Research Journal of Applied Sciences, Engineering and Technology 6(5): 907-913, 2013
DOI:10.19026/rjaset.6.4140
ISSN: 2040-7459; e-ISSN: 2040-7467
© 2013 Maxwell Scientific Organization Corp.
Submitted: October 30, 2012 Accepted: December 21, 2012 Published: June 25, 2013

Research Article

Genetic Algorithm Tuned Fuzzy Logic Controller for Rotary Inverted Pendulum

[1]Tzu-Chun Kuo, [2]Ying-Jeh Huang and [2]Ping-Chou Wu
[1]Department of Electrical Engineering, Ching Yun University,
[2]Department of Electrical Engineering, Yuan Ze University, Chungli 320, Taiwan

Abstract: In this study, a Genetic Algorithm (GA) is proposed to search for the optimal input membership functions of the fuzzy logic controller. With the optimal membership function, the fuzzy logic controller can efficiently control a rotary inverted pendulum. The advantage of the proposed method is tuning the parameters of membership functions automatically rather than tuning them manually. In genetic algorithm, these parameters are converted to a chromosome which is encoded into a binary string. Because the membership functions are symmetric to zero, the length of each chromosome could be reduced by half. The computation time will also be shorter with the shorter chromosomes. Moreover, the roulette wheel selection is chosen as reproduction operator and one-point crossover operator and random mutation operator are also used. After the genetic algorithm completes searching for optimal parameters, the optimal membership function will be introduced to the fuzzy logic controller. Finally, simulation results show that the proposed GA-tuned fuzzy logic controller is effective for the rotary inverted pendulum control system with robust stabilization capability.

Keywords: Fuzzy logic control, genetic algorithm, inverted pendulum

INTRODUCTION

Fuzzy logic controllers have been operated successfully in many un-modeled or nonlinear systems that conventional control theories, like PID controllers, have difficulties to deal with (Klir and Yuan, 1995; Wang, 1996; Tao and Taur, 2000; Zidani *et al*., 2008; Wang and Nie, 2011). Experts use fuzzy logic controller with their own knowledge which is often coached in vague or linguistic statements in the form of IF-THEN rules. These linguistic statements are converted to precise numeric values by using input and output membership functions. This kind of control method has more simple calculation than other intelligent controllers, such as neural network controller. Moreover, fuzzy logic control is suitable for nonlinear systems because it is able to control without exact system transfer functions.

The first genetic algorithm was developed by Holland in 1975 (Goldberg, 1989). Many researches have been extended the application in search, optimization and machine learning. Genetic algorithms are global and robust over a wide range of problems. The searching procedures depend on the mechanics of natural genetics. According to the principle of Darwinian, survival-of-the-fittest, all natural species will survive by adaptation which is the fundamental power of genetic algorithm. A genetic algorithm usually adopts a fitness function to evaluate which chromosomes are worthy to reserve for next generation. By solving a polynomial, a chromosome mathematically stands for a set of roots. Chromosomes are operated by three operations, such as reproduction, crossover and mutation, to generate new offspring for next generation. These three operations make chromosomes to reveal the whole possible solution sets.

A lot of researches on rotary inverted pendulum controller have been proposed. For example, the model predictive control was a feedback control scheme. Further, a gain-scheduling control was proposed which is an effective method when the dynamics of the controlled plants changes critically due to the operating point. After achieving the requirements of every operating point, the gain-scheduling method will control the rotary inverted pendulum effectively.

Genetic algorithms are modern heuristic algorithms, which use natural population genetics to evolve solutions. Based on fuzzy logic control, we developed the genetic algorithm to obtain better parameters of the fuzzy controller to stabilize the rotary inverted pendulum. Membership functions of the proposed fuzzy logic controller will be tuned by genetic algorithms.

