Design and Optimization of Fuzzy Controller for Inverse Pendulum System Using Genetic Algorithm

H. Mehrabian
University of Tehran
h.mehrabian@ece.ut.ac.ir

A. Ashoori
University of Tehran
a.ashoori@ece.ut.ac.ir

Abstract: *The reverse moving pendulum system, because of its nonlinear unstable and non-minimum phase dynamics has always been a good criterion for testing and evaluating control methods. In this paper, a new method based on fuzzy controlling of the system is presented. Several methods are suggested for optimizing the fuzzy controllers such as, genetic algorithm and neural network. We have used genetic algorithm to find the optimum set of fuzzy rules among all possible rules to control the system. This controller not only decreases the settling time, but also eliminates the fluctuations of steady state response acceptably and leads to an even motion of the pendulum, which results in a better controlling of the system in comparison with the existing classic methods.*

KeyWords: Fuzzy Controller, Inverse Pendulum System, Rules Base, Genetic Algorithm (GA), optimization.

1 INTRODUCTION

A fuzzy controller is based on human logic and attempts to behave like a human who controls a system. A fuzzy controller for a desired function requires correct adjustment. The number, shape and operational range of input and output membership functions, table of rules, etc are the parameters that should be adjusted according to the system specifications. Adjusting these parameters for a simple system might be done easily, but for a complicated system it is usually a challenge task, therefore some intelligent methods such as neural network, that has a high level of learning ability and Genetic Algorithm that is a powerful searching method for designing a fuzzy controller are applied.

In this document design of fuzzy controller based on adjusting the rule base using genetic algorithm is presented. Three methods are usually applied in adjusting rule base using Genetic Algorithm[10]

- Michigan method,
- Pittsburgh method
- iterative learning rules

In Michigan method each rule is considered as a chromosome and population consists of a number of rules. In Pittsburgh method the rule base is considered as a chromosome. In the third method chromosomes consist of a group of rules that each rule enters the chromosome after a large number of iterations. In this paper Pittsburgh method has been applied to design a fuzzy controller for inverse pendulum system based on the two methods presented in [1,3]. In part 2 the inverse pendulum system has been described. In part 3 Genetic Algorithm structure is given. Part 4 explains the procedure of providing Genetic Algorithm to achieve appropriate rule base for the fuzzy controller. Finally in part 5 Fuzzy controller design, the Genetic Algorithm optimization and simulations were done using MATLAB™ software. The simulation results verify successful application of our method to real motion situations.

2 THE INVERSE PENDULUM SYSTEM

To solve the problem of inverse pendulum system it is required to get familiar with Lagrange Formula in order to be able to formulate the operation of the system. Therefore we first describe the Lagrange Formula and then using this formula we formulate the movement of the inverse pendulum

2.1 Lagrange Formula

Suppose that $q=(q_1,q_2,...,q_n)$ are indications of orthonormal vectors of the space. If T is the Kinetic Energy of the system and V is the Potential Energy of the system, then $L = T - V$ is called Lagrangian. In this case the equations of the mobility of the system can be achieved from the following equations.

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = F_i$$

$$i = 1,2,\ldots,n$$

For the inverse pendulum system which is shown in figure (1) $q_1 = x$ and $q_2 = \theta$ (where x is position and θ is angle of the pendulum).

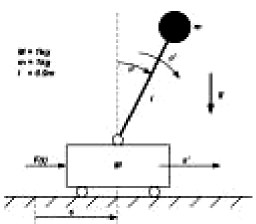

Figure 1: the parameters x, tetta definition

$$T = \tfrac{1}{2} J \dot{\theta}^2 + \tfrac{1}{2} M \dot{x}^2 + \tfrac{1}{2} m\left(\dot{x}_p^2 + \dot{y}_p^2\right)$$

$$V = mgl \times \cos\theta$$

$$x_p = x + l \times \sin\theta \qquad y_p = l \times \cos\theta$$

$$\Rightarrow T = \tfrac{1}{2}\left(J + ml^2\right)\dot{\theta}^2 + \tfrac{1}{2}\left(M + m\right)\dot{x}^2 + ml\,\dot{x}\,\dot{\theta} \times \cos\theta$$

$$L = T - V \tag{1}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) - \frac{\partial L}{\partial x} = F \tag{2}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} = 0 \tag{3}$$

$$(1,2,3) \Rightarrow \begin{cases} (m+M)\ddot{x} + ml\ddot{\theta} \times \cos\theta - ml\dot{\theta}^2 \times \sin\theta = F \\ (J + ml^2)\ddot{\theta} + ml\ddot{x} \times \cos\theta - mgl \times \sin\theta = 0 \end{cases}$$

The system model is also provided in figure (2)

