International Journal of Electrical and Computer Engineering (IJECE)
Vol. 4, No. 5, October 2014, pp. 703~718
ISSN: 2088-8708

Design of Intelligent PID Controller for AVR System Using an Adaptive Neuro Fuzzy Inference System

Kamal Yavarian, Farid Hashemi, Amir Mohammadian
Department of Electrical Engineering, Ardabil Branch, Islamic Azad University, Ardabil, Iran

Article Info

Article history:

Received Jul 27, 2014
Revised Sep 16, 2014
Accepted Sep 30, 2014

Keyword:

Adaptive neuro fuzzy
Automatic voltage regulator
Inference system
Particle swarm optimization
PID Conrtoller
Signal to noise ratio

ABSTRACT

This paper presents a hybrid approach involving signal to noise ratio (SNR) and particle swarm optimization (PSO) for design the optimal and intelligent proportional-integral-derivative (PID) controller of an automatic voltage regulator (AVR) system with uses an adaptive neuro fuzzy inference system (ANFIS). In this paper determined optimal parameters of PID controller with SNR-PSO approach for some events and use these optimal parameters of PID controller for design the intelligent PID controller for AVR system with ANFIS. Trial and error method can be used to find a suitable design of anfis based an intelligent controller. However, there are many options including fuzzy rules, Membership Functions (MFs) and scaling factors to achieve a desired performance. An optimization algorithm facilitates this process and finds an optimal design to provide a desired performance. This paper presents a novel application of the SNRPSO approach to design an intelligent controller for AVR. SNR-PSO is a method that combines the features of PSO and SNR in order to improve the optimize operation. In order to emphasize the advantages of the proposed SNR-PSO PID controller, we also compared with the CRPSO PID controller. The proposed method was indeed more efficient and robust in improving the step response of an AVR system and numerical simulations are provided to verify the effectiveness and feasibility of PID controller of AVR based on SNRPSO algorithm.

Corresponding Author:

Farid Hashemi,
Departement of Electrical Engineering,
Islamic Azad University,
Ardabil Branch, Ardabil, Iran.
Email: farid.hashemi66@gmail.com

1. INTRODUCTION

Nowadays, economic and environmental constraints can lead to higher utilization of existing plant, deferred expenditure on system reinforcement (and longer distance between power plant and load center), with consequent erosion of stability margins. However, it is necessary to ensure that adequate stability margins are maintained for the reliable power supply. Multiple generators in a power station are connected to a common bus bar and each of these generators has an automatic voltage regulator (AVR) whose main objective is to control the primary voltage. Due to system disturbances the electrical oscillations may occur for a long time and might result in system instability. Hence effective control algorithms are required to alleviate these issues. The automatic voltage regulator (AVR) systems are used extensively in exciter control system. The role of an AVR is holding the generator terminal voltage constant under normal operating conditions at various load levels. The AVR loop of the excitation control system employs terminal voltage error for adjusting the field voltage to control the terminal voltage. Control parameters of the automatic voltage regulator (AVR) affect the power system dynamics and stability. Nowadays, more than 90% control loops in industry are PID control. This is mainly due to the fact that PID controller possesses robust performance to meet the global change of industry process, simple structure to be easily understood by engineers, and easiness to design and implement. The PID and its variations (P, PI, PD) still are widely applied in the motion control because of its simple structure and robust performance in a wide range of operating conditions. Unfortunately, it has been quite difficult to tune properly the gains of PID controllers because many industrial plants are often burdened with problems such as high order, time delays, and nonlinearities. Therefore, when the search space complexity increases the exact algorithms can be slow to find global optimum. Linear and nonlinear programming, brute force or exhaustive search and divide and conquer methods are some of the exact optimization methods. Over the years, several heuristic methods have been proposed for the tuning of PID controllers. These methods have several advantages compared to other algorithms as follows: (a) Heuristic algorithms are generally easy to implement; (b) They can be used efficiently in a multiprocessor environment; (c) They do not require the problem definition function to be continuous; (d) They generally can find optimal or near-optimal solutions. Particle swarm optimization (PSO) is an efficient and well known stochastic algorithm which has found many successful applications in engineering problems [1-4]. Signal to noise ratio algorithm doesn’t require a wide solution space, and the large number of searching and iterations were susceptible to related control parameters. On the other hand, this method has an effective appliance and better result for uncertainties conditions and different operation points. SNR has a responsible result in the nonlinear systems optimization. The integral performance criteria in frequency domain were often used to evaluate the controller performance, but these criteria have their own advantages and disadvantages [5-6]. In this study a novel design method for determining the optimal signal to noise ratio algorithm and particle swarm optimization (SNR-PSO) parameters for design the optimal proportional-integral-derivative (PID) controller of an automatic voltage regulator (AVR) system using the hybrid SNRPSO approach such that the controlled system could obtain a good step response output for some event and case that may be happen in the power system. After that we use ANFIS for training and obtained the fuzzy membership function (MF) for fuzzy inference system with result of our optimization. In this paper a fuzzy inference system models which takes K_G and τg as inputs and K_p , K_i and K_d as output. Therefore after make fuzzy inference system when our system inputs change our PID coefficient controller change an intelligently and fed to system and always our system have best operation.

2. LINEARIZED MODEL OF AN AUTOMATIC VOLTAGE REGULATOR (AVR) SYSTEM

Explaining The aim of Automatic Voltage regulator (AVR) control is to maintain the system voltage between limits by adjusting the excitation of the machines. The automatic voltage regulator senses the difference between a rectified voltage derived from the stator voltage and a reference voltage. This error signal is amplified and fed to the excitation circuit. The change of excitation maintains the VAR balance in the network. This method is also referred as Megawatt Volt Amp Reactive (MVAR) control or Reactive-Voltage (QV) control [7-14].

a. PID Controller

The PID controller is used to improve the dynamic response as well as to reduce or eliminate the steady-state error. The PID controller transfer function is:

$$G_{PID}(s) = K_p + \frac{K_i}{S} + K_d S \tag{1}$$

The functionalities of PID controller include: (a) the proportional term provides an overall control action proportional to the error signal through the all pass gain factor (b) The integral term reduces steady-state errors through low-frequency compensation (c) The derivative term improves transient response through high-frequency compensation.

b. Model of an AVR System

The role of an AVR is to hold the terminal voltage magnitude of a synchronous generator at a specified level. A simple AVR system comprises four main components, namely amplifier, exciter, generator, and sensor. For mathematical modeling and transfer function of the four components, these components must be linearized, which takes into account the major time constant and ignores the saturation or other nonlinearities. The reasonable transfer function of these components may be represented, respectively, as follows [14-19].

2.2.1 Amplifier model

The amplifier model is represented by a gain K_A and a τ_A time constant. The transfer function is

$$G_A = \frac{K_A}{1 + \tau_A} \tag{2}$$

Where the typical value of K_A is in the range of [10, 400] and τ_A is very small ranging from 0.02 to 0.1 s.

2.2.2 Exciter model

The transfer function of a modern exciter may be represented by a gain K_E and a single time constant τ_E .

$$G_E = \frac{K_E}{1 + \tau_E} \tag{3}$$

Where the typical value of K_E is in the range of [10, 400] and the time constant τ_E ranges from 0.5 to 1.0 s.

2.2.3 Generator model

The transfer function relating the generator terminal voltage to its field voltage can be represented by a gain K_G and a time constant τ_G

$$G_G = \frac{K_G}{1 + \tau_G} \tag{4}$$

These constants are loads dependent, K_G may vary between 0.1 and 1.0, and τ_G is between 1.0 and 2.0 s.

2.2.4 Sensor model

The sensor circuit, which rectifies, filters, and reduces the terminal voltage, is modeled by the following simple first-order transfer function

$$G_S = \frac{K_S}{1 + \tau_S} \tag{5}$$

Where τ_S range from of 0.001 to 0.06 s.

c. AVR System With PID Controller

The above models provide an AVR system compensated with a PID controller block diagram, which is shown in Figure 1.

