

# Particle Swarm Optimization Based Adaptive Strategy for Tuning of Fuzzy Logic Controller

Sree Bash Chandra Debnath<sup>1</sup>, Pintu Chandra Shill<sup>2</sup> and Kazuyuki Murase<sup>2</sup>

<sup>1</sup>Department of Electrical and Electronic Engineering, Premier University, 1/a O.R Nijam road, Chittagong, Bangladesh.

<sup>2</sup>Department of System Design Engineering, University of Fukui, 3-9-1 Bunkyo, Fukui 910-8507, Japan

sreebash\_eee@puc.ac.bd, {pintu, murase}@u-fukui.ac.jp

## ABSTRACT

*This paper presents a new method for learning and tuning a fuzzy logic controller automatically by means of a particle swarm optimization (PSO). The proposed self-learning fuzzy logic control that uses the PSO with adaptive abilities can learn the fuzzy conclusion tables, their corresponding membership functions and fitness value where the optimization only considers certain points of the membership functions. To exhibit the effectiveness of proposed algorithm, it is used to optimize the Gaussian membership functions of the fuzzy model of a nonlinear problem. Moreover, in order to design an effective adaptive fuzzy logic controller, an on line adaptive PSO based mechanism is presented to determine the parameters of the fuzzy mechanisms. Simulation results on two nonlinear problems are derived to demonstrate the powerful PSO learning algorithm and the proposed method is able to find good controllers better than neural controller and conventional controller for the target problem, cart pole type inverted pendulum system.*

## KEYWORDS

*Fuzzy Control, Optimization, Particle Swarm Optimization, Fuzzy Rule Base System, and Non Linear System.*

## 1. INTRODUCTION

Fuzzy logic systems (FLSs) are fundamental methodologies to represent and process linguistic information, with mechanisms to deal with uncertainty and imprecision. With such remarkable attributes, FLSs have been widely and successfully applied to control [1]-[3], classification [4] and modeling problem and in a considerable number of applications [5]-[7]. The most of the FLSs are organized based on skilled knowledge that is in the form of rule base behavior derived from imprecise heuristic knowledge of experienced control engineers. Such conventional approach is very time-consuming as it requires trial and error method to generate fuzzy rules and membership functions (MFs). Moreover, conventional design does not guarantee generating optimal or near optimal fuzzy rules and MFs [8]-[9]. Clearly, there is a need for developing an automatic FLSs design procedure that would not involve any skilled knowledge base derived from heuristic knowledge of experienced control operators/engineers.

The main factors to consider when designing the FLSs are to determine fuzzy rule bases, MFs of input/output variables, and input/output scaling factors. Many research approaches have been investigated to optimization for a better design, many efforts has been directed to the partial optimization, not the entire system. For example, some works optimize the parameters of MFs, while others optimize the input/output scaling factors. Bio-inspired intelligent computing has been successfully applied to solve the complex problem in recent years such as genetic algorithms

(GAs), tabu search (TS), particle swarm optimization (PSO) and self-organizing feature map (SOFM) can be used to improve the behavior of this parameter optimization problem [9] -[17].

GAs was used by Arslan and Kaya [12] in the determination of MFs of input and output variables. They applied GAs to design a fuzzy logic control system having a single input and output. Bagis [7] proposed a method based on TS algorithm for the determination of MFs. Bai and Chen [10] proposed an automatic method for students' evaluation task. Its purpose was to automatically construct the grade MFs of lenient-type grades, strick-type grades, and normal-type grades of fuzzy rules, respectively. The system performs fuzzy reasoning to infer the scores of students based on the constructed grade-MFs. Meredith et al. [16] presented an approach based on GAs to the tuning of MFs in a FLS for a helicopter. Wong et al. [11] applies PSO to determine appropriate MFs of the fuzzy system (FS) automatically and presented a motion control structure with a distance fuzzy controller and an angle fuzzy controller for the two-wheeled mobile robot. Yang and Bose [13] presented a method for generating fuzzy MFs with unsupervised learning using SOFM. The SOFM approach is a two-step procedure: firstly, generate the proper clusters and secondly, generate the fuzzy MFs according to the clusters in the first step. They applied this method to pattern recognition problems. Ramazan Coban [17] presented a method for designing a closed loop fuzzy logic controller based on the particle swarm optimization algorithm. The designed controller is used for controlling the power level of nuclear research reactors. In his controller, the control rules constructed from numerical experiments made by means of a computer code for the core dynamics calculation and from human operator's experience and knowledge.

Moreover, many researchers have explored the use of optimization algorithms to tune FLC's key factors. The key factors are the membership function parameters, the scaling factors and the fuzzy control rule parameters for the design of FLCs. These methods differ mostly in the order or the selection of the different MF shapes, width and distribution on the performance of a FLC. Moreover, differences between the previous methods lie mainly in the type of coding and the way rule set and the shape and width of membership functions are optimized.

In this paper, we proposed a new adaptive method for nonlinear control strategy based on the combination of fuzzy logic and particle swarm optimization techniques. The purpose is to develop a nonlinear control strategy delivers better performance with respect to pendulums angle. The PSO algorithm is used as an adaptive method for tuning the MFs as well as fuzzy control rules with the aim of minimizing the output error measures or maximizing performance indexes. The results obtained from the simulations clearly show that the PSO-FLS control strategy gives a satisfactory performance with respect to the functional changes of the process. Furthermore, fuzzy based structure strategy gives more flexibility, higher robustness and precise behavior in control action in comparison to the well-known reinforcement learning method called SANE (Symbiotic, Adaptive Neuro-Evolution)[18].

