*Journal of Engineering Sciences, Assiut University, Vol. 40, No. 3, pp. 899-911, May 2012*

# FUZZY CONTROLLER DESIGN FOR DC MOTOR USING ANT COLONY OPTIMIZATION ALGORITHM

***Mohammad El-Bardini; Mohamed Fkirin and Sameh Abd-Elhaleem****

*Faculty of Electronic Engineering, Menouf, 32852, Egypt*
** eng_sah55@yahoo.com*

*(Received February 22, 2012 Accepted March 20, 2012)*

*This paper presents fuzzy controller design using ant colony optimization algorithm (ACO-FC). The objective of ACO-FC is to improve the control performance and design ease of fuzzy controller. In ACO-FC, the antecedent part (IF part) of a fuzzy system is flexibly partitioned in grid type, and the consequent part (THEN part) of each rule is selected by the ants where the route of an ant is regarded as combination of consequent actions selected from every rule. Searching for the best one among all consequence combinations is based mainly on the pheromone matrix among all candidate actions. To verify the control performance of ACO-FC, simulations on position control of a DC motor are performed. Comparison with PID like fuzzy controller demonstrates the advantages of ACO-FC.*

**KEYWORDS:** *Ant colony optimization (ACO), fuzzy controller design, position control.*

## 1. INTRODUCTION

It is well known that fuzzy systems have been widely applied in many applications such as pattern recognition, image processing, cluster analysis, decision analysis and automatic control. However the operation of fuzzy rule derivation is often difficult, consuming time and requires expert knowledge although the human experts find it difficult to examine all the input - output data from a complex system to obtain a set of suitable rules for the fuzzy system hence the performance of the fuzzy controller will need improvement.

In order to solve this problem, several neural fuzzy systems have been proposed to automate the design of fuzzy systems [1] [2], many researchers have proposed optimization methods using meta-heuristic algorithms such as genetic algorithms (GA) [3]-[7], particle swarm optimization(PSO) [7] [8], and ant colony optimization (ACO) [9 ]-[20] . The ACO algorithm is a novel heuristic algorithm, may be considered to be part of swarm intelligence that is inspired by the observation of the behavior of real ants in nature searching for food. The ACO is a multiagent approach where the artificial ant colonies cooperate to find optimum solutions for difficult discrete optimization problems [9]-[11] and continuous optimization problems [12]. ACO algorithms have been successfully applied to a number of different combinatorial optimization problems, such as data mining and network routing [14] [15]. The world of meta-heuristic is wide and developed .several characteristics make ACO a unique approach such as a constructive, population based meta-heuristic which exploits an indirect form of memory of previous performance to avoid the errors in the following stages from the problem.

A fuzzy system consists of a set of fuzzy IF-THEN rules that describe the input-output mapping relationship of the system so the fuzzy controller (FC) design includes designs of antecedent part (IF part) and consequent part (THEN part) as the antecedent part can be partitioned in advance so the bottleneck in the fuzzy system design is the consequent part hence in this paper the ACO will be applied to design the consequent part.

Applying the ACO algorithm to an FC design is called ACO-FC which optimizes the consequent parts. The ACO-FC has a comparable performance to other meta-heuristic algorithms [13] so this paper will focus on using ACO-FC.

Position control of the DC motor has attracted considerable research and several methods have proposed. This paper will apply position control for a DC motor using ACO-FC based PID controller.

This paper is organized as follows. Section 2 describes the fuzzy logic controller to be designed using ACO. The basic concepts of ACO are introduced in Section 3. Section 4 presents the fuzzy controller design by ACO. Section 5 describes the system model. Simulation results are addressed in section 6. Finally, conclusions are presented in section 7.

## 2. FUZZY LOGIC CONTROLLER

Fuzzy control techniques have attracted significant interest and have become an important part of modern control engineering. Fuzzy control is a method of rule-based decision making used for expert systems and process control that emulates the rule-of-thumb thought process used by human beings.

A fuzzy system is characterized by a set of linguistic statements according to expert knowledge that is usually represented in the form of “IF–THEN” rules expressed as:

**IF** (a set of conditions are satisfied).

**THEN** (a set of consequences can be inferred).

The antecedent and the consequence of these IF–THEN rules are associated with fuzzy concepts, so they are often called fuzzy conditional statements. In fact, the antecedent is a condition in its application domain and the consequence is a control action for the system under control. The conditions and actions are linguistic terms represent the values of input and output variables. The Basic configuration of a fuzzy logic controller is shown in Fig.1.