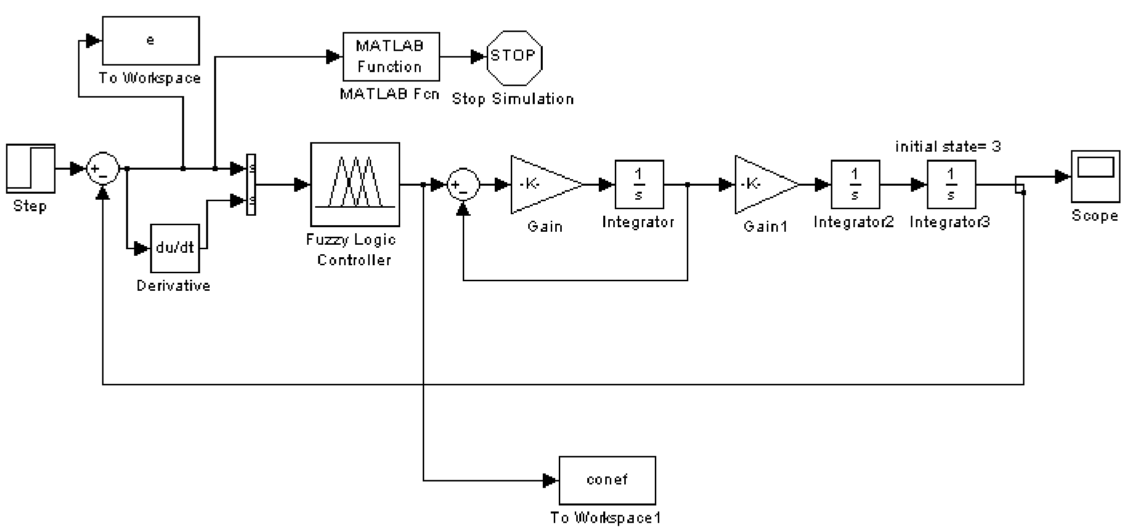

Figure 2: the inverse pendulum system model

3 GENETIC ALGORITHM

Genetic Algorithm (GA) is a powerful searching method that operates based on selecting and producing generation in nature. GA deals with a population of chromosomes and is in search of the one that has the best characteristics. The efficiency of each chromosome is evaluated with a function that indicates its goodness (cost function). In fact in an optimization we are looking for a chromosome that has the best performance. Each chromosome is constructed from a number of smaller parts called genes. Initially, GA chooses some chromosomes randomly and generates the first population. The cost function (how good and efficient the chromosome is) is calculated for each of these chromosomes. To produce next generation, the chromosomes that have bigger cost function values have higher probability to be chosen and GA operators are applied to them in order to produce new chromosomes. GA operators are crossover and mutation. In crossover a number of genes are two chromosomes are selected and exchanged to generate new chromosomes that make next generation and replace the previous generation. In mutation a gene of a chromosome will be selected and replaced by another gene that is produced randomly. Cost function is calculated for these chromosomes and other generations are produced according to its value.

4 PROVIDING GENETIC ALGORITHM

In this part a genetic algorithm for optimizing the rule base will be provided. Accurate preparation of this algorithm for a problem is an important part of solving it, and inaccurate adjustment of this algorithm may yield divergence in optimum response or may lead us to locally optimum responses.

The goal is to get an optimum rule base. Therefore, with a genetic algorithm approach, “an optimum rule base should be found”. To do this, each rule base is considered as a chromosome, the tiniest fracture of a rule base is a rule that describes the output membership function for a specific state of the angle and its derivative (angular velocity). So each rule can be considered as a gene.

A rule can be defined as follows:

If angle belongs to the membership function i and angular velocity belongs to the membership function j, then the control signal belongs to the membership function k . Regarding this definition each rule can be described as a three digit number of ijk.

Initial generation consists of a number of chromosomes that there genes are produced randomly. Genes of a chromosome may be repeated, which means having identical rules in rule base is allowed. Also some rules have identical if parts but different then parts, like 566 and 567, in this case there fuzzy or combination will be appear in the output. Cost function should be defined according to what is as suitable and unsuitability of a chromosome (rule base). The better the rule base is, the better the system will be controlled and the output will follow the input sufficiently. Cost functions may be $I_i = \int_0^t (|e| + |u|)dt$ and $I_i = \int_0^t (|e|^2 + |u|^2)dt$ for continues signals and $I_i = \sum |e| + |u|$ and $I_i = \sum (|e|^2 + |u|^2)$ for discrete signals.