The remaining of the paper is organized as follows: In section II, describe the literature relevant to the fuzzy logic systems. In Section III, the dynamic model of the cart pole type inverted pendulum is described. In Section IV, the new adaptive PSO-FLSs are demonstrated and learning algorithms for PSO-FLSs are discussed in this section as well. In Section V, Experiment results that demonstrate the efficiency of the proposed methodology are given. Finally, in Section VI, some concluding remarks are stated.

## **2. FUZZY LOGIC SYSTEMS (FLS)**

An FLS comprises four components, which are fuzzifier, knowledge base consisting of rule base (RB) and database (DB), fuzzy inference engine, and defuzzifier as depicted in Fig.1. The FLS works as follows, the crisp inputs are first fuzzified into, in general, input fuzzy sets (however, we

will consider only non-singleton fuzzification) which then activate the inference engine and the knowledge base to produce output fuzzy sets. The defuzzifier can then defuzzify fuzzy outputs from the inference engine in order to produce crisp outputs. A brief description of each of the FLSs components is given in the following subsections.

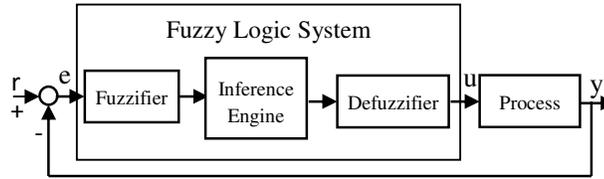


Figure 1. Fuzzy Logic Control

## 2.2. Fuzzifier

Since the input is in crisp normalized values, a fuzzification operator  $fuzz$  is used to fuzzify it in fuzzy form. The fuzzifier maps a crisp input vector with  $p$  inputs  $x = (x_1, \dots, x_p)^T \in X_1 \times X_2 \times \dots \times X_p \equiv X$  into input fuzzy set,  $A_x$ . In Non-singleton fuzzifier:  $\mu_A(x_i)$  realizes maximum value 1 at  $x = x_i$  and decrease from 1 to 0 while moving away from  $x = x_i$ . In this study we use singleton fuzzifier, the singleton fuzzifier of the input variable  $x$  is defined as:

$$A^*(x') = \begin{cases} 1 & x' = x \\ 0 & x' \neq x \end{cases}$$

## 2.3. Fuzzy Inference Engine

The fuzzy inference engine combines rules and gives a mapping from fuzzy sets in the input universe of discourse  $U \in R^n$  to fuzzy sets in the output universe of discourse  $V \in R$  based on the fuzzy logic principle. In the inference engine, multiple antecedents in the rules are connected using AND operation, and the degree of membership in the input sets are combined using those in the output sets using sub-star composition. Multiple rules are then combined by using a join operation.

## 2.4. Defuzzifier

The defuzzifier performs a mapping from the fuzzy sets in  $R$  to crisp points in  $R$ . It involves the selection, from the output fuzzy set, the numerical value  $v^*$  which can be considered as the best representative element of the fuzzy output set. Maximum defuzzifier produces a numerical value which exhibits the highest degree of membership. The most frequently used center of gravity (COG) defuzzifier takes output as the COG of the fuzzy output. The expression generally used for a discrete fuzzy set is as follows:

$$x_{n+1} = \frac{\int_{X \in U_{n+1}} X \mu_U(X) dX}{\int_{X \in U_{n+1}} \mu_U(X) dX} = \theta^{COG} = \frac{\sum_{j=1}^p \theta_j m_o(\theta_j)}{\sum_{j=1}^p m_o(\theta_j)}$$

where  $\theta_j$  denotes the center of gravity (COG) of the consequent label  $C_j$  of the rule  $R_j$ ,  $m_o(\theta_j)$  is the firing strength of this rule, and  $p$  is the number of rules.

## 2.5. Fuzzy Rule Base

The general form of a fuzzy rule used in most FLCs is as follows:

IF (a set of condition is satisfied)  
THEN (a conclusions can be inferred)

Each fuzzy IF-THEN rule has a premise (or IF) part containing several preconditions and the conclusion (THEN) part describing the output action. The premise and conclusion parts are characterized by some fuzzy sets (or MFs). Suppose that we need to design a multiple-input-multiple-output (MIMO) FLC having  $p$  inputs  $x_1 \in X_1, \dots, x_p \in X_p$  and  $c$  outputs  $y_1 \in Y_1, \dots, y_c \in Y_c$  with  $i$ -th fuzzy rule of the form:

$$R_{MIMO}^i : \text{IF } x_1 \text{ is } F_1^i \text{ .....and } x_p \text{ is } F_p^i, \text{ THEN} \\ y_1 \text{ is } G_1^i \text{ ..... } y_c \text{ is } G_c^i, \quad i = 1, \dots, M$$

Where  $F_1^i, \dots, F_p^i$  and  $G_1^i, \dots, G_c^i$  are the antecedent and consequent MFs associated with the linguistic  $p$  input variables and  $c$  output variables, respectively; and  $M$  is the number of rules in the rule base.

