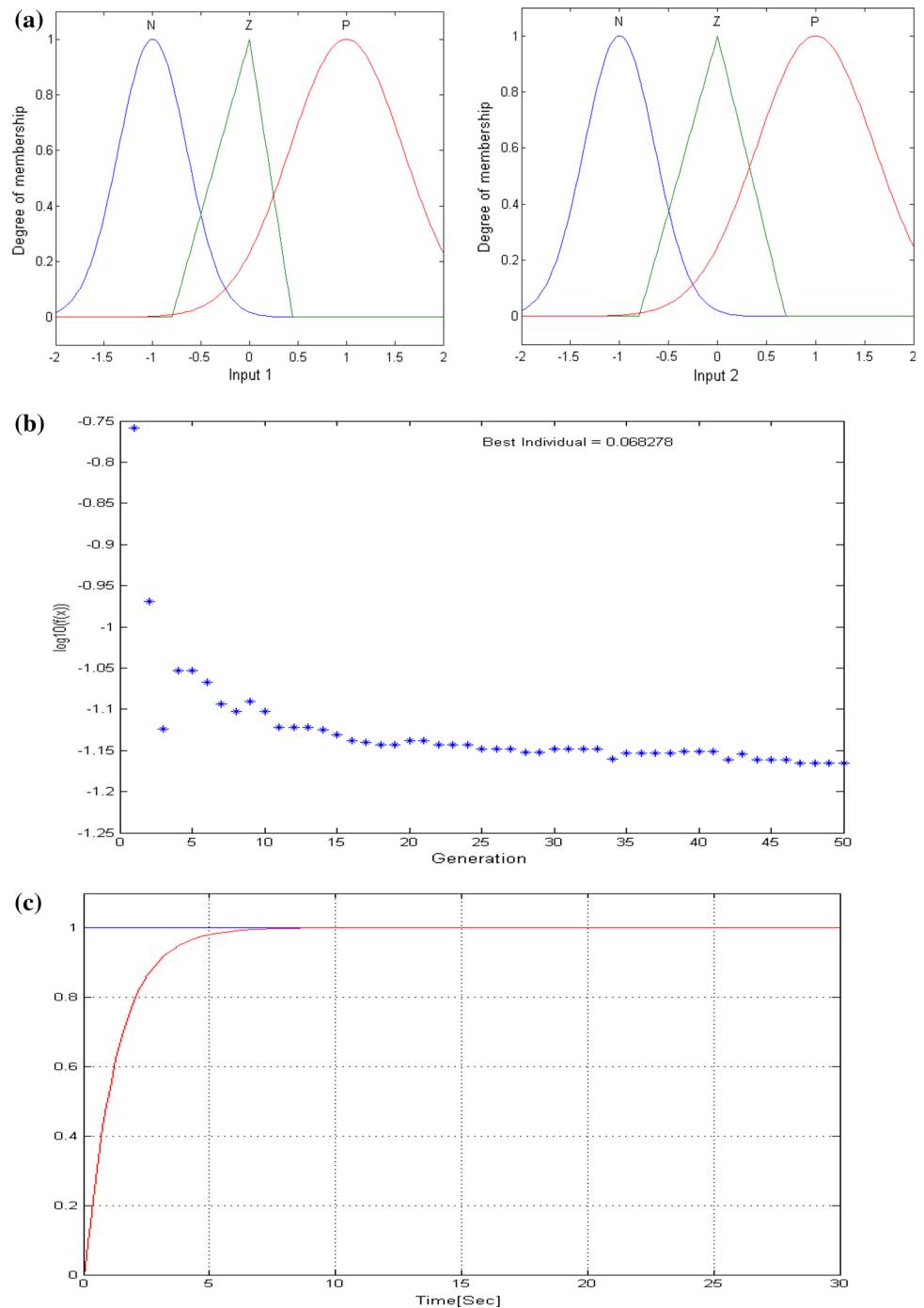


Table 8 Results of the Type-2 FLC obtained by PSO for Plant 1

No	Individuals	Generations	Percentage replace	Cross	Mut	Selection method	GA time exec	Average error
1	55	75	0.7	0.8	0.3	Ruleta	5:56:25	0.067651
2	90	65	0.7	0.7	0.2	Ruleta	8:33:15	0.067995
3	50	80	0.7	0.8	0.2	Ruleta	5:41:16	0.068208
4	90	85	0.7	0.7	0.2	Ruleta	11:18:23	0.068311
5	120	85	0.7	0.7	0.2	Ruleta	14:42:40	0.068749
6	30	55	0.7	0.8	0.3	Ruleta	2:21:10	0.069307

Table 8 continued

No	Individuals	Generations	Percentage replace	Cross	Mut	Selection method	GA time exec	Average error
7	60	40	0.7	0.7	0.2	Ruleta	3:18:27	0.069967
8	45	40	0.7	0.9	0.3	Ruleta	2:26:12	0.073498
9	40	25	0.7	0.8	0.3	Ruleta	1:26:55	0.095847
10	65	35	0.7	0.8	0.2	Ruleta	3:08:03	0.745350

Bold values indicate best result

Fig. 7 **a** Input 1 and input 2 membership functions of the Type-2 FLC, **b** evolution of the GA method and **c** closed-loop response of Plant 1 with the optimized FLC