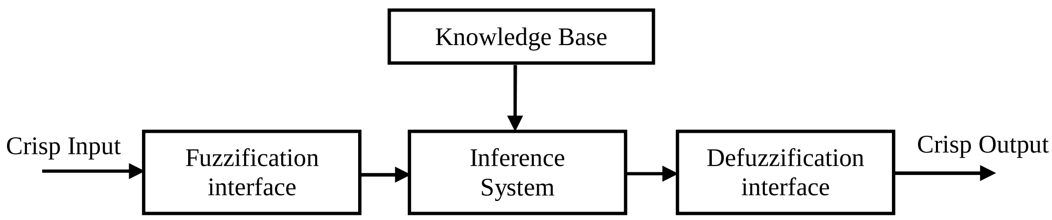



Fig.1: Basic configuration of a fuzzy logic controller

Fuzzy logic controllers are rule based systems consist of two main components:

- The Knowledge Base (KB) that represents the knowledge about the system in the form of fuzzy linguistic IF-THEN rules. The KB consists of the Rule Base (RB) where the RB is a collection of linguistic rules also, and of the Data Base (DB) which contain the sets and the membership functions .the fuzzy linguistic rule structure considered in linguistic fuzzy rule base systems is the following:

$$R_i \text{ : If } X_1 \text{ is } A_{i1} \text{ and } \ldots\ldots\ldots\ldots\text{and } X_n \text{ is } A_{in} \text{ THEN Y is } B_j \qquad (1)$$

Let $i$ be an index for a fuzzy rule number, $X_1$ ,……, $X_n$ and Y are the input and output variables respectively ,and $A_{i1}$,…….., $A_{in}$ and $B_j$ being linguistic labels , each one of them having associated a fuzzy set defines its meaning [16] .

- The inference engine includes three components:
  1. A fuzzification interface: Converting crisp input data into fuzzy sets described by linguistic expressions.
  2. An inference system: uses the output of the fuzzification interface stage together with the KB to calculate fuzzy output i.e. evaluate activation strength of every rule and combine their action sides.
  3. A defuzzification interface: calculates actual output i.e. converts fuzzy output into a precise numerical value to be applied to the process.

In the fuzzy controller of mamdani type which will be used with ACO in this paper the $i$ th rule denoted as $R_i$ is represented in the following form:

$$R_i \text{ : If } X_1(\text{k}) \text{ is } A_{i1} \text{ and } \ldots\text{and } X_n(\text{k}) \text{ is } A_{in} \text{ THEN } u(\text{k}) \text{ is } a_i \qquad (2)$$

Where k is time step, $X_1$(k) ,……, $X_n$(k) are the input variables, $u$ (k) is the output action variable , $A_{in}$ is a fuzzy set and $a_i$ is a crisp value .fuzzy sets $A_{ij}$ uses triangle membership functions .In the inference engine the fuzzy AND operation is implemented by algebraic minimum in fuzzy theory . So given an input data set $\vec{X}$ = ( $X_1 \ldots X_n$) the firing strength $\phi_i(\vec{X})$ of rule $i$ is calculated by

$$\phi_i(\vec{X}) = \min_{j=1,\ldots,n} \mu_{A_{ij}}(x_j) \qquad (3)$$

Where $\mu_A$ is the membership degree for the fuzzy set $A$. If there are $r$ rules in a fuzzy system, the output of the system is calculated by the weighted-average-defuzzification method as

$$u = \frac{\sum_{i=1}^{r} \phi_i(\vec{X}) a_i}{\sum_{i=1}^{r} \phi_i(\vec{X})} \qquad (4)$$

Where $a_i$ is the rule consequent value in equation (2).

## 3. ANT COLONY OPTIMIZATION (ACO)

The ant colony optimization was developed in early 1990s by Dorigo et al [9]. The ACO technique is one of the meta-heuristic optimization methods and is inspired by observation of real ant colonies. Real ants are capable of finding the shortest route from a food source to their nest by exploiting a chemical substance called pheromone which records the information on distance. In real world ants move randomly without using other information initially and lay some pheromone on the paths. After that each individual ant makes a decision of the moving direction based on the strength of the pheromone trails on the ground. The path which have higher amount of the pheromone trails on the ground is the better. While more and more ants walking to the source food, the shortest path accumulates the more pheromone this in turn increases the number of ants choosing the shorter path. Finally the ants will find the shortest path. This phenomenon has inspired the artificial ant colony algorithm in which a colony of ants cooperate to the solution of a problem by exchanging information via pheromone deposited on the paths where the pheromone value represents the memory of the artificial ants .ACO algorithms can be applied to problems that can be described by a graph consisting of nodes and edges connecting the nodes. The edges between the nodes on the graph carry two sources of information which service the ants to choose the correct path. The two sources are the pheromone value $\tau$ and heuristic value $\eta$ where heuristic value represents a priori information about the problem instance definition. The solutions to the optimization problem can be expressed in feasible paths on the graph. The ACO can be used to choose a path among the feasible paths which has minimum cost. The performance measure is based on a quality function $f(\cdot)$. The first member of the ACO algorithm called ant system AS [9]. The AS was successfully applied to the traveling salesman problem (TSP). The TSP plays a central role in ant colony optimization because it was the first problem to be attacked by ACO algorithm [18]. The TSP objective is to find a minimal route length for a salesman to take in visiting $N$ cities with each city being visited once. This problem can be represented by a graph with $N$ nodes represents cities and $E$ being a set of edges fully connecting the nodes. Let $d_{ij}$ be the length of the edge $(i, j) \in E$ that is the distance between cities $i$ and $j$ with $i, j \in N$. The probability of choosing a certain city $j$ is calculated using the mount of pheromone on edge between cities $i$ and $j$, and the distance between these two cities. As in [13] [20] the probability with which an ant $k$ chooses to go from city $i$ to city $j$ is

