International Journal of New Technology and Research (IJNTR)
ISSN: 2454-4116, Volume-6, Issue-11, November 2020 Pages 01-06

Design of Fuzzy Logic Controller Using a Hybrid SFL-Bees Algorithm

Nguyen Duc Hoang

***Abstract*—This paper proposes Hybrid SFL-Bees Algorithm that combines strengths of Shuffled Frog Leaping Algorithm (SFLA) and Bees Algorithms (BA) to tune parameters of the fuzzy logic controller. While SFLA can find optimal solutions quickly because of directive searching and exchange of information, BA has higher random that make it easily escape local optima to find global solutions. Thus Hybrid SFL-Bees Algorithm is able to find optimal solutions quickly like SFLA and escape local optima like BA. Simulation results have shown the effectiveness of the proposed algorithm, its ability to achieve good quality solutions which outperforms the SFLA and BA.**

***Index Terms*—Fuzzy Logic Controller, SFLA, Bees Algorithm.**

I. Introduction

A fuzzy logic controller can be considered as a control expert system which simulates the human thinking. It consists of input and output variables with membership functions, a set of (IF…THEN) rules and an inference system. Designing fuzzy controllers involves choosing input and output variables of the controller, defining membership functions for each input and output variables, constructing the rule base reflecting the linguistic relationship between the inputs and outputs, and tuning the parameters of the membership functions and values of the scaling gains in order to achieve the required performance. Usually, when designing fuzzy controllers these parameters are chosen by trial and error. This manual design method is time-consuming and the control results are not optimal. In order to overcome this problem, optimization techniques are used to tune parameters of fuzzy controller to obtain the best possible solution according to a given criterion or fitness function [1].

Many optimization techniques have been proposed to tune parameters of fuzzy logic controller. In [2], authors used Genetic Algorithm to tune fuzzy control rules. The results showed that the fuzzy control rules obtained greatly improve the behavior of the FLC systems. In [3], authors showed that the PSO can simultaneously tune the premise and consequent parameters of the fuzzy rules for the appropriate design of fuzzy systems. In [4], the Bees Algorithm has been proved to be a useful tool for tuning fuzzy logic controllers to achieve better performance. In [5], Ant Colony Optimization (ACO) was applied to design a fuzzy controller, called ACO-FC. The proposed ACO-FC performance was shown to be better than other evolutionary design methods on one simulation example. In [6], Shuffled Frog Leaping Algorithm was used to optimally tune parameters of a fuzzy logic controller stabilizing a ball and beam system at its equilibrium position. Simulation results show that the designed fuzzy controller is able to balance the ball and beam system around its equilibrium state and the performance of the fuzzy controller is better than that of the well-known LQR controller.

In an attempt to reduce searching time and improve quality of optimal solution, researchers combined the strengths among these optimization algorithms in order to produce hybrid algorithms which are capable of searching for better solutions. For instance, in [7] authors suggested a hybrid algorithm SFLA-GA that combines the advantages of the SFLA, namely an exchange of information among individual members of the group (frogs), and implements a local search using a GA to evaluate the process results. In [8] authors proposed two hybrid Particle Swarm Optimizers combining the idea of the particle swarm with concepts from Evolutionary Algorithms. The hybrid PSOs combine the traditional velocity and position update rules with the ideas of breeding and subpopulations. Simulation results showed that hybrid algorithms have the potential to achieve faster convergence and the potential to find a better solution. In [9] authors proposed a hybrid algorithm HGAPSO combines the new individual generation function of both GA and PSO, which together mimics social behaviors of animals, breeding, and survival of the fittest. The results in temporal sequence production by FCRNN and dynamic plant control problems by TRFN demonstrated the superiority of HGAPSO over GA and PSO.

In this paper, a hybrid algorithm between SFLA and BA has suggested which combines strength of SFLA, namely the ability to find optimal solution rapidly and strength of BA, namely the ability to escape locally optimal solutions to tune parameters of a fuzzy logic controller for balancing a rotary inverted pendulum system in the upright position and compare strengths of this hybrid optimization algorithm with SFLA and BA.