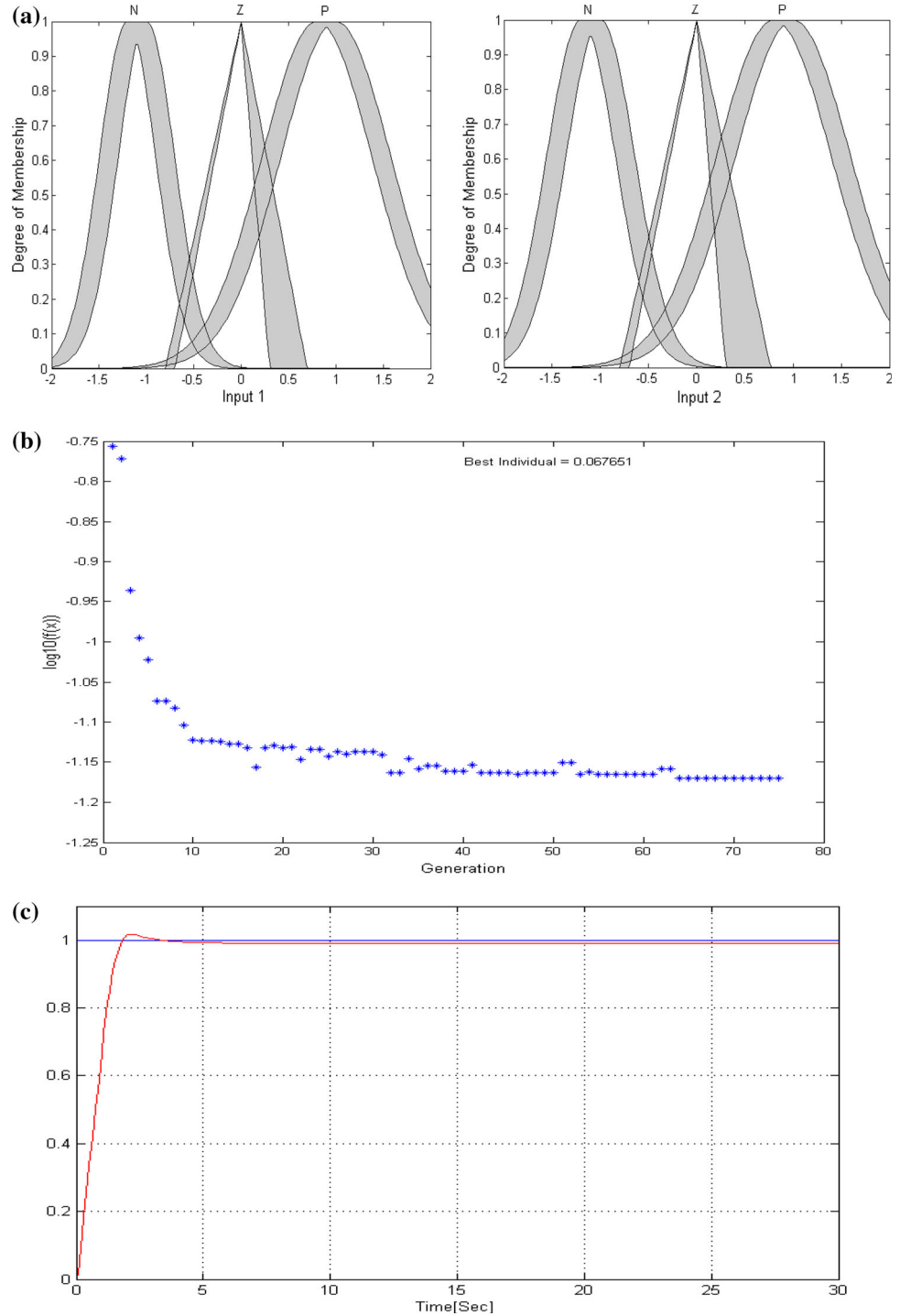


Table 9 Results of the Type-1 FLC obtained by GA for Plant 2

No	Individuals	Generations	Percentage replace	Cross	Mut	Selection method	GA time exec	Average error
1	85	65	0.5	0.5	0.1	Ruleta	0:29:11	0.066594
2	80	50	0.5	0.6	0.1	Ruleta	0:22:18	0.076842
3	55	45	0.5	0.6	0.1	Ruleta	0:39:01	0.077597
4	100	40	0.5	0.3	0.1	Ruleta	0:20:41	0.078355
5	65	35	0.5	0.4	0.2	Ruleta	0:11:53	0.076711
6	200	90	0.5	0.4	0.1	Ruleta	1:33:19	0.065268
7	25	15	0.5	0.8	0.3	Ruleta	0:02:15	0.261787
8	90	35	0.5	0.5	0.2	Ruleta	0:16:26	0.074058
9	40	25	0.5	0.7	0.1	Ruleta	0:05:12	0.134507
10	120	85	0.5	0.4	0.1	Ruleta	0:52:13	0.070636

Bold values indicate best result

$$g(s) = \frac{w_n^2}{s^2 + 2\varepsilon w_n s + w_n^2} \quad \varepsilon = 0.5, w_n = 2 \tag{1}$$

where w_n is the natural frequency and ε is the coefficient damping.

Plant 2 is given by the following transfer function:

$$g(s) = \frac{1}{s^2 + 4} \tag{2}$$

The behavior of Plant 2 is more complex as will be seen later. The main idea in using these two plants is to illustrate the proposed hybrid approach in designing the fuzzy controllers, especially for Type-2 fuzzy logic, which are more complex to design than Type-1. Of course, it is well known that even for designing Type-1 fuzzy controllers the problem could be challenging, but this is even more critical for designing optimal Type-2 fuzzy controllers.

3 Fuzzy logic control design

In this section we show the design of a fuzzy logic controller (FLC), where the optimal controller was found using the proposed hybrid PSO-GA, GAs and the PSO. For the FLC a Takagi–Sugeno type of fuzzy system is used with two inputs (a) error, and (b) error change, with three membership functions for each input, “negative, zero and positive” (Gaussian and triangular), and one output, defined with constant values [7, 9, 13, 18–21, 26, 29]. Figure 1 shows the FLC membership functions for the plant control and Table 1 shows the Fuzzy Rules.

Once we obtained the FLC design, we set the parameters of both methods: the GA chromosome has 17 genes of real values that represent the two inputs, error and error change and one output with constant values

and using different values in the genetic operator’s; mutation and single point crossover and roulette wheel selection method. For PSO method we use different values for the cognitive and social parameters also an inertial value to balance the swarm. For the hybrid PSO-GA method we apply the same parameter configuration used in GA and PSO; from an initial space search we generate two subpopulations to work inside of the hybrid method. The three bio-inspired methods use the same space of solutions (population) that we consider to find the optimal values of the parameters of the membership functions. Table 2 shows the parameters of the membership functions, the minimal and the maximum values in the search range for the Hybrid PSO-GA, GA and PSO to find the best fuzzy controller system for the benchmark plants.

4 Simulation results

In this section we present the simulation results for each bio-inspired method applied to Plant 1 and Plant 2. We organized the presentation of the simulation results as follows:

4.1 Optimal FLC results obtained with the PSO method for Plant 1

The simulation results for the Type-1 FLC obtained with the PSO method applied to Plant 1 are presented in this section. Table 3 contains the configuration values of the PSO method, the execution time of PSO and the average error for each configuration. The first row shows the best controller obtained by PSO.

Fig. 8 **a** Input 1 and input 2 membership functions of the Type-1 FLC, **b** evolution of the GA method and **c** closed-loop response of Plant 2 with the optimized FLC

