International Journal of Computer Applications (0975 – 8887)
Volume 39– No.4, February 2012

An Adaptive Fuzzy Logic Controller Trained by Particle Swarm Optimization for Line of Sight Stabilization

Shahida Khatoon
Department of Electrical Engineering,
F/O engineering and Technology,
Jamia Millia Islamia, New Delhi-25, India

Ravinder Singh
Lastec
DRDO, New Delhi-25, India

ABSTRACT

The design, operation and control of stabilization -tracking systems has been a challenging task for the scientists and engineers with the present day requirements of modern aged sophistication of these systems. The conventional control concepts have been outplayed by the optimal control techniques with the evolution of the modern control theory. Moreover, due to the problems associated with modern control techniques have motivated the engineers to apply intelligent technique to circumvent these difficulties. An attempt has been made to design and feasibility of an adaptive Particle Swarm Optimization (PSO) based fuzzy logic controller for stabilization-tracking system. The simulation results obtained in the study demonstrate the feasibility of the designed controllers.

Keywords

Fuzzy logic controller, Line of sight stabilization, adaptive particle swarm optimization

1. INTRODUCTION

Stabilization-tracking systems are multidisciplinary in nature and are required to maintain the orientation of optical sensors (payload), so that they are pointed in application dependent direction and held steady in inertial space along selected orientation. Thus stabilization tracking system precisely control the position of sensor's line of sight (LOS) and provides isolation from base foundation dynamics or host platform generated vibration. The movement of vehicle carrying the sensors can introduce errors in the attempts to maintain a track on a given target, thus the modulation transfer function of an electro-optical system mounted on a mobile platform decreases very rapidly with an increase in the disturbance on a LOS. The servo control loops determine the ultimate behavior of stabilization-tracking systems.

In recent years, fuzzy logic based control schemes have become a topic of great interest. It is one of the active areas of research in the applications of fuzzy set theory, because conventional controllers cannot be used due to lack of knowledge regarding the input- output models [1,3]. These methods have been used successfully in many real-world applications [2,8]. Fuzzy logic based controllers are generally applicable to systems that do not accurate mathematical models and require only qualitative guidance through experienced operators for their implementation [4]. Fuzzy logic is conceived as a tool for dealing with uncertainty but has proven to be an excellent choice for many control system applications. The fast developments in the technology relating to hardware systems required for implementing fuzzy logic based controllers like fuzzy memory devices, fuzzy computers etc. have made a way for effective utilization of fuzzy logic even in complex and ill defined processes that can be controlled by an expert designer without the knowledge of their underlying dynamics [1,5].

Classic fuzzy modeling and controlling structures are based on extensive expertise of the designer and some heuristic pre-knowledge, in order to avoid these shortcomings fuzzy logic and neural network structure are operated together. This approach involves merging or fusing fuzzy systems and neural networks into an integrated system to reap the benefits of both[1]. The most important neuro-fuzzy model is the Mamdani model. The Mamdani Model incorporates an idea that local dynamics of a non linear system can be represented by different linear dynamic models. In applications of fuzzy-neural networks the learning capability of the neural networks is used for determining optimum values of fuzzy antecedent (membership) and consequent (rule) parameters.

Many supervised learning algorithms have been proposed in the literature for fuzzy controller training [10-13]. The back-propagation (BP) learning algorithm [14] is widely used for training neuro-fuzzy networks by means of error propagation using calculus of variance. However the BP learning algorithm is a powerful training technique that can be applied in networks with feed forward structure to transform them into adaptive systems. But the algorithm may reach the local minima and the global solution may never be found because the steepest descent optimization technique is used in BP training to minimize the error function.

Evolutionary computation has inspired new designs and models for physical systems. Evolutionary computation provides a more robust and efficient approach for solving complex real-world problems. Recently, due to its global optimization capability, the genetic algorithm (GA) has become a useful tool for the automatic design of fuzzy control systems. The GA is employed where desired outputs are not available or costly to obtain. However, the learning performance of the GA may not be satisfactory for complex problems. In most of real-time control problems, the system model is unknown, gradient computation is impossible. Therefore, particle swarm optimization technique is an effective method to avoid this vital drawback.