$$P_{ij}^{k}(t) = \begin{cases} \dfrac{[\tau_{ij}(t)][\eta_{ij}]^{\beta}}{\sum_{l \in N_i^k} [\tau_{il}(t)][\eta_{il}]^{\beta}} & \text{if } \; j \in N_i^k \\ 0 & \text{Otherwise} \end{cases} \tag{5}$$

where $\tau_{ij}(t)$ is the amount of pheromone trails on edge ($i$, $j$) at iteration $t$ ,$\eta_{ij} = 1/d_{ij}$ is the heuristic value of moving from city $i$ to city $j$, $N_i^k$ is the set of neighbors of city *i* for the $k^{th}$ ant, and parameter $\beta$ controls the relative weight of pheromone trial and heuristic value. As in [20] after all ants have completed their tours, the pheromone level is updated by

$$\tau_{ij}(t+1) = (1-\rho).\tau_{ij}(t) + \Delta\tau_{ij}(t) \tag{6}$$

where $0 \le \rho < 1$ is the pheromone trail evaporation rate, $\rho$ must be set to a value < 1 to avoid unlimited accumulation of pheromone [9]. The update value $\Delta\tau_{ij}$ is related to a quality value *F* which is used to measure the performance of each ant path. There are several updating rules for $\Delta\tau_{ij}$ have been studied in [17]. The quality value is calculated based on the application for example the quality function for a control system represents the inverse of the sum of the absolute error.

$$F = \frac{1}{\sum_{k=1}^{n} |e(k)|} \tag{7}$$

where *e(k)* is the error signal and *n* is the number of iteration.

## 4. FUZZY CONTROLLER (FC) DESIGN BY ACO (ACO-FC)

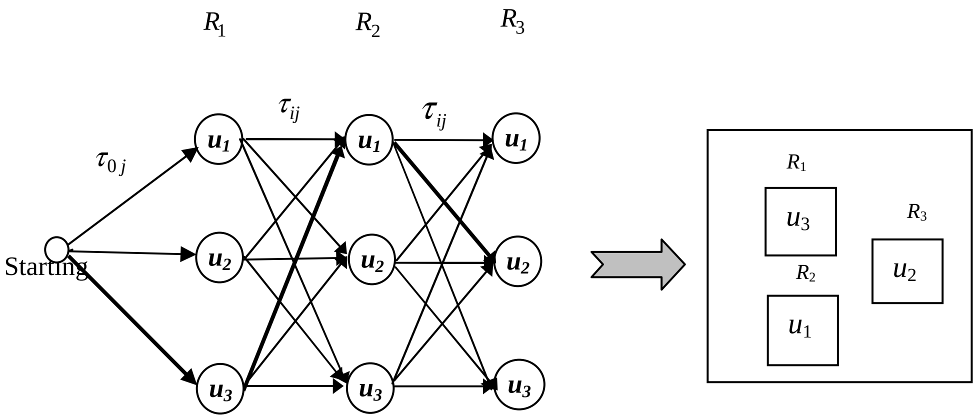
The FC antecedent part can be partitioned in advance without much difficulty. The problem of the design of consequent part can be represented by a graph with *N* nodes which represent the candidate control actions and the edges connecting the nodes. Let the candidate control actions be the set $U= \{u_1, u_2, \ldots, u_N\}$ where the number of candidate consequent values of $a_i$ in equation (2) is *N* for each rule. For each rule, one of the *N* candidate actions is chosen. If there are *r* rules then the complexity of finding the best consequent combination is $N^r$. The ACO algorithm is used to find the solution. The tour of each an ant is regarded as one combination of consequent values selected from every rule.

- **The Algorithm of ACO-FC**

To see the tours of ants through rules as shown in Fig.2, for example there are three rules denoted by $R_1$, $R_2$ and $R_3$ in a fuzzy system and three candidate consequent values $u_1$, $u_2$ and $u_3$ for each rule. The starting from the nest, the ant moves through $R_1$ and $R_2$ and stop at $R_3$. The bold line in Fig.2 indicates the tour of this ant. For each rule the node visited by the ant is selected as the consequent value of the rule. Each ant in the colony will have fuzzy system. For the whole fuzzy system constructed by the ant, the consequent values in $R_1$, $R_2$ and $R_3$ are $u_3$, $u_1$ and $u_2$ respectively. Selection of the consequent values from rule to other is based on the pheromone trails between the rules. The pheromone matrix in Fig.2 includes the first row which represents the initial intensity of pheromone trails. It is set to a small positive constant [9], [13] to select the first fuzzy rule consequent action. The transition probability for selecting consequent value $u_j$ in fuzzy rule *i* in ACO is calculated by:

$$P_{ij}(t) = \frac{\tau_{ij}(t)}{\sum_{m=1}^{N} \tau_{im}(t)} \tag{8}$$

Where $i = 1,...,r$ and $j = 1,...,N$ , in this study it is assumed that no a priori information about the problem is a known so the ACO works without heuristic information for ease implementation equation (8). The pheromone trails $\tau_{ij}$ on the ant path are updated by a quality function *F* in equation (7). The higher *F* value indicates better performance for the controller.