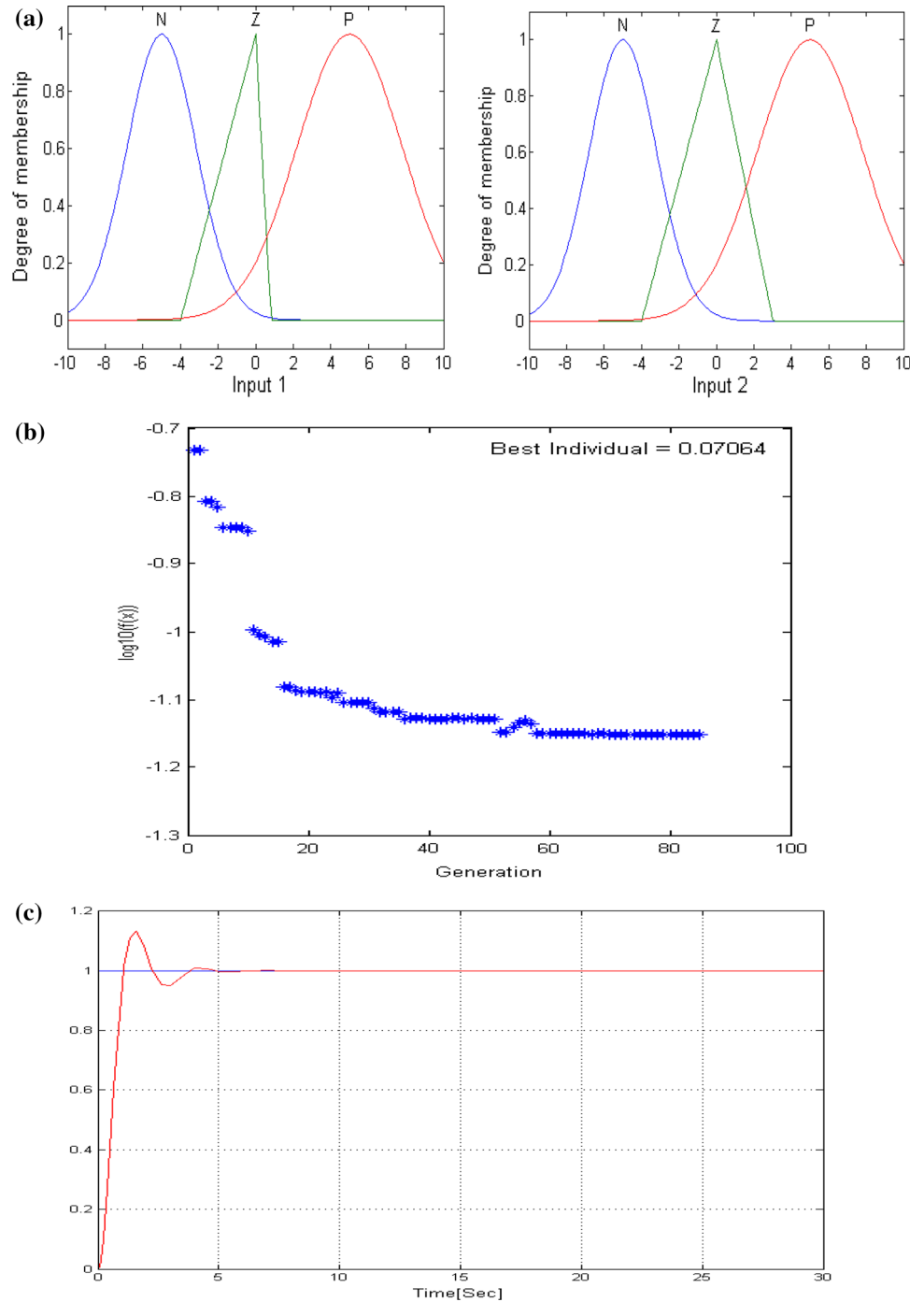


Table 10 Results of the Type-2 FLC obtained by PSO for Plant 2

No	Individuals	Generations	Percentage replace	Cross	Mut	Selection method	GA time exec	Average error
1	90	85	0.5	0.7	0.2	Ruleta	7:59:16	0.071904
2	90	65	0.5	0.7	0.2	Ruleta	5:48:37	0.072523
3	120	85	0.5	0.7	0.2	Ruleta	11:07:54	0.073215
4	45	40	0.5	0.9	0.3	Ruleta	1:51:16	0.076852
5	50	80	0.5	0.6	0.1	Ruleta	4:08:36	0.07786
6	30	55	0.5	0.8	0.3	Ruleta	1:26:11	0.079192

Table 10 continued

No	Individuals	Generations	Percentage replace	Cross	Mut	Selection method	GA time exec	Average error
7	55	75	0.5	0.8	0.3	Ruleta	3:35:16	0.079204
8	40	25	0.5	0.7	0.3	Ruleta	1:04:20	0.079878
9	65	35	0.5	0.6	0.1	Ruleta	2:29:15	0.082048
10	60	40	0.5	0.7	0.2	Ruleta	2:08:08	0.089937

Bold values indicate best result

Fig. 9 **a** Input 1 and input 2 membership functions of the Type-2 FLC, **b** evolution of the GA method and **c** closed-loop response of Plant 2 with the optimized FLC

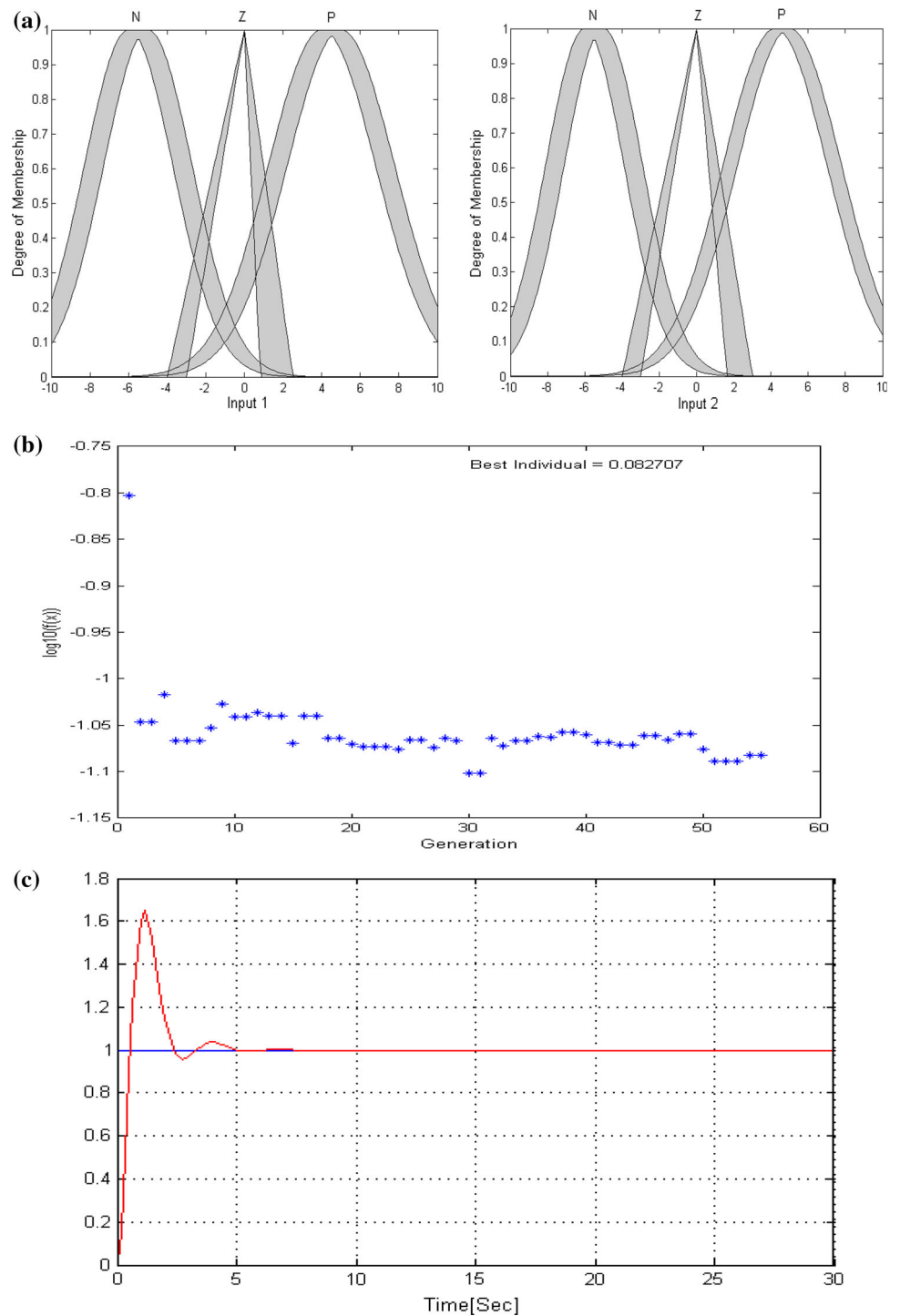


Table 11 Results of the Type-1 FLC obtained by hybrid PSO-GA method for Plant 1

No	Individuals/ particles	Iterations/ generations	C1	C2	Inertia	Percentage remp	Cross	Mut	Time execution	Average error
1	70	200	0.2706	0.0403	0.3254	0.7	0.7	0.2	1:46:07	0.10956
2	200	70	0.1780	0.4863	0.5381	0.7	0.5	0.2	1:20:14	0.11375
3	200	70	0.8963	0.3534	0.6738	0.7	0.7	0.1	1:18:21	0.11517
4	200	70	0.4743	0.5077	0.5163	0.7	0.6	0.2	1:28:50	0.11559
5	200	70	0.4587	0.6339	0.1024	0.7	0.5	0.5	1:42:23	0.11584
6	150	80	0.7293	0.1387	0.8532	0.7	0.7	0.1	1:15:23	0.11971
7	90	75	0.2521	0.3490	0.9708	0.7	0.5	0.2	0:25:14	0.12240
8	200	70	0.8149	0.9059	0.1314	0.7	0.5	0.1	1:33:46	0.12325
9	80	100	0.5159	0.0732	0.1455	0.7	0.6	0.1	0:48:21	0.12488
10	90	35	0.2110	0.9645	0.1126	0.7	0.5	0.2	0:17:52	0.12853

Bold values indicate best result

Figure 2a shows the membership functions of input 1 and input 2 of the optimal FLC, Fig. 2b shows the particles behavior of the PSO method giving the best FLC to control the Plant 1 and Fig. 2c shows FLC Type-1 result for Plant 1 obtaining the control before the 10 s on stable manner.