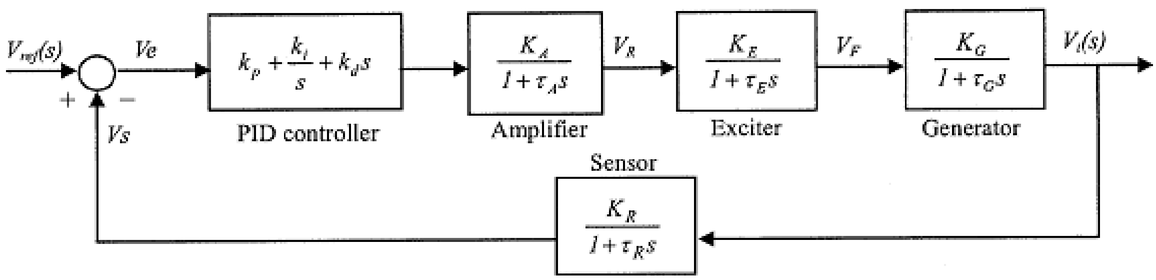

Figure 1. Block diagram of an AVR system with a PID controller.

3. HYBRID SIGNAL TO NOISE RATIO & PARTICLE SWARM OPTIMIZATION

This paper presents a SNR-PSO PID controller for searching the optimal controller parameters of AVR. In this section, a PID controller using the SNRPSO algorithm was developed to improve the step transient response of an AVR system. Signal-to-Noise Ratio (SNR) algorithm are used in this paper to evaluate existence possibility of optimal value in PID parameters. This algorithm does not require a wide solution space, and the large number of searching and iterations were susceptible to related control parameters. On the other hand, this method has an effective appliance and better result for uncertainties conditions and different operation points. Signal-to-Noise Ratio algorithm has a responsible result in the nonlinear systems optimization. Signal-to-Noise Ratio (SNR) is a measure of the variation within a trial when noise factors present. It looks like a response which consolidates repetitions and reflects noise levels into one data point. SNR consolidates several repetitions into one value that reflects the amount of variation present. There SNR are defined depending on the type of characteristic desired, higher is better (HB), lower is better (LB) and nominal is best (NB). The equations for calculating S/N ratios for HB, LB or NB characteristics are given as follows [19]:

a. Higher is better

$$\frac{S}{N_{HB}} = -10\ Log\left[\frac{1}{n\left((1/y_1)^2 + (1/y_2)^2 + \ldots + (1/y_n)^2 \right)} \right] \tag{6}$$

Where $y_1, y2, \ldots, y_n$ refer to the n observations within an experimental condition of *the* controllable factors.

b. lower is better

$$\frac{S}{N_{LB}} = -10\ Log\ \left((1/n) \sum y_i^2 \right) \tag{7}$$

Where n is the number of tests in a trial (number of repetitions regardless of noise levels).

c. Normal is best

$$\frac{S}{N_{NB1}} = -10\ LogV_e \tag{8}$$

$$\frac{S}{N_{NB2}} = +10\ Log\ \left((V_m - V_e)/nV_e \right) \tag{9}$$

The equipment utilization in this study is a "Lower is better" characteristic, since the equipment utilization is to be minimized. So we used the second equation for our response. In general, two arbitrary input considerate for SNR algorithm, one is for signal and the other is for noise. This inputs are selected from [0, 1] interval, due to the naturally of SNR algorithm. Hence, if the signal and noise are stand in this range, the results will be having a same signed and comparison for the best selecting will be without mistake. Signal to Noise Ratio (SNR) algorithm is used to generate the initial solution, it actually widens the search space of PSO besides increasing the efficiency. The position of the next generation is calculated according to PSO algorithm and it is repeated until meeting the end condition. The generation mechanism of solution adopts probabilistic distribution function, and one solution is deeply related to one another. Improper parameters are very likely to trap PSO into a local optimal solution, or make it require more time to find the global optimum. The SNRPSO algorithm was mainly utilized to determine three optimal controller parameters K_p, K_i and K_d such that the controlled system could obtain a good step response output. The design steps of SNR-PSO based PID controller is as follows.

a) Initialize the algorithm parameters like number of generation, population, inertia weight and constants.
b) Initialize the values of the parameters K_p, K_i and K_d randomly via Signal-to-Noise Ratio (SNR) algorithm.

c) Calculate the fitness function of each particle in each generation.
d) Calculate the local best of each particle and the global best of the particles.
e) Update the position, velocity, local best and global best in each generation.

Repeat the steps 3 to 5 until the maximum iteration reached or the best solution is found.

4. ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM (ANFIS)

Artificial intelligence, including neural network, fuzzy logic inference, genetic algorithm and expert systems, has been used to solve many nonlinear classification problems [20-23]. The main advantages of a fuzzy logic system (FLS) are the capability to express nonlinear input-output relationships by a set of qualitative if-then rules. The main advantage of an artificial neural network (ANN), on the other hand, is the inherent learning capability, which enables the networks to adaptively improve their performance. The key properties of neuro-fuzzy network are the accurate learning and adaptive capabilities of the neural networks, together with the generalization and fast learning capabilities of fuzzy logic systems. A neuro-fuzzy (ANFIS) system is a combination of neural network and fuzzy systems in such a way that neural network is used to determine the parameters of fuzzy system. A neural network is used to automatically tune the system parameters. The ANFIS is a very powerful approach for modeling nonlinear and complex systems with less input and output training data with quicker learning and high precision. The neuro fuzzy system with the learning capability of neural network and with the advantages of the rule-base fuzzy system can improve the performance significantly and can provide a mechanism to incorporate past observations into the classification process. In neural network the training essentially builds the system. However, using a neuro fuzzy scheme, the system is built by fuzzy logic definitions and is then refined using neural network training algorithms.

a. ANFIS Architecture

The modeling approach used by ANFIS is similar to many system identification techniques. First, a parameterized model structure (relating inputs to membership functions to rules to outputs to membership functions, and so on) is hypothesized. Next, input/output data is collected in a form that will be usable by ANFIS for training. ANFIS can then be used to train the FIS model to emulate the training data presented to it by modifying the membership function parameters according to a chosen error criterion. Operation of ANFIS looks like feed-forward backpropagation network. Consequent parameters are calculated forward while premise parameters are calculated backward. There are two learning methods in neural section of the system: Hybrid learning method and back-propagation learning method. In fuzzy section, only zero or first order Sugeno inference system or Tsukamoto inference system can be used. This section introduces the basics of ANFIS network architecture and its hybrid learning rule. The Sugeno fuzzy model was proposed by Takagi, Sugeno, and Kang in an effort to formalize a systematic approach to generating fuzzy rules from an input-output dataset. To present the ANFIS architecture, with two inputs, one output and two rules is given in Figure 2. In this connected structure, the input and output nodes represent the training values and the predicted values, respectively, and in the hidden layers, there are nodes functioning as membership functions (MFs) and rules. This architecture has the benefit that it eliminates the disadvantage of a normal feed forward multilayer network, where it is difficult for an observer to understand or modify the network. Here x, y are inputs, f is output, the circles represent fixed node functions and squares represent adaptive node functions.

