International Journal of Innovative Computing, Information and Control
Volume **10**, Number **6**, December **2014**
 ISSN 1349-4198
pp. 2263–2276

THE PID CONTROLLER BASED ON GENETIC ALGORITHM FOR VOLTAGE STABILITY OF THE SYNCHRONOUS MACHINE

Abdullah I. Al-Odienat

Electrical Engineering Department
Mutah University
Karak 61710, Jordan
Odienat@mutah.edu.jo

Received January 2014; revised May 2014

Abstract. *This paper presents a new approach for tuning the fuzzy PID controller based on Genetic Algorithms (GA) to control the terminal voltage during transient conditions of the fourth order Synchronous Generator (SG) Model. In the proposed method, the input and output scaling factors of the fuzzy PID controller are optimized using GA. An appropriate objective function is used to evaluate PID parameters. GA is used to control the error scaling factor, the error change and the control action. The results of the implementation have been compared with the results that were obtained without controller (AVR of SG only). The Integral Square of Error (ISE %) has been taken as a measure of the transient response enhancement. ISE % has been highly reduced which means, a remarkable enhancement in terminal voltage transient response. The best design achieved in this work has been compared to the well-known Ziegler-Nichols tuning method. It has been noted that the proposed tuning mechanism provides smoother system responses, decreases the oscillations and the settling time.*
Keywords: PID controller, Genetic Algorithms, Synchronous Generator, Voltage stability

1. Introduction.

In voltage stability analysis, the speed of the Automatic Voltage Regulator (AVR) of the synchronous generator is of great interest [1]. A considerable lag in control function is introduced due to the high inductance of the generator field winding. This is one of the major obstacles to be overcome in designing of the SG control system [2]. Therefore, the role of controller in enhancing the terminal voltage response becomes more critical and important. The major objectives in power system control design are to prevent the electric power system losing its synchronism after a large sudden fault and to achieve good post fault regulation of the generator terminal voltage [3].

Most of the power system controllers in operation today are PID controllers [4,5]. The tuning process of these controllers is an active area of research. The first significant tuning method was proposed by Ziegler and Nichols [4,6]. Analytical methods to obtain PID parameters based on simple transfer function models were developed by Rivera et al. and Gawthrop and Nomikos.

The analytical methods for tuning the controllers of the linear processes can be derived easily, however, in the real world; processes mostly are non-linear in nature and very complex. The analytical solution to the PID tuning problem of these systems is extremely difficult [7]. The electrical power systems represent a good example of non-linear systems.

In this work, the Fuzzy logic controller based on Genetic Algorithm is used to implement a PID controller. The GAs are global search methods that are based on natural population genetics. They maintain a set of candidate solutions to a given problem that are then left to evolve using genetic operator such as reproduction, crossover and mutation [8].

The Fuzzy logic controllers (FLCs) are rule-based controller. They include rules to direct the decision process and membership function to convert the linguistic variables into precise numeric values. The rule set is gleaned from a human expert knowledge, which is generally based on his experience [9]. In this paper, a Synchronous Machine AVR model is presented, then a Genetic Fuzzy PID Controller (GFPID) Structure is proposed. Next, the result of the proposed approach is addressed and demonstrated.

2. **Linearized Model of an AVR System with a PID Controller.** The model in Figure 1 provides an AVR system compensated with a PID controller.

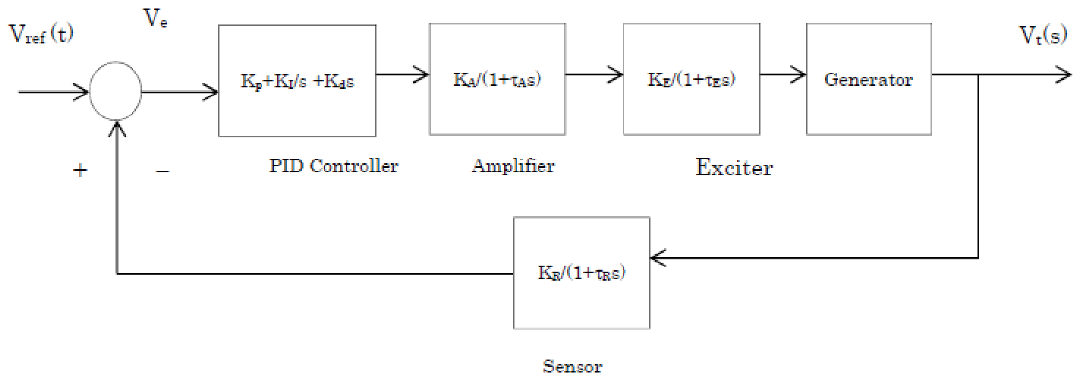

FIGURE 1. Block diagram of an AVR system with a PID controller

In general, the function of an AVR is to hold magnitude of the terminal voltage of a synchronous generator within a specified range. The simple model for the AVR system shown in Figure 1 consists of four main components, namely, an amplifier, an exciter, a generator, and a sensor. This model is linearized taking into account the major time constant and ignoring the saturation or other nonlinearities. A reasonabl derivation of the transfer function of these components may be represented as follows [4].