The remaining paper consists of following sections: section II introduces shortly shuffled frog-leaping algorithm and bees algorithm respectively, a hybrid SFL-Bees algorithm which combines the strengths of SFLA and BA is presented in section III. The description how to design and tune parameters of the fuzzy controller is given in section IV. Simulation results to illustrate strengths of the suggested algorithm is presented in section V and the final section is conclusions and future work.

Nguyen Duc Hoang, Department of Control Engineering & Automation, Faculty of Electrical & Electronics Engineering, Ho Chi Minh University of Technology (HCMUT), 268 Ly Thuong Kiet Street, District 10, Ho Chi Minh City, Vietnam. Vietnam National University Ho Chi Minh City, Linh Trung Ward, Thu Duc District, Ho Chi Minh City, Vietnam

II. OVERVIEW OF SFLA AND BEES ALGORITHM OPTIMIZATION TECHNIQUES

A. Shuffled Frog Leaping Algorithm (SFLA)

The SFLA is a meta-heuristic optimization method that mimics the memetic evolution of a group of frogs when seeking for the location that has the maximum amount of available food. The algorithm contains elements of local search and global information exchange. The SFLA involves a population of possible solutions defined by a set of virtual frogs that is partitioned into subsets referred to as memeplexes. Within each memeplex, the individual frog holds ideas that can be influenced by the ideas of other frogs, and the ideas can evolve through a process of memetic evolution. The SFLA performs simultaneously an independent local search in each memeplex using a particle swarm optimization-like method. To ensure global exploration, after a defined number of memeplex evolution steps (i.e. local search iterations), the virtual frogs are shuffled and reorganized into new memeplexes in a technique similar to that used in the shuffled complex evolution algorithm. The flowchart of the SFLA is illustrated in Fig. 1 [6].

The idea updating frog leaping rule which is expressed as :

$$D = r.c(X_b - X_w) \quad (1)$$

$$X_w(new) = X_w + D \quad (2)$$

where *Xb* and *Xw* are identified as the frogs with the best and the worst fitness respectively; *r* is a random number between *0* and *1*; *c* is a constant chosen in the range between *1* and *2*. [6].

B. Bees Algorithm

The Bees Algorithm is an optimization algorithm inspired by the natural foraging behavior of honey bees to find the optimal. The algorithm requires a number of parameters to be set, namely: number of scout bees (n), number of sites selected out of n visited sites (m), number of best sites out of m selected sites (e), number of bees recruited for best e sites (n2), number of bees recruited for the other (m-e) selected sites (n1), initial size of patches (ngh) which includes site and its neighborhood and stopping criterion. The algorithm starts with the n scout bees being placed randomly in the search space. The fitness of the sites visited by the scout bees are evaluated. Bees that have the highest fitness are chosen as "selected bees" and sites visited by them are chosen for neighborhood search. Then, the algorithm conducts searches in the neighborhood of the selected sites, assigning more bees to search near to the best e sites. The bees can be chosen directly according to the fitness associated with the sites they are visiting. The flowchart of the BA is illustrated in Fig. 1. [4].

III. HYBRID SFL-BEES ALGORITHM

In the SFL algorithm, each memeplex evolves independently to locally search at different regions of the solution space. Then, the memeplexes are shuffled and re-divided into new memeplexes in order to globally search through exchanging the information with each other.

From equation (1) $\Rightarrow$ when $Xw \rightarrow Xb$ $(Xw \rightarrow Xb) \Rightarrow D \rightarrow 0$ $\Rightarrow Xw(new) = Xw \rightarrow Xb$, i.e, when difference in position between *Xw* and *Xb* (*Xg*) become small, the change in position of frog *Xw(new)* is small that makes algorithm trapped in local optimum and leads to premature convergence.

Furthermore, information from the best frog is used only once in each update. For Bees Algorithm, information from good bees (Best Sites) are used many times, remaining bees in population will be replaced by random bees.

