ADAPTATION OF A FUZZY CONTROLLER'S SCALING GAINS USING GENETIC ALGORITHMS FOR BALANCING AN INVERTED PENDULUM

Adrian-Vasile Duka
"Petru Maior" University of Tg. Mures
aduka@engineering.upm.ro

ABSTRACT

This paper examines the development of a genetic adaptive fuzzy control system for the Inverted Pendulum. The inverted pendulum is a classical problem in Control Engineering, used for testing different control algorithms. The goal is to balance the inverted pendulum in the upright position by controlling the horizontal force applied to its cart. Because it is unstable and has a complicated nonlinear dynamics, the inverted pendulum is a good testbed for the development of nonconventional advanced control techniques. Fuzzy logic technique has been successfully applied to control this type of system, however most of the time the design of the fuzzy controller is done in an ad-hoc manner, and choosing certain parameters (controller gains, membership functions) proves difficult. This paper examines the implementation of an adaptive control method based on genetic algorithms (GA), which can be used on-line to produce the adaptation of the fuzzy controller's gains in order to achieve the stabilization of the pendulum. The performances of the proposed control algorithms are evaluated and shown by means of digital simulation.

Keywords: inverted pendulum, fuzzy control, genetic adaptive control, genetic algorithm

1. Introduction

The control of the inverted pendulum system is a challenging problem in Control Engineering. Due to its characteristics (nonlinear, unstable), this system is widely used to demonstrate and test various control algorithms.

Fuzzy control is a practical alternative for controlling the inverted pendulum, and also for other challenging control applications. It provides a convenient method for constructing nonlinear controllers by using heuristic information. In the case of the inverted pendulum, fuzzy control provides an attractive solution, since, for the balancing problem, the fuzzy control rules are easily deducted. While it is easy to describe human knowledge with fuzzy linguistic terms, it is not as easy to tune other parameters of a fuzzy system (e.g. the controller's gains). This is where the genetic algorithm comes into place.

A genetic algorithm (GA) can be seen as a probabilistically guided optimization technique modeled after the principles of genetic evolution.[5]

The GA applied in this case performs a random search on a population of fuzzy controllers to determine which one is the best to implement at specific sampling time periods.

Since the performance of the control system is evaluated by means of digital simulation, a Simulink model of the plant was first developed. Based on this model a direct fuzzy controller was developed and was roughly tuned. As tuning the controller's scaling gains proved difficult, a GA was developed which chooses the best solution for these parameters. The adaptive control algorithm based on the GA involves searching for an optimal controller during each (or every several) sampling periods. The fitness evaluation consists of characterizing the *expected* closed loop performance of each controller (encoded by its scaling gains) in the population based on error analysis. This type of evaluation requires a model of the plant for prediction purposes and a reference model for comparison.[8]

2. Dynamical Model of the Plant

The Inverted Pendulum system consists of a rigid pole attached to a cart by a free joint which allows it one degree of freedom. The cart is constrained to move along a linear horizontal direction when a force is exerted on it. If appropriate forces are applied to the cart, the pole can be kept in its unstable upright position.

The dynamic of the inverted pendulum system shown in Figure 1 is described by the nonlinear equations in (1).

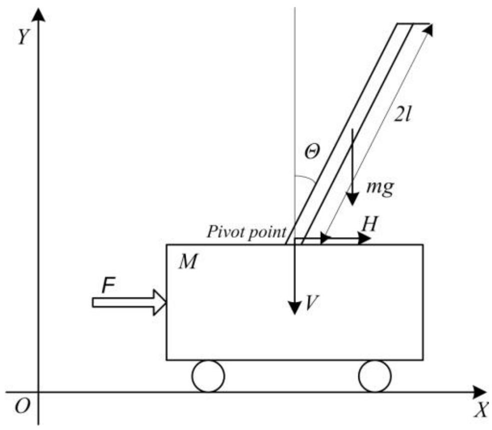

Figure 1. The inverted pendulum system

Assuming the pendulum is a uniform rod, its moment of inertia is $I=ml^2/3$, the nonlinear equations which describe the motion of the inverted pendulum system are:

$$\begin{cases} (M+m)\ddot{X}_p + ml\ddot{\Theta}\cos(\Theta) - ml\dot{\Theta}^2\sin(\Theta) = F \\ mgl\sin(\Theta) - ml^2\ddot{\Theta} - m\ddot{X}_p l\cos(\Theta) = I\ddot{\Theta} \end{cases} \quad (1)$$

Table 1 describes the parameters of equations (1).