Continuing with the simulation results using the PSO method for Plant 1 we present the optimal Type-2 FLC, showing in Table 4 the configuration values of the PSO method, the execution time and the average error of each configuration. The first row shows the best Type-2 controller obtained by PSO.

Figure 3a shows the membership functions of input 1 and input 2 of the optimal FLC, Fig. 3b shows the particles behavior of the PSO method giving the best FLC to control the Plant 1 and Fig. 3c shows FLC Type-2 result for Plant 1 obtaining the control before of the 10 s with a little overshoot.

4.2 Optimal FLC results obtained by the PSO method for Plant 2

In this section we present the simulation results for the Type-1 FLC obtained by the PSO method applied to Plant 2. Table 5 contains the configuration values of the PSO method, the execution time of PSO and the average error for each configuration. The first row shows the best controller obtained by PSO.

Figure 4a shows the membership functions of input 1 and input 2 of the optimal FLC, Fig. 4b shows the particles behavior of the PSO method giving the best FLC to control the Plant 2 and Fig. 4c shows FLC Type-1 result for Plant 2 obtaining the control before of the 5 s on stable manner.

Continuing with the simulation results using the PSO method for Plant 2 we present the optimal Type-2 FLC, showing in Table 6 the configuration values of the PSO method, the execution time and the average error of each configuration. The first row shows the best Type-2 controller obtained by PSO.

Figure 5a shows the membership functions of input 1 and input 2 of the optimal FLC, Fig. 5b shows the particles behavior of the PSO method giving the best FLC to control the Plant 1 and Fig. 5c shows FLC Type-2 result for Plant 1 obtaining the control before of the 5 s but with an overshoot.

4.3 Optimal FLC results obtained by the GA method for Plant 1

This section presents the simulation results for the Type-1 FLC obtained by the GA method applied to Plant 1. Table 7 contains the configuration values of the GA method, the execution time of GA and the average error for each configuration. The third row shows the best controller obtained by GA.

Figure 6a shows the membership functions of input 1 and input 2 of the optimal FLC, Fig. 6b shows the evolution of the GA method giving the best FLC to control the Plant 1 and Fig. 6c shows FLC Type-1 result for Plant 1 obtaining the control before the 10 s on stable manner.

Continuing with the simulation results using the GA method for Plant 1 we present the optimal Type-2 FLC, showing in Table 8 the configuration values of the PSO method, the execution time and the average error of each

Fig. 10 **a** Input 1 and input 2 membership functions of the Type-1 FLC, **b** behavior of the individuals/particles of the hybrid PSO-GA method and **c** closed-loop response of Plant 1 with the optimized FLC

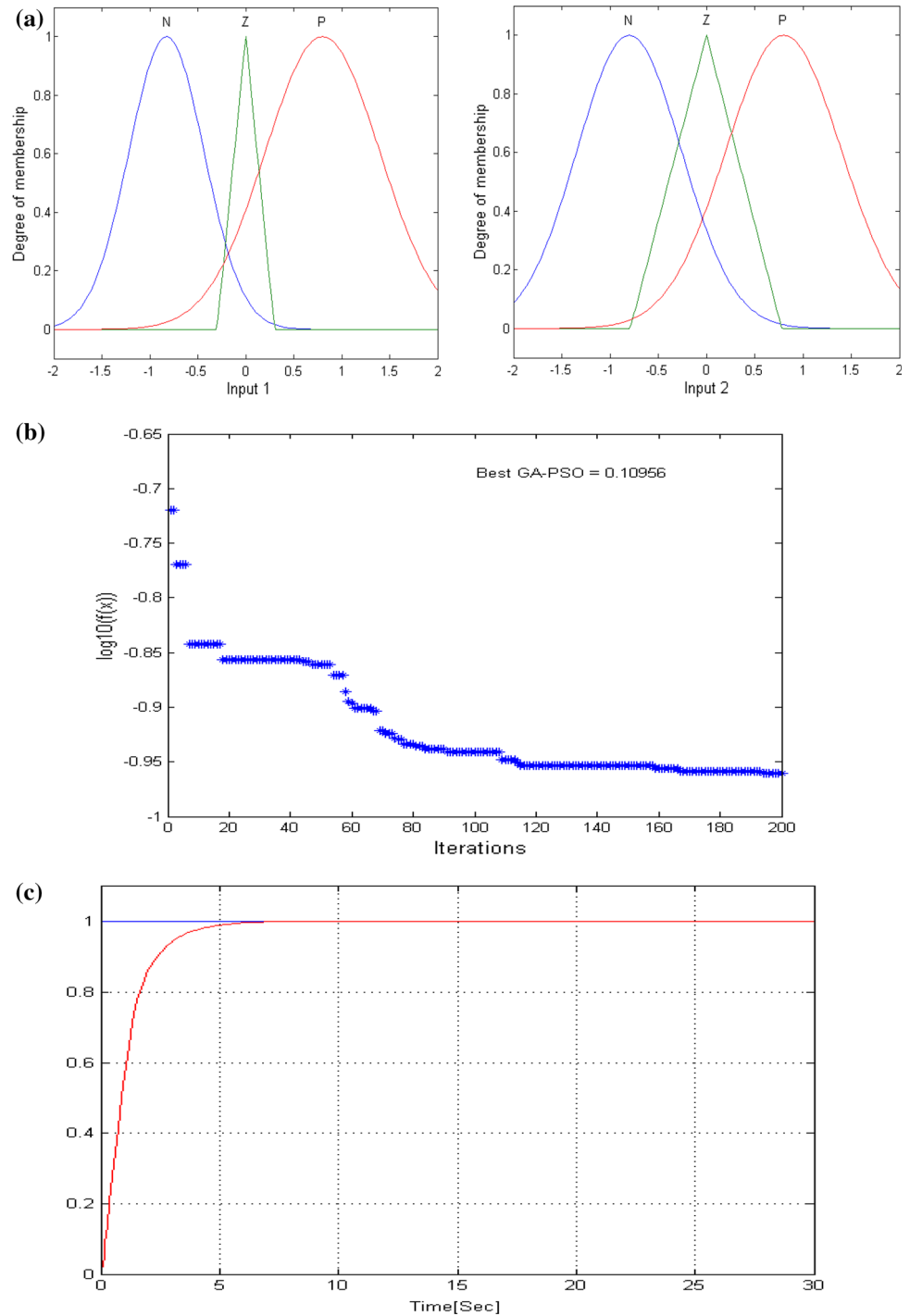


Table 12 Results of the Type-2 FLC obtained by hybrid method for Plant 1

No.	Individuals/ particles	Iterations/ generations	C1	C2	Inertia	Percentage remp	Cross	Mut	Time execution	Average error
1	25	50	0.8149	0.9059	0.1706	0.7	0.8	0.3	11:46:36	0.132233
2	25	50	0.0873	0.4457	0.6676	0.7	0.8	0.3	16:08:47	0.140007
3	45	30	0.9118	0.6906	0.7163	0.7	0.6	0.1	19:08:49	0.147934
4	25	50	0.4928	0.1730	0.7160	0.7	0.8	0.3	20:36:40	0.148900