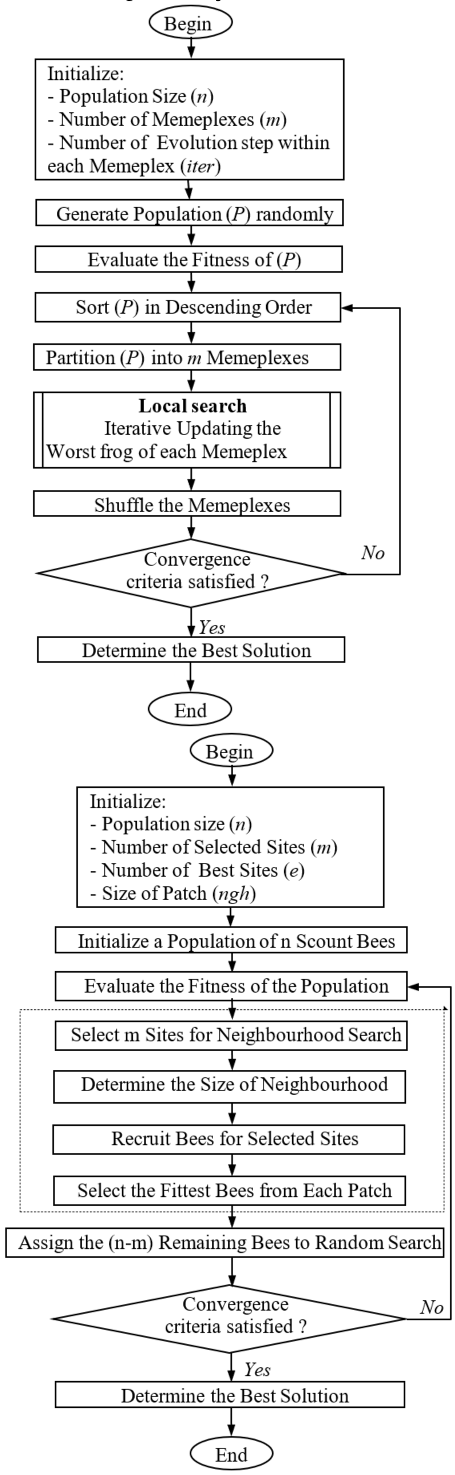

Fig. 1 Flowchart of the SFLA (above) and BA (below)

Thus, BA tries to "exploit" around position of good bees many times comparing with SFLA. That makes BA be able to find solution with better quality but searching time is longer. While SFLA is capable of finding solution quickly thanks to combining local ang global searching.

Based on analyses of the strengths of SFLA and BA, a Hybrid SFL-Bees Algorithm can be produced as follows:

after completing a generation of SFLA, BA will be used with some minor changes. BA updates their new agents around *m* selected best bees and *m* worst bees will be replaced by random bees. (*m*: number of memeplexes in SFLA). The flowchart of the Hybrid SFL-Bees Algorithm is illustrated in Fig. 2.

