Applying Intelligent Controllers for Speed Regulation of DC Motors

Farzan Rashidi†, Mehran Rashidi‡ and Arash Hashemi-Hosseini†

†Engineering Research Institute of Jercen, Tehran, Iran

‡Hormozgan Regional Electric Co. & Faculty of Engineering, Hormozgan University, Bandar-Abbas, Iran

Abstract-- This paper describes the application of several new methods in order to control the speed regulation of DC motors. These methods are: Fuzzy auto tuning, Gas-based PID controller, Gas-Based Fuzzy PID Controller, Fuzzy PID Controller using neural network and Brain Emotional Learning Based Intelligent Controller (BELBIC). All the Control strategies utilize the output speed error and its derivative as feedback damping signals.

Index Terms—DC motor, Speed regulation, Fuzzy controller, Genetic algorithm, BELBIC

I. INTRODUCTION

Due to its excellent speed control characteristics, the DC motor has been widely used in industry even though its maintenance costs are higher than the induction motor [10]. As a result, speed control of DC motor has attracted considerable research and several methods have evolved. Proportional-Integral (PI) controllers have been widely used for speed control of DC motor. Kim et. al. [1,5] surveyed the current state of the PI, PID and command matching controllers for speed regulation of DC motors. To reduce the loading effect and minimize time delay, he added a feed forward controller to the PID controller. The organization of this paper is as follows: The following section formulates the system model of a DC motor. The focus of section III is on Fuzzy PID Controller and how it can be applied to DC motors. A brief review of Genetic Algorithm Based PID Controller is brought up in section IV. Section V, discusses the structure of the GA-Fuzzy controller and the method of incorporating initial knowledge. The focus of section VI is on review of Brain Emotional Learning Based Intelligent Controller (BELBIC) and how it can be applied to DC motors. In section VII, simulation results of the corresponding system are obtained and compared.

II. SYSTEM MODEL

As reference [2] we consider a separately excited DC motors as is shown in figure 1. The equations describing the dynamic behavior of the DC motor are given by:

$$V_a(t) = R_a i_a(t) + L_a \frac{di_a}{dt} + k\omega(t) \tag{1}$$

$$T(t) = J\frac{d\omega(t)}{dt} + \beta\omega(t) + T_1(t) = k i_a(t) \tag{2}$$

Where $\omega(t)$ rotational speed, $i_a(t)$ armature circuit current, $T_1(t)$ constant torque-type load, $R_a(t)$ armature circuit resistance, β coefficient of viscous-friction, k torque coefficient, J moment of inertia, and L_a armature circuit inductance. In state space form, if we let

$$x_a(t) = i_a(t), \qquad x_2(t) = \omega(t)$$

$$u(t) = V_a(t), \qquad \mathrm{d}(t) = \mathrm{T}_1(t) \tag{3}$$

Be our choice state and control variables, then the state space model of system can be represented by the following

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + Ed(t) \\ y(t) &= Cx(t) \end{aligned} \tag{4}$$

Where

$$x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}, \quad \mathrm{C} = \begin{bmatrix} 0 & 1 \end{bmatrix}, \quad \mathrm{E} = \begin{bmatrix} 0 \\ -\frac{1}{\mathrm{J}} \end{bmatrix}$$

$$A = \begin{bmatrix} -\frac{R_a}{L_a} & -\frac{K}{L_a} \\ \frac{K}{J} & -\frac{\beta}{J} \end{bmatrix}, \qquad \mathrm{B} = \begin{bmatrix} \frac{1}{L_a} \\ 0 \end{bmatrix} \tag{5}$$

The load torque is considered as disturbance input.

