International Journal of Computer Applications (0975 – 8887)
Volume 156 – No 11, December 2016

A Comparison of DE and SFLA Optimization Algorithms in Tuning Parameters of Fuzzy Logic Controller

Duc Hoang Nguyen
Faculty of Electrical and Electronics Engineering
HCMC University of Technology
Ho Chi Minh City, Vietnam

ABSTRACT
The paper presents using Differential Evolution (DE) and Shuffled Frog Leaping Algorithm (SFLA) to optimally tune parameters of a fuzzy logic controller stabilizing a rotary inverted pendulum system at its upright equilibrium position. Both the DE and SFLA are meta-heuristic search methods. DE belongs to the class of evolutionary algorithms while SFLA is inspired from the memetic evolution of a group of frogs when seeking for food. In this study, the rule base of the Fuzzy Logic Controller (FLC) is brought by expert experience, and the parameters of the controller, i.e. the membership function parameters and scaling gains, are optimally tuned by the DE and SFLA such that a predefined criterion is minimized. Simulation results show that the designed fuzzy controller is able to balance the rotary inverted pendulum system around its equilibrium state. Besides, convergent rate of SFLA is faster than that of DE but DE has ability to find optimal solutions better than SFLA does.

General Terms
Algorithms.

Keywords
Optimization, DE, SFLA, Fuzzy Controller

1. 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 [7], authors presented an optimized Takagi-Sugeno (TS) fuzzy controller using Differential Evolution (DE) technique for a VSC-HVDC transmission link with a parallel AC line, through the analysis on voltage source converter (VSC) equation in d-q reference frame. The DE technique is used to optimize the rule consequent parameters of a TS fuzzy controller.

In this paper, the author introduces an application of the Differential Evolution and Shuffled Frog Leaping Algorithm in tuning parameters of a fuzzy logic controller for balancing a rotary inverted pendulum system in the upright position and compare the convergent rate as well as the ability to find optimal solution of two these optimization algorithms. This paper is organized as follows. Section 2 introduces the rotary inverted pendulum system and dynamics. Section 3 presents overview of the Differential Evolution and Shuffled Frog Leaping Algorithm. The description how to design and tune parameters of the fuzzy controller is given in section 4. Section 5 shows obtained results and Section 6 concludes this paper.

2. DYNAMICS OF THE ROTARY INVERTED PENDULUM SYSTEM
The rotary inverted pendulum is an ideal experiment when introducing important control concepts such as non-linear systems. It's a natural unstable nonlinear system and a powerful tool to check the control theory and control algorithm. Fig 1. depicts the rotary inverted pendulum system.

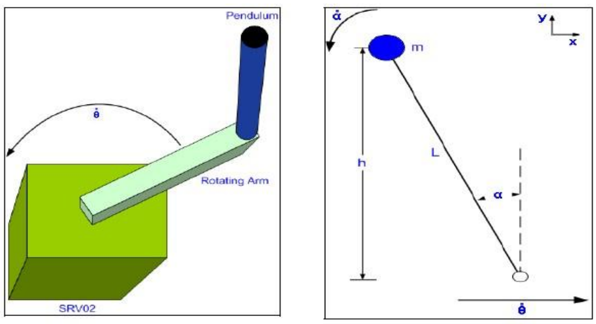

Fig 1: Top View (Left) and Side View (Right) of Rotary inverted pendulum

System dynamic equations:

$$a\ddot{\theta} - bcos(\alpha)\ddot{\alpha} + bsin(\alpha)\dot{\alpha}^2 + G\dot{\theta} = \frac{\eta_m \eta_g K_t K_g}{R_m} V_m \quad (1)$$

$$c\ddot{\alpha} - bcos(\alpha)\ddot{\theta} - dsin(\alpha) = 0 \quad (2)$$

Where:

$$a = J_{eq} + mr^2, b = mLr, c = \frac{4}{3}mL^2, d = mgL,$$

$$E = ac - b^2, G = \frac{\eta_m \eta_g K_t K_m K_g^2 + B_{eq} R_m}{R_m}$$

The parameters of the rotary inverted pendulum system are given in Table 1.