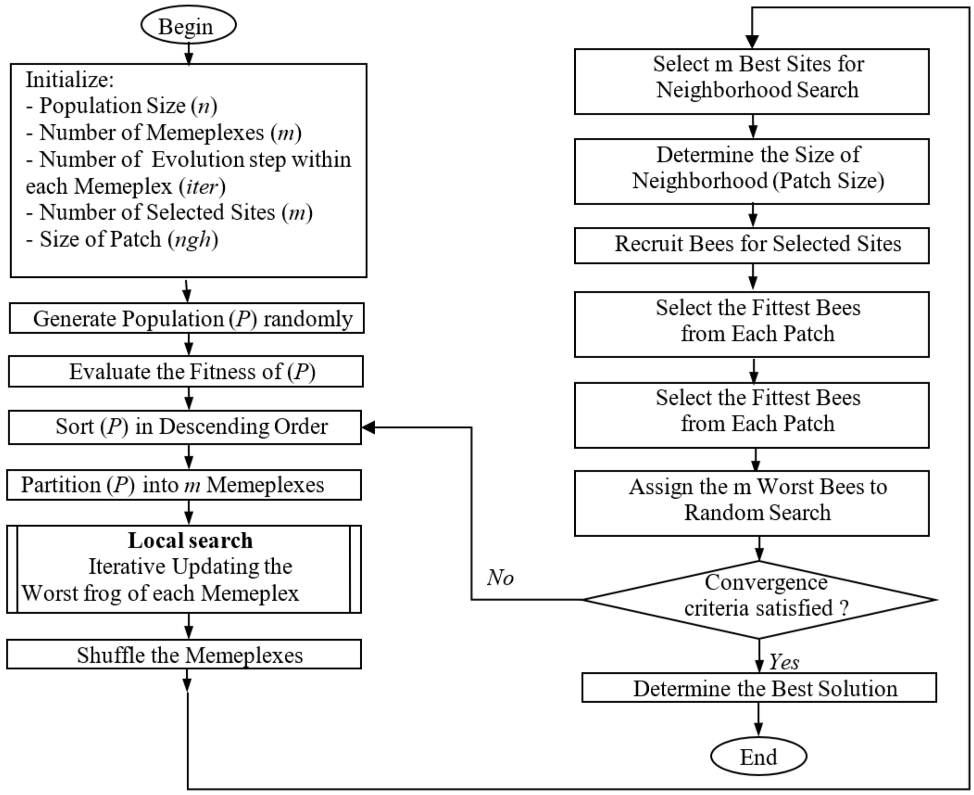

Fig. 2 Flowchart of the Hybrid SFL-Bees Algorithm

IV. Design of Fuzzy Logic Controller

This section describes how to apply the Hybrid SFL-Bees Algorithm for design of fuzzy logic controller to balance the rotary inverted pendulum in the upright position presented in section 2 of [10]. The block diagram of the control system is shown in Fig. 3. The SFLA, BA and Hybrid SFL-Bees Algorithm are used to optimize the fuzzy logic controller obtained from expert knowledge. The controller's inputs and output are chosen to be the 4 states of the process (position and velocity of servo load gear, position and velocity of pendulum arm) and the control moment respectively. Defining 3 linguistic values denoted as NE (Negative), ZE (Zero), and PO (Positive) for each input variables. The linguistic values are qualified by piece-wise membership functions defined in the universe of discourse of [-1, 1] as shown in Fig. 4. The output variable has 9 linguistic values denoted as ZE, Nj (Negative j), Pj (Positive j) (j=1÷4). The dex j represents the strength of the linguistic values such that the higher the index, the stronger the linguistic value.

These output's linguistic values are qualified by singleton membership functions in the universe of discourse of [-1, 1] as illustrated in Fig. 4. Notice that the input's membership functions NE and PO are symmetric about 0. Similarly, the output's membership functions Nj and Pj are symmetric also. By defining symmetric membership functions, the number of adjustable parameters is reduced. As a result, the optimization problem to be solved later is easier.

The Sugeno model is used as the basis of the proposed fuzzy logic controller. The rule base consists of 81 (IF…THEN) rules derived from human knowledge. The complete rule base presented in Table 1. Ideas of rule base system like section IV in [6].

After constructing the structure of the fuzzy controller based on human knowledge, the next step is to optimize its parameters.