The particle swarm algorithm is an optimization technique inspired by social interaction observed among animals such as bird flocking and fish schooling. The kind of social interaction modeled within a PSO is used to guide a population of individuals (so called particles) moving towards the most promising area of the search space [9]. The PSO conducts searches using the population of particles which correspond to individuals in the genetic algorithm.

2. FUZZY LOGIC OVERVIEW

FL was conceived as a better method for sorting and handling data but has proven to be a excellent choice for many control system applications since it mimics human control logic. It can be built into anything from small, hand-held products to large computerized process control systems. It uses an imprecise but very descriptive language to deal with input data more like a human operator. It is very robust and forgiving of operator and data input and often works when first implemented with little or no tuning.

3. MAMDANI FUZZY CONTROLLER

The input variables are error(e) and the rate of change of error(ė). The output of the fuzzy controller gives the input variables are error(e) and the rate of change of error(ė). The output of the fuzzy controller gives the incremental control force(u). The membership functions were defined using the standard Gaussian function,

$$f(x, \sigma, c) = \exp[-(x-c)2 / 2\sigma 2]$$

The proposed membership functions are continuous in the universe of discourse and therefore the inferred control output is smooth.The scaling process is a trial-and-observation procedure. The selection of membership function parameters(c and σ) for different fuzzy sets of a variable is very important issue. The system performance is very sensitive to this selection. Since this selection depends heavily on the knowledge base of the designer, the experience of the designer is very vital.

4. PARTICLE SWARM OPTIMIZATION OVERVIEW

A population of particles is randomly generated initially. Each particle represents a potential solution and has a position represented by a position vector $\vec{p}$ i. A swarm of particles moves through the problem space, with the moving velocity of each particle represented by a velocity vector $\vec{v}$ i. At each step, a cost function Cfi representing a quality measure is calculated by using as input pi. Each particle keeps track of its own best (local best) position, which is associated with the best fitness it has found so far in a vector $\vec{p}$ lb. Furthermore, the best position among the all particles obtained so far in the population is kept track of as global best $\vec{p}$ gb. At each step n, by using the individual best position, $\vec{p}$ lb, and global best position, $\vec{p}$ gb a new velocity for the ith particle is updated by [15,16]

$$\vec{v}\ i(n) = \chi\,(\vec{v}\ i(n-1) + \psi 1\ r2\ (\vec{p}\ lbi - pi\ (n-1))+ \quad (1)$$

$$\psi 2 r2(\vec{p}\ gb - pi(n-1))) \quad (2)$$

where r1 and r2 are uniformly distributed random numbers in the interval of the [0,1].

ψ1 and ψ2 are positive constant learning rates .χ is called the constriction factor and is defined by

$$\chi = \frac{2}{|2- \psi - \sqrt{\psi^2-4\psi}\ |}, \ \ldots\ldots \quad (3)$$

$$\psi = \psi 1 + \psi 2,\quad \psi>4$$

Based on the updated velocities, each particle changes its position according to the following

$$\vec{p}\ i(n) = \vec{p}\ i\ (n-1) + \vec{v}\ i\ (n)\ (4)$$

5. FUZZY CONTROLLER STRUCTURE AND PSO OPTIMIZATION

A standard Mamdani fuzzy inference system [17] with two inputs (error and derivative of error) and one output (control output), 49 first order Mamdani rules, implication min., aggregation max., defuzzification centroid and Gaussian membership functions (7 for error, 7 for derivative of error and 7 for control output) is used in this study. Each membership function has two parameters – mean and variance. Hence total 42= (2*(7+7+7)) parameters have been selected as a position vector having 42 positions. The purpose of this study is to train these 42 parameters using PSO. The ith particle's cost function, Cfi, is computed according to the following equation in this study

$$\text{Cfi} = \frac{1}{2N}\sum_{m=1}^{N} e_m^2 \quad (5)$$

Where N is the number of discrete time Samples. It is expected to minimize this cost function value, while the PSO algorithm iteration goes forward. All parameters of the FC are updated at every final time. Functions for the input variables (e and ė) and the output variable (u). These are the standard values of the membership function.The rule base forms an important element to process the fuzzified inputs. The expert knowledge is usually in the form of "if-then" rules, which are easily implemented by fuzzy conditional statements in fuzzy logic. The collection of fuzzy control rules constitutes the rule base. The set of rules which we have used in this project are given in the form of a matrix in Table 2. There are total of 49 numbers for all the possible combinations of input fuzzy sets.