ROTARY INVERTED PENDULUM DYNAMICS

The structure of rotary inverted pendulum is shown in Fig. 1. The rotary inverted pendulum is operated by a rigid rod pendulum rotating in a vertical plane. The rigid rod is attached to a pivot arm that is mounted on

Corresponding Author: Tzu-Chun Kuo, Department of Electrical Engineering, Ching Yun University, Chungli 320, Taiwan

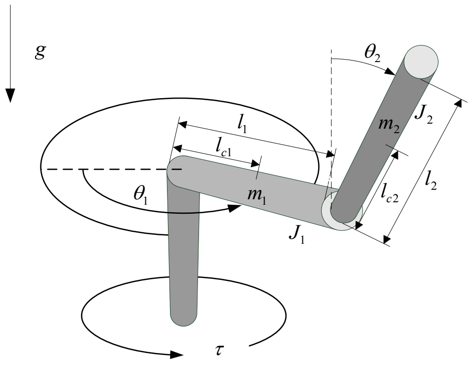

Fig. 1: Sketch of the rotary inverted pendulum

the shaft of the servo-motor. The pivot arm can be rotated in the horizontal plane by the servo-motor. The pendulum hangs downwards at a standstill situation. The goal here is to swing up the pendulum from the downward position and then to balance the pendulum in the vertical-upright position.

The system variables and parameters are defined as follows: m_1 is the mass of the rotary arm, m_2 is the mass of the inverted pendulum, l_1 is the length of the rotary arm, l_2 is the length of the inverted pendulum, l_{c1} is the distance of mass of the rotary arm to the extreme point, l_{c2} is the distance of mass of the inverted pendulum to the extreme point, J_1 is the inertia pendulum, θ_1 is the angle of the rotary arm, θ_2 is the angle of the inverted pendulum, $\dot{\theta}_1$ is the angle velocity of the rotary arm, $\dot{\theta}_2$ is the angle velocity of the inverted pendulum and g is the gravitational constant.

Applying the Lagrange equation to derive the dynamic equations of the rotary inverted pendulum yields:

$$\frac{d}{dt}\frac{\partial T}{\partial \dot{\theta}_i} - \frac{\partial T}{\partial \theta_i} + \frac{\partial V}{\partial \theta_i} = \tau_i \, , \tag{1}$$

where,
T = The kinetic energy
V = The potential energy
θ_i = The included angle of link i
τ_i = The moment of force of link i

For this rotary inverted pendulum, one can get kinetic energy and potential energy as follows:

$$\begin{aligned} T = & \frac{1}{2}m_1[(l_{c1}\dot{\theta}_1 \sin\theta_1)^2 + (l_{c1}\dot{\theta}_1 \cos\theta_1)^2] + \frac{1}{2}J_1\dot{\theta}_1^2 \\ & + \frac{1}{2}m_2[(-\dot{\theta}_1 l_1 \sin\theta_1 - \dot{\theta}_1 l_{c2} \cos\phi \cos\theta_1 + \dot{\phi} l_{c2} \sin\phi \sin\theta_1)^2 \\ & (\dot{\theta}_1 l_1 \cos\dot{\theta}_1 - \dot{\theta}_1 l_{c2} \cos\phi \sin\theta_1 - \dot{\phi} l_{c2} \sin\phi \sin\theta_1)^2 \\ & + (\dot{\phi} l_{c2} \cos\phi)^2] + \frac{1}{2}J_2\dot{\phi}_2 \end{aligned} \tag{2}$$

$$V = -m_2 g l_{c2}(1 - \sin\phi) \tag{3}$$

where, ϕ is defined as $\frac{\pi}{2} - \theta_2$.

Substituting (2) and (3) into (1) yields the following expression:

$$\begin{aligned} \tau_1 = & (m_1 l_{c1}^2 + m_2 l_1^2 + m_2 l_{c2}^2 \cos^2\phi + J_1)\ddot{\theta}_1 \\ & - 2m_2 l_{c2}^2 \cos\phi \sin\phi \dot{\theta}_1 \dot{\phi} - m_2 l_1 l_{c2} \cos\phi \dot{\phi}^2 \\ & + (m_2 l_1 l_{c2} \sin\phi)\ddot{\phi} \end{aligned} \tag{4}$$

$$\begin{aligned} 0 = & -(m_2 l_1 l_{c2} \sin\phi)\ddot{\theta}_1 + (m_2 l_{c2}^2 + J_2)\ddot{\phi}_2 \\ & + m_2 l_{c2}^2 \cos\phi \dot{\theta}_1^2 + m_2 g l_{c2} \cos\phi. \end{aligned} \tag{5}$$

The dynamics of the rotary inverted pendulum control system can be rewritten as:

$$M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + G(\theta) = \tau \tag{6}$$

where,

$\theta \in R^2$	: Vector respecting to angle
$\dot{\theta} \in R^2$	: Vector respecting to angle velocity
$\ddot{\theta} \in R^2$	: Vector respecting to angle acceleration
M (θ) $\in R^{2\times2}$	: The inertial matrix
C (θ,$\dot{\theta}$) $\in R^{2\times2}$	: A matrix which is relative to centripetal force and Coriolis force
G (θ) $\in R^2$	: A vector of gravity term
$\tau \in R^2$	: A vector of control input torque