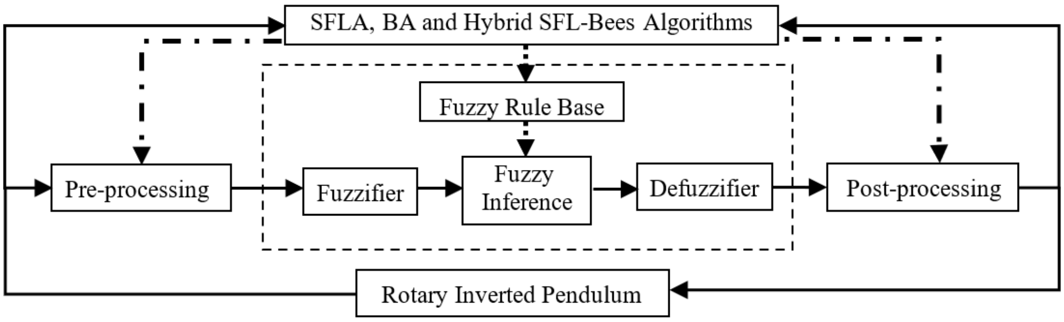

Fig. 3 Ideas tune Parameters of Fuzzy Logic Controller

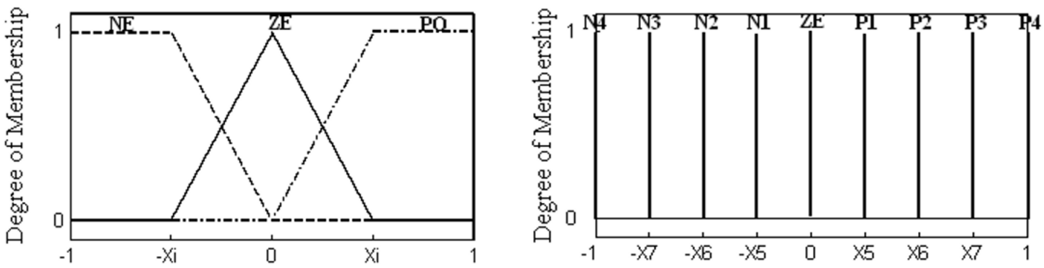

Fig. 4 Input membership functions (i=1÷4) (left) and Output membership functions (right)

TABLE I.

RULE BASE SYSTEM

#	θ	α	$\dot{\theta}$	$\dot{\alpha}$	u	#	θ	α	$\dot{\theta}$	$\dot{\alpha}$	u
1	NE	NE	NE	NE	P1	42	ZE	ZE	ZE	PO	P1
2	NE	NE	NE	ZE	P2	43	ZE	ZE	PO	NE	N2
3	NE	NE	NE	PO	P3	44	ZE	ZE	PO	ZE	N1
4	NE	NE	ZE	NE	ZE	45	ZE	ZE	PO	PO	ZE
5	NE	NE	ZE	ZE	P1	46	ZE	PO	NE	NE	P1
6	NE	NE	ZE	PO	P2	47	ZE	PO	NE	ZE	P2
7	NE	NE	PO	NE	ZE	48	ZE	PO	NE	PO	P3
8	NE	NE	PO	ZE	P1	49	ZE	PO	ZE	NE	ZE
9	NE	NE	PO	PO	P2	50	ZE	PO	ZE	ZE	P1
10	NE	ZE	NE	NE	P2	51	ZE	PO	ZE	PO	P2
11	NE	ZE	NE	ZE	P3	52	ZE	PO	PO	NE	N1
12	NE	ZE	NE	PO	P4	53	ZE	PO	PO	ZE	ZE
13	NE	ZE	ZE	NE	P1	54	ZE	PO	PO	PO	P1
14	NE	ZE	ZE	ZE	P2	55	PO	NE	NE	NE	N4
15	NE	ZE	ZE	PO	P3	56	PO	NE	NE	ZE	N3
16	NE	ZE	PO	NE	ZE	57	PO	NE	NE	PO	N2
17	NE	ZE	PO	ZE	P1	58	PO	NE	ZE	NE	N4
18	NE	ZE	PO	PO	P2	59	PO	NE	ZE	ZE	N3
19	NE	PO	NE	NE	P2	60	PO	NE	ZE	PO	N2
20	NE	PO	NE	ZE	P3	61	PO	NE	PO	NE	N4
21	NE	PO	NE	PO	P4	62	PO	NE	PO	ZE	N3
22	NE	PO	ZE	NE	P2	63	PO	NE	PO	PO	N2
23	NE	PO	ZE	ZE	P3	64	PO	ZE	NE	NE	N2
24	NE	PO	ZE	PO	P4	65	PO	ZE	NE	ZE	N1
25	NE	PO	PO	NE	P2	66	PO	ZE	NE	PO	ZE
26	NE	PO	PO	ZE	P3	67	PO	ZE	ZE	NE	N3
27	NE	PO	PO	PO	P4	68	PO	ZE	ZE	ZE	N2
28	ZE	NE	NE	NE	N1	69	PO	ZE	ZE	PO	N1
29	ZE	NE	NE	ZE	ZE	70	PO	ZE	PO	NE	N4
30	ZE	NE	NE	PO	P1	71	PO	ZE	PO	ZE	N3
31	ZE	NE	ZE	NE	N2	72	PO	ZE	PO	PO	N2
32	ZE	NE	ZE	ZE	N1	73	PO	PO	NE	NE	N2
33	ZE	NE	ZE	PO	ZE	74	PO	PO	NE	ZE	N1
34	ZE	NE	PO	NE	N3	75	PO	PO	NE	PO	ZE
35	ZE	NE	PO	ZE	N2	76	PO	PO	ZE	NE	N2