Table 1 Parameters of fuzzy membership functions.

Variables	e	ė	u
Fuzzy Sets	cσ	cσ	cσ
nb	-1.0 0.35	-1.0 0.141	-1.0 0.141
Nm	-0.25 0.1	-0.66 0.141	-0.57 0.142
Ns	-0.1 0.04	-0.2 0.12	-0.15 0.1
Z	0.0 0.013	0.0 0.05	0.0 0.007
Ps	0.1 0.04	0.1 0.04	0.1 0.04
pm	0.25 0.1	0.66 0.141	0.57 0.142
pb	1.0 0.35	1.0 0.141	1.0 0.141

Particle Swarm optimization Algorithm

Initialize

- Set constants kmax, c1, c2.
- Randomly initialize particle positions x0iε D in IRn for i=1,…..,p.
- Randomly initialize particle velocities 0≤v0i≤v0max for i=1,……,p.
- Set k=1

Optimize

- Evaluate function value fki using design space coordinates xk.
- If fki ≤ fbesti then fbesti=fki,pki=xki.
- If fki≤fbestg then fbestg=fki,pkg=xki.
- If stopping condition is satisfied then goto 3.
- Update all particle velocities vki for i=1,…..,p with rule (2.1).
- Update all particle positions xki for i=1,…..,p with rule (2.2).
- Increment k.
- Goto 2(a).

Terminate.

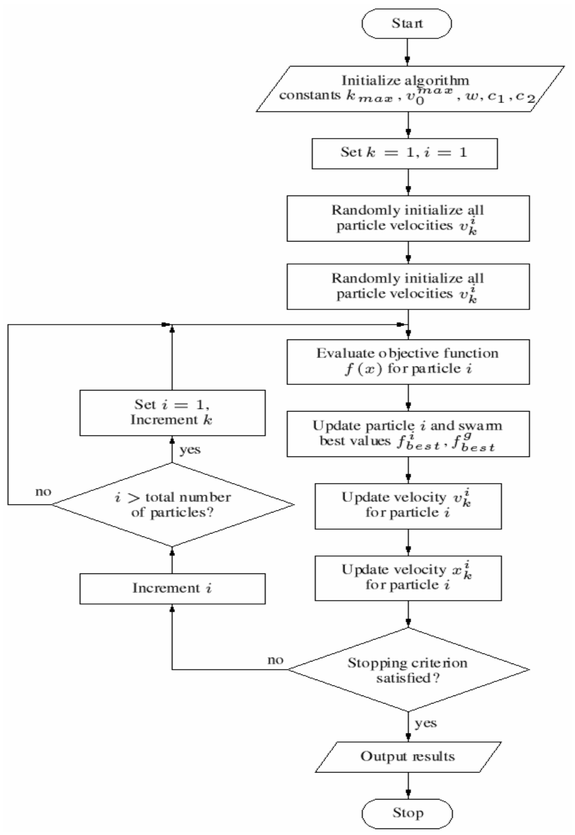
Start
Initialize algorithm constants $k_{max}, v_0^{max}, w, c_1, c_2$
Set $k = 1, i = 1$
Randomly initialize all particle velocities v_k^i
Randomly initialize all particle velocities v_k^i
Evaluate objective function $f(x)$ for particle i
Update particle i and swarm best values f_{best}^i, f_{best}^g
Update velocity v_k^i for particle i
Update velocity x_k^i for particle i
Stopping criterion satisfied?
no
yes
Output results
Stop
Set $i = 1$, Increment k
yes
$i >$ total number of particles?
no
Increment i