The form of these matrices and vectors can be completely presented as follows:

$$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}, \dot{\theta} = \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}, \ddot{\theta} = \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} \tag{7}$$

$$M(\theta) = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}, C(\theta,\dot{\theta}) = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \tag{8}$$

$$G(\theta) = [G_1 \quad G_2]^T, \tau = [\tau_1 \quad 0]^T \tag{9}$$

where,

$$\begin{aligned} M_{11} &= m_1 l_{c1}^2 + m_2 l_1^2 + m_2 l_{c2}^2 \sin^2\theta_2 + J_1 \\ M_{12} &= m_2 l_1 l_{c2} \cos\theta_2 \\ M_{21} &= m_2 l_1 l_{c2} \cos\theta_2 \\ M_{22} &= m_2 l_2^2 + J_2 \\ C_{11} &= -\frac{1}{2} m_2 \, l_{c2}^2 \dot{\partial}_2 \sin(2\theta_2) \\ C_{12} &= -\frac{1}{2} m_2 l_{c2}^2 \dot{\theta}_1 \sin(2\theta_1) + m_2 l_1 l_{c2} \dot{\theta}_2 \sin\theta_2 \\ C_{22} &= 0, \ G_1 = 0 \\ C_{21} &= -\frac{1}{2} m_2 l_{c2}^2 \dot{\theta}_1 \cos(2\theta_2) \\ G_2 &= m_2 g l_{c2} \sin\theta_2 \end{aligned}$$

The input signal is the voltage feeding into the rotary inverted pendulum. A voltage signal is supplied to a PWM driver amplifier which drives the servo-motor to control the pendulum. The relationship of the control input τ_1 and the voltage v is:

$$\tau_1 = (k_b / R_m)v - (k_b^2 / R_m)\dot{q}_1 \tag{10}$$

where,
R_m : The armature resistance
k_b : Is the back-emf constant

Recalling (6) and (10), the system dynamic equation is presented as:

$$M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + G(\theta) = u \tag{11}$$

where, u = [v 0] and each element in above can be described in detail as follows:

$$C_{11} = (R_m / k_b)(-0.5 m_2 l_{c2}^2 \dot{\theta}_2 \sin(2\theta_2) + k_b^2 / R_m)$$

$$M_{11}(new) = (R_m / k_b)M_{11}, \quad M_{21} = m_2 l_1 l_{c2} \cos\theta_2$$

$$C_{22} = 0, \quad G_1 = 0, \quad M_{12}(new) = (R_m / k_b)M_{12}$$

$$C_{21} = -0.5 m_2 l_{c2}^2 \dot{\theta}_1 \cos(2\theta_2)$$

$$C_{12}(new) = (R_m / k_b)C_{12}$$

$$G_2 = m_2 g l_{c2} \sin\theta_2$$

FUZZY LOGIC CONTROLLER

The proposed fuzzy logic controller consists of four components including fuzzification, fuzzy rule base, fuzzy inference engine and defuzzification. The fuzzification converts a crisp value into a suitable linguistic value within a certain universe of discourse. The Gaussian-type membership functions are used in the fuzzification process:

$$\mu(p) = \exp\left[-\frac{1}{2}\left(\frac{p - m_i}{\sigma_i}\right)^2\right] \tag{12}$$

where,
p : The input variable of fuzzy logic controller
m_1 : The center of membership function
σ_i : The dispersion of membership function

Define the tracking error dynamics as:

$$e = y_d - y \tag{13}$$

$$\dot{e} = \dot{y}_d - \dot{y} \tag{14}$$

where $y_d = [y_{d1} \;\; y_{d2}]^T$ consists of the desired output of each joint and denotes the practical output of each joint:

$$y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \tag{15}$$

Table 1: Fuzzy rule base

e/ė	NB	NS	ZO	PS	PB
NB	NB	NB	NS	ZO	PS
NS	NB	NS	ZO	PS	PS
ZO	NS	NS	ZO	PS	PS
PS	NS	NS	ZO	PS	PB
PB	NS	ZO	PS	PB	PB

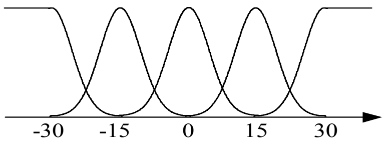

Fig. 2: Initial input membership functions of the fuzzy logic controller

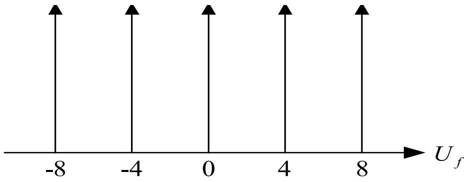

Fig. 3: Output membership functions of the fuzzy logic controller

Let the error e and the error differential $\dot{e}$ be the input linguistic variables of the fuzzy logic and the control force u be the output linguistic variable and the associated fuzzy sets for e, $\dot{e}$ and u are expressed as follows:

- **For e, $\dot{e}$ [the antecedent part]:** Negative Big (NB), Negative Small (NS), Zero (ZE), Positive Small (PS) and Positive Big (PB).
- **For u [the consequent part]:** Negative Big (NB), Negative Small (NS), Zero (ZE), Positive Small (PS) and Positive Big (PB).

The fuzzy rule base consists of a set of fuzzy IF-THEN rules:

Rule 1: IF e is NB and $\dot{e}$ is NB, THEN u is NB
Rule 2: IF e is NS and $\dot{e}$ is NB, THEN u is NS
⋮
Rule 25: IF e is PB and $\dot{e}$ is PB, THEN u is PB

Table 1 shows twenty-five linguistic rules. The universes of the input and output variables are [-30, 30] and [-8, 8], respectively. In this study, Gaussian-type input membership functions, singleton output membership functions and center average defuzzification method are adopted as they are computationally simple and easy for implementation.

Then, the control force can be obtained by the fuzzy inference mechanism as follows:

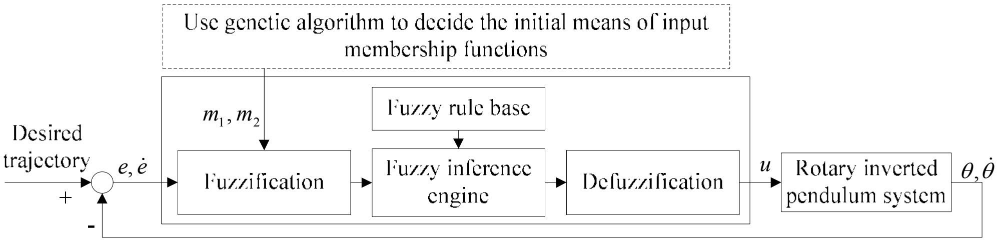

Fig. 4: Structure of the GA-tuned fuzzy logic controller

$$u = \frac{\sum_{i=1}^{25} w_i r_i}{\sum_{i=1}^{25} w_i} \tag{16}$$

where,

w_i = The firing strength of the corresponding rule

r_i = The center of the corresponding output membership function

Note that the universes of input and output variable affect directly the distributions of input and output membership functions. The input membership functions are shown in Fig. 2. The singleton output membership functions are uniformly distributed between -8 and 8 as shown in Fig. 3.

The input membership functions of the proposed GA-tuned fuzzy logic controller are tuned by genetic algorithm. Genetic algorithm gives a set of mean values in input membership functions of the fuzzy logic controller. Then, the fuzzy logic controller generates the control force u. Consequently, with substituting control force u, error e and error differential $\dot{e}$ into fitness function, the fitness value is obtained.

The block diagram of the proposed computation structure is shown in Fig. 4. Based on each fitness values of each chromosome in a generation, the genetic algorithm selects the best set of means for input membership functions. Further, the rotary inverted pendulum system is controlled by fuzzy logic controller according to the means of input membership functions selected by genetic algorithm.

GENNTIC ALGORITHM

The use of genetic algorithms is widespread in applications to engineering, scientific and business fields. Before genetic algorithms are applied, the optimization problem should be converted to a suitably described function which is usually called fitness function that may consist of one or more variables. The fitness function represents a performance index. If a genetic algorithm is applied to a control problem, the higher the fitness value is, the better the system performance will be. Genetic algorithms are to imitate the genetic operation processes, e.g., reproduction, crossover, or mutation, to obtain a solution corresponding to the fitness value.