36	ZE	NE	PO	PO	N1	77	PO	PO	ZE	ZE	N1
37	ZE	ZE	NE	NE	ZE	78	PO	PO	ZE	PO	ZE
38	ZE	ZE	NE	ZE	P1	79	PO	PO	PO	NE	N3
39	ZE	ZE	NE	PO	P2	80	PO	PO	PO	ZE	N2
40	ZE	ZE	ZE	NE	N1	81	PO	PO	PO	PO	N1
41	ZE	ZE	ZE	ZE	ZE						

The parameters to be optimized consist of the input membership function parameters X1,X2,X3, X4 (see Fig. 4), the output membership function parameters X5,X6,X7 (see Fig. 4), and the scaling gains X8, X9,X10, X11,X12 (see Fig. 5). The parameters of the fuzzy controller are optimized according to the quadratic criterion (3), in which the weighting matrices Q and R are positive definite.

$$J_{LQR} = \int_0^{\infty} (x^T Q x + u^T R u) dt \qquad (3)$$

The SFLA, BA or Hybrid SFL-Bees Algorithm discussed are employed to solve this optimization problem

V. Results and Discussions

Matlab and Simulink are used to implement the SFLA, BA or Hybrid SFL-Bees Algorithm based fuzzy controller. Simulation schematic of the rotary inverted pendulum as in Fig. 5.

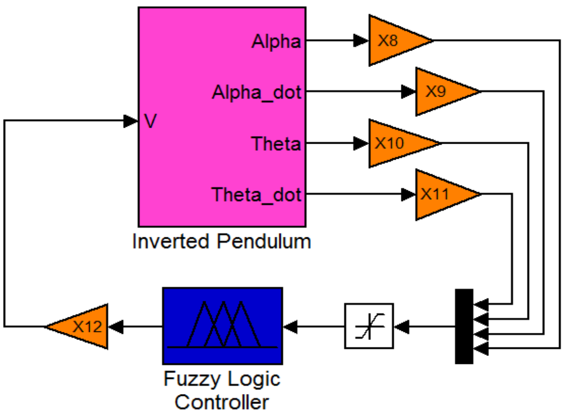

Fig. 5 Simulation schematic of the rotary inverted pendulum

The parameters of FLC that need to be tuned are consist of 12 variables from X1 to X12. Parameters of SFLA, BA and Hybrid SFL-BA are given in Table 2. These parameters are chosen based on many simulations having best results. The weighting matrices in (3) reflecting the desired control performance are chosen to be Q=diag[10,1,20,1] and R=0.1 through a "trial and error" process.

TABLE II.

THE SFLA, BA AND HYBRID SFL-BA PARAMETERS

	N	G	c	m	iter	e	n1	n2	ngh
SFLA	50	500	2	10	10				
BA	50	500		15		5	10	20	0.1
SFL-BA	50	500	2	10	10	5	10	20	0.1

Evolution of quadratic performance index are presented in Fig. 6. Closed responses of system in case of using SFLA as in Fig. 7.