To define cost function first each chromosome (rule base)and then the linear model of the inverse pendulum system together with the Fuzzy controller are put in a loop and are evaluated for several states of the input and output, after that the weighted summation of the obtained errors are calculated due to this equation: $f(RB_j) = \frac{1}{\sum_i \alpha_i \times I_i^j}$. In this equation I_i^j is the performance of the system for j^{th} rule base for i^{th} initial and final state, in which α_i is the weight of each I_i^j regarding the initial and final values. In this case the cost function will be maximized. Now the genetic algorithm operations can be exerted on chromosomes. Choosing operation is done according to the cost function and with a rotary method.

In the mutation operation, first a gene will be chosen randomly and another gene that is randomly produced will replace it in the chromosome. Therefore mutation changes a rule in the rule base.

Fuzzy controller has two inputs (error and its derivative) and one output signals. In the fuzzy controller Mamdani method and also operand MIN for AND and operand MAX for OR have been used.

Since 5 membership functions have been chosen for error and derivative of error, length of a chromosome that is the number of genes is assumed 25. Considering the 7 membership functions for the output, all of the forms that a rule can have are equal to $5 \times 5 \times 7 = 175$, therefore it is better to choose the number of population about 175.

As described above, each rule (gene) can be represented in 175 forms, since each chromosome has 25 genes and identical rules are allowed, all of the forms that a chromosome may be represented is 17525 .

5 PROVIDING THE OPTIMAL RULE BASE

In this part we perform GA with population number of 150, crossover probability of 1, mutation probability of 0.15 and simulation time of 15 seconds, for 200 generation, and analyze the results. Step response of the inverse pendulum system with a fuzzy controller and the best achieved rule base for a number of generations is provided and depicted in figure 4. As can be seen in the figure, the response does not have overshoot and its steady state error is zero.

As it is obvious from figure 5, it is probable that we don't get the best response in the last generation. The best response (maximum cost function value) is given in figure 5. Although there are some fluctuations in it, but it is increasing which means better responses will be obtained in newer generations.

The rule base correspondent to the best answer is provided in table (1). In this table the output membership functions are as follows.

- PL (stands for Positive Large),
- PM (stands for Positive Medium),
- PS (stands for Positive Small),
- Z (stands for zero),
- NL (stands for Negative Large),
- NM (stands for Negative Medium),
- NS (stands for Negative Small).

It can be seen that some boxes of the table are empty that occurs under three circumstances:

The inverse pendulum system does not include such error and error dot, so the fuzzy controller doesn't need to this type of rule.

Because of the overlap of the membership functions, this box is affected by the neighbors' boxes rules.

The dynamic of system orders that under this condition it doesn't require a rule and waits until the condition changes.

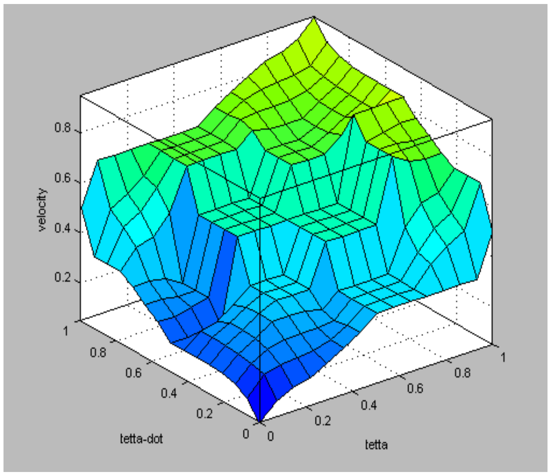

Figure 3: the optimum rule base

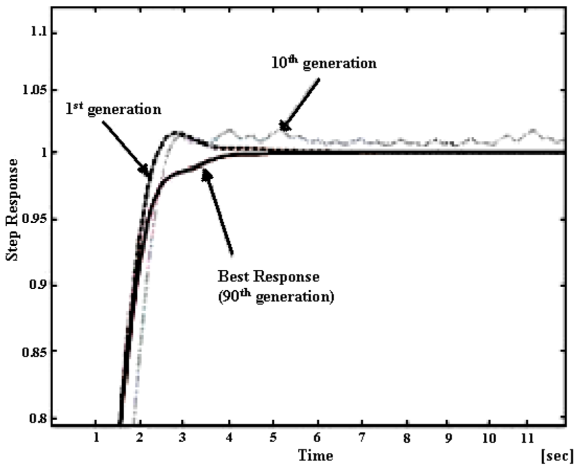

Figure 4: the step response for the optimum rule base

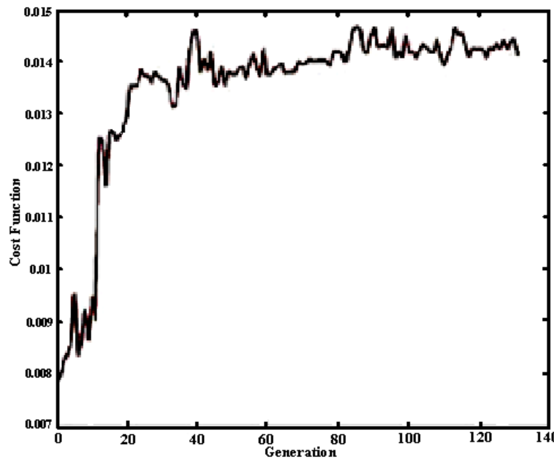

Figure 5: the cost function for the optimum response(rule base) in each generation