### 3. PROBLEM STATEMENT: INVERTED PENDULUM SYSTEMS

The inverted pendulum control problem is a standard test application for complex control approaches, with its complexity originating from the non-linear nature of the problem [19]. Being an under-actuated mechanical system and inherently open loop unstable with highly non-linear dynamics, the inverted pendulum system is a perfect test-bed for the design of a wide range of classical and contemporary control techniques. The cart pole balancing problem is a popular demonstration of using feedback control to stabilize an open-loop unstable system with fewer control inputs than the degrees of freedom. The cart-pole task involves a balancing pole hinged to a motion less cart that travels left or right along a straight bounded track as shown in Fig. 2. The pole is free to rotate only in the vertical plane of the cart and track. There are no sidelong resultant forces on the pole and it remains balanced as shown in Fig 2.

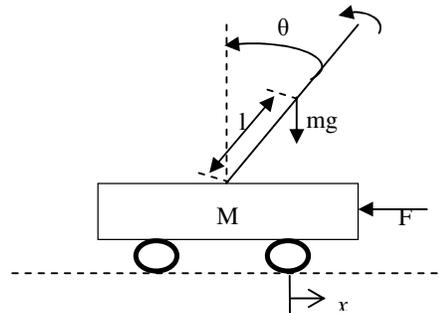


Figure 2. Inverted Pendulum Systems

The control objective is to apply a sequence of left or right forces of fixed magnitude to the wheeled cart so that it swings up the pendulum from its natural pendant position and stabilizes in the inverted position; once it reaches the upright equilibrium point (Fig.3). The cart must also be homed to a reference position on the rail. Here, the system state is specified by four real-valued variables:  $x$ -the horizontal position of the cart;  $\Delta x$  -the velocity of the cart (rate at which the error of position changes);  $\theta$ , the angle of the pole/shaft with respect to the vertical line;  $\Delta\theta$ , the angular velocity of the pole/shaft. The force  $F \in [-10, 10]$  newton's is applied to the cart and a zero magnitude force is not permitted. The dynamics of the cart-pole system are modeled by the following non-linear differential equations:

$$\begin{aligned} \dot{x} &= \Delta x \\ \Delta \dot{x} &= F + ml(\sin(\theta)\Delta\theta^2 - \Delta\theta \cos(\theta))/(M + m) \\ \dot{\theta} &= \Delta\theta \\ \ddot{\theta} &= \frac{g \sin \theta + \cos \theta \left( \frac{-F - ml\theta^2 \sin \theta}{M + m} \right)}{l \left( \frac{4}{3} - \frac{m \cos^2 \theta}{M + m} \right)} \end{aligned}$$

where  $g$  (acceleration due to gravity) is  $9.8 \text{ m/s}^2$ ,  $M$  (mass of cart) is  $1 \text{ kg}$ ,  $m$  (mass of pole) is  $0.5 \text{ kg}$ . if we linearized the above equation then we get the following equation:

$$\ddot{\theta} = \frac{g\theta - \left( \frac{F}{M + m} \right)}{l \left( \frac{4}{3} - \frac{m}{M + m} \right)}$$

From the above, we can find the state equation as follows:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 15.776 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1.463 \end{bmatrix} U$$

Where  $x_1$  is  $\theta$  and  $x_2$  is  $\dot{\theta} = \Delta\theta$  and  $U$  is the control force  $F$ . Our control objective here is to balance the pole without regard to the cart's position and velocity.

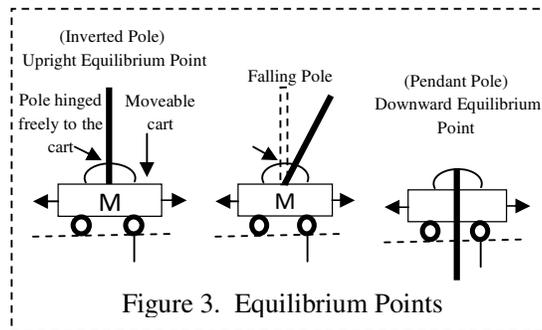


Figure 3. Equilibrium Points

## 4. INTEGRATED ARCHITECTURE: PSO AND FLSS

PSO is an optimization method that finds the optimal solution using a population of particles [20] developed by Dr. Kennedy and Dr. Eberhart in 1995, inspired by social behavior of bird flocking or fish schooling [15],[21].

### 4.1. The standard particle swarm optimization algorithm

The notation and operators used in the PSO algorithm is described as follows:

#### 4.1.1. Particle ( $x_i$ )

Each individual in the swarm is represented as particle, contains the parameters of velocity and position. All particles act under the same conditions and principles: finding the best personal and best overall location while constantly checking the value of its current location. The particle is

represented a candidate solution, m-dimensional real valued vector, where m is the number of optimized parameters. The particle  $x_i$  is:

$$x(t) = x_1(t), x_2(t), \dots, x_n(t)$$

#### 4.1.2. Particle velocity and Position

Each particle has a position vector ( $P_i$ ) and a velocity vector ( $V_i$ ) in the search space. At every learning cycle, each particle's position and velocity are updated by the two best positions. The new velocity and position of particle are calculating by the following two equations:

$$V_i(t+1) = wV_i(t) + c_1r_1(P_{pbest_i}(t) - P_i(t)) + c_2r_2(P_{gbest}(t) - P_i(t))$$

$$P_i(t+1) = P_i(t) + V_i(t+1)$$

Where ( $pbest_i$ ) is the  $i$ th particle's best solution that has achieved so far, ( $gbest$ ) is the overall best value from all particles in current generation,  $c_1 = c_2 = 2$  are acceleration constant,  $r_1$  and  $r_2$  are the two normally distributed random numbers between 0 or 1.