The amplifier model is represented by a gain K_A and a time constant τ_A, the transfer function is:

$$V_R(s)/V_e(s) = K_A/(1 + s\tau_A) \tag{1}$$

where V_{ref} is the reference voltage, V_S is the output voltage of the sensor, V_R is the output voltage of the AVR and V_e is the input voltage to the controller.

Typical values of K_A are in the range of 10 to 400. The amplifier time constant τ_A is very small and ranging from 0.02 to 0.1 s.

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

$$V_F(s)/V_R(s) = K_E/(1 + s\tau_E) \tag{2}$$

where V_F is the field voltage. Typical values of K_E are in the range of 10 to 400. The range of the time constant τ_E is from 0.5 to 1 s. In the linearized model, the transfer function relating the generator terminal voltage to its field voltage can be represented as $K_G/(1 + \tau_G s)$. Hence, the transfer function for the simplest first order SG model is:

$$V_t(s)/V_F(s) = K_G/(1 + \tau_G s) \tag{3}$$

While the fourth order SG model can be represented as:

$$\begin{aligned} V_t(s)/V_F(s) = K_G&(1 + s\tau_{z1})(1 + s\tau_{z2}) \\ &(1 + s\tau_{z3})(1 + s\tau_{z4})/(1 + s\tau_{p1})(1 + s\tau_{p2}) \\ &(1 + s\tau_{p3})(1 + s\tau_{p4}) \end{aligned} \tag{4}$$

The model constants are load dependent and may vary between 0.7 to 1.0 for K_G, and between 1.0 and 2.0 s for τ_G for full load to no load, respectively.

The sensor is represented by a simple first-order transfer function as

$$V_S(s)/V_t(s) = K_R/(1 + s\tau_R) \tag{5}$$

K_R is very small, ranging from 0.001 to 0.06 s. Figure 2 illustrates a fourth order SG-AVR model without a controller.

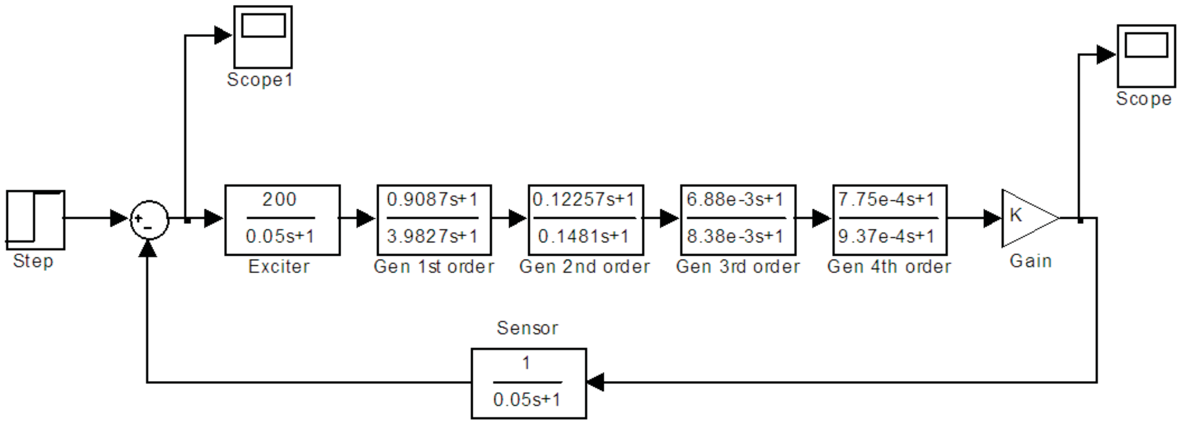

FIGURE 2. Fourth order SG-AVR model without controller

3. **Genetic Fuzzy PID Controller (GFPID) Structure.** Different designs are found in literature for fuzzy PID controllers [10-12]. Fuzzy PI controllers are more practical than fuzzy PD controllers for their effectiveness in removing steady state error. However, the performance of the fuzzy PI controllers is poor in transient response for higher order processes due to the internal integration operation. Thus, in practice, the Genetic Fuzzy PID controllers are more useful. It is intuitive and convenient to combine PI and PD actions together to form a GFPID controller [12]. One way of constructing a GFPID controller is achieved by summing the fuzzy PD controller output and its integrated part. A fuzzy PID controller with a single rule-base is adopted in this study. The closed-loop control structure considered in the study is shown in Figure 3. The output of the fuzzy PID controller is given by:

$$u = \delta U + \gamma \int U dt \tag{6}$$

where U is the output of the GFC, δ and γ are constants.

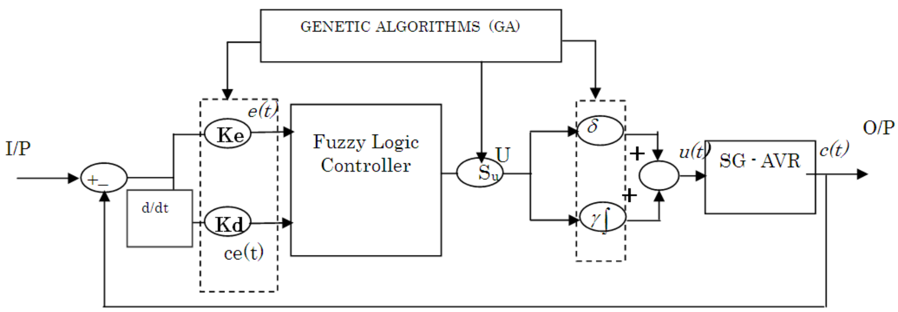

FIGURE 3. Closed loop control with Genetic Fuzzy PID controller (GFC)