Table 1. Rotary Inverted Pendulum System Model Parameters Used In Simulation

Symbol	Description	Value	Unit
L	Length to Pendulum's Center of mass	0.335/2	m
M	Mass of Pendulum	0.125	kg
R	Length of arm that attaches to SRV02	0.158	m
G	Gravitational Constant	9.81	m/s^2
R_m	Motor Armature Resistance	2.6	Ohm
K_t	Motor Torque Constant	0.00767	N.m/A
K_m	Motor Back-EMF Constant	0.00767	V.s/rd
K_g	Total Gear Ratio	70	
η_m	Motor Efficiency	0.69	
η_g	Gearbox Efficiency	0.90	
B_{eq}	Equivalent Viscous Damping Coefficient as seen at the Load	4e-3	N.m.s/rd
J_{eq}	Equivalent Inertia as seen at the Load	0.0036	$kg.m^2$
V_m	Control Signal	0 ÷ 24	V

Further information about this system, refer to [8].

The purpose of the paper is to design a FLC that will balance the inverted pendulum at the upright position when the initial condition is not zero. Parameters of FLC will be tuned by SFLA and DE methods.

3. OVERVIEW OF DE AND SFLA OPTIMIZATION TECHNIQUES

3.1 Differential Evolution

Differential Evolution grew out of Ken Price's attempts to solve the Chebychev Polynomial fitting Problem that had been posed to him by Rainer Storn [9-10]. DE adopted for various optimization scenarios including constrained, large-scale, multi-objective, multimodal and dynamic optimization, hybridization of DE with other optimizers, and also the multi-faceted literature on applications of DE [11-16].

DE belongs to the class of evolutionary algorithms which use bio-inspired operations of crossover, mutation, and selection on a population in order to minimize an objective function. These operations will be briefly described in this section.

Mutation: Mutation operator is the prime operator of DE and it is the implementation of this operation that makes DE different from other evolutionary algorithms. The mutation process at each generation begins by randomly selecting three individuals in the population. There are many mutation strategies implemented in the DE, however in this paper the following strategy is used.

$$V_i^k = X_{r0}^k + F\left(X_{r1}^k - X_{r2}^k\right) \quad (3)$$

Where X_{r0}^k, X_{r1}^k and X_{r2}^k are randomly selected and satisfy: $X_{r0}^k \neq X_{r1}^k \neq X_{r2}^k$;

Crossover: after the mutation phase is complete, the crossover process is applied to target vector X and mutated vector V in order to generate trial vector U by using the equation (4).

$$U_i^k = \langle U_i^k(j) \rangle = \begin{cases} \langle V_i^k(j) \rangle & if\ \left(rand_j(0,1) \le p_c\right)\ or\ j = rnbr(i) \\ \langle X_i^k(j) \rangle & otherwise \end{cases} \quad (4)$$

Selection: The population for the next generation is selected from the individual in current population and its corresponding trial vector according to the rule (5).

$$X_i^{k+1} = \begin{cases} U_i^k & if\ \ f\left(U_i^k\right) \le f\left(X_i^k\right) \\ X_i^k & otherwise \end{cases} \quad (5)$$

Where *f(.)* is the objective function.

The flowchart of the DE is illustrated in Fig.2. Further information about DE, refer to [12].

3.2 Shuffled Frog Leaping Algorithm

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.