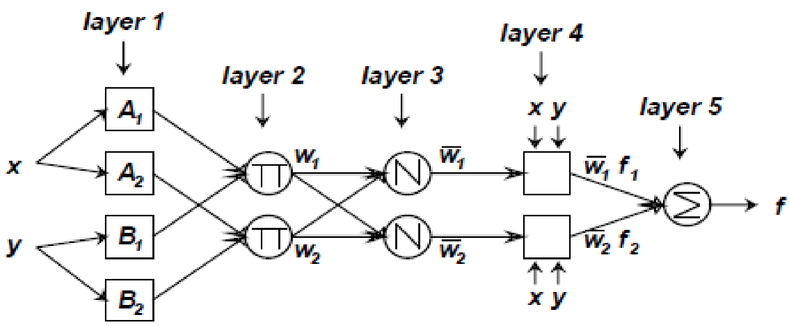

Figure 2. ANFIS architecture

Consider a first order Sugeno fuzzy inference system which contains two rules:

$Rule1: If$ X is A_1 and Y is B_1 , then $f_1 = p_1x + q_1y + r_1$

$Rule2: If$ X is A_2 and Y is B_2 , then $f_2 = p_2x + q_2y + r_2$

Where, p_1 , p_2 , q_1 , q_2 , r_1 , r_2 are linear parameters and A_1 , A_2 , B_1 , B_2 are nonlinear parameter. ANFIS is an implementation of a fuzzy logic inference system with the architecture of a five-layer feed-forward network. The system architecture consists of five layers, namely, fuzzy layer, product layer, normalized layer, de-fuzzy layer and total output layer. With this way ANFIS uses the advantages of learning capability of neural networks and inference mechanism similar to human brain provided by fuzzy logic. The operation of each layer is as follows: Here the output node *i* in layer *l* is denoted as O_i^l .

Layer 1 is fuzzification layer. Every node *i* in this layer is an adaptive node with node function

$$O_{1,i} = \mu A_i(x), \qquad for \ i = 1,2$$
$$O_{1,i} = \mu B_i(x), \qquad for \ i = 3,4 \tag{10}$$

Where *x* is the input to i_{th} node, O_i^l is the membership grade of *x* in the fuzzy set *Ai*. Generalized bell membership function is popular method for specifying fuzzy sets because of their smoothness and concise notation, and defined as

$$\mu_{A_i}(x) = \frac{1}{1 + \left| \frac{x - c_i}{a_i} \right|^{2b_i}} \tag{11}$$

Here $\{a_i, b_i, c_i\}$ is the parameter set of the membership function. The center and width of the membership function is varied by adjusting c_i and a_i . The parameter b_i is used to control the slopes at the crossover points. This layer forms the antecedents of the fuzzy rules (*IF* part).

Layer 2 is the rules layer. Every node in this layer is a fixed node and contains one fuzzy rule. The output is the product of all incoming signals and represents the firing strength of each rule.

$$O_i^2 = w_i = \mu_{A_i}(x)\mu_{B_i}(y), \quad i = 1,2 \tag{12}$$

Layer 3 is normalization layer. Every node in this layer is a fixed node and the i_{th} node calculates the ratio of the i_{th} rule's firing strength to the sum of all rules' firing strengths. Outputs of this layer are called normalized firing strengths computed as:

$$O_i^3 = \overline{w}_i = \frac{w_i}{w_1 + w_2} \quad i = 1,2 \tag{13}$$

Layer 4 is consequent layer. Every node in this layer is an adaptive node and computes the values of rule consequent (*THEN* part) as:

$$O_i^4 = \overline{w_i} f_i = \overline{w_i}(p_i x + q_i y + r_i) \tag{14}$$

Layer 5 is summation layer and consists of single fixed node which calculates the overall output as the summation of all incoming signals as:

$$O_i^5 = \sum_i \overline{w_i} f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \tag{15}$$

It can be observed that there are two adaptive layers in this ANFIS architecture, namely the first layer and the fourth layer. In the first layer, there are three modifiable parameters {*ai*, *bi*, *ci*}, which are related to the input membership functions. These parameters are the so-called premise parameters. In the fourth layer, there are also three modifiable parameters {*pi*, *qi*, *ri*}, pertaining to the first order polynomial. These parameters are the so-called consequent parameters [22-23].

b. Learning algorithm of ANFIS

The task of the learning algorithm for this architecture is to tune all the modifiable parameters, namely {*ai*, *bi*, *ci*} and {*pi*, *qi*, *ri*}, to make the ANFIS output match the training data. When the premise parameters *ai*, *bi* and *ci* of the membership function are fixed, the output of the ANFIS model can be written as:

$$f = \frac{w_1}{w_1 + w_2} f_1 + \frac{w_2}{w_1 + w_2} f_2 \tag{16}$$

Substituting Eq. (4) into Eq. (7) yields:

$$f = \overline{w_1} f_1 + \overline{w_2} f_2 \tag{17}$$

Substituting the fuzzy if-then rules into Eq. (8), it becomes:

$$f = \overline{w_1}(p_1 x + q_1 y + r_1) + \overline{w_2}(p_2 x + q_2 y + r_2) \tag{18}$$

After rearrangement, the output can be expressed as:

$$f = (\overline{w_1} x) p_1 + (\overline{w_1} y) q_1 + (\overline{w_1}) r_1 + (\overline{w_2} x) p_2 + (\overline{w_2} y) q_2 + (\overline{w_2}) r_2 \tag{19}$$

Which is a linear combination of the modifiable consequent parameters *p*1, *q*1, *r*1, *p*2, *q*2 and *r*2. The least squares method can be used to identify the optimal values of these parameters easily. When the premise parameters are not fixed, the search space becomes larger and the convergence of the training becomes slower. A hybrid algorithm combining the least squares method and the gradient descent method is adopted to solve this problem. The hybrid algorithm is composed of a forward pass and a backward pass. The least squares method (forward pass) is used to optimize the consequent parameters with the premise parameters fixed. Once the optimal consequent parameters are found, the backward pass starts immediately. The gradient descent method (backward pass) is used to adjust optimally the premise parameters corresponding to the fuzzy sets in the input domain. The output of the ANFIS is calculated by employing the consequent parameters found in the forward pass. The output error is used to adapt the premise parameters by means of a standard backpropagation algorithm. It has been proven that this hybrid algorithm is highly efficient in training the ANFIS [21-23].

5. OBJECTIVE FUNCTION DEFINITION

In the design of a PID controller, the performance criterion or objective function is first defined based on some desired specifications and constraints under input testing signal. Some typical output specifications in the time domain are overshoot, rise time, settling time, and steady-state error. In general, three kinds of performance criteria, the integrated absolute error (IAE), the integral of squared-error (ISE), and the integrated of time weighted- squared-error (ITSE) are usually considered in the control design under step input testing, as they can be evaluated analytically in the frequency domain. It is worthy to notice that using different performance indices probably makes different solutions for PID controllers. The three integral performance criteria in the frequency domain have their own advantages and disadvantages. For example, a disadvantage of the IAE and ISE criteria is that their minimization can result in a response with relatively small overshoot but a long settling time. Although the ITSE performance criterion can overcome the disadvantage of the ISE criterion, the derivation processes of the analytical formula are complex and time-consuming [4]. The IAE, ISE, ITAE and ITSE performance criteria formulas are as follows:

Integral of absolute error (IAE): $J = \int |\Delta e| dt$ (20)

Integral of squared error (ISE): $J = \int (\Delta e)^2 \, dt$ (21)

Integral of time weighted absolute error (ITAE): $J = \int t|\Delta e| dt$ (22)

Integral of time weighted squared error (ITSE): $J = \int t(\Delta e)^2 \, dt$ (23)

Each performance index has its own advantages and disadvantages and will result in different system performance. The ISE is a typical performance criterion used in a number of control applications. It tends to penalize all errors with respect to the given weighting factors. The ITAE is also widely used in control applications and includes the time, t, in order to penalize the settling time of the controlled system. The minimization of ISE and IAE can result in a response with small overshoot but longer settling time and is seen as a disadvantage. Hence selection of a performance index should be based on the desired performance aspects for the overall system. The fitness function (objective function) for SNR-PSO is defined as:

$$CostFunction = (O_{sh} \times 1000)^2 + t_{st}{}^2 + \frac{0.001}{(\max - dv)^2} \tag{24}$$

In this paper, the desired performance aspects are to minimization of cost function with the help of any optimization technique corresponds to minimum overshoot (O_{sh}), minimum settling time (t_{st}) and $\max - dv$. Therefore, it becomes an unconstrained optimization problem to find a set of decision variables by minimizing the objective function. Maximum population size = 50, maximum allowed iteration cycles = 100, $C_1 = C_2$ = 2.05. The parameters of the block diagram are chosen as K_A = 10, $K_e = K_s$ =1.0, τ_a = 0.1 s, τ_e = 0.4 s, τ_S = 0.01 s, τ_g = 1.0 s. Only K_G and τg are load dependent.