The mathematical formula relating the input and the output variables of the Fuzzy Logic Controller is given by [12,13]:

$$U_f = H + PE + F\overset{\bullet}{E} \tag{7}$$

In this study, the output of the fuzzy controller is multiplied by the output scaling factor (S_u)

$$U = S_u * \left(H + PE + F\overset{\bullet}{E}\right) \tag{8}$$

where $E = K_e.e =$ the error signal (1st input to controller), $\overset{\bullet}{E} = K_d.e^{\bullet}$ is the change in error signal, $\overset{\bullet}{E}$ is the error change (2nd input to controller), H, P, and F are constants. Therefore, from (7) and (8) the controller output is obtained as follows [8]:

$$U = \delta HS_u + \gamma HtS_u + \delta K_e pS_u e + \gamma K_d HS_u e + \gamma K_e PS_u \int edt + \delta K_d FS_u e \tag{9}$$

While the continuous PID controller equation is

$$u(s) = \frac{K_d s^2 + K_p s + K_I}{s} \tag{10}$$

Thus, the equivalent control components of the Genetic Fuzzy PID Controller are obtained as follows:

$$K_p = \delta K_e pS_u + \gamma K_d FS_u, \quad K_I = \gamma K_e PS_u, \quad K_d = \delta K_d FS_u$$

where K_p, K_I, and K_d are proportional, integral, and derivative gains respectively. In this paper, δ, γ, K_e, K_d, and S_u are obtained by Genetic Algorithms.

3.1. **Genetic Algorithm configuration.** For a given optimization problem, the structure of the Genetic Algorithm requires specifying many problem dependent elements and parameters. These parameters include the *coding scheme* used for the encoding of the parameters to be optimized, the *resolution* and *range* of each parameter, the *population size*, the *crossover* and *mutation* probabilities, the *generation gap*, the type of *reinsertion*, and others [14,15].

GAs are search tools and optimization procedures that were devised on the principles of natural evolution and population genetics. The advantages of GA over other traditional optimization techniques may be summarized as follows:

- GA works on a coding of the parameters to be optimized.
- It searches the problem space using a population of trials representing possible solutions to the problem, not a single point.
- It uses a performance index assessment to guide the search in the problem space.
- GA uses probabilistic rules to make decisions.

GA includes various operations such as *reproduction*, *crossover*, and *mutation*. In reproduction process a new generation of population is formed by selecting the fittest individuals in the current population. Crossover operator is responsible for producing new offspring by selecting two strings and exchanging portions of their structures. The function of mutation is to alter the value of a random position in a string [16,17].

The Genetic Algorithm parameter and operators in the research are.

- Type of Selection: Roulette Wheel Selection
- Type of Crossover: Single point Crossover
- Type of mutation : Uniform mutation
- Probability of Crossover: 0.95
- Probability of Mutation: 0.01
- Max. No. of Generation: 1000

- Population size: 80

The main steps of GA algorithm are listed in the flowchart shown in Figure 4.

The change in the scaling gains at the inputs (S_e, S_c) and output (U) of the fuzzy controller can have significant impact on the performance of the resulting fuzzy control system, and hence they are convenient parameters for tuning. In this work, these gains are found using GAs. The chromosome representation for GFPID controller is shown in Figure 5.

The length (L) of chromosome is 5 genes. δ and γ genes are constants, K_e is the scaling factor for error, K_d is the scaling factor for change in error and S_u is the scaling factor for control action. The most important step in applying GA is to choose the appropriate objective function to evaluate the fitness of PID parameters.

The most common objective functions are the following [18]:

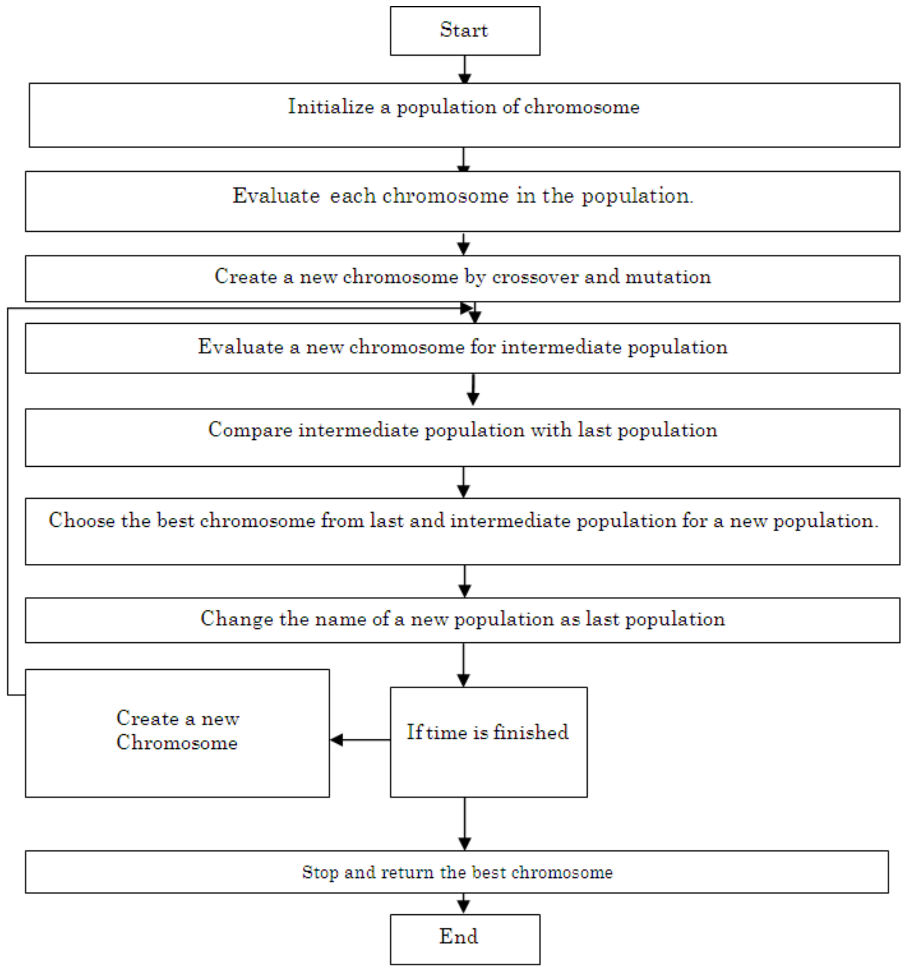

FIGURE 4. Flowchart for algorithm steps of GA

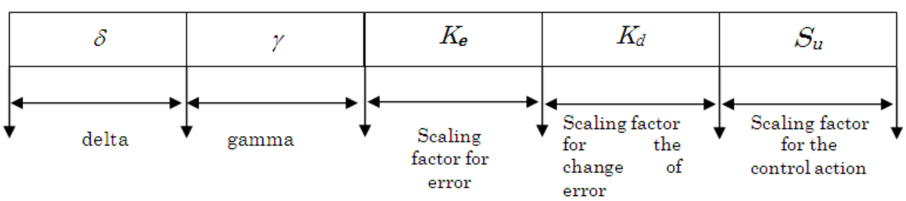

FIGURE 5. Chromosome representation for GFPID controller

Integration of error

$$S_1 = \int_0^\infty |e(t)|dt \tag{11}$$

Integration of error squared

$$S_2 = \int_0^\infty e^2(t)dt \tag{12}$$

Integration of time delay error squared

$$S_3 = \int_0^\infty te(t)dt \tag{13}$$

The objective is to have a response with small overshoot, fast rise time and short settling time. This can be achieved by selecting the optimal values of PID parameters. In this paper the following objective function is used:

$$S_4 = (1 + sh)(n_r t_r + n_l t_l) \tag{14}$$

where sh, t_r, and t_l are the over shoot, rise time and settling time, respectively. The constants n_r and n_l are to be selected by the user. Using S_4 in Equation (14), the fitness was found to be around 0.018 and after about 60 generations the fitness increased to an optimum value of about 0.01.

3.2. **Fuzzy logic controller.** In the implementation mode, the number of the fuzzy rules is 49. The membership functions used in both antecedent and consequent parts of the fuzzy rules are triangular and trapezoidal membership functions. In general the fuzzy logic controller consists of:

a) Preprocessing. The inputs are most often hard or *crisp*, measurement from some measuring equipment rather than linguistic [13]. In this work, the Normalization or scaling for fuzzy input is used as a preprocessing.

b) Fuzzification. The first block inside the controller is *fuzzification*, which converts each piece of input data to degrees of membership by a lookup in one or several membership functions. The fuzzification block thus matches the input data with the conditions of the rules to determine how well the condition of each rule matches that particular input instance. There is a degree of membership for each linguistic term that applies to that input variable [10].

c) Knowledge Base. It is comprised of two components, a Data Base (DB), containing the definition of scaling and fuzzy membership function of the fuzzy sets specifying the meaning of the linguistic terms, and a Rule Base (RB) consisted by collection of fuzzy rules. The Rule Base is a set of IF-THEN rules [12]. A single IF-THEN rule has a general form of IF (antecedent or premise) THEN (consequent or conciliation). If error is Neg and change in error is Neg then output is NB. If error is Neg and change in error is Zero then output is NM. The names = Zero, Neg., Pos. are labels of fuzzy sets as well as *NB*, *NM*, *PB* and *PM* (negative big, negative medium, positive big, and positive medium respectively) [19-21].

d) Inference Engine. It emulates the human ability to interpret and apply knowledge about how best to control the plant [12,22]. There are two types of Inference Engine: composition based inference and individual rule based inference.

e) Defuzzification. It translates the fuzzy control action into real control action. It is a process of producing a crisp output on the base of fuzzy input. Different defuzzifucation methods have been developed and applied such as (Center of gravity, Center of largest, and Center of sum) [12,23-25].

3.3. **The plant.** The final model for the synchronous generator AVR control system has been developed. The proposed controller (GFPIDC) combines the two powerful tools together in a sequential application; firstly applying Fuzzy PID controller to the plant under consideration, and then using the GAs for optimizing the values of the controller parameters to enhance the voltage stability under transient conditions. This controller must be inserted in the AVR voltage control system to achieve the required stability improvement.