Fig. 1 Flow Chart For The PSO

Table 2 Final updated values of the 42 parameters of the Mamdani model

Final updatedvalues of theMamdanimodel

```
gbest(:,:,499)=
Columns 1 through 8
0.0155   0.0132   0.0130   0.0706   0.3500   0.236   0.0130
0.0500
Columns 9 through 16
0.0501   0.0500   0.0500   0.0500   0.0501   0.0501   0.1004
0.1003
Columns 17 through 24
0.1000   0.1000   0.1088   0.1002   0.1024   0.9989   0.7546
0.7622
Columns 25 through 32
1.0000   0.1004   0.9395   -0.9479   1.0000   -0.0467   1.0000
0.3252
Columns 33 through 40
0.7391   -0.0535   0.8855   0.9963   0.8942   0.3070   0.4855   -
0.6341
Columns 41 through 42
0.2294   -0.3357
```

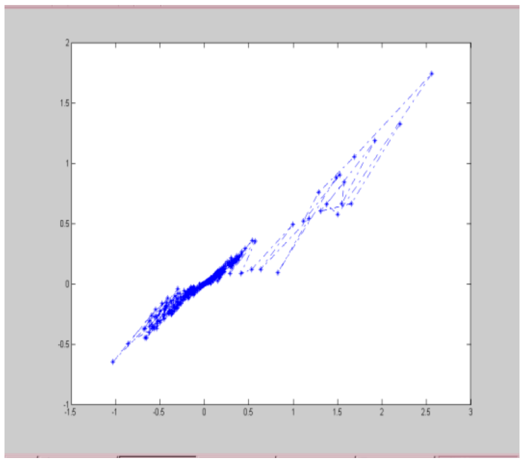

Fig. 2Particle Swarm optimization algorithm result.

In beginning the particles scattered everywhere but finally they get Converge at one position. The present graph is the plotting of particle between their positions and velocities.

x axis- particle position.

y axis – particlevelocity.

6. PROBLEM STATEMENT

In this paper, the plant under consideration consists of a gimbaled payload that is driven by a permanent magnet dc motor [6,7]. A dual axis dynamically tuned gyro is used to sense the inertial angular rate of the gimbal in elevation and azimuth. The relevant parameters of gimbal / electronics system are as follows:

- gimbal inertia = 0.5 kg- m2
- weight of payload= 35 kg
- load pole= 1 Hz

- gimbal resonance=140 Hz
- torque rating=3.5nm (peak)
- torque sensitivity(kt) = 0.786 Nm/A
- backemf constant(kb)=0.786 V/(rad/sec)
- gyro scale factor=5.73 V/rad.
- gyro dynamics, singe pole at 100 Hz
- data acquisition resolution, 16 bits (max. input = ± 10 V)
- dead band due to stiction friction, 10% of the peak torque
- digital-to-analog converter resolution = 16 bits (max. input = ± 10 V)

The design considerations were as follows:

- steady state error for step response is <= 0.1%
- percent overshoot is <= 40%
- rise time, <=50 ms
- typical disturbance frequencies are 0.1 to 0.5 Hz
- typical amplitude of disturbance input is 0.2 rad/sec

Results of the proposed fuzzy controller trained by PSO

Here, in Fig. 3, the block diagram of PSO trained fuzzy controlled stabilization loop is shown.