#### 4.1.3. Inertia Weight

The inertia weight is used as a control parameter in PSO to control the impact of the previous velocities on the current velocity. In this study we used the inertia weight that linearly decreasing from a relatively large value to a small value through the course of the PSO run to improve the performance of PSO. It can be expressed as:

$$w = w_{\max} - gen \frac{w_{\max} - w_{\min}}{\max gen}$$

Where  $w$  is the inertia weight,  $w_{\max} = 0.8$  and  $w_{\min} = 0.4$ ,  $gen$  is the evolutionary generation number. Initially  $w$  is set to 0.8 and  $w$  decreases while the evolutionary generation number increases. In the beginning, the particle swarm has the capability of global searching and at the end it has local searching capability. In PSO, there is no crossover and mutation operator; hence it is very easy to be realized.

## 4.2. Tuning of MFs distribution and Fuzzy control rules

The approach of using a PSO for control rules and their corresponding MFs tuning in FLSs is shown in Fig. 4. In order to achieve the desired level of robust performance for a controller, exact tuning of the membership functions is very important. Thus, to reduce the design effort and find a better fuzzy system control, membership functions are designed automatically by PSO algorithms. The control algorithms consist of the following basic steps:

Step-1: Firstly, select the general structure of the fuzzy logic controller according to the purpose of control and dynamics of the process.

Step-2: Define the number of fuzzy sets (Number of membership functions) for each fuzzy variable and set up initial fuzzy control rules.

Step-3: We set the initial individuals of PSO for the control parameters of fuzzy logic controller.

Step-4: Here, all control parameters of fuzzy logic controllers are tuned through PSO algorithms.

In the proposed PSO process, each particle is shaped to represent the fuzzy control rules and their corresponding MFs parameters of the FLC's inputs and outputs. Each particle represents a potential solution. The parameters are the center (a) value and width (b) of each fuzzy MFs as shown in Fig. 5. These parameters are used to define the particles of PSO algorithm and looking for the global best fitness.

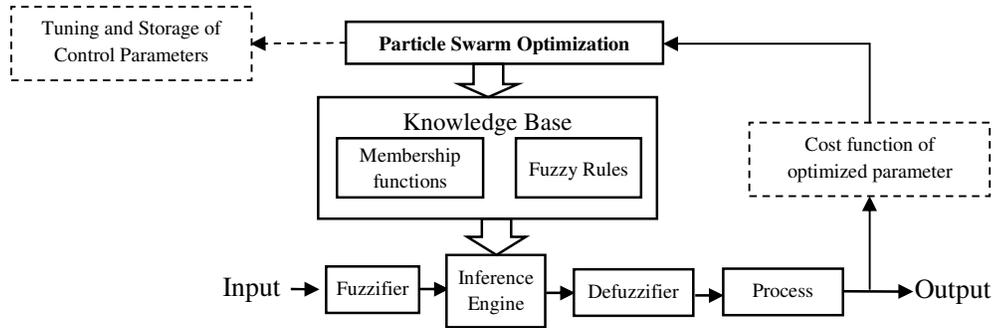


Figure 4. The scheme of auto-tuning of the fuzzy controller involving estimation of the Control parameters

### 4.3. Encoding Schema

One of the most attractive problems in PSO is coding the solution space. There are two parameters to define the membership functions ( $a$  and  $b$  parameters) according to Fig. 5. A chromosome represents a candidate solution of the control problem. In this method, a solution candidate is expressed by real coding. Consequently, the  $a$  and  $b$  parameters for the input and output membership are expressed in terms of strings consisting real numbers as shown in Fig. 6. In this study represents the fuzzy rule base and their corresponding MFs with appropriate chromosome. For PSO, each particle represents a fuzzy rule base and their corresponding MFs. The number of active rules in a rule base and the parameters of rule base should be contained in the fuzzy rule base.

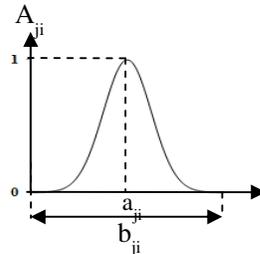


Figure 5. Fuzzy Sets defining parameter

Assume that the chromosome contains  $n$  MFs then a potential solution to it is encoded as a  $2n$ -dimensional real-valued vector, as demonstrated in Fig. 6.