6. METHODOLOGY OF THE PROPOSED ALGORITHM

In this study, we propose to use a hybrid intelligent system called ANFIS for design the optimal proportional-integral-derivative (PID) controller of an automatic voltage regulator (AVR) system. We combine the ability of a neural network (NN) to learn with fuzzy logic (FL) to reason in order to form a hybrid intelligent system called ANFIS. The goal of ANFIS is to find a model or mapping that will correctly associate the inputs with the target. The fuzzy inference system (FIS) is a knowledge representation where each fuzzy rule describes a local behavior of the system. The network structure that implements FIS and employs hybrid-learning rules to train is called ANFIS. The concept of the proposed technique is based on recognizing the patterns of the sensitivities of some indices to prescribed credible events since every event could have a signature on the patterns of these indices. The following independent variables are defined with respect to this target location. Table 2 has been computed to illustrate the comparative performance characteristics of SNRPSO PID controller. K_G has been varied from 0.7 to 1.0 in steps of 0.1. τ_g has been varied from 1.0 to 2.0 in steps of 0.2. Thus, Table 2 includes 24 different sets of input conditions of AVR system. Each input corresponds to nominal optimal PID gain as output. From Table 2 it may be noted that SNRPSO based optimization technique offers (a) lesser overshoot of change in terminal voltage (O_{sh}), (b) lesser settling time of change in terminal voltage (t_{st}), and (c) more maximum derivative of change in terminal voltage ($\max - dv$). The behavioral model of the proposed technique can be represented within the fuzzy inference system as follows:

$$Data_{in} = \begin{bmatrix} [K_g, T_g]^1 \\ [K_g, T_g]^2 \\ \dots\dots\dots\dots \\ \dots\dots\dots\dots \\ \dots\dots\dots\dots \\ [K_g, T_g]^M \end{bmatrix}_{M*1} \qquad Data_{out} = \begin{bmatrix} Output \;\; ([K_g, T_g]^1) \\ Output \;\; ([K_g, T_g]^2) \\ \dots\dots\dots\dots\dots\dots \\ \dots\dots\dots\dots\dots\dots \\ \dots\dots\dots\dots\dots\dots \\ Output \;\; ([K_g, T_g]^M) \end{bmatrix}_{M*1}$$

$$[K_p, K_i, K_d] = [Data_{out}] \qquad \mathrm{i} = 1,2,...,\mathrm{M}$$

$$\mathrm{S} = [Data_{in} \; Data_{out}]$$

That:

K_g : Under the i^{th} event;

T_g : Under the i^{th} event;

M : The Number of performed tests

In this proposed methodology, extensive prescribed events are simulated off-line in order to capture the essential features of the system behavior that produce the ANFIS. These prescribed events are defined in the event database from Table.2 (24 different sets) which the network simulator executes the required events.

Table 2. Anfis rule base table, optimized PID gains and transient response parameters

K_G	τg	Type of controller	K_p	K_i	K_d	O_{sh}	t_{st}	$\max - dv$	MF
0.7	1	SNR-PSO	0.7762	0.5274	0.2474	9.7953e-8	0.5341	0.1103	0.3675
	1.2	SNR-PSO	0.8430	0.5102	0.2750	1.9037e-6	0.5922	0.1131	0.4292
	1.4	SNR-PSO	0.8629	0.4680	0.2893	8.7316e-8	0.6802	0.1045	0.5544
	1.6	SNR-PSO	1.0964	0.5284	0.4003	1.8994e-8	0.5312	0.1205	0.3511
	1.8	SNR-PSO	1.1862	0.5291	0.4439	2.5885e-7	0.5609	0.1187	0.3856
	2	SNR-PSO	1.2800	0.5184	0.4852	4.5202e-7	0.5532	0.1171	0.3790
0.8	1	SNR-PSO	0.6795	0.4616	0.2167	3.7172e-7	0.5333	0.1130	0.3628
	1.2	SNR-PSO	0.6736	0.4146	0.2698	2.3390e-7	0.6633	0.0960	0.5486
	1.4	SNR-PSO	0.8655	0.4619	0.3052	1.3507e-7	0.5353	0.1197	0.3562
	1.6	SNR-PSO	0.9560	0.4608	0.3496	6.9746e-7	0.5343	0.1200	0.3549
	1.8	SNR-PSO	1.0127	0.4490	0.3666	2.8920e-7	0.5616	0.1146	0.3909
	2	SNR-PSO	0.9181	0.3815	0.3089	1.5401e-8	0.7772	0.0934	0.7187
0.9	1	SNR-PSO	0.5701	0.3913	0.1776	9.1740e-9	0.5966	0.1096	0.4392
	1.2	SNR-PSO	0.6382	0.3887	0.2048	1.5431e-9	0.6217	0.1037	0.4795
	1.4	SNR-PSO	0.7699	0.4108	0.2716	6.6844e-7	0.5326	0.1191	0.3543
	1.6	SNR-PSO	0.8269	0.3986	0.3017	5.1328e-8	0.5688	0.1169	0.3967
	1.8	SNR-PSO	0.8953	0.3963	0.3265	2.2475e-10	0.5716	0.1144	0.4032
	2	SNR-PSO	0.7700	0.3202	0.2579	1.8179e-7	0.8434	0.0920	0.8295
1	1	SNR-PSO	0.4938	0.3278	0.1407	4.7599e-6	0.6330	0.1053	0.4931
	1.2	SNR-PSO	0.6147	0.3681	0.2066	8.4153e-8	0.5435	0.1123	0.3760
	1.4	SNR-PSO	0.6466	0.3239	0.2094	6.2972e-8	0.6050	0.1014	0.4633
	1.6	SNR-PSO	0.7342	0.3577	0.2579	8.7420e-8	0.5713	0.1125	0.4054
	1.8	SNR-PSO	0.8367	0.3668	0.3112	1.6682e-7	0.5330	0.1188	0.3549
	2	SNR-PSO	0.8829	0.3586	0.3308	2.2754e-7	0.5689	0.1152	0.3989

7. ARCHITECTURE OF THE PROPOSED ALGORITHM

The architecture of the proposed Intelligent-based for design the optimal PID controller of AVR system is shown in Figure 3. It is consists of three main modules, namely the input module, fuzzy inference system, and the output module. These modules are described as follows:

a. Input Module

The input to this module are K_G & τg .

b. Fuzzy Inference system (FIS)
This module is the fuzzy inference system software model of design the optimal PID controller of AVR system. This module has already been discussed in Section II.

c. Output Module
This is an output unit which include K_p , K_i and K_d .

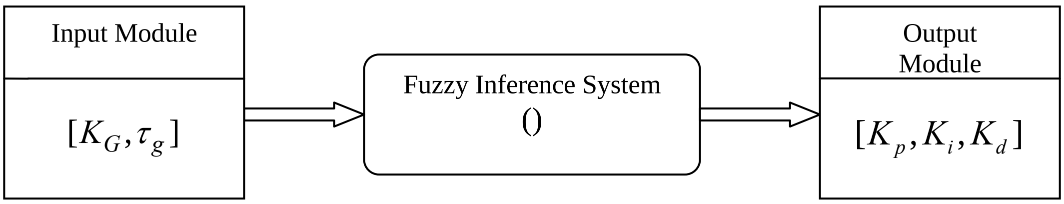

Figure 3. Architecture of the proposed intelligent-based for design the optimal PID controller of AVR system

This approach use and fed to the ANFIS for training and obtained the fuzzy membership function (MF) without need to determine of type and number of membership function. In this paper a fuzzy inference system models which takes K_G and τg as inputs and K_p , K_i and K_d as output. Figures 4-6 show the fuzzy membership function for K_p , K_i and K_d obtained only from dataset for all conditional. The result obtained to indicate that ANFIS is effective method for design an intelligent PID controller.