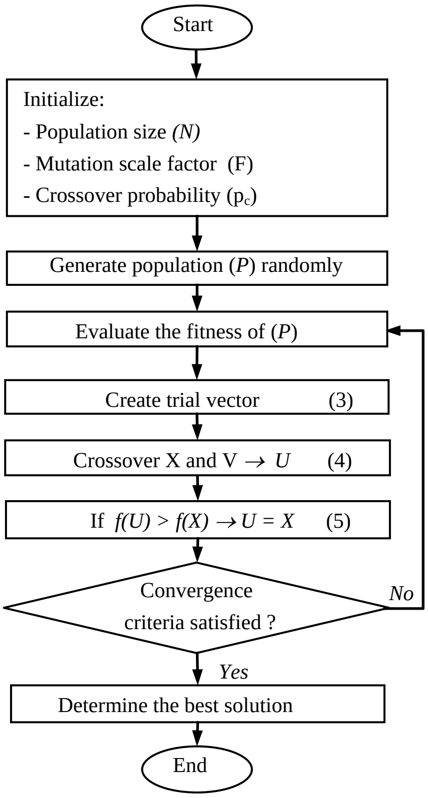

Fig. 2. Flowchart of the DE

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. 3.

The idea updating frog leaping rule which is expressed as :

$$D = r.c(X_b - X_w) \quad (6)$$

$$X_w(new) = X_w + D, \qquad \|D\| \le D_{max} \quad (7)$$

Where X_b and X_w 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]

SFLA has been successfully applied to solve various optimization problems. [17-22].

4. DESIGN OF FUZZY LOGIC CONTROLLER

This section discusses the design of a fuzzy logic controller for balancing the rotary inverted pendulum in the upright position presented in section 2. The block diagram of the control system is shown in Fig. 4.

Defining 3 linguistic values denoted as NE (Negative), ZE (Zero), and PO (Positive) for each input variables.

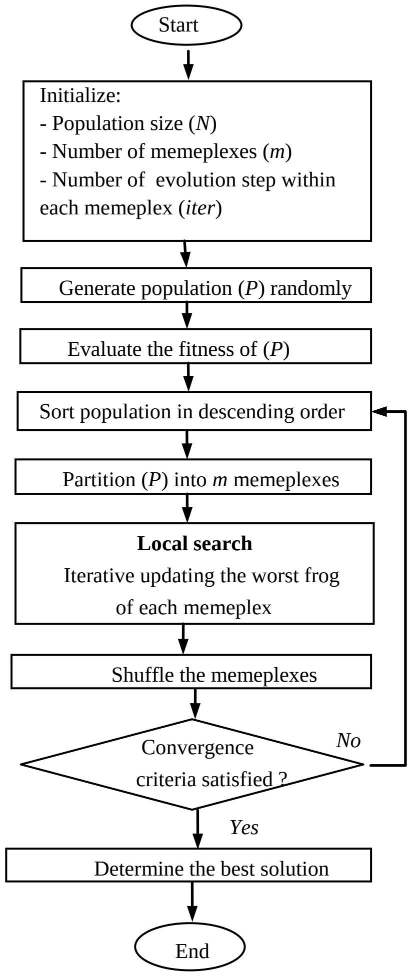

Fig. 3. Flowchart of the SFLA

The linguistic values are qualified by piece-wise membership functions defined in the universe of discourse of [-1, 1] as shown in Fig. 5. The output variable has 9 linguistic values denoted as ZE, Nj (Negative j), Pj (Positive j) (j=1÷4). The index 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. 5. 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 2. 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. The parameters to be optimized consist of the input membership function parameters X1,X2,X3, X4 (see Fig. 5), the output membership function parameters X5,X6,X7 (see Fig. 5), and the scaling gains X8, X9,X10, X11,X12 (Fig. 6). The parameters of the fuzzy controller are optimized according to the quadratic criterion (8), in which the weighting matrices Q and R are positive definite.

$$J_{LQR} = \int_0^\infty (x^T Q x + u^T R u)dt \quad (8)$$

The DE and SFLA methods discussed in section 3 are employed to solve this optimization problem.