A. Numerical Values

The DC motor under study has the following specifications and parameters

a) Specifications

1hp, 220 volts, 4.8 amperes, 1500rpm

b) Parameters:

$R_a = 2.25\Omega, \mathrm{L_a} = 46.5mH, \mathrm{J} = .07\mathrm{kg.m}^2$

$\beta = 0.002 N.m. \frac{\sec}{rad}$, $\mathrm{K} = 1.1\mathrm{V}\frac{\sec}{\mathrm{rad}}$

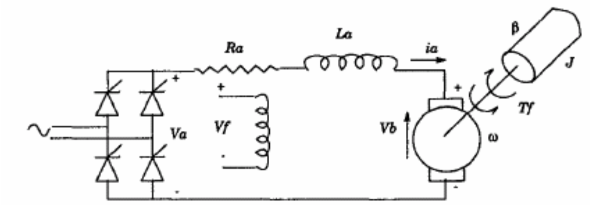


Figure1: The drive system of the separately excited DC motor [2]

III. FUZZY PID CONTROLLER

As reference [3,4,7] a tuning algorithm proposed in this paper should obtain the output of the plant from Ziegler-Nichols tuning with the relay feedback experiment in Astrom and Haglund. As we compute the error and the change error rate of the response, we can find the proportional gain and integral time using the error and change error rate of the output response as a membership function of fuzzy auto-tuner. The error change rate computes from the error of output by the equation (6)

$$\Delta e(k) = e(k) - e(k-1) \tag{6}$$

Fuzzy auto-tuner determine, the $K_p(k)$ and $\Delta\tau_i(k)$ by equation (7) from reasoning the change of proportional gain $K_p(k)$ and the change of integral time $\tau_i(k)$ from output of plant (7)

$$\begin{aligned} K_p(k) &= K_p(k-1) + \Delta K_p(k) \\ \tau_i(k) &= \Delta\tau_i(k-1) + \Delta\tau_i(k) \end{aligned} \tag{7}$$

A. Fuzzy rules

The fuzzy rules to determine $K_p(k)$ and $\tau_i(k)$ are the following equations (8) and (9)

1. The rule for $\Delta K_p(k)$
2. R^i : If $e(k)$ is A_j and $\Delta e(k)$ is B_j then $\Delta K_p(k)$ is $C_{j,i}$ (8)

Where $i(i = 1,...,9)$ is the number of rules

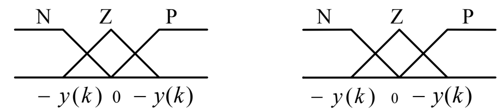
2. The rule for $\Delta\tau_i(k)$

R^i : If $e(k)$ is A_j and $\Delta e(k)$ is B_j then $\Delta\tau_i(k)$ is $D_{j,i}$

(9)

B. The membership function of the Premise

The membership function of the premise of the error $\Delta e(k)$ and the change error $e(k)$ are defined in triangle type in fig. 2(a) and (b)



(a) error (b) The change of error

Figure (2): Membership Function of premise

C. The parameter of the consequence

The parameter of the consequence defined by the fuzzy linguistic variable from the performance index shows that the increased proportional gains. Results in large overshot, a decrease of rise time, and the increased integral time τ_i results in a decreased rise time [6].

Table 1: Consequence Variable of K_p

$e(k)/\Delta e(k)$	N	Z	P
N	N	P	N
Z	P	Z	N
P	P	N	P

Table 2: Consequence Variable of τ_i

$e(k)/\Delta e(k)$	N	Z	P
N	P	N	P
Z	N	Z	P
P	N	P	N

IV. GAS-BASED PID CONTROLLER

The genetic Algorithm is based on the mechanics of natural selection and natural genetics the combines the notion of survival of the fittest, random and yet structured research and parallel evaluation of the nods in the search space. In GA, a result of problem is represented by a chromosome which usually is encoded into a binary bit string in computer programming. Many chromosomes make up of a population evolves [9]. According to the principle of survival of fittest, the population reproduces, crossover and mutates, and produce a new generation. Those processes are done again and again until the fittest chromosome is found. So the result of the problem is got. In GA, there is several working principle:

Encoding: The process in which the parameters are translated in to binary bit string is encoding. When encoding is finished, the population is formed and evolution can be done.