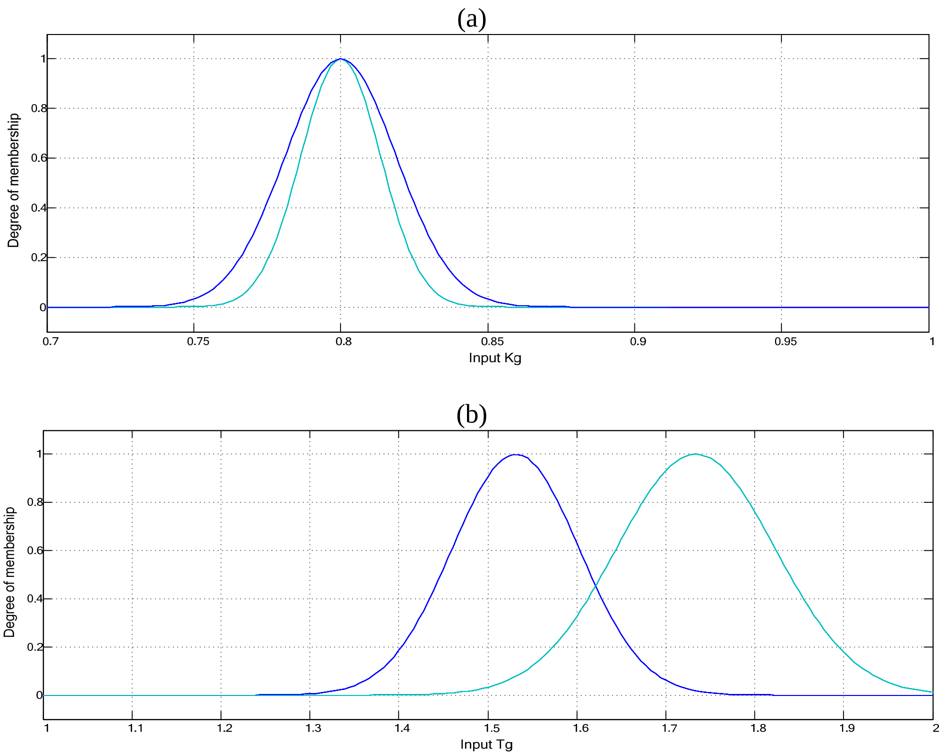

Figure 4. The fuzzy membership function obtained from ANFIS for K_p : (a) Input K_G ; (b) Input τg

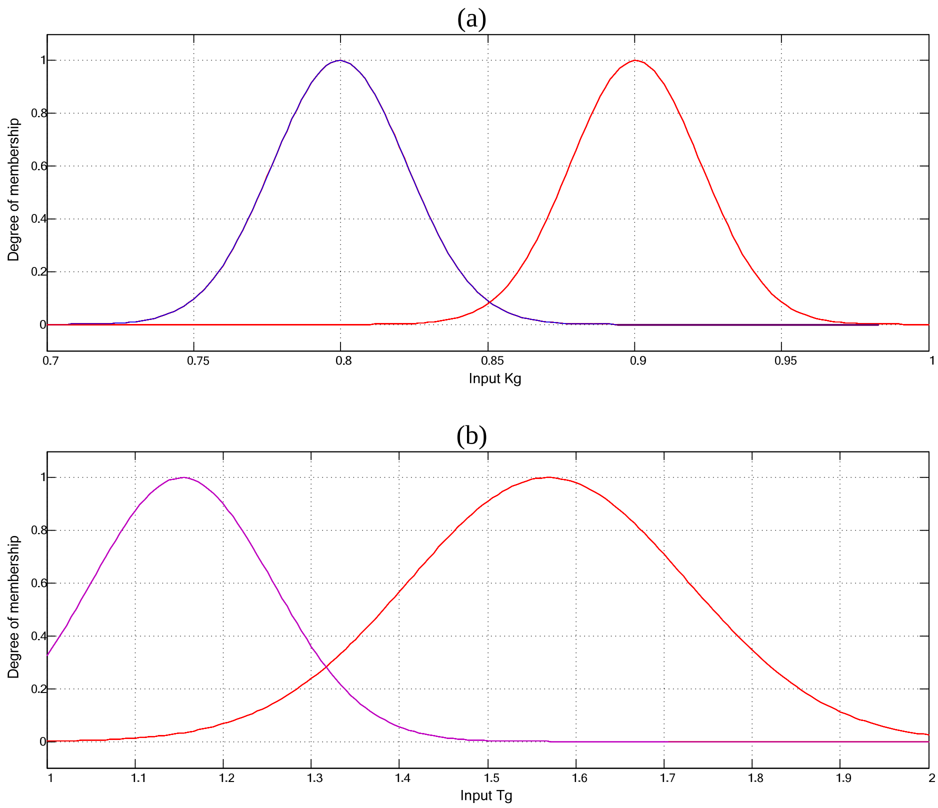

Figure 5. The fuzzy membership function obtained from ANFIS for K_i : (a) Input K_G ; (b) Input τ_g

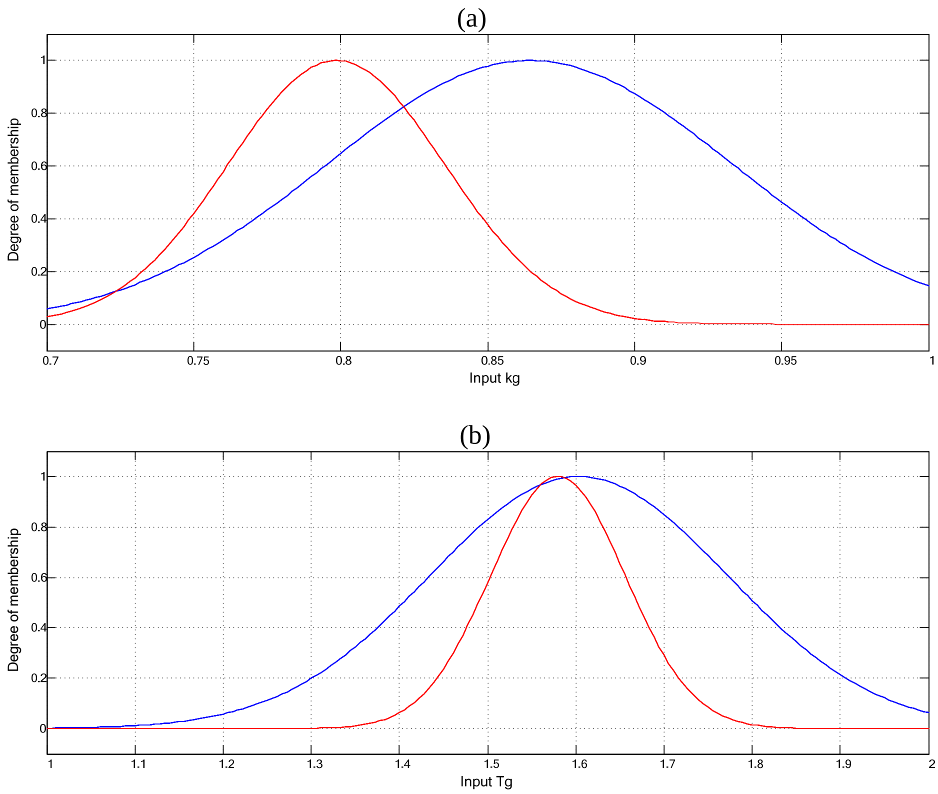

Figure 6. The fuzzy membership function obtained from ANFIS for K_d : (a) Input K_G ; (b) Input