| | | |
|---|---|---|
| $\tau_{01}$ | $\tau_{02}$ | $\tau_{03}$ |
| $\tau_{11}$ | $\tau_{12}$ | $\tau_{13}$ |
| $\tau_{21}$ | $\tau_{22}$ | $\tau_{23}$ |
| $\tau_{31}$ | $\tau_{32}$ | $\tau_{33}$ |

Pheromone matrix

Fig.2: FC constructed by ant tour and the corresponding pheromone matrix.

After the construction of a fuzzy controllers with number of ants $N_{ants}$, select the one with the highest *F* from the initial until now as in the ACS [11], if a new global best ant is found in this iteration then the pheromone trails on the tour traveled by the global best ant are updated otherwise no pheromone update is performed in this iteration. The new pheromone trail $\tau_{ij}(t)$ is updated by:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t)\, if\ (i, j) \in\ global\ \ best\ \ tour \tag{9}$$

$$\Delta \tau_{ij}(t) = c_2 \cdot F_{K^*} \tag{10}$$

where the global best ant is denoted as $K^*$ with quality value $F_{K^*}$, and $C_2$ is the parameter for controlling the amount of update. The flow chart of ACO-FC algorithm is shown in Fig.3.

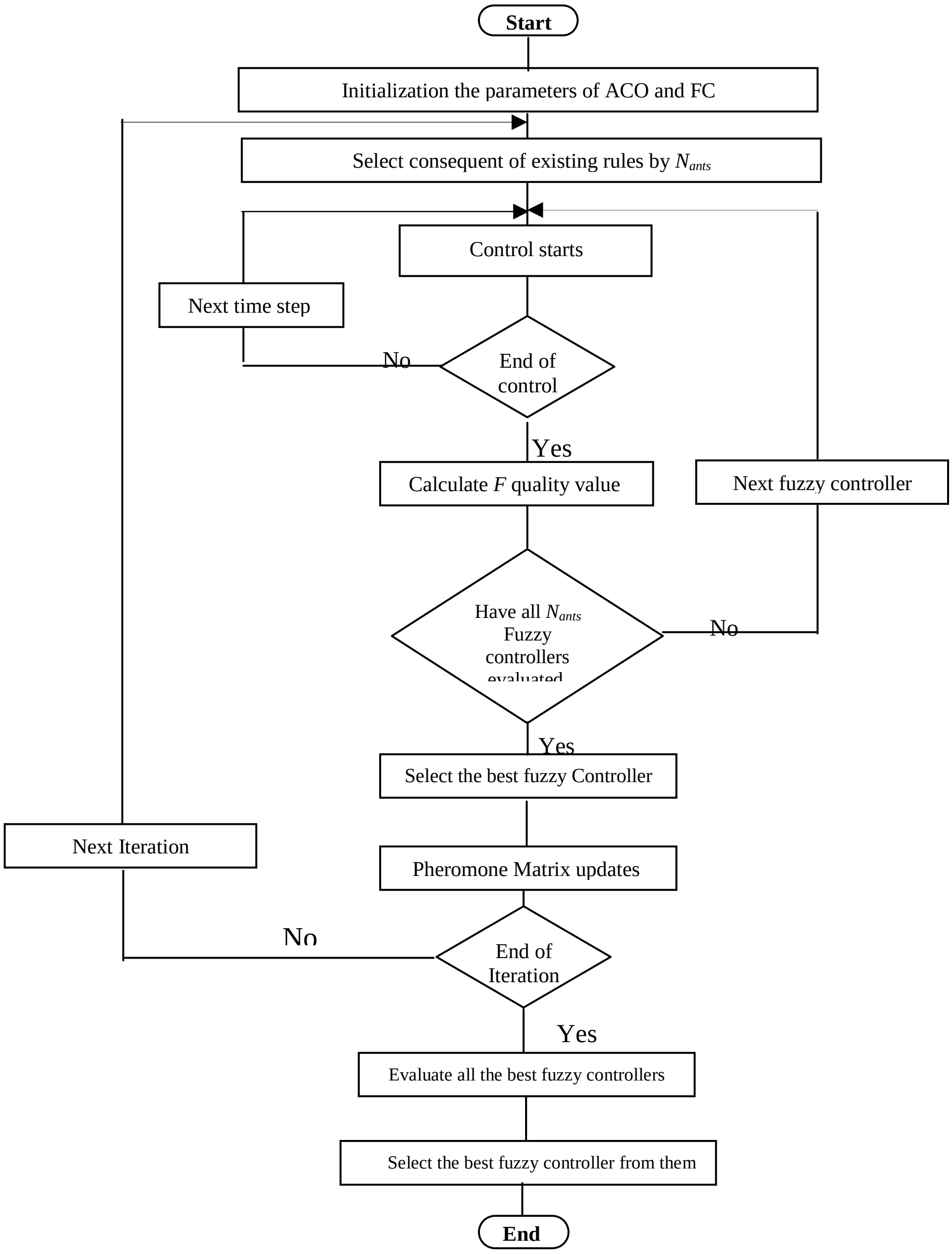



Fig.3: Flow chart of ACO for fuzzy controller design.