Table 1: Plant parameters

θ	Angle of the pendulum	
F	Force applied to the cart	
X_p	Position of the pivoting point	
m	0.5 kg	Mass of pendulum
M	1 kg	Mass of cart
l	0.5 m	Distance between the pivot point *p* and the centre of gravity *cg* of the pendulum
m	9.81 m/s^2	Gravitational constant
I	0.0833 kg·m^2	Inertia of pendulum

A linear model of the plant was developed. This model will be used later by the genetic module to predict the behavior of the various controllers encoded in the population. Based on the response of this model, the controllers will be evaluated and the GA will choose the best solution which will control the actual plant.

To obtain the model of the plant used for prediction, equations (1) were linearized about the equilibrium point $\Theta = 0$. This linearization assumes the the following approximations for small angles:

$$\sin(\Theta) = 0;\ \cos(\Theta) = 1;\ \frac{d^2\Theta}{dt^2} = 0 \quad (2)$$

Considering the assumptions in (2), the following linear ISO (input state output) model resulted for the inverted pendulum:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \dfrac{g}{\frac{4}{3}l - \frac{ml}{m+M}}x_1 - \dfrac{1}{(m+M)(\frac{4}{3}l - \frac{ml}{m+M})}u \\ y = x_1 \end{cases} \quad (3)$$

where $x_1 = \Theta$ is the angle of the pendulum, x_2 is the rotational speed of the rod, $u = F$ is the input to the system and $y = x_1$ is the system's output.[2]

For implementation on digital systems, which is required for the computation of the GA, the ISO model of the plant in equation (3) was digitized with a sampling rate $T=0.001s$. The discrete ISO model is given in equation (4).

$$\begin{cases} \underline{x}(k+1) = \phi\underline{x}(k) + \Gamma u(k) \\ y(k) = H\underline{x}(k) \end{cases} \quad (4)$$

where: $\underline{x}(k)$ represents the state vector at sampling time kT, $u(k) = F$ is the control input of the plant at time kT and $y(k)$ is the system's output (the angle of the pendulum). Matrixes ϕ, Γ, H have the following values:

$$\phi = \begin{pmatrix} 1 & 0.001 \\ 0.0196 & 1 \end{pmatrix}$$
$$\Gamma = \begin{pmatrix} 0 \\ -0.0013 \end{pmatrix} \quad (5)$$
$$H = (1 \quad 0)$$

3. Development of the Fuzzy Controller

This section discusses the design of the direct fuzzy controller as applied to the inverted pendulum system.

Generally, a fuzzy controller presents itself as a nonlinear input-output mapping. The design of a fuzzy controller can be resumed to: choosing and processing the inputs and outputs of the controller and designing its four component elements (the rule base, the inference engine, the fuzzification and the defuzzification interfaces)

For the designed fuzzy controller, the error and change in error were considered as inputs, and the force applied to the cart was considered the output variable, thus resulting a PD type fuzzy controller. The crisp values for the input variables are computed as follows:

$$e(kT) = r - y = \Theta_r - \Theta$$
$$c(kT) = \frac{e(kT) - e(kT - T)}{T} \quad (6)$$

where $r=\Theta_r=0$ represents the setpoint of the system, which coincides to the upright position of the pendulum.
The universe of discourse of the variables (that is, their domain) was normalized to cover a range of [-1, 1] and scaling gains (g_e, g_c, g_u) were used to normalize. A standard choice for the membership functions was used with five membership functions for the three fuzzy variables (meaning $25 = 5^2$ rules in the rule base) and symmetric, 50% overlapping triangular shaped membership functions (Figure 2), meaning that only 4 ($=2^2$) rules at most can be active at any given time.