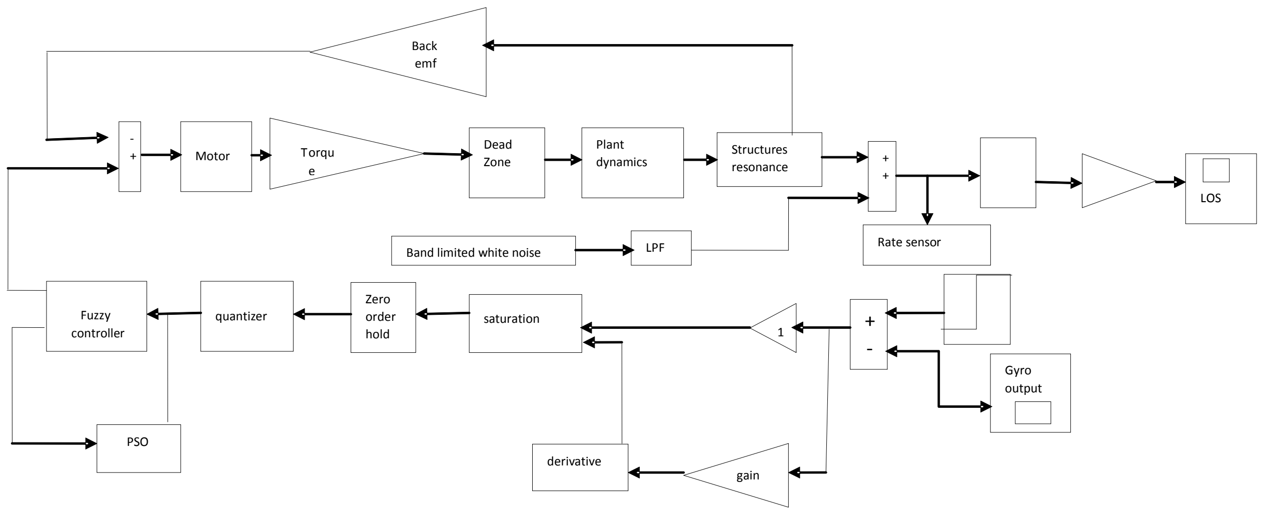

Fig.3 Block diagram of the stabilization loop using PSO trained fuzzy controller

Characteristics	PSO Trained Fuzzy Controller
Rise Time (ms)	36
Settling Time (ms)	53
Peak Overshoot (%)	0.8
Steady State Error (%)	0.1

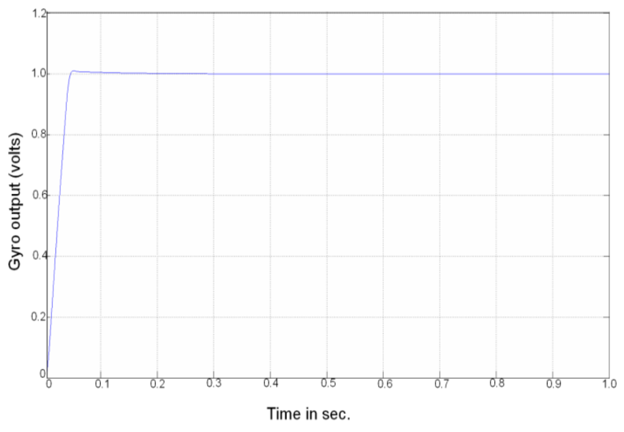

Fig. 4 Step response with saturation (PSO trained fuzzy controller)

Table 3 Error between standard output and observed output which after certain iterations becomes zero

Error between standard output and observed output

```
err1=
Columns 1 through 9
-1.4627 -0.9118 -0.5660  -0.8313  -0.8054  -1.1752  -0.6383 -0.7754 -
1.8635
Columns 10 through 18
-1.0405 -0.6990 -0.5460  -0.9532  -0.9181  -0.2645  -0.6284 -0.1102 -
0.5311
Columns 19 through 27
-0.1628 -0.1406 -0.3670  -0.1165  -0.2596  -0.2220  -0.2132 -0.1889 -
0.3895
Columns 28 through 36
-0.1610 -0.2194 -0.1535  -0.1560  -0.2170  -0.1600  -0.3106 -0.2480 -
0.2039
Columns 37 through 45
-0.2180 -0.2911 -0.1703  -0.1978  -0.0418  -0.0885  -0.2095 -0.1400 -
0.1813
Columns 46 through 54
-0.1150 -0.0518 -0.1191  -0.0626  -0.1824  -0.0474  -0.0843 -0.1295 -
0.0847
Columns 55 through 63
-0.1441 -0.0443 -0.1117  -0.0248  -0.0642  -0.0622  -0.1179 -0.0757 -
0.0554
Columns 64 through 72
-0.0693 -0.0857 -0.0492  -0.0753  -0.0329  -0.0373  -0.0753 -0.1060 -
0.0697
```

Final updatedweights

```
Columns 775 through 783
0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
Columns 784 through 792
0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
Columns 793 through 801
0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 w0 =
      1.3322   1.4054   1.4786   1.5518
      1.3322   1.4054   1.4786   1.5518
w1 =
     -0.3296  -1.1015  -1.1101  -0.8805
      1.2944   0.3489   -0.1621  -0.1158
v  =
     -0.2127   0.0596   0.3320    0.6043
```