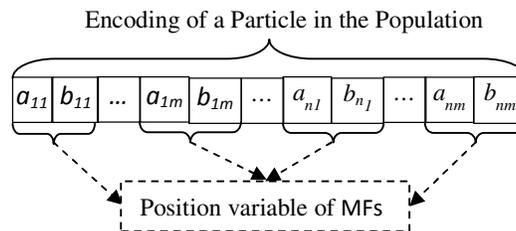


Figure 6. Illustration of the encoding scheme of PSO algorithms

#### 4.4. Evolutionary Strategy

As a PSO is computationally expensive, this optimization is carried out off-line, for a 20 input scenarios selected at random. Once the optimized FLS is obtained, it can be used for making on-line prediction of the outputs. The proposed algorithm, evolutionary strategy works as follows (Fig. 7):

- (i) The algorithm starts by setting the initial set of parameters. In this case, generate randomly  $m$  particles. It becomes an initial swarm. At the same time, initial positions and velocities of all particles are generated randomly within constrains in  $[-v^{\max}, v^{\max}]$ .
- (ii) Each particle represented potential FLCs. The initial set of parameter is used to build the potential FLSs. Then the potential FLCs are evaluated through the objective function. Search for the best value of the objective function  $pbest$ . In initial  $pbest$ ,  $gbest$  is set as the best value of the objective function. If the control system performance satisfied the termination criteria then stop and get the optimal fuzzy set values. Otherwise go to step (iii).
- (iii) Update the fuzzy set parameter value through the optimization algorithm. In this way get the new parameters of control rules and their corresponding (MFs). PSO doing this update by calculating the inertia weight and velocity of the particle. At the same time check the maximum velocity of particle.
- (iv) Then the updated parameter is used to evaluate the performance of fuzzy control system. If the control systems performance satisfied the termination criteria then stop and get the optimal fuzzy set values. Otherwise go to step (iii).
- (v) Optimal parameter for building the optimal FLCs : The particle that generates the latest  $gbest$  is an optimal parameter and represented the optimal parameter of FLCs.

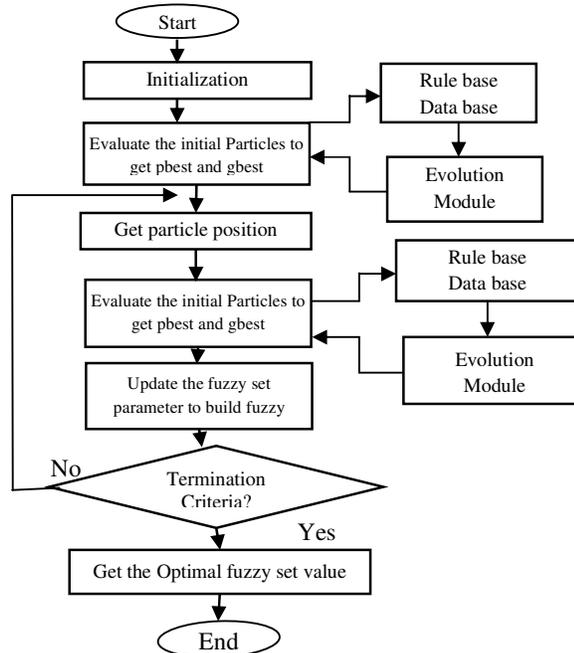


Figure 7. Flow chart of the proposed algorithm for designing the FLSs

In the PSO-FLS process each particle is formed according to the MFs parameters of the FLC's inputs and outputs. The parameters that define the MFs are the mean value  $m$  and the standard deviation of each MF. The MFs is defined as:

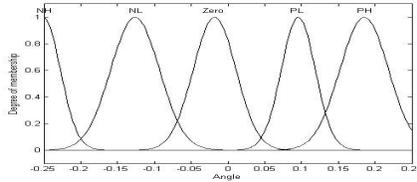
$$f_{mf}(x) = e^{-\frac{(x-m)^2}{2\sigma^2}}$$

Compared to other optimization algorithms [19-21], the PSO algorithm is easy to implement, there are few parameters to adjust and exhibits stable convergence. Compared to other traditional optimization techniques the advantages of PSO can be summarized as follows:

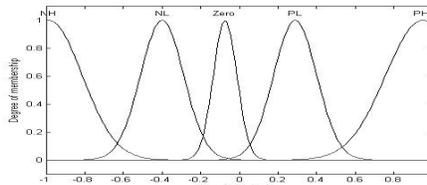
- (a) PSO to be less susceptible to being trapped in local minima due to the implicit parallelism.
- (b) To guide the search in the search space, the PSO algorithm uses performance index or objective function. For this reason PSO algorithm can easily deal with nonlinear and non-differentiable objective functions.
- (c) PSO algorithms use probabilistic rather than deterministic transition rules. Hence PSO algorithms are a kind of stochastic optimization. This makes PSO algorithms more flexible and robust than traditional optimization techniques.

Table 1: Optimized Fuzzy Control rules

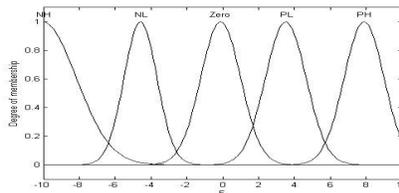
Control Action		Angle				
		NH	NL	Zero	PL	PH
Angular velocity	NH	NL	NL	NH	NH	Zero
	NL	NH	PL	NL	Zero	PL
	Zero	NH	PL	Zero	PL	PH
	PL	NL	Zero	PL	NH	PH
	PH	Zero	PL	PH	PH	NL



(a)



(b)



(c)

Figure 8. Gaussian Shape MFs (a) Angle ( $\Theta$ ), (b) Angular Velocity ( $\Delta\Theta$ ) and (c) Control Action Force (F)

#### 4.5. Objective function

Selecting the objective function is the one of the main factor of the PSO algorithm. The purpose of the algorithm is to improve the control system performance. Any function that can achieve this goal may be selected as the objective function. Different objective functions promote different PSO behaviors, which generate fitness values providing a performance measure of the problem considered. The PSO algorithm is to minimize the control error of the FLSs; the objective function of PSO is defined as:

$$f(x(k)) = \sum_{i=0}^{t_f} \varepsilon^2$$

where  $t_f$  is the total running time of the FLC,  $\varepsilon$  is the control error.