Table 12 continued

No.	Individuals/ particles	Iterations/ generations	C1	C2	Inertia	Percentage remp	Cross	Mut	Time execution	Average error
5	20	15	0.9458	0.3435	0.6244	0.7	0.8	0.3	7:18:33	0.151596
6	40	25	0.8149	0.9059	0.1706	0.7	0.7	0.4	1:19:48	0.157026
7	20	15	0.0910	0.2991	0.8237	0.9	0.8	0.3	2:46:26	0.159745
8	40	25	0.8149	0.9059	0.1706	0.7	0.7	0.4	14:28:04	0.163500
9	15	60	0.8149	0.9059	0.1706	0.7	0.7	0.1	14:05:55	0.178253
10	25	50	0.8149	0.9059	0.1706	0.7	0.8	0.3	11:31:26	0.179500

Bold values indicate best result

Fig. 11 **a** Input 1 and input 2 membership functions of the Type-2 FLC, **b** individuals/particles behavior of the hybrid PSO-GA method and **c** closed-loop response of Plant 1 with the optimized FLC

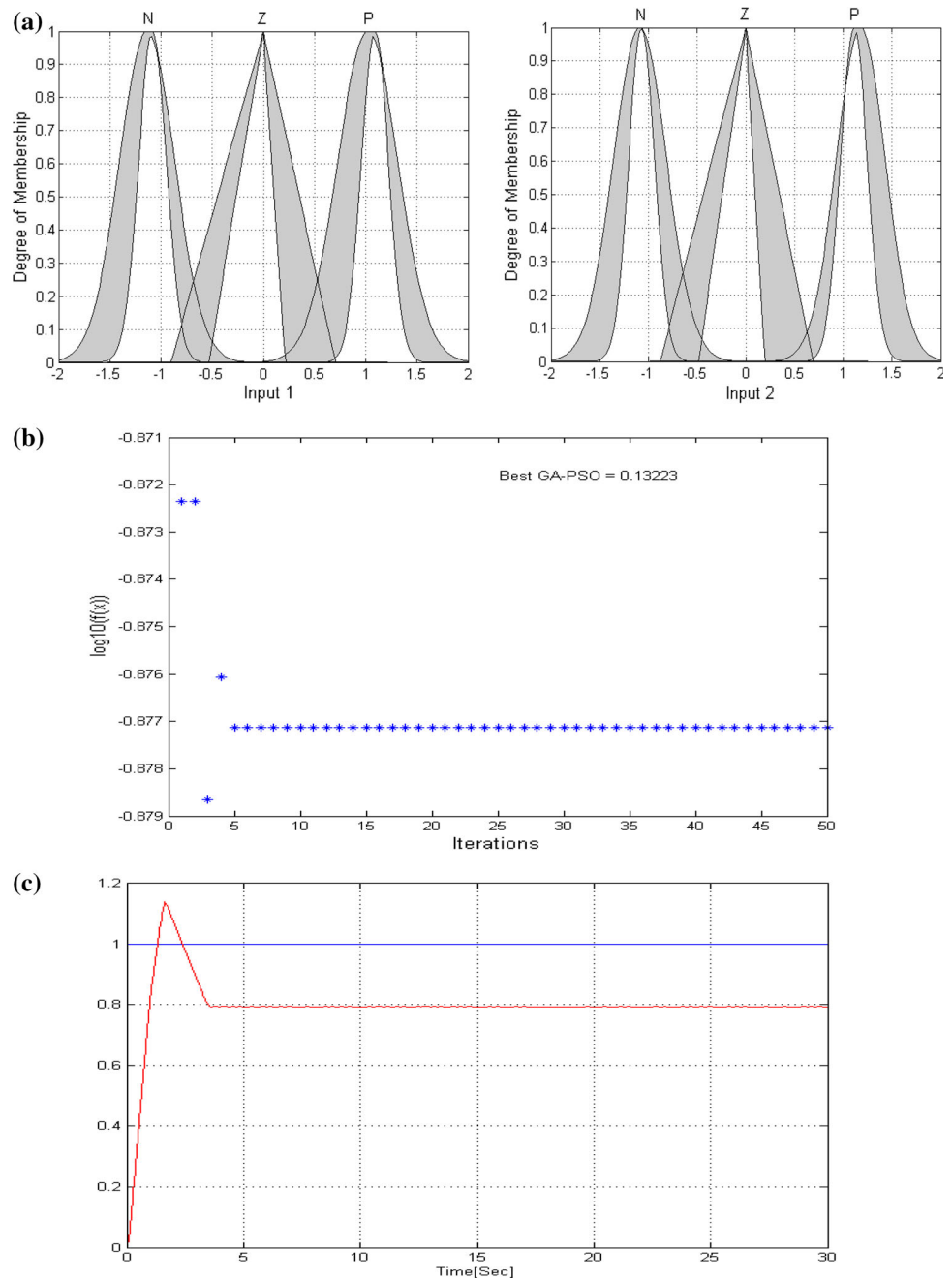


Table 13 Results of the Type-1 FLC obtained by hybrid PSO-GA method for Plant 2

No.	Individuals/ particles	Iterations/ generations	C1	C2	Inertia	Percentage remp	Cross	Mut	Time execution	Average error
1	200	70	0.05	0.05	1	0.7	0.5	0.2	1:10:15	0.045568
2	200	70	1	1	1	0.7	0.5	0.1	0:50:02	0.045656
3	150	80	0.5171	0.5210	0.8254	0.7	0.7	0.1	1:11:57	0.046050
4	80	120	0.7372	0.8160	0.7541	0.7	0.5	0.2	1:00:36	0.046147
5	100	150	0.6608	0.1696	0.8976	0.7	0.5	0.2	1:41:48	0.046674
6	200	70	0.25	0.25	1	0.7	0.7	0.1	1:53:35	0.054476
7	200	70	0.8149	0.9059	0.1706	0.7	0.8	0.3	1:32:57	0.055365
8	50	45	0.8192	0.0754	0.8559	0.7	0.7	0.1	0:13:43	0.055597
9	50	150	0.1934	0.4693	0.7081	0.7	0.6	0.2	0:55:20	0.057434
10	30	80	0.3163	0.2037	0.5234	0.7	0.8	0.3	0:16:05	0.057775

Bold values indicate best result

configuration. The first row shows the best Type-2 controller obtained by PSO.

Figure 7a shows the membership functions of input 1 and input 2 of the optimal FLC, Fig. 7b shows the evolution of the GA method giving the best FLC to control the Plant 1 and Fig. 7c shows FLC Type-2 result for Plant 1 obtaining the control before of the 5 s with a small overshoot.

4.4 Optimal FLC results obtained by the GA method for Plant 2

This section presents the simulation results for the Type-1 FLC obtained by the GA method applied to Plant 2. Table 9 contains the configuration values of the GA method, the execution time of GA and the average error for each configuration. The tenth row shows the best controller obtained by GA.

Figure 8a shows the membership functions of input 1 and input 2 of the optimal FLC, Fig. 8b shows the evolution of the GA method giving the best FLC to control the Plant 2 and Fig. 8c shows FLC Type-1 result for Plant 2 obtaining the control before the 5 s with a small overshoot and undershoot.

Continuing with the simulation results using the GA method for Plant 2 we present the optimal Type-2 FLC, showing in Table 10 the configuration values of the PSO method, the execution time and the average error of each configuration. The sixth row shows the best Type-2 controller obtained by PSO.