Fitness: Function: In the current generation, each of strings is decoded to be its corresponding actual parameters. Then these parameters are sent to a judgment machine, which yields a measure of the solution's quality evaluated with some objective functions and assigned individually with fitness value. This judgment machine is usually called fitness function $f(x)$. To a chromosome Xi, its fitness value is $fitness[i] = f(Xi)$.

Reproduction: According to fitness, the fitter the chromosome is, the more it is reproduced. The reproduce number of a chromosome is determined by an equation as follow:

$$\text{The Choromosome } Xi\text{'s reproduction number} = \frac{Fitness[i]}{\sum_{i}^{N} Fitness[i]} * N \tag{10}$$

Where N .is the number of all the chromosome. Thus, through reproduction operator, the chromosomes with high fitness are converted and the chromosomes with low fitness are deleted. This is survival of fitness.

Crossover: According to a certain proportion Pc, randomly selected two chromosomes from the population, randomly selected the position in which crossover is done, and the two chromosomes exchange a part of their bit string. So new gene combinations are produced and new chromosomes are produced. The evolution is doing on varying chromosomes, and the fitness of chromosomes is improved. The value of Pm usually ranges from 0.25 to 1.

Mutation: According to certain proportion Pm, randomly selected a chromosome, randomly selected a position of the chromosome. The number in the position is changed. That is 1 to 0, 0 to 1. Only depending on mutation operator can not get the optimization of the problem, but it can bring random value and make population cover the whole solving space. So we can get the global optimization. The value of Pm ranges from 0.01 to 0.3 From above introduction, we can see, GA is a search algorithm that continuously repeats these steps:

Reproduction, Crossover, and mutation, then makes the new generation fitter than the old generation, until the requirements are completed. So in this paper GAs are used to optimized PID parameters K_p, K_i and K_d. Firs, K_p, K_i and K_d are encoded to 16 bits string. The length total chromosome is 48 bits. It is supposed that K_p, K_i and K_d are bounded in the closed intervals $[0 \;\; K_{pm}]$, $[0 \;\; K_{im}]$ and $[0 \;\; K_{dm}]$ respectively. The decimal values of their corresponding binary strings are linearly related to their to range boundaries K_{pm}, K_{im} and K_{dm} Secondly, according to GAs operation: evaluation, crossover, mutation, K_p, K_i and K_d are optimized. After a prespecified number of generations, K_p, K_i and K_d are suitable enough to make system have good steady-state and dynamic performance.

GAs-Based Fuzzy Control Algorithm

GAs-Fuzzy algorithm consists of Encoding, Evolution, and Decoding sub-system [8]. GAs generates entire population of points (typically fixed-length chromosome like character strings) each with associated fitness value, test each point independently, and combines qualities from existing points to form a new population can be the fuzzy membership function parameters. The fitness value is computed using the information concerning the quality of the solution the produced by members of the population (objective function value). This adaptive evolutionary leaning process relates to the evolutionary selection procedure of genetic chromosomes. Genetic algorithms simulate this process, over generations, and identifies the most suitable candidate, i.e. gain the best membership function parameters. The Encoding sub-system uses a concatenated, mapped, using binary coding. Gaussion membership function is used for the fuzzy control module, two parameters, namely mean and standard deviation, control the Gaussion membership function. The value of each parameter can be encoded as a six bit binary number (for example, 000111). There is a membership function associated with every linguistic variable of a state or action in fuzzy Control rule. Each of these N m-bit strings represent one possible set of parameters. The input signals to the fuzzy logic controller consist of the error: $e = y - SV$ Between the reference input SV and actual output values y and its change $\dot{e}$. The fuzzy logic algorithm can be described as follows:

Determine the linguistic descriptions of the input $e, \dot{e}$ and the control variable u, and their membership function. For a given real-valued input e and $\dot{e}$, determine the membership degree corresponding two each linguistic description of e and $\dot{e}$ by means of the fuzzy membership functions, i.e., obtain vectors $e = [e_1, e_2]$ and $ec = [ec_1, ec_2]$ where e_1, e_2, ec_1 and ec_2 are the membership degree of the e and ec individually. Determine the fuzzy relation matrix R between the linguistic descriptions of e, ec and u. These relations are based on the expert knowledge which is given in terms of linguistic description. Calculate the vector $U = [k_1, k_2, k_3, k_4]$ in terms of the vector x and the fuzzy relation matrix R. Using centroid method, transform the vector U to the corresponding real number u. The output signal from the fuzzy logic controller u_f is determined by the Min-Max inference algorithm and the fuzzy rule base as shown in Table 1. The fuzzify section in the fuzzy logic controller change the true physical quantities to fuzzy quantities based on membership function. The defuzzification is carried out by the centroid method.

$$u = \frac{k_1\mu_1 + k_2\mu_2 + k_3\mu_3 + k_4\mu_4}{\mu_1 + \mu_2 + \mu_3 + \mu_4} k_\mu \tag{11}$$

Where μ_i is the minimum value of the membership function between μ_e and μ_{ec}. In this paper the exponential function is used as the membership function of fuzzy logic controller. The membership function is defined as following [4]:

$$\mu(x) = \begin{cases} 1 & 0 \le \mathrm{x} \le \mathrm{c} \\ e^{-k(x-c)^2} & \mathrm{x} > \mathrm{c} \end{cases} \quad (12)$$

When x is NB, $\mu(x) = e^{-k(x-c)^2}$ When x is Zero ,NS, NM, PS or PM.

$$\mu(x) = \begin{cases} 0 & \mathrm{x} \le \mathrm{c} \\ 1 - e^{-k(x-c)^2} & \mathrm{x} > \mathrm{c} \end{cases} \quad (13)$$

When x is PB. The shape of membership function of $\dot{e}$ is same as the shape of e. the membership function can be adjusted by changing its mean c and parameter k. If k is decreased, the membership function of this type become flatter. so there are thirteen parameters to need learning. They are

$K_{ZE}, K_{SE}, K_{ME}, K_{ZC}, K_{MC}, K_{SC}$

$K_c, K_u, K_{RBE}, K_{LBE}, K_{RBC}$

and K_{LBC}, where K_{ZE} is the value of k when e is Zero, K_{SE} is the value of k when e is NS or PS, due to the symmetry of the problem. K_{LBE} is the value of k when e is NB. Other parameters are similar to K_{ZE}, K_{SE} and K_{LBE}. As six bits are used to encode a parameter, the total chromosome length amounts 78 bits.

Table 3: The Fuzzy Logic Control Rules

$e \setminus \dot{e}$	NB	NM	NS	ZO	PS
NB	PVB	PB	PM	PS	PVS
NM	PB	PM	PS	PVS	PZ
NS	PM	PS	PS	PZ	Z
ZO	PS	PVS	PZ	Z	NZ
PS	PVS	PZ	Z	NZ	NVS
PM	PZ	Z	NZ	NVS	NS
PB	Z	NZ	NVS	NS	NM

V. PROPOSED NEURAL NETWORK ARCHITECTURE

In order to automate the determination of the Fuzzy-PID control gains, a neural network is trained to find the five control gains for the specific under consideration, given the desired plant output. The problem is basically a parameter estimation one; that is, given the desired plant output, we would like to find the five Fuzzy-PID control gains that define such an output. Determined the number of neurons in the input layer. The outputs of the network were the five control gain parameters, which stipulated a network with five outputs unit. The number of units in the hidden layer was determined individually for the particular problems here by using the ALADIN algorithm [11], and ranged from 3 to 48 neurons. The neural network was trained with 15 exemplars and then tested on 3 new ones.

The ALADIN algorithm used here determines the architecture of a feed forward analog output neural network with one layer of hidden units by selectively deactivating redundant hidden units during the training process. The ALADIN algorithm employs the back-propagation algorithm for training. The training process started with 99 hidden neurons for each problem. The ALADIN algorithm then pruned the neurons in hidden layer down to the point where the network could still learn and generalize to within an acceptable error criterion.