### 5. SIMULATION RESULTS AND EVALUATION

To demonstrate the efficiency of the proposed adaptive algorithm we carried out a series of simulations and simulation results are presented for the cart pole type inverted pendulum system described earlier in this section. In order to get an accurate result as well as a fast convergence of the algorithm, the values of the aforementioned parameters are considered as follows: the number of particles 50, searching iterations 100, inertia weight 0.75 and velocity 1. We prove the stability of the fuzzy logic controller in time occurs with the tuning fuzzy control rule sets obtained by the proposed approach.

#### 5.1. Fuzzy control rules

PSO are powerful nonlinear optimization algorithms. However, sometimes the powerful performance is obtained at the cost of expensive computational complexity and much time. To overcome this type of problem, first, we have selected the several initial angular positions over the working space and obtained the auto tuned control parameters by means of PSO algorithms according to the change of each selected initial angular positions, then we obtained the desirable fuzzy control rules as shown in table-1, this optimized fuzzy rule is very important in the development of fuzzy control systems, which are usually decided upon subjectively. These control rules makes an interface between the inputs and outputs and is used to control the inverted pendulum angle.

#### 5.2. Membership Functions

In our simulations of this paper, we used five antecedent and consequent fuzzysets “Negative High”, “Negative Low”, “Zero”, “Positive Low” and “Positive High” in. Fig 8. In this study, the fine-learning process is introduced to optimize parameters in the membership functions since those parameters are only decided by experience control engineers/operators previously, and different sets of membership functions parameters would have a tremendous different impact on the final control effect. The designed fuzzy logic controller coupled the point-valued MAX–MIN fuzzy inference engine product rule to combine the membership values for each rule (Mamdani type), and the centre of area (COA) defuzzifier scheme to obtain the crisp value.

#### 5.3. Controller Response

Based on the obtained controller, the tracking control of the inverted pendulum is demonstrated in Fig. 9, and an excellent tracking performance is also achieved. The control algorithm has been tested for this nonlinear control problem on the following initial angles:  $+20^{\circ}$ ,  $-20^{\circ}$ ,  $15^{\circ}$ ,  $-15^{\circ}$ ,  $5^{\circ}$  and  $-5^{\circ}$  degrees. Figure 9 shows the angular displacement responses of the proposed controller. From this Fig. 9, we have shown that from any starting pendulum angle the inverted pendulum goes to the desired angle  $0^{\circ}$ . These results further reflect that the generated adaptive controller through PSO algorithms could drive the system toward an implicit equilibrium position and

velocity. In the proposed controller, the nonlinearity of the controlled object can be compensated by the PSO algorithms.

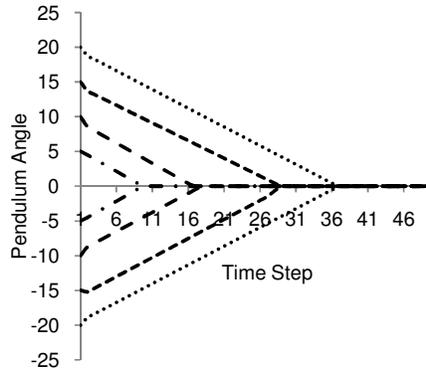


Figure 9. The dynamics of output of the system controlled by optimized controller during switching from swing-up to stabilization

#### 5.4. Control Surface

Control signals have been generated by the adaptive fuzzy controller based on the rules base and membership functions presented above. Fuzzy logic control toolbox generated smooth nonlinear control surface, which is presented in Fig. 10. From this control surface, we conclude that the output of the optimized fuzzy logic controller is the smooth function of its input variables as long as the neighboring MFs in the antecedent have enough overlap. This smooth nonlinear control surface controls the reactor with satisfying results. The nonlinear property of the control surface is essential to achieve the good performance. Note that the Fuzzy controller comes with a significant nonlinear mapping between the inputs and output.

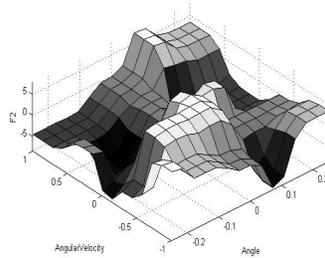


Figure 10. Control Surface (input-output relationship) generated by proposed controller

For this PSO based optimization technique, Fig. 11 shows the average and best fitness corresponding to the number of iterations (or generations) that has passed through for the PSO approaches. This clearly highlights very good convergence performance of PSO for this inverted pendulum problem.