8. SIMULATION RESULTS

In this section, the efficiency and effectiveness of the introduced SNR-PSO is validated. The block diagram of the AVR system with a SNRPSO PID controller and the system parameters values are shown in Figure 2. The three controller parameters K_p , K_i and K_d all range from 0.2 to 2. Step perturbation of 1p.u. of reference voltage has been applied to get the transient response of incremental change in terminal voltage in the present work. In order to emphasize the advantages of the proposed SNRPSO PID controller, we also compared with the CRPSO PID controller [14]. In this compared the characteristics of the two controllers had the same evaluation function and individual definition.

a. Case 1: K_G =0.7 , τg =1.6

In this case study for step response without controller, percent of overshoot *(M P* %) is 21.54% and settling time (*Ts)* is very major because we have steady state error. The terminal voltage step responses of the AVR without controller, SNRPSO PID controller and CRPSO PID controller are shown in Figure 7. In [14] the three controller parameters (K_p , K_i and K_d) for this case study are 1.0989,0.4480 and 0.3067 respectively. As can be seen, the intelligent PID controller could create very perfect step response of the AVR system, indicating that the intelligent PID controller is better than the CRPSO PID controller.

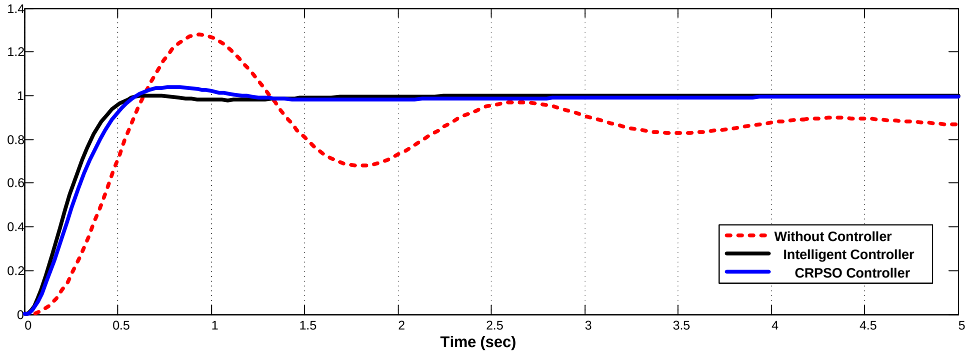

Figure 7. The step response of AVR system without controller and SNRPSO controller and CRPSO controller

b. Case 2: K_G =0.8 , τ_g =1

In this case study for step response without controller, percent of overshoot *(M P* %) is 38.59% and settling time (*Ts)* is very major because we have steady state error. The terminal voltage step responses of the AVR without controller, SNRPSO PID controller and CRPSO PID controller are shown in Figure 8. In [14] the three controller parameters (K_p , K_i and K_d) for this case study are 0.5884, 0.4005 and 0.2261 respectively. As can be seen, the intelligent PID controller could create very perfect step response of the AVR system, indicating that the intelligent PID controller is better than the CRPSO PID controller.

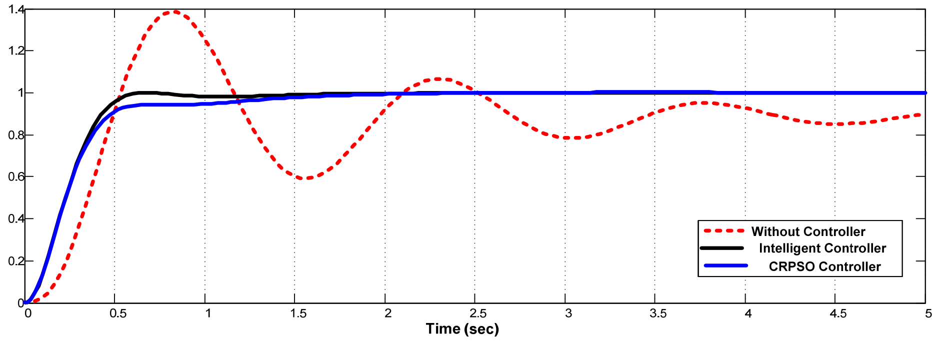

Figure 8. The step response of AVR system without controller and SNRPSO and CRPSO controllers

c. Case 3: K_G **=0.9 ,** τ_g **=1.6**

In this case study for step response without controller, percent of overshoot *(M P %)* is 33.39% settling time (*Ts)* is very major because we have steady state error. The terminal voltage step responses of the AVR without controller, SNRPSO PID controller and CRPSO PID controller are shown in Figure 9. In [14] the three controller parameters (K_p , K_i and K_d) for this case study are 0.6357, 0.3148 and 0.1963 respectively. As can be seen, the intelligent PID controller could create very perfect step response of the AVR system, indicating that the intelligent PID controller is better than the CRPSO PID controller.

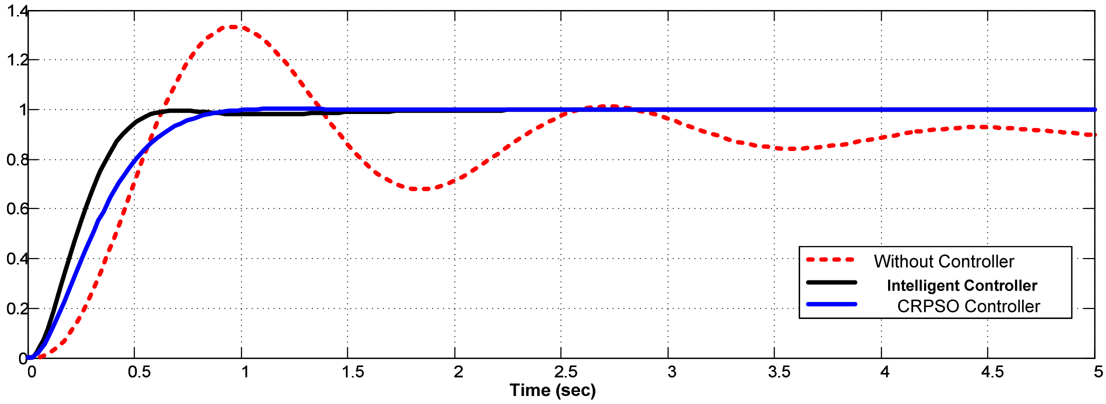

Figure 9. The step response of AVR system without controller and SNRPSO and CRPSO controllers

d. Case 4: K_G **=1 ,** τ_g **=1.8**

In this case study for step response without controller, percent of overshoot *(M P %)* is 35.43% and settling time (*Ts)* is very major because we have steady state error. The terminal voltage step responses of the AVR without controller, intelligent PID controller and CRPSO PID controller are shown in Figure 10. In [14] the three controller parameters (K_p , K_i and K_d) for this case study are 1.2730, 0.5952 and 0.5671 respectively. As can be seen, the intelligent PID controller could create very perfect step response of the AVR system, indicating that the intelligent PID controller is better than the CRPSO PID controller.