VI. BRAIN EMOTIONAL LEARNING BASED INTELLIGENT CONTROLLER (BELBIC)

BELBIC is the abbreviation for brain emotional learning based intelligent controller, whose description is reported in [12]. Motivated by the success in functional modeling of emotions in control engineering applications, in [12] a structural model based on the limbic system of mammalian brain [13], for decision making and control engineering applications has been developed. The schematic structure of BELBIC is illustrated in figure 3. The main parts that are responsible for performing the learning algorithms are orbitofrontal cortex and amygdala whose detailed learning formulas are given in [12,13]. BELBIC is essentially an action generation mechanism based on sensory inputs and emotional cues. In general, these can be vector valued, although in the benchmarks discussed in this paper for the sake of illustration, one sensory input and one emotional signal (stress) have been considered. The emotional learning occurs mainly in amygdala. The learning rule of amygdala is given as follows:

$$\Delta G_a = k_1 . \max(0, EC - A) \quad (14)$$

where G_a is the gain in amygdala connection, k_1 is the learning step in amygdala and EC and A are the values of emotional cue function and amygdala output at each time. The term max in the formula (14) is for making the learning changes monotonically, implying that the amygdala gain can never be decreased. This rule is for modeling the incapability of unlearning the emotion signal (and consequently, emotional action), previously learned in the amygdala [12,13]. Similarly, the learning rule in orbitofrontal cortex is shown in formula (15).

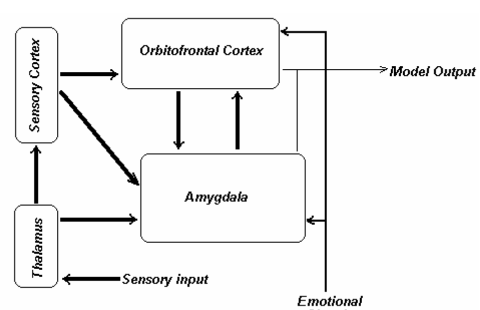


Figure 3- The abstract structure of BELBIC

$$\Delta G_o = k_2.(MO - EC) \quad (15)$$

where G_o is the gain in orbitofrontal connection, k_2 is the learning step in orbitofrontal cortex and MO is the output of the whole model, where it can be calculated as formula (16)

$$MO = A - O \quad (16)$$

in which, O represents the output of orbitofrontal cortex. In fact, by receiving the sensory input S, the model calculates the internal signals of amygdala and orbitofrontal cortex by the relations in (17) and (18) and eventually yields the output.

$$A = G_a.S \quad (17)$$

$$O = G_o.S \quad (18)$$

Since amygdala does not have the capability to unlearn any emotional response that it ever learned, inhibition of any inappropriate response is the duty of orbitofrontal cortex.

VII. COMPUTER SIMULATION RESULT

A. The result of fuzzy auto-tuning

The computer simulation results of the fuzzy auto tuning for ideal type PID controller are shown in fig.4. The overshoot, the rise-time and settling-time of the response are 8%, 1.6sec and 5sec respectively. It was improvement the overshot 19% and decreased the settling-time.

B. The results of the Gas-Based PID Controller

In this case we assume that the following performance specifications for the parameters K_P, K_i and K_d:

a) Steady-state error as small as possible.
b) Overshot<10%.
c) Settling-time<3 seconds.
d) Rise-time as small as possible.

The process of GA-based PID controller design is very simple suppose the range of K_P, K_i and K_d is from -30 to +30 and the required precision is 16 decimal places for all variables. Then the total length of the chromosome is 64 bits. Also the performance criterion is defined here as

J=rise-time + settling-time + overshoot +ITAE (integral of time multiplied by absolute error), and the fitness function (with non negative value) is defined as

Fitness=1/(1+J).

The computer simulation result of the GAs-based PID controller is shown in fig. 5.