#### 5.5. Evaluating our work

In order to evaluate our new PSO-FLS approach, we compare it against well-known reinforcement learning method called SANE (Symbiotic, Adaptive Neuro-Evolution)[15]. SANE is evolutionary neural networks based cooperative model that has been applied to cart pole type inverted pendulum system with very promising results [15]. The closer inspection of the study in [15] reveals that the system is stabilized within 46 time steps for the best case and 4461 for the worst cases from initial pole angles  $\pm 15^0$  degrees and random cart positions  $\pm 2.4$  meters. For the

same initial condition, the number of time steps required to stabilize the cart-pole system is 29 for best case and 2964 is for the worst case while using our proposed approach. The time steps required for cart and pole stabilization were computed over 20 simulations for each method. The advantage of using this PSO-FLSs structure is that the number of rules used in the fuzzy knowledge base has been reduced substantially also.

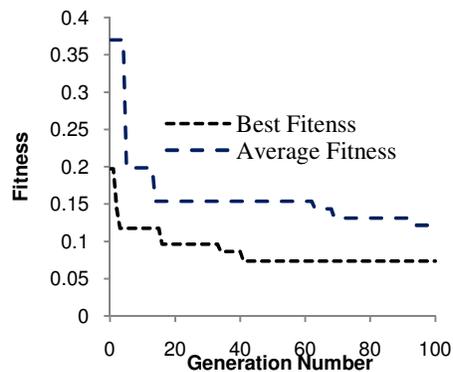


Figure 11. Convergence characteristics of PSO on the average and best Fitness in finding the solution

## 6. CONCLUSION

An adaptive evolutionary particle swarm optimization is proposed to design fuzzy rule base with their corresponding MFs automatically. To reduce fuzzy system effort and increase cost saving, a PSO has been used to tune the fuzzy control rules and their corresponding membership functions. PSO is able to generate an optimal set of parameter concerning the fuzzy controller including the fuzzy membership functions automatic adjustment and then the adjusted fuzzy system achieves better performance for inverted pendulum problem. It is verified that the Fuzzy-PSO controller has better control performance inverted pendulum system. Furthermore, implementation of the FLC tuning with PSO is much easier than the conventional methods because there is need neither derivative knowledge nor complex mathematical equations. Finally, the simulation results reveal that the PSO algorithm has stable convergence with good computational efficiency while comparing the SANE [15].

## 7. REFERENCES

- [1] Pierre Guillemin (1996) "Fuzzy Logic Applied to Motor Control", *IEEE Trans. on Industry application*, vol. 31, no.1, pp. 51-56.
- [2] Hu Songlin, Zhang Yunning, Yin Xiuxia and Du Zhaoping (2013) "T-S fuzzy-model-based robust stabilization for a class of nonlinear discrete-time networked control systems", *Nonlinear Analysis: Hybrid Systems*, vol. 8, p.69-82, 2013, doi:10.1016/j.nahs.2012.11.001.
- [3] Rahul L. Navale, and Ron M. Nelson (2010) "Use of genetic algorithms to develop an adaptive fuzzy logic controller for a cooling coil", *Energy and Buildings*, Vol. 42, Issue 5, pp. 708-716.
- [4] Shang-Ming Zhou and John Q. Gan (2007) "Constructing L2-SVM-Based Fuzzy Classifiers in High Dimensional Space with Automatic Model Selection and Fuzzy Rule Ranking", *IEEE Trans. on Fuzzy Syst.*, vol. 15, no. 3, pp. 398-409.
- [5] Z. Liu and H.X. Li (2005) "A probabilistic fuzzy logic system for modeling and control", *IEEE Trans. Fuzzy Systems*, vol. 13, pp. 848-859.
- [6] P. C. Shill, M.A.H. Akhand, M. F. Amin, K. Murase (2011) "Optimization of Fuzzy Logic Controller for Trajectory Tracking using Genetic Algorithm", *J. Advanced Computational Intelligence and Intelligent Informatics (JACIII)*, Vol. 15, No. 6, pp.639-651.