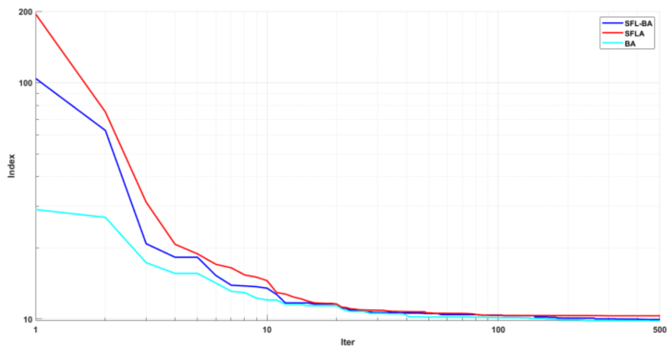

Fig. 6 Evolution of index

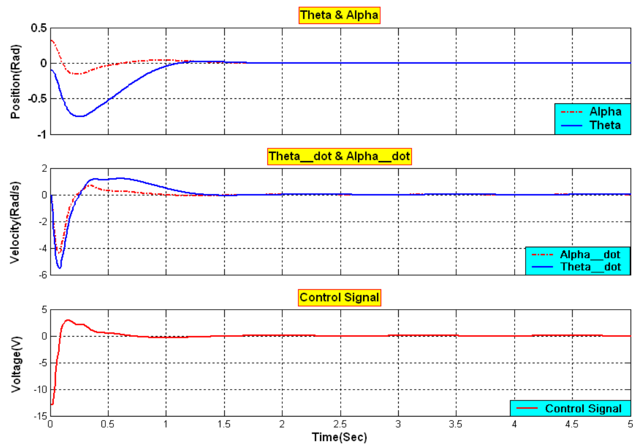

Fig. 7 Closed response of system

Remarks

Results show that convergent rate of Hybrid SFL-Bees Algorithm is faster then that of BA (similar to that of SFLA) while objective function value (index) of Hybrid SFL-Bees Algorithm is greater than value of BA but smaller than value of SFLA (Hybrid SFL-Bees Algorithm has strength of BA, i.e. be able to globally search for solution)

VI. Conclusions

In this paper, a novel algorithm called Hybrid SFL-Bees Algorithm is proposed that combine strengths of SFLA and BA, namely ability to find global optimal solution quickly.

SFLA, BA and Hybrid SFL-Bees Algorithm are employed for tuning the parameters of FLC. Overall, the results indicate that these algorithms can be used in the optimizing the parameters of a fuzzy logic controller to stabilize a rotary inverted pendulum system at its upright equilibrium position. It can be observed that, in terms of convergent rate, the Hybrid SFL-Bees Algorithm is similar to that of SFLA approach, both are faster than BA. Besides, the Hybrid SFL-Bees Algorithm and BA give smaller value of objective function comparing to SFLA.

ACKNOWLEDGMENT

We acknowledge the support of time and facilities from Ho Chi Minh University of Technology (HCMUT), VNU-HCM for this study.

REFERENCES

[1] K.M. Passino, and S. Yurkovich, *Fuzzy logic*, Menlo Park, CA: Addison-Wesley Longman, 1998.

[2] F. Herrera, M. Lozano, and J.L. Verdegay, *Tuning fuzzy logic controllers by genetic algorithms*, International Journal of Approximate Reasoning, 12: 299- 315, 1995.

[3] Feng M, *Particle swarm optimisation learning fuzzy systems design*, Proceedings of the Third International Conference on Information Technology and Applications (ICITA'05) Volume 2, 2005.

[4] D.T. Pham, A.Haj Dqrwish, E.E. Eldukhri, and S. Otri, *Using the Bees Algorithm to tune a fuzzy logic controller for a robot gymnast*, Proceedings of the 3rd Virtual International Conference on Intelligent Production Machines and Systems, 2007.

[5] C.-F. Juang; H.-J. Huang; C.-M. Lu, *Fuzzy Controller Design by Ant Colony Optimization*, IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2007), pp.1-5, 2007.

[6] Duc-Hoang Nguyen and Thai-Hoang Huynh, *A SFLA-Based Fuzzy Controller for Balancing a Ball and Beam System*, Tenth IEEE International Conference on Control, Automation, Robotics and Vision (ICARCV 2008), Hanoi, Vietnam, 17-20, 2008.

[7] Cheng-San Yang, Li-Yeh Chuang, Chao-Hsuan Ke and Cheng-Hong Yang, *A Combination of Shuffled Frog-Leaping Algorithm and Genetic Algorithm for Gene Selection*, Journal of Advanced Computational Intelligence and Intelligent Informatics, Vol.12 No.3, 2008.

[8] Morten Løvbjerg, Thomas Kiel Rasmussen and Thiemo Krink, *Hybrid Particle Swarm Optimiser with Breeding and Subpopulations*, GECCO, 2001.

[9] Chia-Feng Juang, *A Hybrid of Genetic Algorithm and Particle Swarm Optimization for Recurrent Network Design*, IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics, Vol. 34, no. 2, April 2004.

[10] Duc-Hoang Nguyen, Manh-Dung Ngo, *Comparing convergence of PSO and SFLA optimization algorithms in tuning parameters of fuzzy logic controller*, AETA2015, 2015.