4. **Results and Discussion.** The proposed GFPID controller is implemented here as an auxiliary enhancement with the SG-AVR control system. The simulation has been performed using MATLAB. The open loop transfer function of the plant has been converted into a discrete model using Zero Order Hold (ZOH) and the sampling time has been chosen based on Shannon sampling theorem. MATLAB function is used to perform the transformation from continuous to discrete system [26]. The data for the SG-AVR model are given in Tables 1 and 2. Figure 6 illustrates the 4th order SG-AVR with GFPID controller.

TABLE 1. Optimum time constants

Rotor Circuit	**Time Constants**	
	Poles	**Zeros**
F	3.9517	0.9087
J	0.1481	0.1257
K	0.00838	0.00688
L	0.000937	0.000775

TABLE 2. Values in per units of the constants required for excitation control system

V_{ref}	K_E	K_G	K_R	K_1	K_2	K_3	K_4	K_5	T_E	K_R	ΔV_L
1	200	1	1	0.2	1.5	1.4	-0.1	0.5	0.05	0.05	0.05

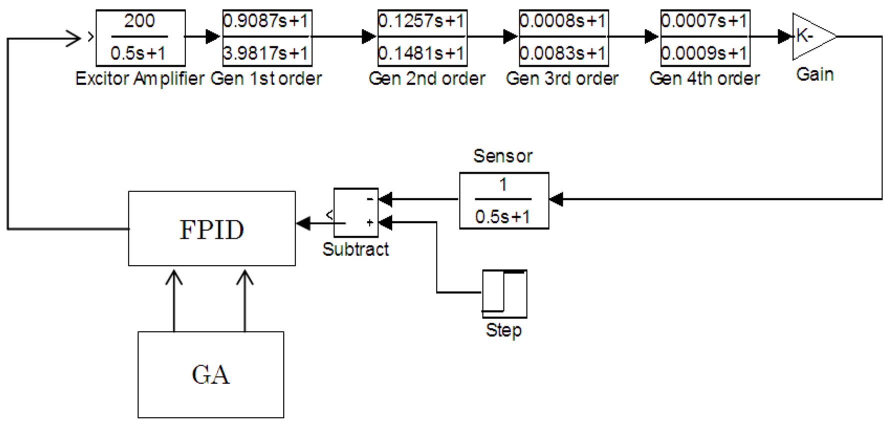

FIGURE 6. Fourth order SG-AVR with GFPIDC

4.1. **GFPIDC-1 scaling factors tuned genetically.** Figure 7 shows the Chromosome Representation for GFPID controller with genetic optimized scaling factors, where in this type of controller, the chromosome is represented by five Gens, namely, scaling factor for error (S_e), scaling factor for change in error (S_{ce}), scaling factor for control action (S_u), Alpha coefficient (δ) and Beta coefficient (γ). The obtained optimized values are then used in the final stage of tuning.

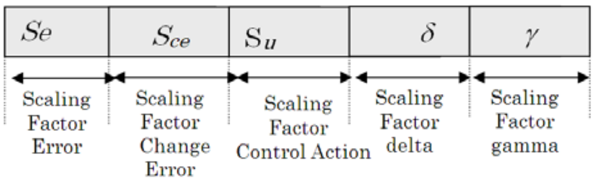

FIGURE 7. Chromosome representation for the GFPIDC with genetic optimized scaling factors

4.2. **GFPIDC-2 memberships functions tuned genetically.** This new method will use five values for the scaling gains obtained from the first step with tuning the membership functions for the input error, input deviation of error and the control action. Figure 8 illustrates the chromosome representation for this controller, where this chromosome consists of 49 genes as follows: Chromosome length (L) = 21 genes for error memberships + 21 genes for change in errors memberships + 7 genes for control actions memberships.

e1	e2	……	e21	ce1	ce2	…….	ce21	u1	u2	…….	u7

FIGURE 8. Chromosome representation for the GFPIDC where the memberships tuned genetically

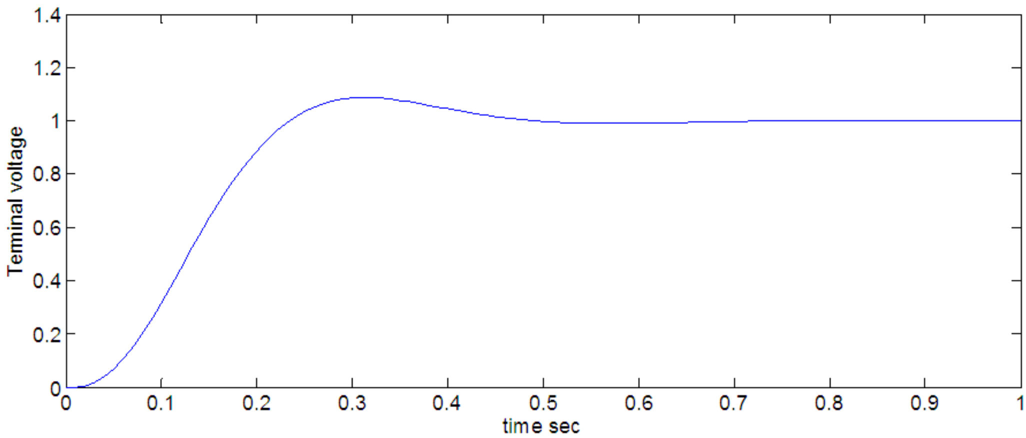

FIGURE 9. Terminal voltage step response of the 4th order SG model with GFPIDC, scaling factors tuned genetically

Figure 9 shows the terminal voltage response of the synchronous generator for step input, with the GFPIDC, where the five scaling factors had been tuned genetically. Figure 10 depicts the response by applying the same controller but the difference here is that; the memberships for the errors, derivative of errors and the control actions are tuned genetically, through applying the control system expert that was obtained from the first step by using the optimized values of the mentioned scaling factors.