## 5. SYSTEM MODEL

We consider a DC shunt motor where the transfer function between the output angular displacement of this motor shaft $y(s)$ and its input control action $u(s)$ after simplification is given by [21]:

$$\frac{y(s)}{u(s)}=\frac{K_b}{JL_aS^3+(R_aJ+BL_a)S^2+(K_b^2+R_aB)S} \tag{11}$$

Where $R_a$ is the armature resistance in ohm, $L_a$ is the armature inductance in Henry, $K_b$ is the back emf constant in volt/(rad/sec), $J$ is the moment of inertia of motor and load in Kg-m2/rad, and $B$ is the frictional constant of motor and load in N-m/(rad/sec).by using the following parameters $R_a$ =2.45 ohm, $L_a$ =0.035 H, $K_b$ =1.2 volt/(rad/sec), $J$ =0.022 Kg.m$^2$/rad , $B$ =0.5*$10^{-3}$ N.m/(rad/sec) as given in [21].The overall transfer function of the system is given below,

$$\frac{y(s)}{u(s)}=\frac{1.2}{0.00077\,s^3+0.0539\,s^2+1.441s} \tag{12}$$

## 6. SIMULATION RESULTS

In this section, the ACO-FC and PID like fuzzy controller are utilized for position control of DC motor. In ACO-FC, the number of ants is $N_{ants}$ =50, and parameter $\rho$ and $c_2$ are set to 0.9 and 1.9, respectively. The initial pheromone trails $\tau(0)s'$ are all set to one. The objective is to control the output $y(k)$ to track the desired $y_d(k)$ which is a unit step in task 1 and a sine wave signal in task 2. The controller input variables are error $e(k)$ and change of error $\Delta e(k)$, where $e(k)=y_d(k)-y(k)$ and $\Delta e(k)=e(k)-e(k-1)$. For each input variable seven fuzzy sets used five triangles and two trapezoidal. To increase control accuracy, the candidate action number $N$ is selected to be larger or equal to the rules number $r$. By this way, no fuzzy rules are forced to share the same consequent action [13]. There are 49 fuzzy rules, for each fuzzy rule the Control action is selected from *49* candidates, where the set *U=0.7*{-1,-0.958,-0.916,….,0,...,0.916, 0.958,1}*. The quality function $F$ is defined as in equation (7) where *n* represents the number of iteration equals 1000, with a total number of 15 runs.

In the simulation a comparison between the ACO-FC and the PID like fuzzy controller is carried. The IF-THEN rule base for the PID like fuzzy controller is presented in [22].

### Task 1. The desired trajectory is:

A unit step $y_d(k)=1$

***Case1: the system without disturbance***

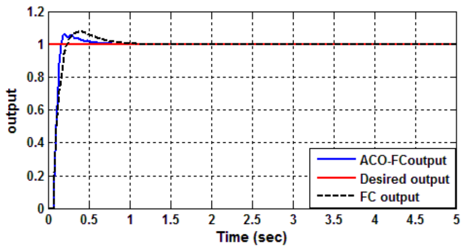
Figure 4 shows the simulation results of ACO-FC versus FC under step response. Table1 shows that The output of DC motor using ACO-FC is better than FC where the maximum overshoot is smaller, the system performance is faster where settling time and rise time is smaller in case ACO-FC and the error function for performance evaluation is defined to be the root mean square error (RMSE)

$$RMSE = \sqrt{\frac{\sum_{k=1}^{n} (y_d(k) - y(k))^2}{n}} \tag{13}$$

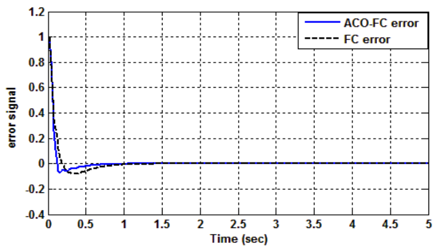
The RMSE for ACO-FC is better than FC. Although the execution time for ACO-FC program is greater than FC program but the performance of ACO-FC is better than FC, in addition to automating the design of fuzzy systems and avoiding the difficult of derivation of rules in fuzzy systems. Here the execution time on a PC with Intel core 2 duo, 2.1 G- HZ.