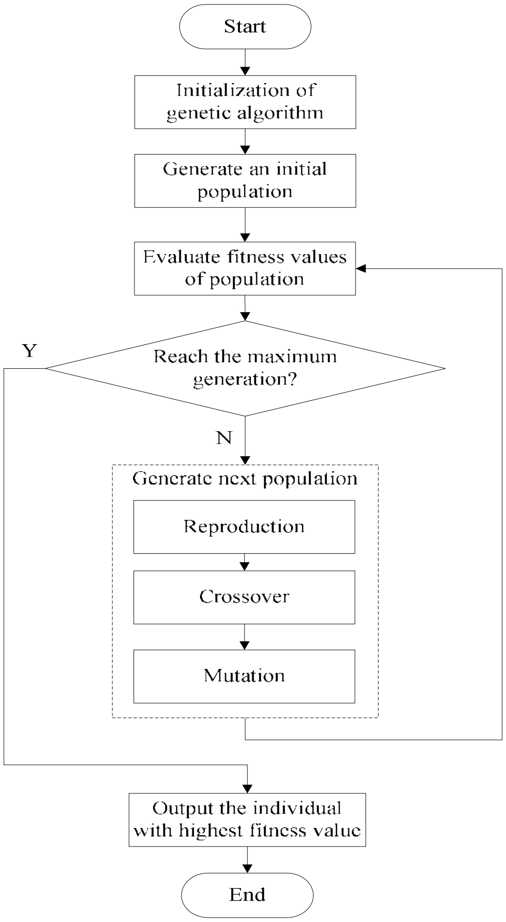
Start
Initialization of genetic algorithm
Generate an initial population
Evaluate fitness values of population
Reach the maximum generation?
Y
N
Generate next population
Reproduction
Crossover
Mutation
Output the individual with highest fitness value
End

Fig. 5: Flowchart of genetic algorithm approach

The flowchart of the genetic algorithm is shown in Fig. 5. Each block of the proposed genetic algorithm in the flowchart is described as follows.

Define a string of a chromosome: According to the accuracy and the range of parameter, one is able to determine the number of segments. Note that each factor may have different number of bits according to its range. The number of segments is the bit length. Define a fitness function which is a kind of index to evaluate how good a chromosome is. One has to determine the population size and the number of generation.

Assume the variable x_i are located on the interval [a_j, b_j] and the numbers are accurate to ten-thousandths. Thus, the variable should be at least divided into (b_j - a_j) ×10^4 parts. The required bit length m_j of the variable x_i could be obtained:

$$2^{m_j -1} < (b_j - a_j) \times 10^4 \leq 2^{m_j} -1 \tag{17}$$

Moreover, the real value could be obtained:

$$x_j = a_j + decimal(substring_j) \times \frac{b_j - a_j}{2^{m_j} -1} \tag{18}$$

where, decimal (substring_j) is simply converting binary to decimal integer and it can be evaluated with MTALAB command *bin2dec*.

Generate an initial population: Generate an initial population. N sets of chromosomes should be randomly generated before using a Genetic Algorithm (GA) operation. These chromosomes are called the initial population. Generally, larger population size requires fewer generations to come to an optimal solution. Moreover, the total computation time counts on N times the generation numbers. To create an initial population, the MATLAB command *randint* is adopted to generate matrix of uniformly distributed random binary numbers. However, *randint* generates numbers not strings.

Define the fitness function: Define the fitness function: The fitness function is the performance index of a GA to resolve the viability of each chromosome. The design of the fitness function is according to the performance requirements, e.g., convergence value, error, rise time[17]. In this study, the control objective is to achieve the stability as well as the robustness of the rotary inverted pendulum. We also expect the control force is small after achieving the objective. Thus, Due to the fitness function is related to control force and error, the fitness function is defined as:

$$Fitness = \frac{1}{(|\sigma| + |u|) + \varepsilon} \tag{19}$$

Evaluation: Calculate the fitness value of each chromosome after converting the string to the real value through Eq. (18). According to the fitness values of chromosomes, the proper chromosomes will be selected for further reproduction, crossover and mutation.