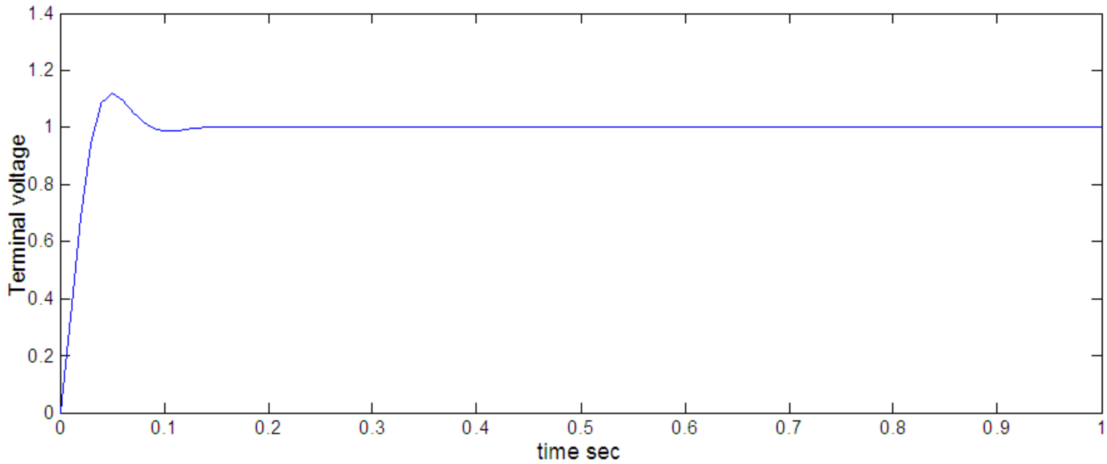

FIGURE 10. Terminal voltage step response of the 4th order SG model with GFPIDC, memberships tuned genetically

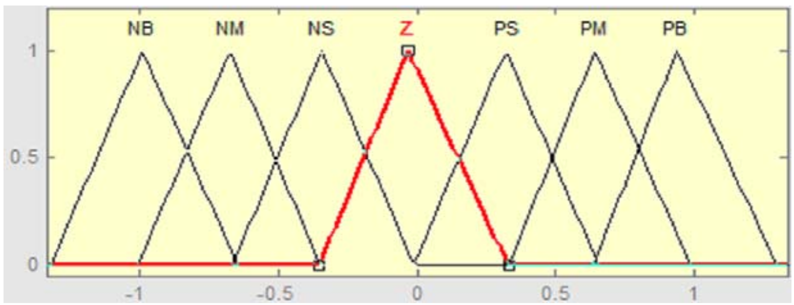

FIGURE 11. Error membership distribution before tuning

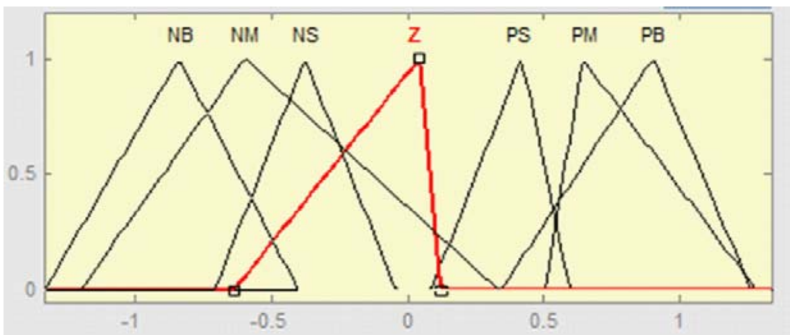

FIGURE 12. Error membership distribution after tuning

Figures 11 and 12 illustrate the resultant membership functions for error before and after tuning respectively, while Figures 13 and 14 show these memberships for the change of error both for the 4th order model.

Table 3 shows the performance indexes of the terminal voltage response for the various order models of SG with GFPIDC-1 (scaling factors tuned genetically), and GFPIDC-2 (membership functions tuned genetically) together with the cases without controller and with PID controller.

The percentage reduction in ISE % resulted from these controllers are calculated and demonstrated in Table 3 where

$$\text{ISE reduction \%} = \frac{\text{ISE without controller} - \text{ISE with controller}}{\text{ISE without controller}} \times 100\%$$

It is clear that performance index through the ISE is reduced for each case which means a good improvement in system stability as the enhancement percentage is proportional directly to the ISE % reduction.