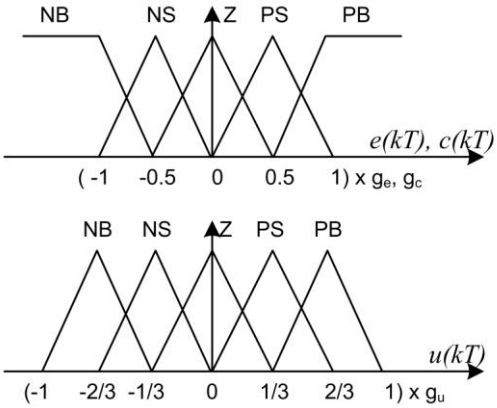

Figure 2. Membership functions for the input and output variables of the fuzzy controller

The fuzzy controller implements a rule base made of a set of IF-THEN type of rules. These rules were determined heuristically based on the knowledge of the plant and are very intuitive. An example of IF-THEN rule is the following:

IF e is negative big (NB) and c is negative big (NB) THEN u is positive big (PB)

This rule quantifies the situation where the pendulum is far to the right of the vertical and it is moving clockwise, hence a large force (to the right) is needed to counteract the movement of the pendulum so that it moves toward zero.[2]

The resulting rule table is shown in the Table 2, having a form which is specific to fuzzy PD systems.[1,3,9]

Table 2: The fuzzy controller's rule base

"force" u		"change in error" c				
		NB	NS	Z	PS	PB
"error" e	NB	PB	PB	PB	PS	Z
	NS	PB	PB	PS	Z	NS
	Z	PB	PS	Z	NS	NB
	PS	PS	Z	NS	NB	NB
	PB	Z	NS	NB	NB	NB

For the inference engine the min-max method was chosen and as defuzzification method the center of gravity (COG) was used. The resulting crisp value represents the controller output.
While determining the rule base of the controller, based on heuristic knowledge was quite intuitive, choosing the scaling gains (g_e, g_c, g_u) proved difficult and was done by roughly estimating the variation domain of the controller's input and output variables.
The results in Figures 4, 5 and 6 show the system's response (angle of the pendulum) when fuzzy controllers with different values for the scaling gains are applied to the plant. These results were obtained by simulation using the Simulink model of the closed loop system shown in Figure 3.

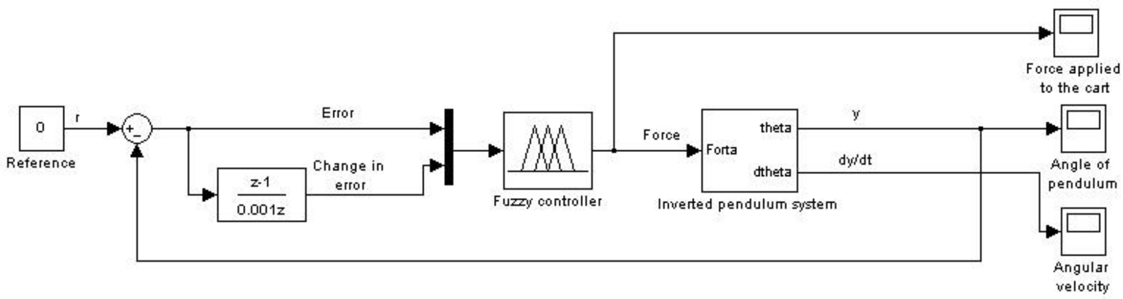

Figure 3. Simulink model of the closed loop system controlled by a fuzzy controller