Figure 9a shows the membership functions of input 1 and input 2 of the optimal FLC, Fig. 9b shows the evolution of the GA method giving the best FLC to control the

Plant 2 and Fig. 9c shows FLC Type-2 result for Plant 2 obtaining the control before of the 5 s with a large overshoot and a small undershoot.

4.5 Optimal FLC results obtained by the hybrid PSO-GA method for Plant 1

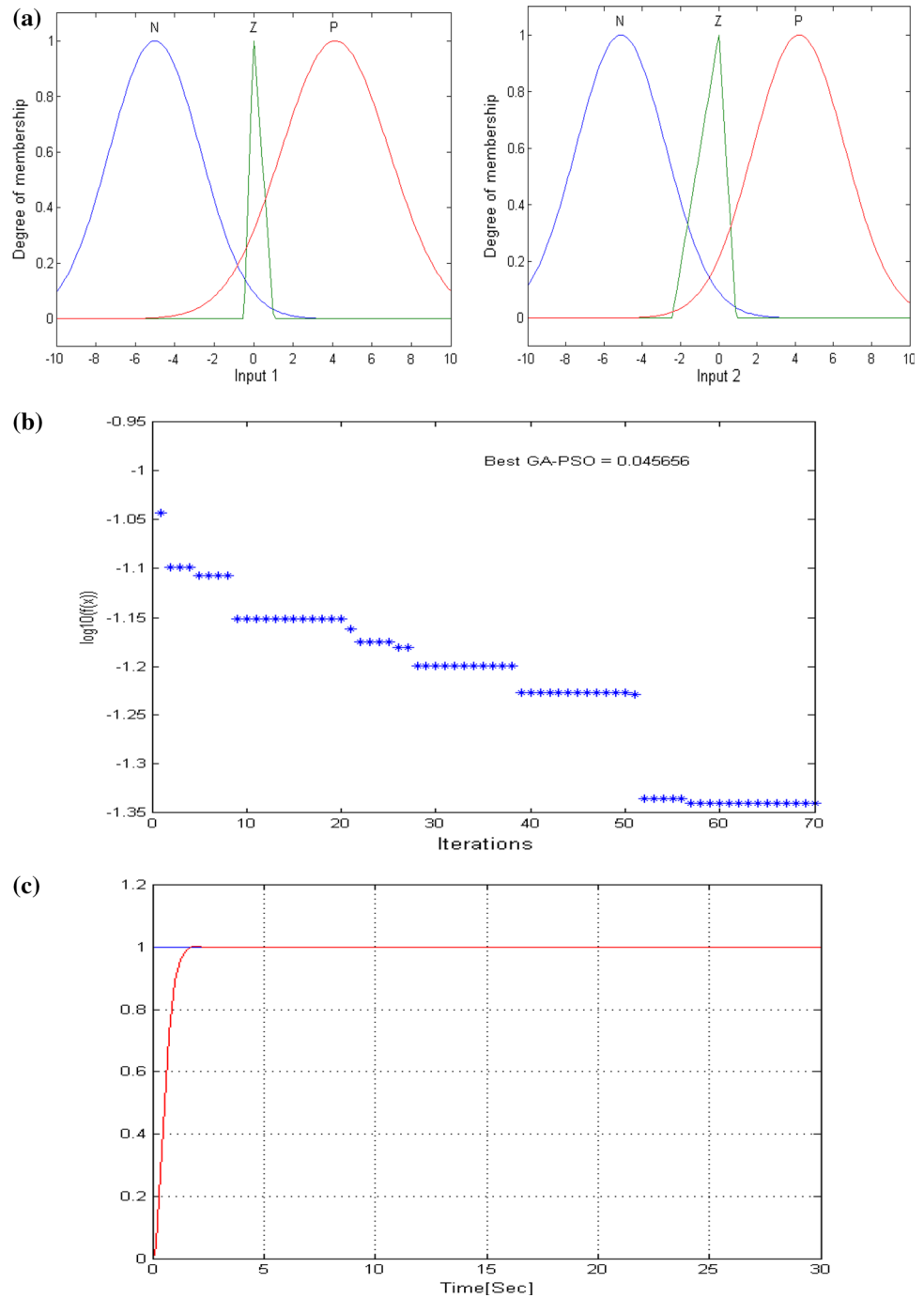
This section presents the simulation results of the proposed method called hybrid PSO-GA to obtain the Type-1 FLC applied to Plant 1. Table 11 contains the configuration values of the hybrid method, the execution time of hybrid PSO-GA and the average error for each configuration. The first row shows the best controller obtained by the proposed method.

Figure 10a shows the membership functions of input 1 and input 2 of the optimal FLC, Fig. 10b shows the behavior of the individuals/particles of the hybrid method giving the best FLC to control Plant 1 and Fig. 10c shows FLC Type-1 result for Plant 1 obtaining the control before 10 s in a stable manner.

Continuing with the simulation results using the hybrid method for Plant 1 we present the optimal Type-2 FLC, showing in Table 12 the configuration values of the hybrid PSO-GA method, the execution time and the average error of each configuration. The first row shows the best Type-2 controller obtained by the hybrid PSO-GA.

Figure 11a shows the membership functions of input 1 and input 2 of the optimal FLC, Fig. 11b shows the individuals/particles behavior of the hybrid PSO-GA method giving the best FLC to control the Plant 1 and Fig. 11c shows FLC Type-2 result for Plant 1 obtaining the control before 5 s with a large overshoot and a small undershoot.

Fig. 12 **a** Input 1 and input 2 membership functions of the Type-1 FLC, **b** behavior of the individuals/particles of the hybrid PSO-GA method and **c** closed-loop response of Plant 2 with the optimized FLC



4.6 Optimal FLC results obtained by the hybrid PSO-GA method for Plant 2

This section presents the simulation results of the proposed method called hybrid PSO-GA to obtain the Type-1 FLC applied to Plant 2. Table 13 contains the configuration values of the hybrid method, the execution time of hybrid

PSO-GA and the average error for each configuration. The second row shows the best controller obtained by the proposed method.

Figure 12a shows the membership functions of input 1 and input 2 of the optimal FLC, Fig. 12b shows the behavior of the individuals/particles of the hybrid method giving the best FLC to control the Plant 1 and Fig. 12c

Table 14 Results of the Type-2 FLC obtained by hybrid method for Plant 2

No.	Individuals/particles	Iterations/generations	C1	C2	Inertia	Percentage ramp	Cross	Mut	Time execution	Average error
1	90	75	0.6423	0.5047	0.2825	0.9	0.5	0.1	11:46:50	0.054486
2	100	150	0.8585	0.5916	0.6241	0.7	0.7	0.2	23:52:43	0.055428
3	45	30	0.7364	0.2531	0.2844	0.7	0.6	0.2	1:52:47	0.055463
4	50	45	0.7788	0.2838	0.3317	0.7	0.7	0.1	3:36:40	0.055628
5	50	150	0.0660	0.1742	0.7760	0.7	0.6	0.2	13:24:31	0.056913
6	120	70	0.4290	0.7808	0.4866	0.7	0.5	0.1	14:38:23	0.057324
7	150	80	0.2078	0.5113	0.6797	0.7	0.7	0.1	23:47:11	0.057343
8	80	120	0.6423	0.6964	0.7362	0.7	0.5	0.2	16:18:49	0.057476
9	70	45	0.1723	0.9135	0.9778	0.7	0.5	0.1	5:59:03	0.059703
10	90	35	0.4734	0.0731	0.0842	0.7	0.5	0.2	5:30:13	0.059950

shows the FLC Type-1 result for Plant 2 obtaining the control before 5 s in a stable manner.