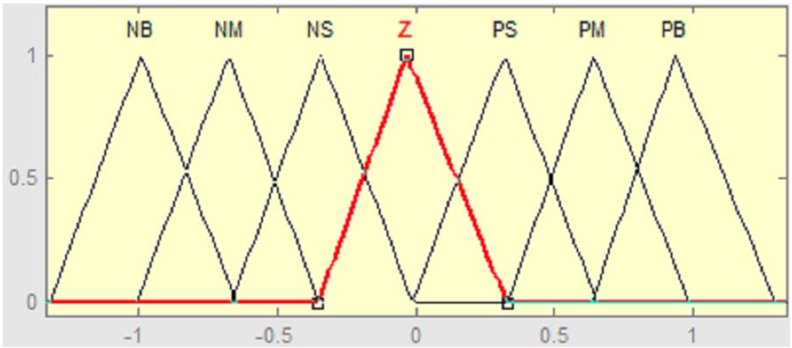

FIGURE 13. Error change membership before tuning

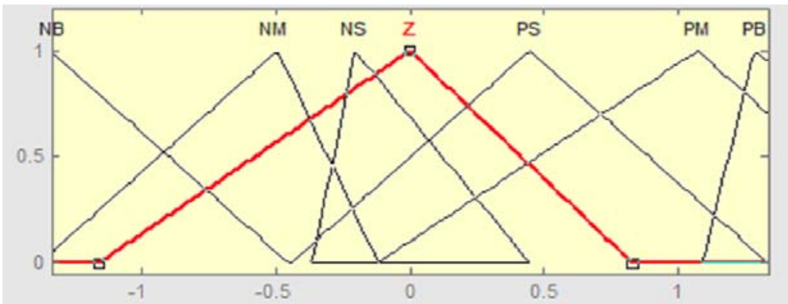

FIGURE 14. Error change membership after tuning

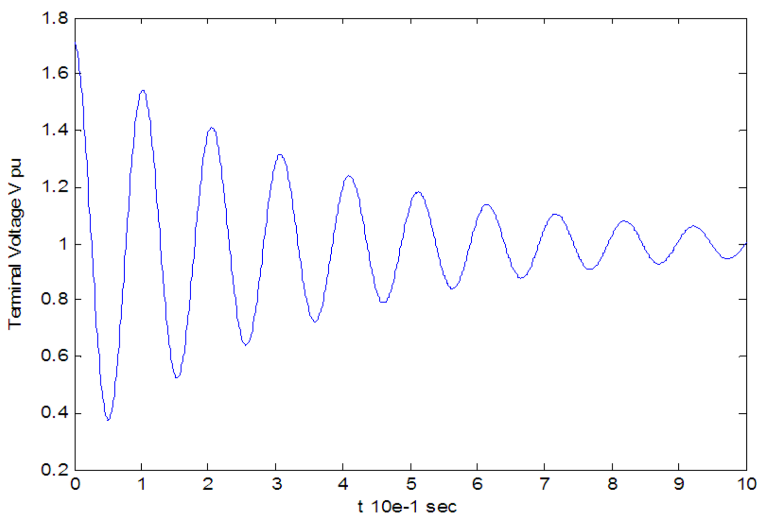

FIGURE 15. Voltage responses of the 4th order SG model: Response without controller

Figure 15 depicts the voltage responses of the 4th order SG model without controller, while the voltage responses for the same SG model with PID controller is shown in Figure 16. Figures 17-19 illustrate terminal voltage transient responses with GPID controller, GFC and the suggested technique GFPIDC-2, respectively. It is clear that the suggested method shows the best terminal voltage response.

The performance characteristics of the proposed controller has been compared to a well-known tuning method: Ziegler-Nichols (ZN) as shown in Figure 20.

ZN is a continuous cycling method for controller tuning. It was the first efficient method to tune a PID controller. The method is based on experiments executed on a control loop of a real system (or simulated system). According to ZN method the controller is firstly tuned as P-mode only at a gain makes the control system oscillatory. This gain is referred to as the ultimate gain K_u and the oscillation period as the ultimate period P_u. ZN approach determines the ultimate gain and period. The parameters of the PID controller are then determined from values of K_u and P_u based on ZN setting rules.

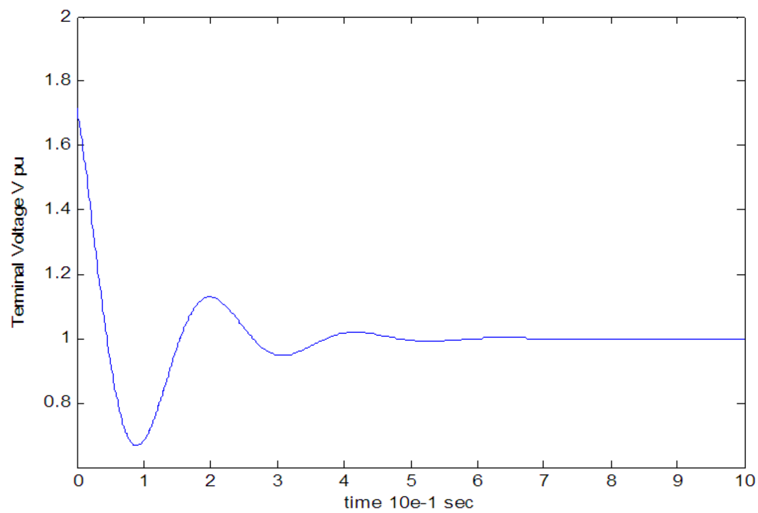

FIGURE 16. Voltage responses of the 4th order SG model: response with PID controller