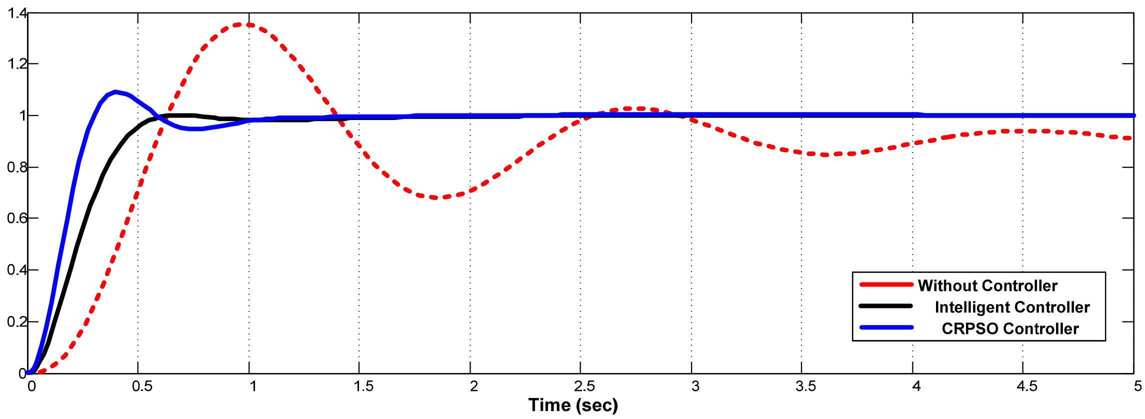

Figure 10. The step response of AVR system without controller and SNRPSO and CRPSO controllers

In this section we fed different K_G and τg as inputs that they were not in our train date for make a fuzzy inference system. When our inputs data fuzzy inference system change our PID coefficient controller change an intelligently and fed to system and always our system have best operation. In order to confirm the performance of the proposed method and verify the presented technique four case studies are performed.

e. Case 5: K_G =0.77 , τ_g =1.33

In this case study for step response without controller, percent of overshoot *(M P* %) is 30.23 % and settling time (*Ts)* is very major because we have steady state error. The output value intelligent controller based on an adaptive neuro fuzzy inference system when fed $K_G = 0.77$, $\tau_g = 1.33$ to that are K_p =0.8721, K_i =0.4509 and K_d =0.2969 respectively. The terminal voltage step responses of the AVR without controller and intelligent PID controller are shown in Figure 11. As can be seen, the intelligent PID controller could create very perfect step response of the AVR system.

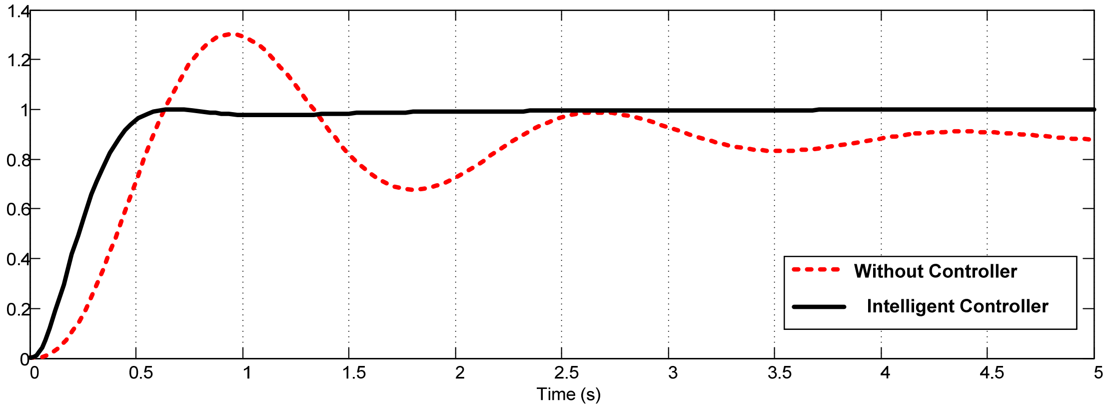

Figure 11. The step response of AVR system without controller and intelligent controller

f. Case 6: K_G =0.87 , τ_g =1.89

In this case study for step response without controller, percent of overshoot *(M P* %) is 27.56 % and settling time (*Ts)* is very major because we have steady state error. The output value intelligent controller based on an adaptive neuro fuzzy inference system when fed $K_G = 0.87$, $\tau_g = 1.89$ to that are K_p =0.8389, K_i = 0.4021 and K_d =0.3145 respectively. The terminal voltage step responses of the AVR without controller and intelligent PID controller are shown in Figure 12. As can be seen, the intelligent PID controller could create very perfect step response of the AVR system.

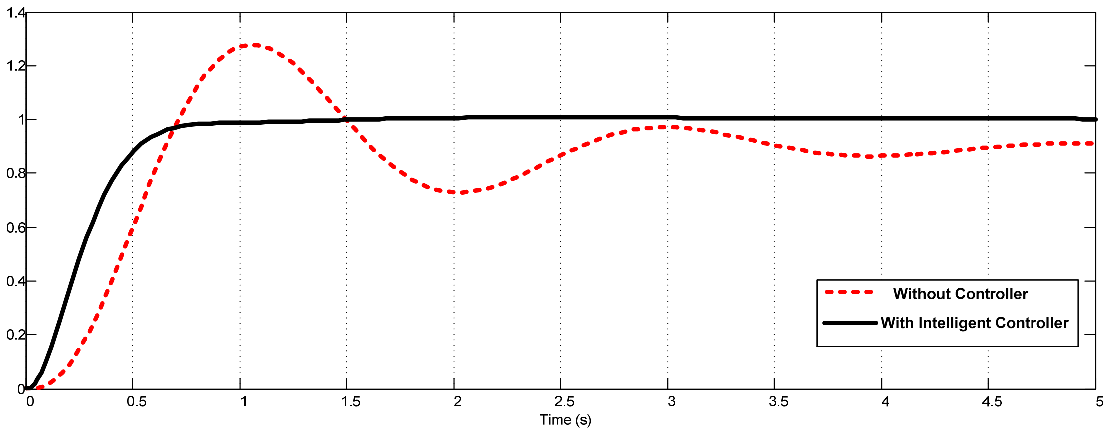

Figure 12. The step response of AVR system without controller and SNRPSO controller

g. Case 7: Kg=0.95 , τ_g =1.67

In this case study for step response without controller, percent of overshoot *(M P* %) is 34.85 % and settling time (*Ts)* is very major because we have steady state error. The output value intelligent controller based on an adaptive neuro fuzzy inference system when fed K_G *=0.95,* τ_g *=1.67* to that are K_p =0.7934, K_i = 0.3729and K_d =0.2928 respectively. The terminal voltage step responses of the AVR without controller and SNRPSO PID controller are shown in Figure 13. As can be seen, the SNRPSO PID controller could create very perfect step response of the AVR system.

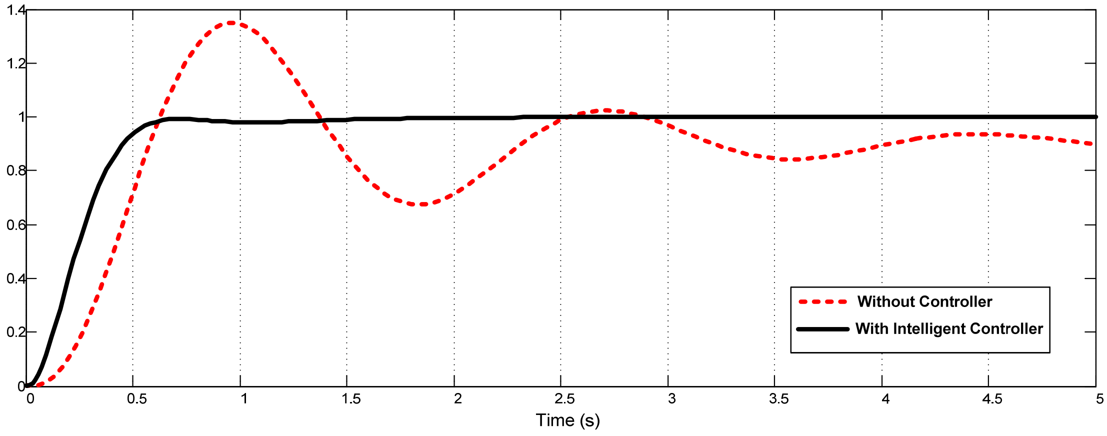

Figure 13. The step response of AVR system without controller and SNRPSO controller

h. Case 8: K_G =0.72 , τ_g =1.42

In this case study for step response without controller, percent of overshoot *(M P* %) 25.52 % and settling time (*Ts)* is very major because we have steady state error. The output value intelligent controller based on an adaptive neuro fuzzy inference system when fed K_G *=0.72,* τ_g *=1.42* to that are K_p =0.8787, K_i = 0.4733 and K_d =0.2976 respectively. The terminal voltage step responses of the AVR without controller and presented intelligent controller are shown in Figure 14. As can be seen, the SNRPSO PID controller could create very perfect step response of the AVR system.