**Table 1.Comparison of results**

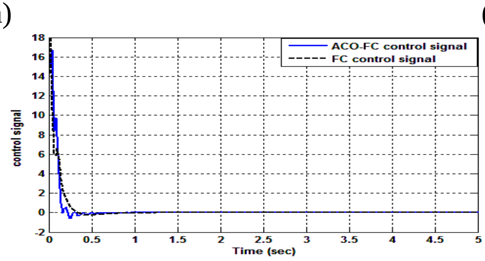
| Methods | ACO-FC | FC |
|---|---|---|
| Settling time | 0.48 | 0.78 |
| Rise time | 0.08 | 0.12 |
| Maximum over shoot | 0.07 | 0.08 |
| Peak time | 0.18 | 0.36 |
| RMSE | 0.0449 | 0.0471 |
| Execution time | 7.4 sec | 2 sec |



(a)



(b)



(c)

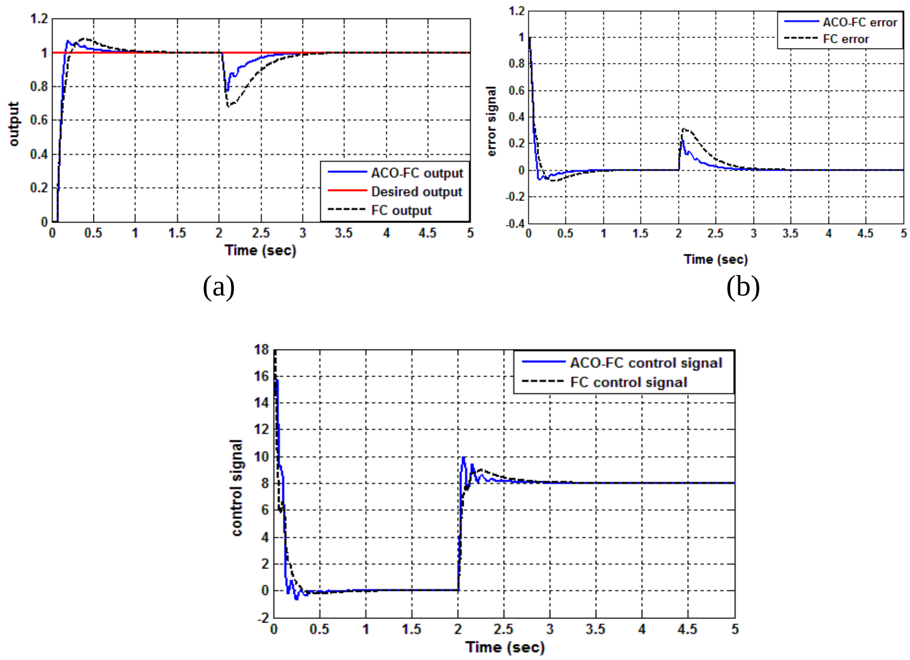
Fig.4: Simulation results for position control of DC motor without disturbance
a) output response, b) error signal, c) control signal

### *Case2: the system with disturbance*

Figure 5 shows the simulation results of ACO-FC versus FC under step response with step output disturbance of magnitude 10% at t=2 sec. From the simulation results, Table 2 shows that the ACO-FC still makes the system faster, maximum overshoot is smaller, and the system became capable of overcoming the disturbance better than FC where RMSE for the system using ACO-FC is better than FC. Although the execution time for ACO-FC program is greater than FC program but the performance of ACO-FC is better than FC, in addition to automating the design of fuzzy systems and avoiding the difficult of derivation of rules in fuzzy systems. Here the execution time on a PC with Intel core 2 duo, 2.1 G- HZ.

**Table2. Comparison of results**

| **Methods** | **ACO-FC** | **FC** |
|---|---|---|
| Settling time | 2.56 | 2.86 |
| Rise time | 0.08 | 0.12 |
| Maximum over shoot | 0.07 | 0.08 |
| Peak time | 0.18 | 0.36 |
| RMSE | 0.0484 | 0.0587 |
| Execution time | 7.4 sec | 2 sec |




(a) (b)

(c)

Fig.5: Simulation results for position control of DC motor with disturbance a) output response, b) error signal c) control signal