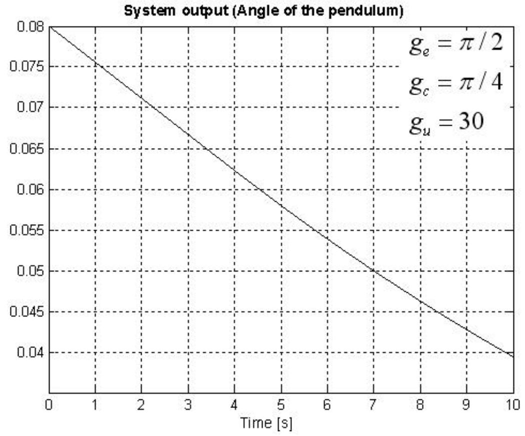

Figure 4. System output *(g_e, g_c, g_u)=(π/2, π/2, 30)*

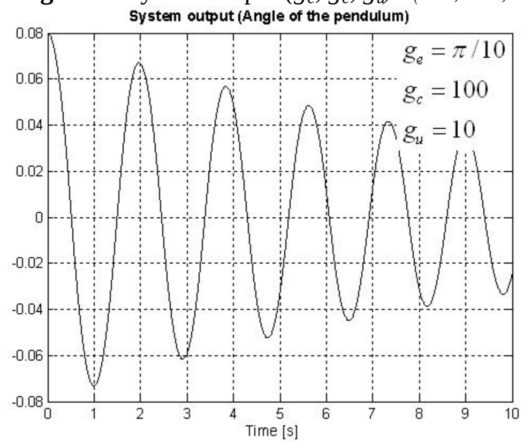

Figure 5. System output *(g_e, g_c, g_u)=(π/10,100,10)*

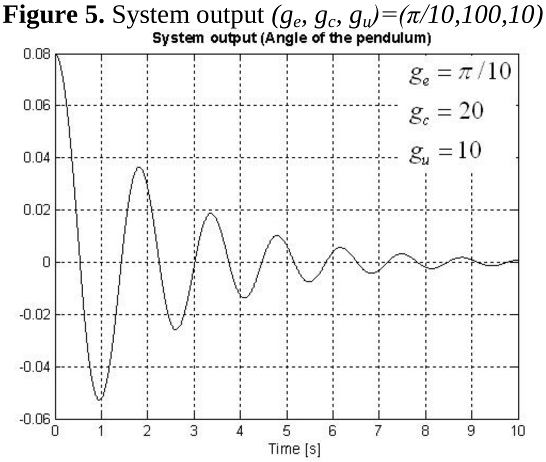

Figure 6. System output *(g_e, g_c, g_u)=(π/10,20,10)*

For all three simulations the same initial condition was considered. This corresponded to a 10 degrees displacement of the pendulum. As seen from the results, testing the system for various values of the scaling gains can lead to more satisfactory results. However, this task is difficult and time consuming and will be handled by the GA presented in the next section.

4. Fuzzy Controller Tuning by a Genetic Algorithm

A Genetic Algorithm (GA) is a parallel search technique modeled after the principles of genetic evolution, which tries to find optimal solutions to complex optimization problems. Finding the best values for the scaling gains of a fuzzy controller can be considered such a problem. The GA presented in this section is used to tune the three scaling gains of the direct fuzzy controller developed in section 3.

4.1. The Proposed Genetic Algorithm

The GA performs a random search for the fittest element of a population within a search space. The population consists of strings of binary numbers, called chromosomes, which hold possible solutions of the problem. In the case of the inverted pendulum control system the chromosomes encode the scaling gains of the controller.

The members of the population are manipulated cyclically through the following genetic operations: selection, crossover, mutation and replacement, to produce a new generation (a new population) that tends to have higher overall fitness evaluation. By creating successive generations which continue to evolve, the GA will tend to search for an optimal solution for the controller's parameters.

The key to the search is the fitness function. This function is computed for each member (chromosome) of the population and characterizes how well that particular member solves the given problem. Parents for the next generation are selected based on this fitness value. That is, chromosomes with a higher fitness value are more likely to be chosen as parents for the next generation. During the crossover stage of the GA the selected members of a population are randomly paired together to "mate" and share genetic information. The resulting population suffers mutation and at the final stage in the GA the replacement operation introduces the new population.[4,6,7,8]

The implemented GA, in the case of the inverted pendulum control system, is performed as follows:

1. A population of n chromosomes is generated. Each chromosome has m genes (the information encoded by the chromosome is encoded as a binary m-bit string)
2. The fitness value for each chromosome in the population is computed
3. Based on the Roulette-Wheel selection algorithm the fittest chromosomes are selected.
4. For each of the selected chromosomes, a probability of crossover p_c is computed. Based on this probability some of the chromosomes will become the parents for the next generation and others will go to the next generation without producing any off-springs.
5. The resulting parents are paired together and a single-point crossover is performed.