Continuing with the simulation results using the hybrid method for Plant 2 we present the optimal Type-2 FLC, showing in Table 14 the configuration values of the hybrid PSO-GA method, the execution time and the average error of each configuration. The first row shows the best Type-2 controller obtained by the hybrid PSO-GA.

Figure 13a shows the membership functions of input 1 and input 2 of the optimal FLC, Fig. 13b shows the individuals/particles behavior of the hybrid PSO-GA method giving the best FLC to control the Plant 2 and Fig. 13c shows the FLC Type-2 result for Plant 1 obtaining the control before 10 s with a small overshoot.

4.7 Comparison of results of the optimization methods

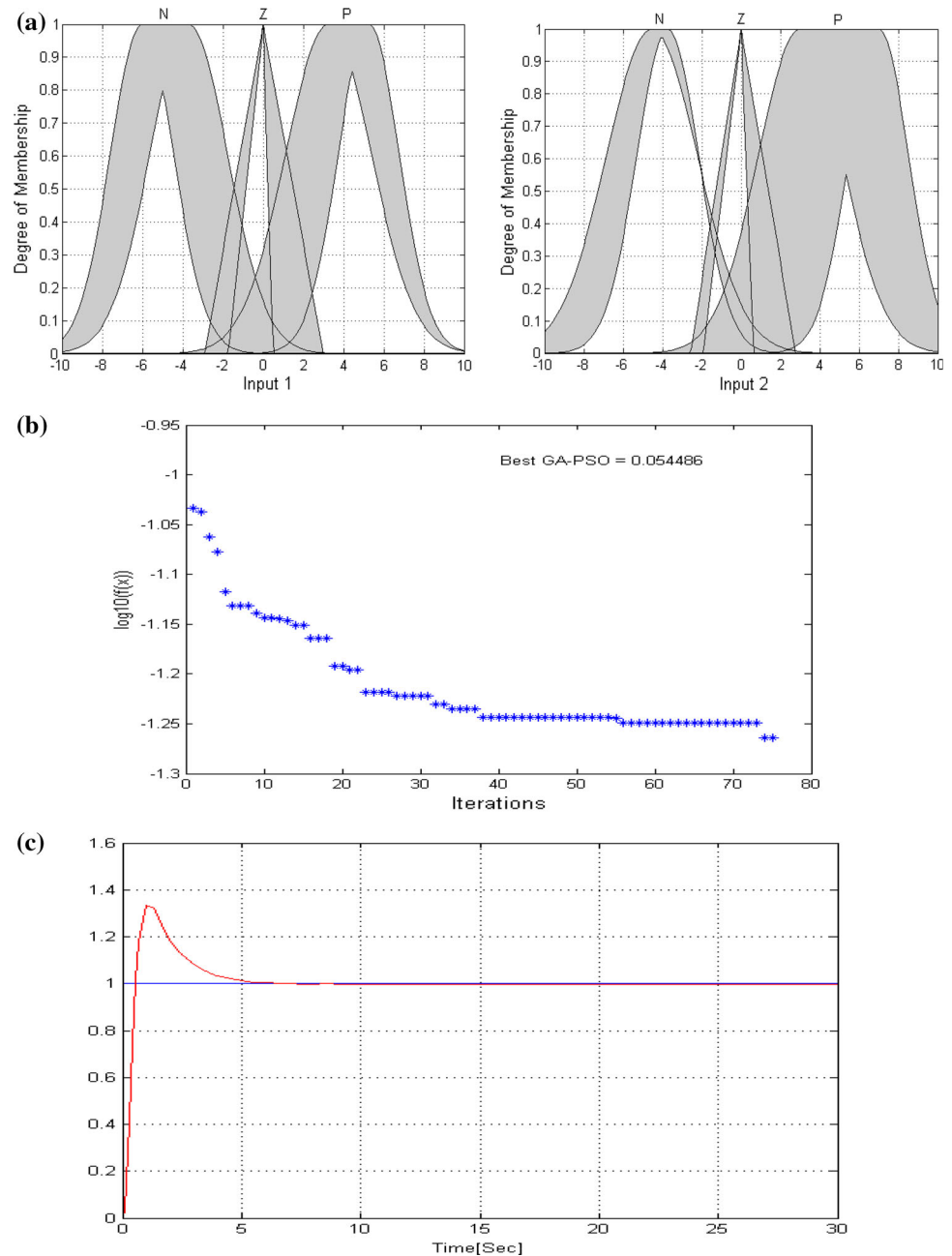
In this section we present a comparison of results among bio-inspired methods, based on the average error obtained for the simulations. Table 12 show the comparison of results of the Best FLC obtained with the GA, PSO and the hybrid PSO-GA method applied to Plant 1 and Plant 2.

In Table 12 we show the comparison of the optimization methods and we can observe that the FLCs of Type-1 and Type-2 show the best results with the PSO and Hybrid PSO-GA, however the Hybrid PSO-GA shows a significant difference in the results with respect to the GA and PSO results obtained separately in Plant 2, which is more complex to solve (Table 15).

5 Conclusions

We described in this paper the application of bio-inspired methods to design optimized fuzzy logic controllers using genetic algorithms, particle swarm optimization and the proposed hybrid PSO-GA method. To test these optimized FLCs we use two different plants. In particular we presented results of a genetic algorithm (GA), particle swarm optimization (PSO) and a hybrid PSO-GA, using two benchmark problems with different levels of complexity. The main result shows that the optimal FLC gets stability in less than 10 s. On the other hand, the FLCs obtained by GA are better than the FLCs obtained by the PSO because the controller obtained with the GA shows more stability and less steady state error with respect to the reference, the plots of the results shows this difference. We can observe that the Type-1 FLCs obtained by GA and PSO show lower steady state errors than the FLCs obtained by the hybrid

Fig. 13 **a** Input 1 and input 2 membership functions of the Type-2 FLC, **b** individuals/particles behavior of the hybrid PSO-GA method and **c** closed-loop response of Plant 2 with the optimized FLC



PSO-GA due to the simplicity of the problem. However the Type-1 and Type-2 FLCs obtained by the proposed hybrid method applied to Plant 2 improved with respect of the FLCs obtained by GA and PSO obtaining lower steady state error and better stability, that we can observe in the simulation results. Based in this fact we

expect that the hybrid method will obtain better results when used in more complex problems.

We have achieved satisfactory results with the GA, PSO and the hybrid PSO-GA; the next step is to solve the problem in a perturbed environment and considering multiple objective optimization to obtain better results in more

Table 15 Comparison of the average error of the Best FLC obtained by the optimization methods

No	Method	Type of FLC	Time exec	Average error
Plant 1				
1	PSO	Type-1	0:26:26	0.081179
2	PSO	Type-2	7:05:20	0.080280
3	GA	Type-1	0:23:01	0.071366
4	GA	Type-2	5:56:25	0.067651
5	Hybrid PSO-GA	Type-1	1:46:07	0.109560
6	Hybrid PSO-GA	Type-2	11:46:36	0.132233
Plant 2				
1	PSO	Type-1	0:34:32	0.131933
2	PSO	Type-2	8:51:28	0.083388
3	GA	Type-1	0:52:13	0.070636
4	GA	Type-2	1:26:11	0.079192
5	Hybrid PSO-GA	Type-1	0:50:02	0.045656
6	Hybrid PSO-GA	Type-2	11:46:50	0.054486

complex problems. Moreover, in the future we will extend the results for nonlinear systems.