Reproduction: By the performance of index (the fitness value), better chromosomes will be selected from preceding population to generate offspring. Reproduction is the operator carrying old strings through into a new population, depending on the fitness value. Chromosomes with high fitness values obtain a high possibility of copies in the next generation. In this study, we choose roulette wheel selection as reproduction operator. In roulette wheel selection, the expected value of an individual is that fitness divided by the actual fitness of the population. Each individual can be assigned as a slice of the roulette wheel and the size of the slice being proportional to the individual's fitness. The wheel is spun N times, where N is the number of individuals in the population. On each spin, the individual under the wheel's marker is selected to be in the pool of parents for the next generation. If the population has N individuals, the fitness value of the i^{th} individual is f (C_i). Therefore, the probability for the individual being selected is:

$$P_s(C_i) = \frac{f(C_i)}{\sum_{j=1}^{M} f(C_i)} \tag{20}$$

After evaluating the probability of each individual, we randomly produce N numbers between 0 and 1 for determining which individuals are available to make their next offspring. These chromosomes are applied to further crossover and mutation operations.

Crossover operator: In this study, one-point crossover operator is adopted. The crossover operator exchanges the chromosome string of two selected parent chromosomes starting from a random index. It is an effective way of exchanging information and recombining segments from high-fitness chromosomes. The crossover procedure is to randomly select a pair of strings from the mating pool, then randomly determine the crossover position. Note that the randomly selection is the roulette wheel selection. There is also an example. The parent chromosomes are C_1 = 11011 and C_2 = 01100. There are only four crossover positions, because the length of each chromosome is five. If the crossover position is 3, then the offspring chromosomes are 11000 and 01111.

Mutation operator: The mutation operator is used to avoid the possibility of mistaking a local optimum for a global one. The local optimum implies premature convergence of the population. It is an occasional random change at some string position based on the mutation probability. Mutation of a bit involves flipping a bit, changing 0 to 1 and vice versa. It is worthy to mention that mutation operation is performed to produce new offspring after crossover operation. Different from crossover operator, the mutation operator has five mutation positions if the length of each chromosome is five. For example, if the parent chromosome is 11011 and the mutation position is

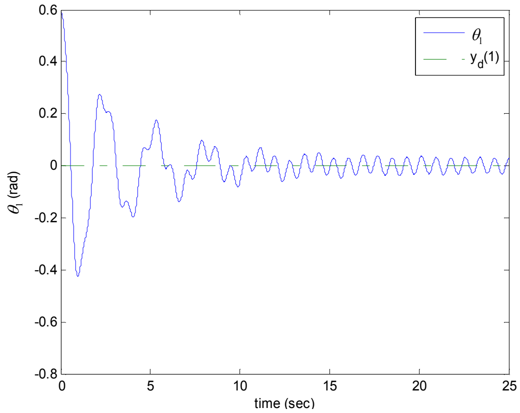

Fig. 6: Tracking performance 1 (without optimal membership function)

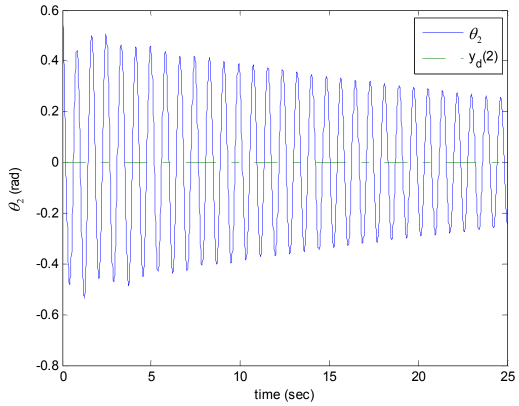

Fig. 7: Tracking performance 2 (without optimal membership function)

5, then the fifth bit will be switched. Thus the offspring chromosome is 11010.

Generate next generation or stop: The genetic algorithm uses the operations of reproduction, crossover and mutation to generate the next generation until the number of generation is satisfied.

SIMULATION RESULTS

The proposed fuzzy logic controller has five membership functions for each input. The initial means of the Gaussian-type input membership functions in (12) are -30, -15, 0, 15 and 30, respectively. Therefore, these ten membership functions have to be initialized since the fuzzy logic controller has two inputs, e and $\dot{e}$. All the standard deviations of the Gaussian-type input membership functions are 7.5. The fuzzy logic inference engine utilizes the product inference engine. The center average defuzzification method is adopted. Since the input membership functions are formulated to be symmetric to zero, the number of optimal mean values could be reduced from 5 to 2, which are both positive. Putting negative sign to the two mean values turns out another five input membership functions.