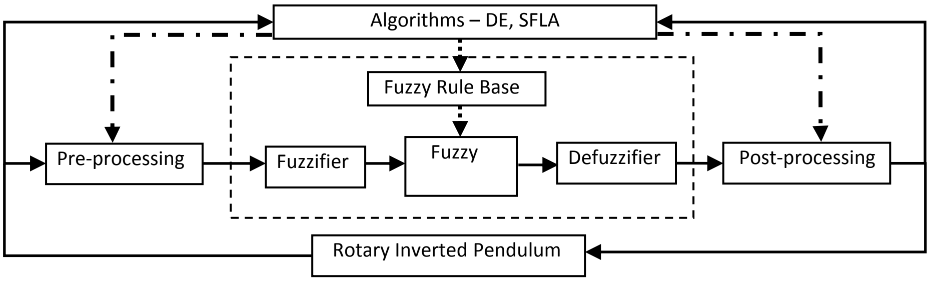

Fig. 4. Ideas tune Parameters of Fuzzy Logic Controller

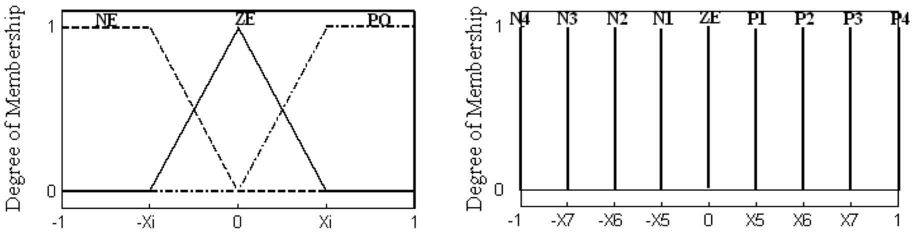

Fig. 5. Input membership functions (i=1÷4) (left) and Output membership functions (right)

Table 2. Rule Base System

#	θ	α	$\dot{\theta}$	$\dot{\alpha}$	u	#	θ	α	$\dot{\theta}$	$\dot{\alpha}$	u	#	θ	α	$\dot{\theta}$	$\dot{\alpha}$	u
1	NE	NE	NE	NE	P1	28	ZE	NE	NE	NE	N1	55	PO	NE	NE	NE	N4
2	NE	NE	NE	ZE	P2	29	ZE	NE	NE	ZE	ZE	56	PO	NE	NE	ZE	N3
3	NE	NE	NE	PO	P3	30	ZE	NE	NE	PO	P1	57	PO	NE	NE	PO	N2
4	NE	NE	ZE	NE	ZE	31	ZE	NE	ZE	NE	N2	58	PO	NE	ZE	NE	N4
5	NE	NE	ZE	ZE	P1	32	ZE	NE	ZE	ZE	N1	59	PO	NE	ZE	ZE	N3
6	NE	NE	ZE	PO	P2	33	ZE	NE	ZE	PO	ZE	60	PO	NE	ZE	PO	N2
7	NE	NE	PO	NE	ZE	34	ZE	NE	PO	NE	N3	61	PO	NE	PO	NE	N4
8	NE	NE	PO	ZE	P1	35	ZE	NE	PO	ZE	N2	62	PO	NE	PO	ZE	N3
9	NE	NE	PO	PO	P2	36	ZE	NE	PO	PO	N1	63	PO	NE	PO	PO	N2
10	NE	ZE	NE	NE	P2	37	ZE	ZE	NE	NE	ZE	64	PO	ZE	NE	NE	N2
11	NE	ZE	NE	ZE	P3	38	ZE	ZE	NE	ZE	P1	65	PO	ZE	NE	ZE	N1
12	NE	ZE	NE	PO	P4	39	ZE	ZE	NE	PO	P2	66	PO	ZE	NE	PO	ZE
13	NE	ZE	ZE	NE	P1	40	ZE	ZE	ZE	NE	N1	67	PO	ZE	ZE	NE	N3
14	NE	ZE	ZE	ZE	P2	41	ZE	ZE	ZE	ZE	ZE	68	PO	ZE	ZE	ZE	N2
15	NE	ZE	ZE	PO	P3	42	ZE	ZE	ZE	PO	P1	69	PO	ZE	ZE	PO	N1
16	NE	ZE	PO	NE	ZE	43	ZE	ZE	PO	NE	N2	70	PO	ZE	PO	NE	N4
17	NE	ZE	PO	ZE	P1	44	ZE	ZE	PO	ZE	N1	71	PO	ZE	PO	ZE	N3
18	NE	ZE	PO	PO	P2	45	ZE	ZE	PO	PO	ZE	72	PO	ZE	PO	PO	N2
19	NE	PO	NE	NE	P2	46	ZE	PO	NE	NE	P1	73	PO	PO	NE	NE	N2
20	NE	PO	NE	ZE	P3	47	ZE	PO	NE	ZE	P2	74	PO	PO	NE	ZE	N1
21	NE	PO	NE	PO	P4	48	ZE	PO	NE	PO	P3	75	PO	PO	NE	PO	ZE
22	NE	PO	ZE	NE	P2	49	ZE	PO	ZE	NE	ZE	76	PO	PO	ZE	NE	N2
23	NE	PO	ZE	ZE	P3	50	ZE	PO	ZE	ZE	P1	77	PO	PO	ZE	ZE	N1
24	NE	PO	ZE	PO	P4	51	ZE	PO	ZE	PO	P2	78	PO	PO	ZE	PO	ZE
25	NE	PO	PO	NE	P2	52	ZE	PO	PO	NE	N1	79	PO	PO	PO	NE	N3
26	NE	PO	PO	ZE	P3	53	ZE	PO	PO	ZE	ZE	80	PO	PO	PO	ZE	N2
27	NE	PO	PO	PO	P4	54	ZE	PO	PO	PO	P1	81	PO	PO	PO	PO	N1