This way two new chromosomes are produced which will replace their parents.

6. The population resulting after the selection and crossover stages suffers mutation with a probability p_m. This genetic operation randomly selects a gene in the population and changes its value.
7. The resulting population becomes the new generation
8. The algorithm is repeated from step 2 until an optimum is achieved or until a predefined number of generations are produced.

The proposed GA uses an additional genetic operation, called *elitism*, which assures that the chromosome with the highest fitness value in a population will propagate to the next generation without being modified.[8,10]

4.2. The Adaptive System

The control algorithm based on the GA developed in this paper involves searching for an optimal controller on every *N* sampling periods. This control algorithm is based on the GMBC (Genetic Model Based Control) structure shown in Figure 7.

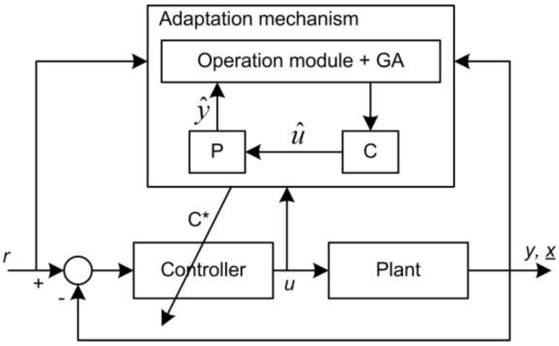

Figure 7. Genetic Model Based Control System[10]

The functional diagram in Figure 7 contains the following blocks:

1. The plant described by equations (1)
2. The fuzzy controller with the structure presented in section 3
3. The adaptation mechanism which produces the adaptation of the controller's gains based on the GA described in section 4.1

The GMBC uses the GA described earlier to evolve a population of possible controllers. Each controller in the population is evaluated using a fitness function that characterizes its expected performance based on error analysis. This type of evaluation requires a model of the plant for prediction purposes and a reference model for comparison. The prediction model *P* is the linear discrete model developed in section 2, equations (4) and (5). The reference model characterizes the desired response of the system. For balancing the inverted pendulum the obvious choice for the reference model is $y_m=r=0$.

The prediction model *P* uses the output $\hat{u}_i$ of each controller in the population ($C_i, i=\overline{1,n}$) and produces an estimated output $\hat{y}_i$ which is then compared to the reference model producing the error:

$$\varepsilon_i = y_m - \hat{y}_i \tag{7}$$

This prediction is performed over a horizon of 10 samples. At the end, the fitness of each controller in the population is evaluated and the fittest controller $C^* \in \{C_i, i=\overline{1,n}\}$ is selected to control the plant for the next 10 samples. Based on the fitness values associated to each controller, the population evolves to the next generation.

Since the control objective is to balance the pendulum, the fitness function used by the genetic algorithm considers the minimization of the square error for each controller in the population and it is computed over the selected horizon according to equation (8):

$$J_i = \sum_{j=1}^{N} \varepsilon_j^2 \tag{8}$$

where: ε_j denotes the estimation error at moment $j \in [1, N]$ corresponding to controller C_i in the population and *N* denotes the length of the horizon. In this case $N=10$.

For balancing the inverted pendulum the GA was used to tune the three scaling gains of the fuzzy controller as follows:

1. Each controller parameter was represented as a unique trait. Accordingly, there were 3 traits, represented with m_i genes per trait, making the total chromosome length for each individual $m=30$ genes. The probabilities of crossover and mutation were set to $p_c=0.8$ and $p_m=0.3$. The GA was run with a population size of $n=20$ chromosomes. As a starting point, the initial population was randomly chosen such that each trait was within pre-specified allowable ranges for each controller parameter (Table 3).