The GAs-PID parameters were:
Number of generations: 500;
Population size: 100
Crossover probability: 0.96
Mutation probability: .001
Length of chromosomes: 46

the resulting performance parameters are overshoot = 0%, rise time = 0.6, settling-time = 2.0 sec

C. Results of GAs-Based Fuzzy Controller

The computer simulation results of GAs-based Fuzzy PID controller is shown in fig.6. The GAs-fuzzy parameters were:
Number of generations: 500;
Population size: 100
Crossover probability: 0.96
Mutation probability: .001
Length of chromosomes: 78

The parameters were searched for within the following intervals:

$K_{ZE}, K_{SE}, K_{ME}, K_{MC}, K_{SC}, K_{RBE}$
K_{LBE} and $K_{RBC} \in [0.1 \quad 2]$

$K_e \in [1.0 \quad 10], K_c \in [0.05 \quad 0.25]$
$K_u \in [0.1 \quad 0.5]$

$K_{ZC} \in [0.02 \quad 1], K_{SC} \in [0.5 \quad 2.9]$

The resulting performance parameters are overshoot = 0%, Rise-time = 0.8 sec, settling-time = .3 sec

D. The result of the fuzzy PID controller using neural network

The computer simulation results of fuzzy PID controller using neural network is shown in fig. 7. Presented below for this example are the output response obtained using the control gain parameter that the neural network generated. For this example the network was trained with 15 exemplars (off-line parameter estimation) and then tested on three new exemplars. For each set of the three testing data, the neural network gave the same control gain value, hence we show one output plot for this problem. The five control gains generated by the neural network are shown in below. Figure 7 shows the output of the Fuzzy-PID control system using the aforementioned gains.

Initial hidden neurons = 100, Hidden neurons after pairing =14, Input neurons =201, output neurons =5, Training exemplars =15, Testing Exemplars=3

E. The result of BELBIC

The computer simulation result of the BELBIC is shown in fig.8. The result shows the power of the proposed algorithm in the control of speed regulation of DC motors.

VIII. CONCLUSION

In this paper several effective speed regulation schemes for DC motor are presented. These methods were:

Fuzzy auto tuning, GAs-based PID controller, GAs-Based Fuzzy PID Controller, fuzzy PID controller using neural network and BELBIC. Simulation results shown that fuzzy

PID controller using neural network and BELBIC have better performance rather than other methods.

IX. REFERENCE

[1] Y. T. Kim & S. H. Baek, “The Speed Regulation of a DC Motor Drive System with a PI, PID, and Command Matching Controllers”, Dongguk Journal, Vol. 29, 1990, pp.525-541

[2] M.R.Akhbarzadeh and et.al, “Evolutionary fuzzy speed regulation for a DC motor”, IEEE 1997.

[3] Astrom and Hang, “Refinements of the Ziegler Nichols tuning formula for PID auto-tuners”, pros ISA conf., USA.

[4] G. Chen, “Conventional and Fuzzy PID Controllers: An Overview”, 1996.

[5] H. Yousef & H. M. Khalil, “Fuzzy logic-based Control of Series Connected DC Motor Drives”, Proceedings of the IEEE international Symposium on Industrial Electronics, pp. 495-499, 1995

[6] D. Misir, H. A. Malki, and G. Chen, “Design and Analysis of fuzzy PID Controllers”, The Jornal of Fuzzy Sets and System, May 1996

[7] W. Pedrycz, “An Identification Algorithm in Fuzzy Relational System”, Fuzzy Sets Syst, Vol.13, pp.153-167, 1994

[8] Jingzhe Yu, Zhen Ye, Chen Gue, “Using Genetic Algorithms in Fine-Tuning of Fuzzy Logic Controller”, in Proceeding of the IEEE ICIPS’97

[9] D. E. Goldberg, “Genetic Algorithms in Search optimization and Machine Learning”, Reading, MA: Assison-Wesley Publishing Company, Inc, 1991

[10] M. E. Fisher, A. Ghosh, & A. M. Sharaf, "Intelligent Control Strategies for permanent Magnet DC Motor Drives" , Proceedings of the IEEE International Conference on Power Electronics, New Delhi, India, 1996

[11] N.B.Karayiannis, “ALADIN Algorithms for learning and architecture determination”, IEEE Trans. Analog and digital signal processing, Vol.41, No.11, 1994.