5. SIMULATION RESULTS

5.1 Parameter settings

Matlab and Simulink are used to implement the DE or SFLA based fuzzy controller. Simulation schematic of the rotary inverted pendulum as in Fig. 6.

The parameters of FLC that need to be tuned are divided into 2 groups. Group 1 consist of 5 variables from X8 to X12 need to be tuned (remaining parameters have fixed value : X1=X2=X3=X4 = 0.5; X5 = 0.25, X6 = 0.50, X7 = 0.75). Group 2 consist of 12 variables from X1 to X12 are tuned simultaneously. Parameters of DE and SFLA is given in Table 3. These parameters are chosen based on many simulations having best results. The weighting matrices in (8) 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 3. The DE and SFLA parameters

	N	G	p_c	F, c	m	iter	D_{max}
DE	50	500	0.5	0.8			
SFLA	50	500		2	10	10	∞

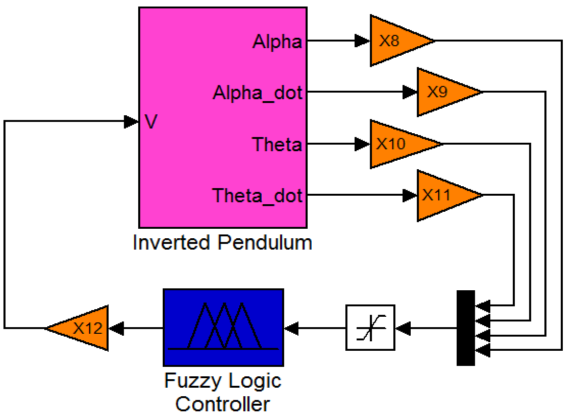

Fig. 6. Simulation schematic of the rotary inverted pendulum

5.2 Results and remarks

Evolution of quadratic performance index in case of tuning 5 and 12 parameters are presented in Fig. 7 and 8, respectively. Closed responses of system in case of tuning 5 and 12 (typically chosen as using SFLA) parameters as in Fig. 9 and 10, respectively.