By omitting the repeated rules we can achieve a table, which is optimized regarding number of rules too. The fuzzy controller used in this system has two inputs and one output, so the plot of the graph (rules surface) can be shown in a three dimensional space. Figure (3) indicates the relationship between the error, the error dot and the output control signal.

error / error-dot	NL	NS	Z	PS	PL
NL	NL	NM	NS	NS	-
NS	NM	NS	-	-	PS
Z	NM	-	-	-	PM
PS	NS	-	-	PS	PM
PL	-	PS	PS	PM	PL

Table 1: optimum rule base

6 IMPLEMENTING THE RESULTS ON THE SYSTEM

The optimum fuzzy controller explained in the previous part is applied to the inverse pendulum in this section in real time mode.
The results using Simulink Toolbox of MATLAB software for some initial conditions are shown in figure 6 and figure7.

As it can be seen it has a small steady state error, which is because of the format of the cost function.
As can be seen in these figures the step response of the system has small overshoot when the error is big and the controller can control the pendulum with a very small overshoot while the error is small. Also the system has a small settling time.

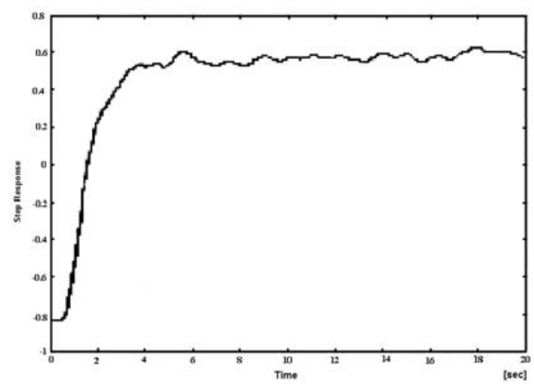

Figure 6: step response for initial state 1
[error , error-dot]= [0.2 , 0.05]

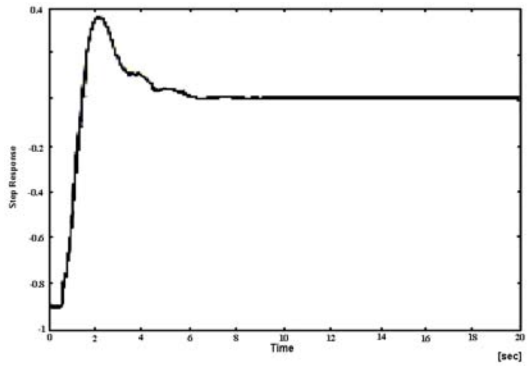

Figure 7: step response for initial state 2
[error , error-dot]= [0.5 , 0.05]

7 CONCLUSION

In this paper a fuzzy controller is designed for the system of the inverse pendulum system. To optimize the fuzzy controller we used Genetic Algorithm. In this method not only the rules, but also the number of rules is optimized. The results showed small overshoot and also small settling time which indicates the good performance of the designed system. The Genetic Algorithm succeeded to find the desired chromosome in 90th generation between 17525 chromosomes (after searching 12750 chromosomes) that shows the power of genetic algorithm.

8 REFERENCES

[1] A.Fatehi, C.Lucas, “Optimization of the Rule Base and Parameters of a Fuzzy Logic Controller by Genetic Algorithms”, Proc.Int.Conf Application of Fuzzy systems (ICAFS), Tabriz, Iran, pp236-240, 1994 .

[2] O.Cordon, F.Herrera, M.Lozano, “Genetic Algorithms and Fuzzy Logic in Control Processors”, University of Granada, Technical Report #DECSAI-98107, July, 1998 .

[3] R.Schleiffer, H.J.Sebastian, E.K.Antonsson, “Genetic Algorithms in Fuzzy Engineering Design”, 1999 ASME Design Engineering Technical Conferences, Las Vegas, Nevada, USA .

[4] O.Cordon, F.Herrera, F.Gomide, F.Hoffmann, L.Magdalena, “Ten Years of Genetic Fuzzy Systems: Current Framework and New Trends”, Universidad de Granada .

[5] F.Herrera, M.Lozano, J.L.Verdegay, “A Learning Process for Fuzzy Control Rules Using Genetic Algorithms”, Dept. of computer science and AI, University of Granada, Technical Report, #DECSAI-95108, February,1995.