Table 3: Encoding of the scaling gains as chromosomes

Parameter	Number. of genes (m_i)	Lower limit (a_i)	Upper limit (b_i)
g_e	11	0	π/10
g_c	10	0	10
g_u	9	0	30

2. A sampling rate of $T=0.001s$ was chosen and a prediction horizon of $N=10$ samples was set
3. The initial conditions for the plant, the prediction model and the controller were set.
4. The reference input was set to $r(k)=0$ and the states of the system $\underline{x}(k)$ were measured. These were fed to the adaptive mechanism.
5. for each member of the population $C_i, i=\overline{1,n}$ the controller output $\hat{u}_i$ is computed using the known structure of the controller and then the predicted output $\hat{y}_i$ is determined using the prediction model of the plant P
6. ε_j is determined
7. We move on to the next sampling period $t=t+T$. As long as $j<N$ step 5 is executed
8. when $j=N$, the fitness function J_i for each member of the population is evaluated. This function captures the behavior of each controller if it had been controlling the plant for the duration of the selected horizon
9. the controller C_i with the highest fitness value becomes C^* and will control the plant for the duration of the next horizon.
10. the next generation of controllers evolves based on the GA and the cycle is repeated from step 4

The simulation of the GMBC for the inverted pendulum system was done using the Simulink model in Figure 8.

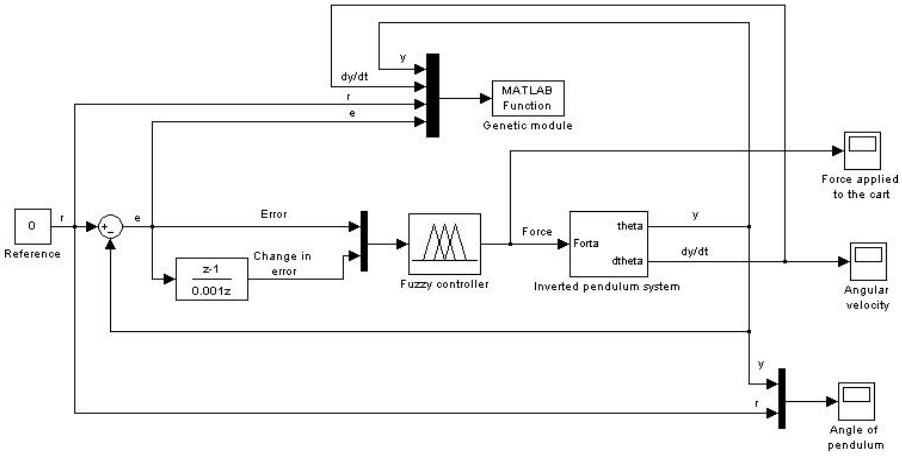

Figure 8. Simulink model of the Genetic Model Based Control system for the inverted pendulum

The initial conditions of the simulation considered $(g_e, g_c, g_u)=(1,5,5)$ as scaling gains of the fuzzy controller and a 10 degree displacement of the pendulum. The objective of the system was to balance the pendulum.

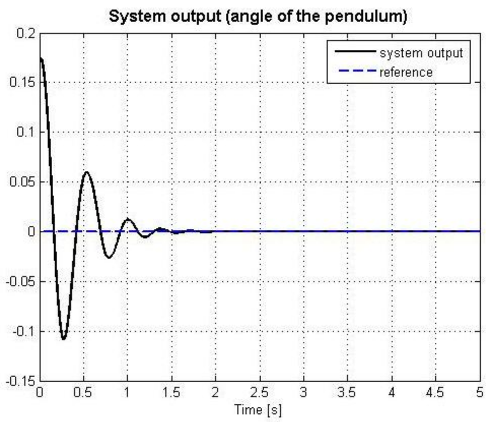

Figure 9. GMBC system output

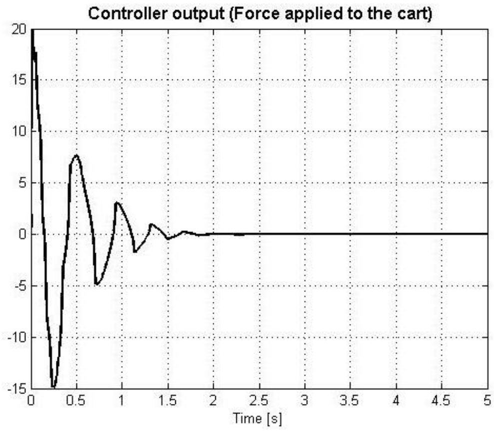

Figure 10. GMBC controller output

The output of the system (angle of the pendulum) is shown in Figure 9 and the output of the fuzzy controller (force applied to the cart) is shown in Figure 10.