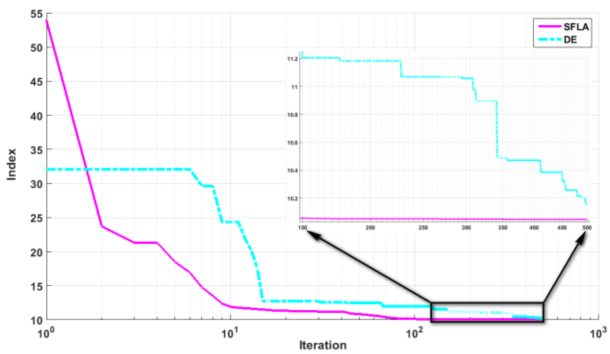

Fig.7. Evolution of index in case of tuning 5 variables

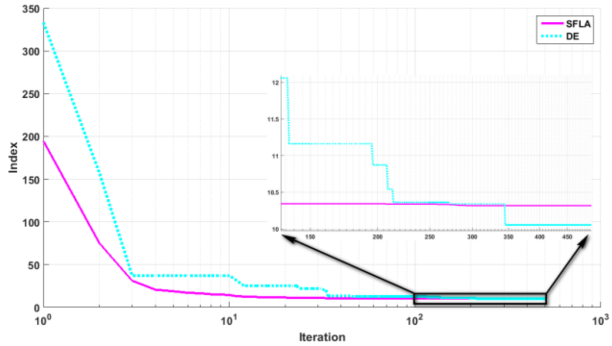

Fig.8. Evolution of index in case of tuning 12 variables

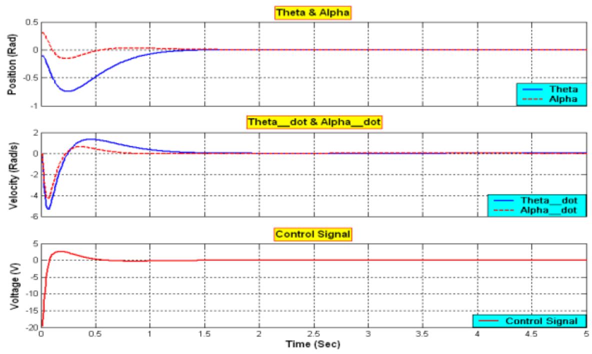

Fig.9. Closed response of system in case of tuning 5 parameters

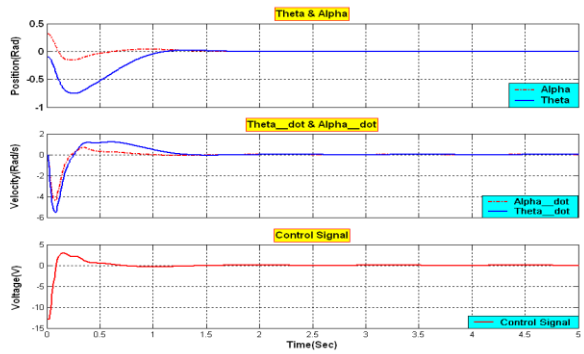

Fig.10. Closed response of system in case of tuning 12 parameters

5.3 Remarks

- In both cases, it can be observed that SFLA has convergent rate is faster than DE.
- From 100th iteration onwards, performance index of SFLA is almost unchanged.
- In the case of tuning 12 parameters, DE has value of objective function is smaller than SFLA.

Above remarks show that SFLA is better than DE in terms of convergent rate. However, SFLA is easily trapped in local optimal solutions while DE can escape them to find better solutions. These remarks can be explained as follows: SFLA can find optimal solutions quickly because of directive searching and exchange of information, DE has higher random that make it easily escape local optima to find global solutions.

6. CONCLUSION

DE and SFLA are techniques have proved to be effective solutions to optimization problems. The objective of this paper is to compare the convergent rate and ability to find optimal solution of these two optimization techniques for a fuzzy logic controller design. Both DE and SFLA are employed for tuning the parameters of FLC in two cases: 5 and 12 parameters of FLC are tuned. Overall, the results indicate that both DE and SFLA 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, SFLA approach is faster than DE. Besides, DE technique has smaller value of objective function in the case of tuning 12 parameters.