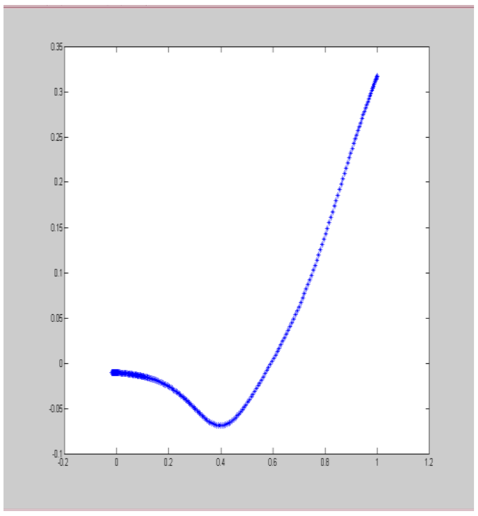

Fig. 5 Fuzzy curves obtained from neuro fuzzy controller.

7. CONCLUSION

This paper presents the results of a Mamdani type fuzzy logic based controller trained using PSO for a real time defense systems. The proposed fuzzy logic based controller eliminates the hit and trial procedure for tuning the fuzzy controller parameters thus reducing the effort and time required in both design and optimization. Moreover , implementation of fuzzy controller trained using PSO is much easier than the traditional methods, especially in embedded systems. The proposed approach may further be improved and can be used for the rejection of torque disturbances on the line of sight.

8. REFERENCES

[1] C.C Lee, " Fuzzy logic in control systems: Fuzzy logic controller – part 1,. IEEE Trans. Syst. Man Cybern. 20(2),404-418 (1990)

[2] C.C Lee, " Fuzzy logic in control systems: Fuzzy logic controller – part 2,. IEEE Trans. Syst. Man Cybern. 20(2),419-435 (1990) Charles Elkan - The Paradoxical Success of Fuzzy Logic.

[3] M. Hellmann- Fuzzy Logic Introduction.

[4] Chun-Yi Su and Yury Stepanenko - Adaptive Control of a Class of Nonlinear Systems with Fuzzy Logic.

[5] Lotfi A. Zadeh - Is There a Need for Fuzzy Logic.

[6] Jiasheng Zhang, Dingwen Yu and Shiqing Qi Structural research of fuzzy PID controllers.

[7] J. A. R. Krishna Moorty, Rajeev Maratheand Hari Babu - Fuzzy controller for line-of-sight stabilization systems.

[8] Constantin von Altrock - Recent Successful Fuzzy Logic Applications in Industrial Automation.

[9] Cihan Karakuzu- Fuzzy controller training using particle swarm optimization for nonlinear system control,ISA Trans. 47, 229-239(2008)

[10] Jang JS, self-learning fuzzy controllers based on temporal back propogation, IEEE Trans. Neural Network 1992;3(5):723-41

[11] Reeves CR. Modern heuristic techniques for combinational problems. New York:Wley:1993.

[12] Lin CT,Lee CSG, Supervised and unsupervised learning with fuzzy similarity for neural network-based fuzzy logic control systems, In:Fuzzy sets,neural network, and soft computing . New York: Van Nostrand Reinhold; 1994, 85-125

[13] Chen J, Rine D. Training fuzzy logic based software components by combining adaptation algorithms.Int J Softw Comput 1998;2(2):48-60

[14] Lin CT, Lee CSG. Neural fuzzy systems: a neuro-fuzzy synergism to intelligent system. Englewood Cliffs (NJ):Prentice Hall;1996.

[15] Parrott Daniel, Li Xiadong. Locating and tracking multiple dynamic optima by a particle swarm model using speciation . IEEE Trans. Evol Comput 2006;10(4):440-58.

[16] Clerc M, Kennedy J. The particle swarm-explosion, stability, and convergence in multidimensional complex space. IEEE Trans Evol Comput 2002;6(1):58-73

[17] Jang J-SR.ANFIS; adaptive networkbased fuzzy inference system. IEEE trans Syst Man Cybern 1993;23(3):665-85.