[12] Lucas, C., Shahmirzadi, D., Sheikholeslami, N., 2002 “Introducing A Controller Based on Brain Emotional Learning Algorithm: BELBIC (Brain Emotional Learning Based Intelligent Controller)” Submitted to International Journal of Intelligent Automation and Soft Computing (AutoSoft), August

[13] Moren, J., Balkenius, C., 2000, “A Computational Model of Emotional Learning in the Amygdala” From animals to animals 6: Proceedings of the 6th International conference on the simulation of adaptive behaviour, Cambridge, Mass.

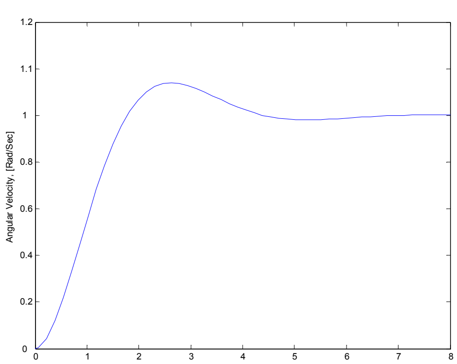


Figure 4: The result for Fuzzy auto tuning

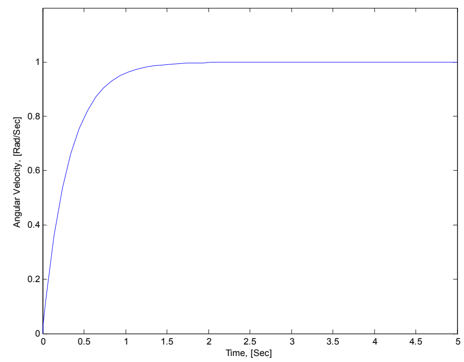


Figure 5: The result for GA-Based PID Controller

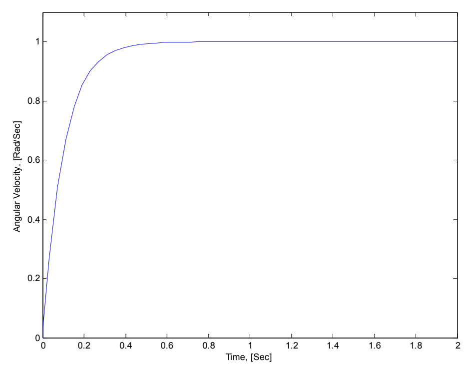


Figure 6: The result for GA-Based Fuzzy PID Controller

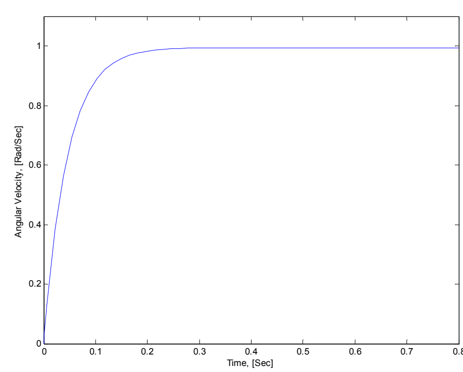


Figure 7: The result for fuzzy PID Controller using neural network

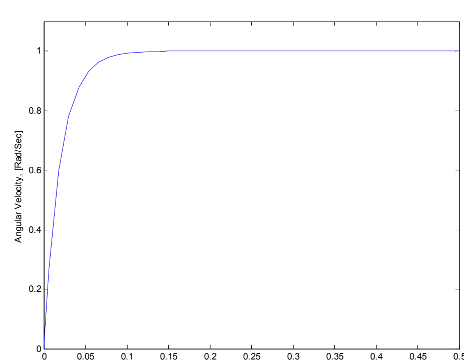


Figure 8: The result for BELBIC