7. 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 - Volume 02.

[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] Niranjan Nayak, Sangram Keshari Routray, Pravat Kumar Rout, "Design of Takagi-Sugeno fuzzy controller for VSC-HVDC parallel AC transmission system using differential evolution algorithm", International Journal of Computer Aided Engineering and Technology 2016, Volume 8, Issue 3.

[8] Quanser.: SRV02-Series Rotary Experiment # 7, "Rotary Inverted Pendulum".

[9] R. Storn and K. Price, "Differential Evolution - A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces," Tech. Report, International Computer Science Institute (Berkeley), 1995.

[10] R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces", Journal of Global Optimization, vol. 11, Dec. 1997, pp. 341-359.

[11] Neri, F. & Tirronen, "Recent advances in differential evolution: A survey and experimental analysis", Artificial Intelligence Review 33(1-2): 61-106. V. 2010.

[12] Swagatam Das, Sankha Subhra Mullick, and P. N. Suganthan, "Recent Advances in Differential Evolution – An Updated Survey", Swarm and Evolutionary Computation, Volume 27, April 2016, Pages 1–30.

[13] K. V. Price, R. M. Storn, and J. A. Lampinen, "Differential Evolution: A Practical Approach to Global Optimization", Springer-Verlag, Berlin, Heidelberg, second edition, 2006

[14] M.-F. Han et al,"Differential evolution with local information for neuro-fuzzy systems optimization", Knowledge-Based Systems 44 (2013) 78–89, Elsevier 2013.

[15] Cheng-Jian Lin, Chih-Feng Wu, Hsueh-Yi Lin & Cheng-Yi Yu, "An Interactively Recurrent Functional Neural Fuzzy Network with Fuzzy Differential Evolution and Its Applications", Sains Malaysiana 44(12)(2015): 1721–1728.

[16] Patricia Ochoa, Oscar Castillo and José Soria, "Differential Evolution with Dynamic Adaptation of Parameters for the Optimization of Fuzzy Controllers", Recent Advances on Hybrid Approaches for Designing Intelligent Systems Studies in Computational Intelligence 547, Springer 2014.

[17] Muzaffar Eusuff, Kevin Lansey and Fayzul Pasha, "Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization", Engineering Optimization Vol. 38, No. 2, March 2006, 129–154.

[18] A. Darvishi, A. Alimardani, B. Vahidi , S.H. Hosseinian, "Shuffled Frog-Leaping Algorithm for Control of Selective and Total Harmonic Distortion", Journal of Applied Research and Technology, Volume 12, Issue 1, February 2014, Pages 111–121.

[19] Kaushik Kumar Bhattacharjee , S.P. Sarmah, "Shuffled frog leaping algorithm and its application to 0/1 knapsack problem", Applied Soft Computing, Volume 19, June 2014, Pages 252–263.

[20] Yi Han, et al,"Shuffled Frog Leaping Algorithm for Preemptive Project Scheduling Problems with Resource Vacations Based on Patterson Set", Journal of Applied Mathematics,Volume 2013 (2013), Article ID 451090, 15 pages.

[21] Daniel Mora-Melia, Pedro L. Iglesias-Rey, F. Javier Martínez-Solano and Pedro Muñoz-Velasco, "The Efficiency of Setting Parameters in a Modified Shuffled Frog Leaping Algorithm Applied to Optimizing Water Distribution Networks", Water 2016, 8, 182, MDPI.

[22] Dina M. Said, Nabil M. Hamed, Almoataz Y. Abdelaziz, "Shuffled Frog Leaping Algorithm for Economic Dispatch with Valve Loading Effect", International Electrical Engineering Journal, Vol 7 No 3, 30 JUL, 2016.