### Task 2. The desired trajectory is:

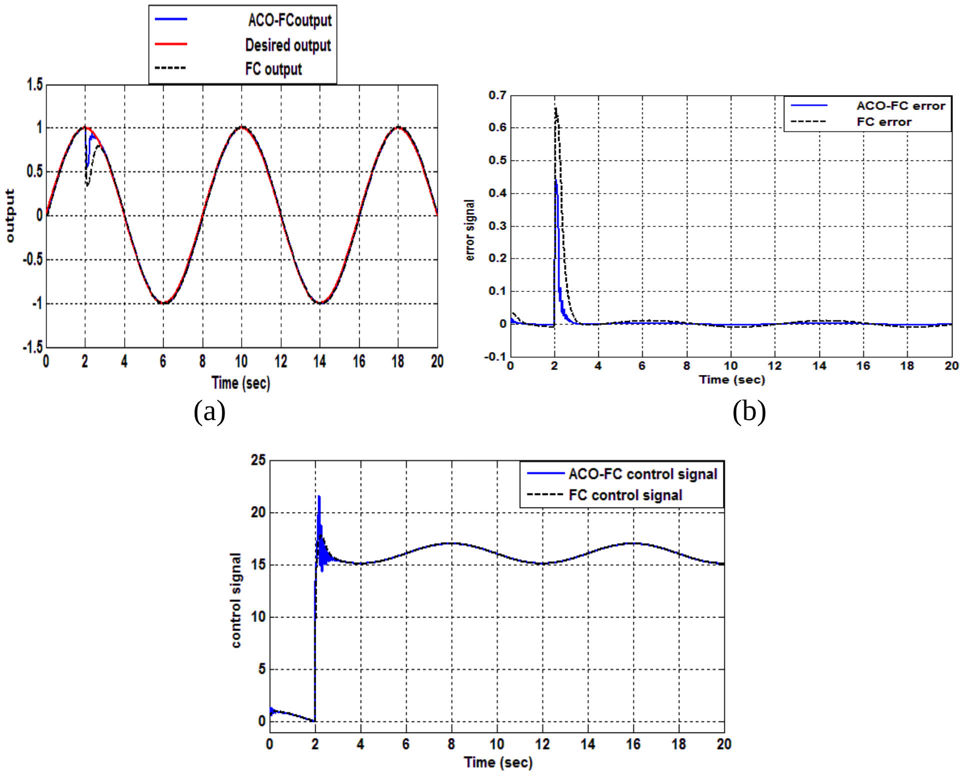
A sine wave $y_d(k) = \sin(\frac{\pi k}{200})$, $k = 1, \ldots, 1000$

Figure 6 shows the simulation results of ACO-FC versus FC with disturbance of magnitude 20% at t=2 sec. From the simulation results, Table 3 shows that the RMSE and the standard deviation (STD) of the sum of absolute error for ACO-

FC are better than FC. Although the execution time for ACO-FC program is greater than FC program but the performance of ACO-FC is better than FC, in addition to automating the design of fuzzy systems and avoiding the difficult of derivation of rules in fuzzy systems. Here the execution time on a PC with Intel core 2 duo, 2.1 G- HZ.

**Table3. Comparison of results**

| Methods | ACO-FC | FC |
|---|---|---|
| RMSE | 0.0375 | 0.0752 |
| STD | 0.0370 | 0.0729 |
| Execution time | 7.9 sec | 2.18 sec |




(a) (b)

(c)

Fig.6: Simulation results for position control of DC motor with disturbance a) output response, b) error signal c) control signal

## 7. CONCLUSIONS

Applying the ACO algorithm to a FC design, called ACO-FC, is presented in this paper which optimizes FC consequent parts. The Simulation results for the position control of the DC motor indicate that the ACO-FC provides better control performance than classical FC especially during disturbance. Using the ACO makes the design of FC easy and effective. The continuous ant colony optimization may improve optimization performance and will be studied in the future.

## REFERENCES

[1] W. Li and Y. Hori, “An algorithm for extracting fuzzy rules based on RBF neural network,” IEEE Trans. Industrial Electronics, Vol. 53, No. 4, PP. 1269– 1276, Aug. 2006.

[2] M. S. Ballal, Z. J. Khan, H. M. Suryawanshi, and R. L. Sonolikar, “Adaptive neural fuzzy inference system for the detection of inter-turn insulation and bearing wear faults in induction motor,” IEEE Trans. Industrial Electronics, Vol. 54, No. 1, PP. 250–258, Feb. 2007.

[3] C. H. Chou, “Genetic algorithm-based optimal fuzzy controller design in the linguistic space”, IEEE Trans. Fuzzy Systems, Vol. 14, No. 3, PP. 372–385, 2006.

[4] C. Cotta, E. Alba and J. M. Troya, “Evolutionary design of fuzzy logic controllers”, IEEE International Symposium on Intelligent Control, PP. 127-132, September 1996.

[5] C. J. Lin, “A GA-based neural fuzzy system for temperature control”, Fuzzy Sets Syst., Vol. 134, PP. 311–333, 2004.

[6] E. G. Mansoori, M. J. Zolghadri, and S. D. Katebi, “SGERD: A steady state genetic algorithm for extracting fuzzy classification rules from data”, IEEE Trans. Fuzzy Syst., Vol. 16, No. 4, PP. 1061–1071, Aug. 2008.

[7] M. Clerc and J. Kennedy, “The particle swarm—explosion, stability, and convergence in a multidimensional complex space”, IEEE Trans. Evolutionary Computation, Vol. 6, No. 1, PP. 58–73, 2002.

[8] A. Chatterjee, K. Pulasinghe, K. Watanabe, and K. Izumi, “A particle swarm-optimized fuzzy-neural network for voice-controlled robot systems”, IEEE Trans. Industrial Electronics, Vol. 52, No. 6, PP. 1478–1489, Dec. 2005.