- [7] I-Hsum Li, Lian-Wang Lee (2011) “A hierarchical structure of observer-based adaptive fuzzy-neural controller for MIMO systems”, *Fuzzy Sets and Systems*, vol.185, issue 1, pp. 52–82.
- [8] A.Bagis (2003) “Fuzzy and PD controller based intelligent control of spillway gates of dams”, *Journal of Intelligent Fuzzy Systems*, vol.14, no.1, pp.25–36.
- [9] A.Bagis (2003) “Determining fuzzy membership functions with tabu search: An application to control”, *Fuzzy Sets and Systems*, vol. 139, pp.209–25.
- [10] S. Bai.and S Chen (2008) “Automatically constructing grade membership functions of fuzzy rules for students evaluation”, *Expert Systems with Applications*, vol. 35, pp.1408–1414.
- [11] C.C. Wong, H.Y. Wang and S.A. Li (2008) “PSO-based motion fuzzy controller design for mobile robots”, *International Journal of Fuzzy Systems*, vol. 10, no.1, pp. 24–32.
- [12] A. Arslan, and M. Kaya (2001) “Determination of fuzzy logic membership functions using genetic algorithms”, *Fuzzy Sets and Systems*, vol. 118, pp.297–306.
- [13] Yang and Bose, N. K. (2006) “Generating fuzzy membership function with self-organizing feature map”, *Pattern Recognition Letters*, vol. 27, pp.356–365.
- [14] G. Coath and S. Halgamuge (2003) “A Comparison of Constraint-handling Methods for the Application of Particle Swarm Optimization to Constrained Nonlinear Optimization Problems”, *IEEE Congress on Evolutionary Computation*, Canberra, Australia. pp. 2419-2425.
- [15] Randy L. Haupt, Sue Ellen Haupt (2004) *Practical Genetic Algorithms*, 2<sup>nd</sup> Edition, John Willy & Sons Inc.
- [16] D.L. Meredith, C.L. karr, and K. Krishna Kumar (1992) “ The Use of Genetic Algorithms in the design of Fuzzy logic Controllers”, *In proc. 3<sup>rd</sup> Workshop on Neural Networks (WNN’92) Academic/Industrial Defence*, pp. 1105-1109.
- [17] Ramazan Coban (2011) “A fuzzy controller design for nuclear research reactors using the particle swarm optimization algorithm”, *Nuclear Engineering and Design*, vol. 241, Issue 5, pp. 1899–1908.
- [18] D. E. Moriarty and R. Miikkulainen (1996) “Efficient Reinforcement Learning Through Symbiotic Evolution”, *Machine Learning*, vol. 22, pp. 11–32
- [19] S. Brock, “Practical Approach to Fuzzy Control of Inverter Pendulum”, *ICIT Slovenia*, 2003.
- [20] J. Kennedy and R.C. Eberhart (1995) “Particle swarm optimization”, *IEEE International Conference on Neural Networks* , Perth, Australia , IEEE Service Centre.
- [21] X. F. Xie, W. J. Zhang, and Z. L. Yang (2002) “A Dissipative Particle Swarm Optimization”, *Congress on Evolutionary Computation*, Hawaii, USA, pp. 1456-1461.
- [22] Chang, W.-D. (2009). “PID control for chaotic synchronization using particle swarm optimization”, *Chaos, Solitons & Fractals*, vol. 39, no. 2, pp. 910–917.
- [23] Chiou, J.-S., and Liu, M.-T.(2009) “Numerical simulation for Fuzzy-PID controllers and helping EP reproduction with PSO hybrid algorithm”, *Simulation Modelling Practice and Theory*, vol. 17, no. 10, pp.1555–1565.
- [24] Toscano, R., and Lyonnet, P. (2009) “Robust PID controller tuning based on the heuristic Kalman algorithm”, *Automatica*, vol. 45, no. 9, pp. 2099–2106.
- [25] S. Yuhui, and R. Eberhart, (1998) “A modified swarm optimizer”, *in: Proceedings of IEEE International Conference on Evolutionary Computation*, IEEE Press, Piscataway, NJ, pp. 69–73.
- [26] M.Clerc, (2006), *Particle Swarm Optimization*, ISTE Ltd., New Port Beach, CA.
- [27] A. M. El-Zonkoly , A. A. Khalil and N.M. Ahmied (2009) “Optimal tuning of lead-lag and fuzzy logic power system stabilizers using particle swarm optimization”, *Expert Systems with Applications*, Vol. 36, Issue 2, Part 1, pp. 2097–2106.
- [28] Chia-Feng Juang, Yu-Cheng Chang , Chia-Hung Hsu and I-Fang Chung (2011) “Fuzzy controller design using group-crossover particle swarm optimization for truck reversing control ”, *IEEE International Conference on Fuzzy Systems*, Taiwan, pp. 2664 – 2667.

[29] Juing-Shian Chiou, Shun-Hung Tsai and Ming-Tang Liu (2012) “A PSO-based adaptive fuzzy PID-controllers”, *Simulation Modelling Practice and Theory*, Vol. 26 , pp. 49–59.

## Authors

**Sree Bash Chandra Debnath** received the B.Sc. degree in Electrical & Electronic Engineering (EEE) from Chittagong University of Engineering and Technology (CUET), Bangladesh in 2011. He joined as a lecturer at the Department of EEE, Premier University, Chittagong, in February, 2012 and currently he is serving as an Lecturer. His research interest includes evolutionary computation, fuzzy logic and artificial neural networks.



**Pintu Chandra Shill** received the B.Sc. degree in Computer Science Engineering (CSE) from Khulna University of Engineering and Technology (KUET), Bangladesh in 2003 and M.Sc. degree in Computer Engineering from Politecnico di Milano, Italy in 2008. He joined as a lecturer at the Department of CSE, KUET in 2004 and currently he is serving as an Assistant Professor. He has published several research papers in some reputed Journal and Conference. His research interest includes evolutionary computation, fuzzy logic and artificial neural networks.



**Kazuyuki Murase** is a Professor at the Department of Human and Artificial Intelligence Systems, Graduate School of Engineering, University of Fukui, Fukui, Japan, since 1999. He received ME in Electrical Engineering from Nagoya University in 1978, PhD in Biomedical Engineering from Iowa State University in 1983. He Joined as a Research Associate at Department of Information Science of Toyohashi University of Technology in 1984, as an Associate Professor at the Department of Information Science of Fukui University in 1988, and became the professor in 1992. He is a member of The Institute of Electronics, Information and Communication Engineers (IEICE), The Japanese Society for Medical and Biological Engineering (JSMBE), The Japan Neuroscience Society (JSN), The International Neural Network Society (INNS), and The Society for Neuroscience (SFN). He serves as a Board of Directors in Japan Neural Network Society (JNNS), a Councilor of Physiological Society of Japan (PSJ) and a Councilor of Japanese Association for the Study of Pain (JASP).