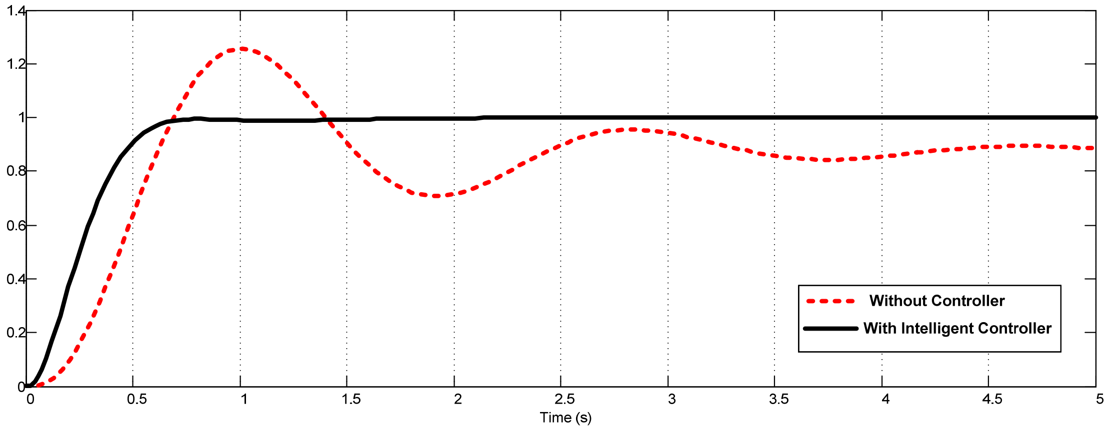

Figure 14. The step response of AVR system without controller and SNRPSO controller

9. CONCLUSION

The quality of the power supply is determined by the constancy of frequency and voltage. Minimum frequency deviation and good terminal voltage response are the characteristics of a reliable power supply. The conventional controllers used for this problem have large settling time, overshoot and oscillations. Hence, when evolutionary algorithms are applied to control system problems, their typical characteristics show a faster and smoother response. The proposed SNR-PSO PID controller provides a satisfactory stability between frequency overshoot and transient oscillations with zero steady state error. By using this algorithm the speed of convergence and accuracy can be increased and the system can be used for many real time applications. Superior performance, robustness, and efficiency of the proposed method have been proved through extensive simulation results. The results show that this approach is robustness and suitable for optimizing various control problems including adaptive control system with large scale dimensions.

REFERENCES

[1] M Zamani, Masoud Karimi-Ghartemani, Nasser Sadati, Mostafa Parniani. “Design of a fractional order PID controller for an AVR using particle swarm optimization”. *Control Engineering Practice*. 2009; 17(12): 1380–1387.

[2] Indranil Pan, Saptarshi Das. “Chaotic multi-objective optimization based design of fractional order $PI^{\lambda}D^{\mu}$ controller in AVR system”. *International Journal of Electrical Power & Energy Systems*. 2012; 43(1): 393–407.

[3] Hui Zhu, Lixiang Li, Ying Zhao, Yu Guo, Yixian Yang. “*CAS algorithm-based optimum design of PID controller in AVR system*”. Chaos, Solitons & Fractals. 2009; 42(2): 792–800.

[4] Dong Hwa Kim. “Hybrid GA–BF based intelligent PID controller tuning for AVR system”. *Applied Soft Computing*. 2011; 11(1): 11–22.

[5] Fabrizio Padula, Antonio Visioli. “Tuning rules for optimal PID and fractional-order PID controllers”. *Journal of Process Control*. 2011; 21(1): 69–81.

[6] Leandro dos Santos Coelho. “*Tuning of PID controller for an automatic regulator voltage system using chaotic optimization approach*”. Chaos, Solitons & Fractals. 2009; 39(4): 1504–1514 .

[7] S Panda, BK Sahu, PK Mohanty. “Design and performance analysis of PID controller for an automatic voltage regulator system using simplified particle swarm optimization”. *Journal of the Franklin Institute*. 2012; 349(8): 2609–2625.

[8] Saptarshi Das, Indranil Pan, Shantanu Das, Amitava Gupta. “A novel fractional order fuzzy PID controller and its optimal time domain tuning based on integral performance indices”. *Engineering Applications of Artificial Intelligence*. 2012; 25(2): 430–442.

[9] H Zhu, L Li, and Y Zhao. “CAS algorithm-based optimum design of PID controller in AVR system”. *Chaos, Solitons and Fractals*. 2009; 42: 792–800.

[10] Devaraj D, Selvabala B. “Real-coded genetic algorithm and fuzzy logic approach for real-time tuning of proportional-integral - derivative controller in automatic voltage regulator system”. Generation, Transmission & Distribution, *IET*. 3(7): 641 – 649

[11] Hasanien HM. “Design Optimization of PID Controller in Automatic Voltage Regulator System Using Taguchi Combined Genetic Algorithm Method”. *Systems Journal, IEEE*. 2012; PP(99): 1,

[12] ZL Gaing. “A Particle Swarm Optimization Approach for Optimum Design of PID Controller in AVR System”. *IEEE Transactions On Energy Conver*sion. 2004; 19(2).

[13] V Mukherjee, and SP Ghoshal. “Comparison of intelligent fuzzy based AGC coordinated PID controlled and PSS controlled AVR system”. *Electrical Power and Energy Systems*. 2007; 29: 679–689.

[14] V Mukherjee, and SP Ghoshal. “Intelligent particle swarm optimized fuzzy PID controller for AVR system”. *Electric Power Systems Research*. 2007; 77: 1689–1698.

[15] J Faiz, Gh Shahgholian, M Arezoomand. “ Analysis and Simulation of the AVR System and Parameters Variation Effects”. POWERENG 2007 in Setubal, Portugal. 2007: 450-453.

[16] MS Widyan. "On the effect of AVR gain on bifurcations of subsynchronous resonance in power systems”. *Electrical Power and Energy Systems*. 2010; 32: 656–663.

[17] Yi Tang, M Cui, C Hua, L Li, and Y Yang. “Optimum design of fractional order PID controller for AVR system using chaotic ant swarm”. *Expert Systems with Applications*. 2012; 39: 6887–6896.

[18] SP Ghoshal. “Optimizations of PID Gains by Particle Swarm Optimizations in Fuzzy based Automatic Generation Control”. *Electric Power Systems Research*. 2004; 70(3): 203-212.

[19] Ifeanyi E Madu, Christian N Madu. "Design optimization using signal-to-noise ratio". *Simulation Practice and Theory 7*. 1999: 349-372.

[20] Gupta MM, Rao DH. Neuro-control systems: theory and applications. Piscataway, NJ: *IEEE press*. 1994.

[21] Yen J, Langari R, Zadeh LA. Industrial applications of fuzzy logic and intelligent systems. New York, NY: *IEEE Press*. 1995.

[22] Jang J-SR. ANFIS adaptive-network-based fuzzy inference system. *IEEE Trans Syst Man Cyber*. 1993; 23(3): 665–85.

[23] Gupta MM, Rao DH. Newo-control systems: theory and applications. Piscataway, NJ: *IEEE Press*. 1994.