The results show a faster stabilization of the pendulum than the one shown in Figures 4, 5 and 6.

To quantify the genetic evolution of the system, Figure 11 shows the evolution of the square error J_i of the fittest controller in each population.

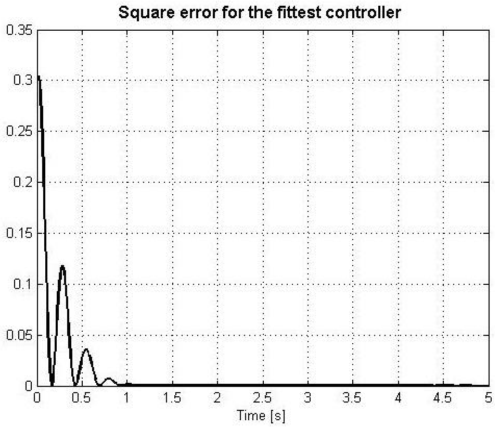

Figure 11. Evolution of the fitness value for the best controller in each population

5. Conclusion

Designers are especially attracted to fuzzy systems because fuzzy systems allow them to capture domain knowledge quickly using rules that contain fuzzy linguistic terms. However, tuning the parameters of these controllers heuristically is often difficult. The genetic adaptive technique presented in this paper offers a solution for the online adaptive tuning of a fuzzy controller's scaling gains. The plots shown throughout this paper represent the response of the adaptive schemes for one trial. They are not guaranteed to perform in the same manner given the same reference inputs and plant initial conditions. While all trials showed similar responses, the graphs presented here are not meant to convey any type of average behavior or guarantee of such.

References

[1] Butkiewicz, B., *About Robustness of Fuzzy Logic PD and PID Controller Under Changes of Reasoning Methods*, ESIT 2000, Aachen, Germany.

[2] Duka A.V., Oltean S.E., Dulău M., *Model Reference Adaptive vs. Learning Control for the Inverted Pendulum. A Comparative Case Study*, Journal of Control Engineering and Applied Informatics, December 2007, Vol 9, number 3-4, p.67-75

[3] Fayez, F. M, Maged N. F., *PID-Fuzzy Logic Position Tracking Controller for Detuned Field-Oriented Induction Motor Servo Drive*, WSEAS Transaction on Systems, 2004

[4] *Koza, John R., Genetic Programming. On the Programming of Computers by Means of Natural Selection*; The MIT Press, 1998

[5] Lee, Michael A., Takagi, Hideyuki, *Integrating Design Stages of Fuzzy Systems using Genetic Algorithms*, Proc. of the 2nd Int'l Conf. on Fuzzy Systems (FUZZ-IEEE'93), Vol.1, pp. 612-617, 1993

[6] Michalewicz, Zbigniew, *Genetic Algorithms + Data Structures = Evolution Programs*; 3rd edition, Springer-Verlag, 1999

[7] Mitchell, Melanie, *An Introduction to Genetic Algoithms*; The MIT Press, 1999

[8] Moore, Mathew L.; Musacchio John T., *Genetic Adaptive Control for an Inverted Wedge: Experiments and Comparative Analyses*; Engineering Applications of Artificial Intelligence, Vol. 14, No. 1, pp. 1-14, Feb. 2001

[9] Orhan, A., Akin, E., Karaköse, M., *DSP Implementation of Fuzzy Controlled Magnetic Levitation System*, Fuzzy Days 2001, LNCS 2206, p. 950–958.

[10] Porter, L.; Passino, K.M., *Genetic Adaptive and Supervisory Control*; Int. Journal of Intelligent Control and Systems, Vol. 2, No. 1, p. 1–41, 1998.