References

1. Abd-El-Wahed WF, Mousa AA, El-Shorbagy MA (2011) Integrating particle swarm optimization with genetic algorithms for solving nonlinear optimization problems. *J Comput Appl Math* 235(5):1446–1453
2. Alcalá R, Alcalá-Fdez J, Herrera F (2007) A proposal for the genetic lateral tuning of linguistic fuzzy systems and its interaction with rule selection. *IEEE Trans Fuzzy Syst* 15(4):616–635
3. Alcalá R, Gacto MJ, Herrera F, Alcalá-Fdez J (2007) A multi-objective genetic algorithm for tuning and rule selection to obtain accurate and compact linguistic fuzzy rule-based systems. *Int J Uncert Fuzziness Knowl Based Syst* 15(5):539–557
4. Angeline PJ (1998) Evolutionary optimization versus particle swarm optimization: philosophy and performance differences. *Evolutionary programming VII, Lecture Notes in Computer Science* 1447, Springer, pp 601–610
5. Angeline PJ (1998) Using selection to improve particle swarm optimization. In: *Proceedings 1998 IEEE world congress on computational intelligence, Anchorage, Alaska, IEEE*, pp 84–89
6. Casillas J, Cordon O, del Jesús MJ, Herrera F (2005) Genetic tuning of fuzzy rule deep structures preserving interpretability and its interaction with fuzzy rule set reduction. *IEEE Trans Fuzzy Syst* 13(1):13–29
7. Chi Z, Yan H, Pham T (1996) *Fuzzy algorithms: with applications to image processing and pattern recognition*. World Scientific, Singapore
8. Cordon O, Gomide F, Herrera F, Hoffmann F, Magdalena L (2004) Ten years of genetic fuzzy systems: current framework and new trends. *Fuzzy Sets Syst* 141(1):5–31
9. Driankov D, Hellendoorn H, Reinfrank M (1993) *An introduction to fuzzy control*. Springer, Berlin
10. Engelbretht AP (2005) *Fundamentals of computational swarm intelligence*, England, Wiley, pp 5–129
11. Eberhart RC, Kennedy J (1995) A new optimizer using particle swarm theory. In: *Proceedings of the sixth international symposium on micromachine and human science, Nagoya, Japan*, pp 39–43
12. Fogel DB (1994) An introduction to simulated evolutionary optimization. *IEEE Trans Neural Netw* 5(1):3–14
13. Fukao T, Nakagawa H, Adachi N (2000) Adaptive tracking control of a nonholonomic mobile robot. *IEEE Trans Robotics Automat* 16(5):609–615
14. Kao Y-T, Zahara E (2008) Hybrid genetic algorithm and particle swarm optimization for multimodal functions. *Appl Soft Comput* 8(2):849–857
15. Kennedy J, Mendes R (2002) The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans Evol Comput* 6(1):58–73
16. Kennedy J, Mendes R (2002) Population structure and particle swarm performance. *Proceedings of IEEE conference on evolutionary computation*, In, pp 1671–1676
17. Kim DH (2011) Hybrid GA–BF based intelligent PID controller tuning for AVR system. *Appl Soft Comput* 11(1):11–22
18. Lee TH, Leung FHF, Tam PKS (1999) Position control for wheeled mobile robot using a fuzzy controller. *IEEE, Los Alamitos*, pp 525–528
19. Liang Q, Mendel JM (2000) Interval Type-2 fuzzy logic systems: theory and design. *IEEE Trans Fuzzy Syst* 8(5):535–550
20. Martínez R, Castillo O, Aguilar LT (2008) Intelligent control for a perturbed autonomous wheeled mobile robot using Type-2 fuzzy logic and genetic algorithms. *J Automat Mobile Robotics Intell Syst* 2:1–10
21. Martínez R, Castillo O, Aguilar LT, Rodríguez A (2009) Optimization of Type-2 fuzzy logic controllers for mobile robots using evolutionary methods. In: *Proceedings of the 2009 IEEE international conference on systems, man, and cybernetics, San Antonio, TX, USA*, pp 4909–4914
22. Marinakis Y, Marinaki M (2010) A hybrid genetic: particle swarm optimization algorithm for the vehicle routing problem. *Expert Syst Appl* 37(2):1446–1455
23. Melin P, Castillo O (2004) A new method for adaptive control of non-linear plants using Type-2 fuzzy logic and neural networks. *Int J General Syst* 33(2): 289–304, 1563–5104
24. Montiel O, Castillo O, Melin P, Sepúlveda R (2005) Black box evolutionary mathematical modeling applied to linear systems. *Int J Intell Syst* 20(2):293–311

25. Montiel O, Castillo O, Melin P, Díaz AR, Sepúlveda R (2007) Human evolutionary model: a new approach to optimization. *Inform Sci* 177(10):2075–2098
26. Montiel O, Sepulveda R, Melin P, Castillo O, Porta Garcia MA, Meza Sanchez IM (2008) Performance of a simple tuned fuzzy controller and PID controller on a DC motor. In: *Foundations of computational intelligence, IEEE symposium series on computational intelligence (FOCI'07)*, Honolulu, Hawaii, USA, pp 531–537, April 1–5
27. Muhammad S, Saira Z, Muddassar F (2009) A hybrid GA-PSO fuzzy system for user identification on smartphones, GECCO'09. In: *Proceedings of the 11th annual conference on genetic and evolutionary computation, Montreal-Canada*, pp 1617–1624, July 8–12
28. Niu Q, Jiao B, Gu X (2008) Particle swarm optimization combined with genetic operators for job shop scheduling problem with fuzzy processing time. *Appl Math Comput* 205(1):148–158
29. Pedrycz W (1996) *Fuzzy modelling: paradigms and practice*. Kluwer Academic Press, Dordrecht
30. Sepúlveda R, Montiel O, Castillo O, Melin P (2010) Optimizing the MFs in Type-2 fuzzy logic controllers, using the human evolutionary model. In: *International review of automatic control (IRE-ACO), theory and applications*, volume 3, no. 1, Italy, pp 1–10
31. Tripathi PK, Bandyopadhyay S, Kumar Pal S (2007) Multi-Objective Particle Swarm Optimization with time variant inertia and acceleration coefficients. *Inf Sci* 177(22):5033–5049
32. Valdez F, Melin P, Castillo O (2011) An improved evolutionary method with fuzzy logic for combining particles swarm optimization and genetic algorithms. *Appl Soft Comput Impact Soft Comput Progr Artif Intell* 11(2):2625–2632
33. Herrera F (2008) Genetic fuzzy systems: taxonomy, current research trends and prospects. *Evol Intel* 1:27–46
34. Herrera F, Lozano M, Verdegay JL (1995) Tuning fuzzy logic controllers by genetic algorithms. *Int J Approx Reason* 12(3–4): 299–315
35. Fazendeiro P, de Oliveira JV, Pedrycz W (2007) A Multiobjective design of a patient and anaesthetist-friendly neuromuscular blockade controller. *IEEE Trans Biomed Eng* 54(9): 1667–1678
36. Lin Chih-Min, Li Ming-Chia, Ting Ang-Bung, Lin Ming-Hung (2011) A robust self-learning PID control system design for nonlinear systems using a particle swarm optimization algorithm. *Int J Mach Learn Cybernet* 2(4):225–234
37. Rana S, Jasola S, Kumar R (2012) A boundary restricted adaptive particle swarm optimization for data clustering. *Int J Mach Learn Cybernet*. doi:[10.1007/s13042-012-0103-y](https://doi.org/10.1007/s13042-012-0103-y)
38. Wang Xizhao, He Yulin, Dong Lingcai, Zhao Huanyu (2011) Particle swarm optimization for determining fuzzy measures from data. *Inform Sci* 181(19):4230–4252