[9] M. Dorigo, V. Maniezzo, and A. Colorni, “Ant system: Optimization by a colony of cooperating agents”, IEEE Trans. Systems, Man and Cybernetics, Part B, Vol. 26, No. 1, PP. 29–41, Feb. 1996.

[10] T. Stutzle and H. H. Hoos, “MAX MIN ant system”, Journal of Future Generation Computer Systems, Vol. 8, No. 16, PP. 889–914, 2000.

[11] M. Dorigo and L. M. Gambardella, “Ant colony system: A cooperative learning approach to the traveling salesman problem”, IEEE Trans. Evolutionary Computing, Vol. 1, No. 1, PP. 53–66, Apr. 1997.

[12] C. Blum and K. Socha, “Training feed-forward neural networks with ant colony optimization: an application to pattern classification”, 5th International Conference, Hybrid Intelligent Systems, 2005, 6 PP.

[13] C. F. Juang, C. M. Lu, C. Lo, and C. Y. Wang, “Ant colony optimization algorithm for fuzzy controller design and its FPGA implementation”, IEEE Trans. Industrial Electronics, Vol. 55, No. 3, PP. 1453–1462, Mar. 2008.

[14] R. S. Parpinelli, H. S. Lopes, and A. A. Freitas, "Data mining with an ant colony optimization algorithm," IEEE Trans. Evolutionary Computing, Vol. 6, No. 4, PP. 321-332, Aug. 2002.

[15] K. M. Sim and W. H. Sun, "Ant colony optimization for routing and Load-balancing: survey and new directions", IEEE Trans. Systems, Man, and Cybernetics, Part A: Systems and Humans, Vol. 33, No. 5, PP. 560-572, Sept. 2003.
[16] J. Cassillas, O. Cordon, and F. Herrera, "Learning fuzzy rules using ant colony optimization algorithms", International Workshops on Ant Algorithms From Ant Colonies to Arti Ants, PP. 13–21, 2000.
[17] M. Dorigo, M. Birattari and T. Stuzle, "Ant Colony Optimization, Artificial Ants as a Computational Intelligence Technique", IEEE Computational Intelligence Magazine, November 2006.
[18] M. Dorigo and G. D. Caro, "Ant colony optimization: A new meta-heuristic", IEEE Congress. Evolutionary Computing, Vol. 2, PP. 1470–1477, Jul. 1999
[19] M. Dorigo and T. Stutzle, "Ant Colony Optimization", Cambridge, MA: MIT, Jul. 2004.
[20] C. F. Juang and C. Lo, "Fuzzy systems design by clustering-aided ant colony optimization for plant control", International Journal of General Systems, Vol. 36, No. 6, PP. 623–641, Mar. 2007.
[21] N. Thomas, Dr. P. Poongodi, " Position Control of DC Motor Using Genetic Algorithm Based PID Controller", World Congress on Engineering, Vol. 2, july 2009.
[22] R. Malhotra and T. Kaur, "DC Motor Control Using Fuzzy Logic Controller", International Journal of Advanced Engineering Sciences and Technologies, Vol. 8, No. 2, PP. 291-296, 2011.

## تصميم متحكم غيمي للمحرك المستمر باستخدام طريقة الامثلية المبنية علي سلوك مستعمرة النمل

هذا البحث يقدم تصميم لمتحكم غيمى باستخدام طريقة الامثلية المبنية على سلوك مستعمرة النمل والهدف من ذلك تسهيل تصميم المتحكم الغيمى وكذلك تحسين اداؤه وتفادى الصعوبات التى كانت تواجه الباحثين فى تصميم المتحكم الغيمى .والمتحكم الغيمى يقوم اساسا على مجموعة من القواعد (rules) التى بها يتم التحكم فى النظام. والمشكلة تكمن فى استنتاج هذه القواعد وبمساعدة طريقة الامثلية المبنية على سلوك مستعمرة النمل تقوم كل نملة باستنتاج القواعد وتصميم متحكم غيمى خاص بها. والمستعمرة ككل بمساعدة مادة كميائية تسمى pheromone تتركها النملة ورائها اثناء مرورها لاستنتاج rules تتعاون مع بعضها فى الوصول الى افضل متحكم غيمى وذلك بتحديث طريق النملة التى وقع عليها الاختيار كافضل متحكم غيمى ثم تكرار العملية بعدالتحديث ليتم الوصول لحل افضل . وفى النهاية تتم مقارنة المتحكمات الغيمية واختيار الافضل منها .وتعتمد المقارنة بين المتحكمات الغيمية على دالة الجودة التى بها يمكن الحكم على اداء كل متحكم غيمى.وتم فى هذا البحث تطبيق هذه الطريقة على المحرك المستمر لعمل position control ومقارنة النتائج مع متحكم غيمى مصمم بالطريقة الكلاسيكية فوجد أن طريقة التصميم للمتحكم الغيمى المعتمدة على طرق الامثلية المبنية على سلوك مستعمرة النمل أفضل من الطريقة الكلاسيكية .