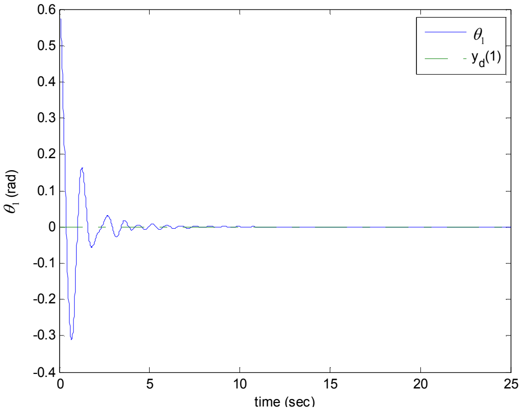

Fig. 8: Tracking performance 1 of the rotary inverted arm

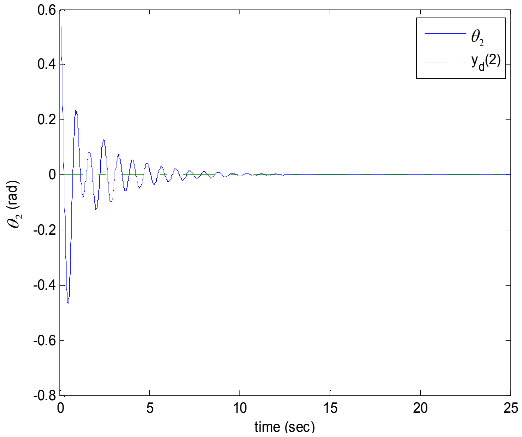

Fig. 9: Tracking performance 2 of the rotary inverted pendulum

The initial positions of the rotary arm and the inverted pendulum are both located at 0.5 rad. Let the tracking command be y_d $[0\ 0]^T$. In other words, the rotary arm and the inverted pendulum will be driven toward zero.

Firstly, the fuzzy controller without genetic algorithm is adopted. Figure 6 and 7 show the tracking performances of the rotary arm and the inverted pendulum, respectively. If the proposed genetic algorithm is involved, the corresponding results are shown in Fig. 8 and 9. Comparing Fig. 6 with Fig. 8, it is obvious that the rotary arm with GA-tuned fuzzy controller converged to zero after 10 sec, which is much better than that without GA-tuned controller. As for the inverted pendulum, Fig. 7 demonstrates that it was not stable to zero in 25 sec. Yet, the performance with GA-tuned controller is better. Figure 9 shows the inverted pendulum completely converged to zero after 13 sec.

CONCLUSION

In this study, a GA-tuned fuzzy logic controller for nonlinear model of the rotary inverted pendulum system has been presented. The mathematical model of the rotary inverted pendulum system has been driven. The proposed fuzzy logic controller adopted Gaussian-type input membership functions, singleton output membership functions and center average defuzzification method. The fitness function is the performance index of the genetic algorithm, which is with respect to tracking error and control force. We used genetic algorithm to automatically tune the membership functions of the fuzzy logic controller. Note that Roulette wheel selection was adopted as the reproduction operator. The crossover operator, a recombination operator, is applied to the mating pool to create a better offspring.

Simulation results demonstrate that the proposed controller for rotary inverted pendulum is effective in searching for optimal membership functions rather than trial and error method. It is worth to be mentioned, the length of each chromosome could be reduced by half because the membership functions are symmetric to zero. With the shorter chromosomes, the computation time will also be shorter. It is clear that the performance of the proposed fuzzy logic controller with genetic algorithm is better than that of fuzzy logic controller without GA.

REFERENCES

Goldberg, D.E.., 1989. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Boston, MA.

Klir, G.J. and B. Yuan, 1995. Fuzzy Sets and Fuzzy Logic: Theory and Applications. Prentice Hall, Eaglewood Cliffs, NJ.

Tao, C.W. and J.S. Taur, 2000. Flexible complexity reduced PID-like fuzzy controllers. IEEE T. Syst. Man Cy. B, 30: 510-516.

Wang, L.X., 1996. Stable adaptive fuzzy controllers with application to inverted pendulum tracking. IEEE T. Syst. Man Cy. B, 26: 677-691.

Wang, Y. and L. Nie, 2011. Application of fuzzy control on smart car servo steering system. Adv. Sci. Lett., 4: 2099-2103.

Zidani, F., D. Diallo, M.E.H. Benbouzid and R. Naït-Saïd, 2008. A fuzzy-based approach for the diagnosis of fault modes in a voltage-fed PWM inverter induction motor drive. IEEE T. Ind. Electron., 55: 586-593.