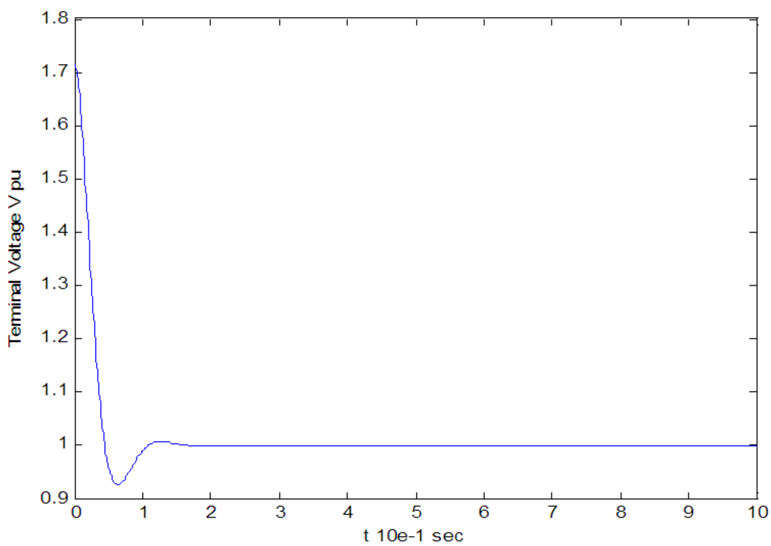

FIGURE 17. Voltage responses of the 4th order SG model: response with GPID controller

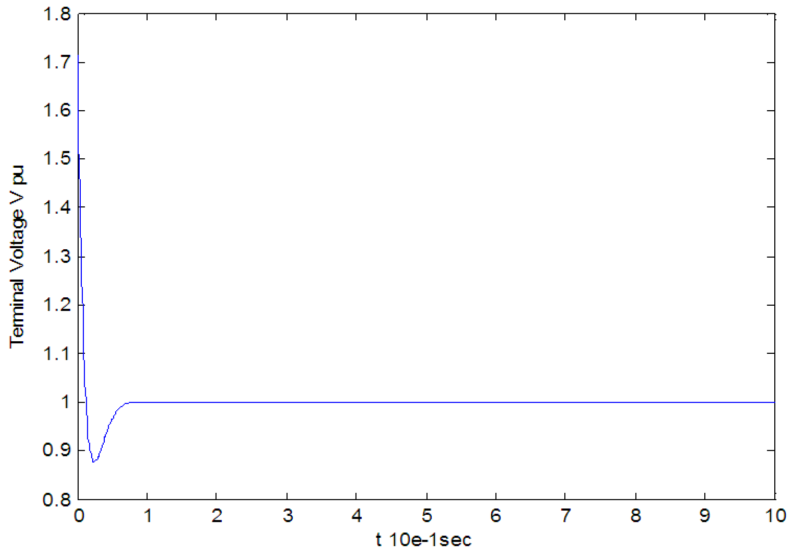

FIGURE 18. Voltage responses of the 4th order SG model: response with GFC

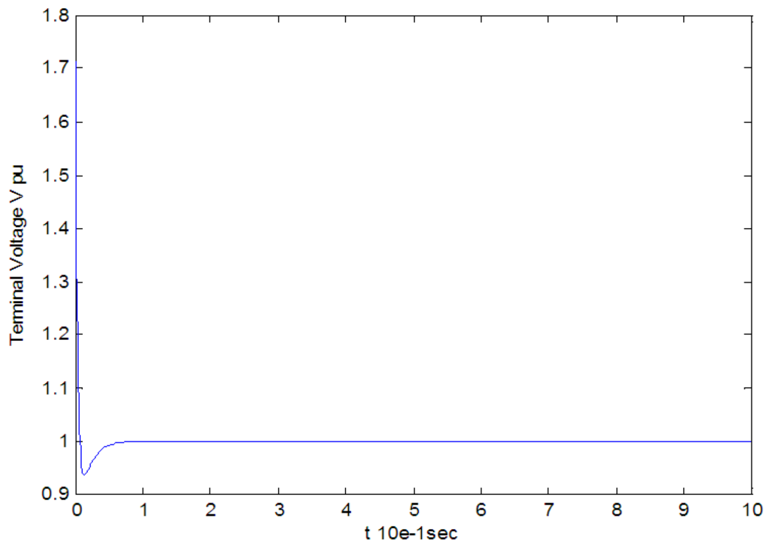

FIGURE 19. Voltage responses of the 4th order SG model: response with GFPIDC-2

Simulation results demonstrate the proposed tuning method has a better control performance compared with the conventional one (Ziegler-Nichols tuning method). Figure 20 shows that ZN method has a relatively very slow response (long settling time) as compared to GFPIDC-2.

5. **Conclusions.** The simulation results confirmed effectiveness of the GFPID controller to control the terminal voltage response. It provides improvements in the response, decrement in steady state error, decrement in rise time and decrement in settling time. Figures 11 to 14 show the effect of tuning the Fuzzy PID controller by GA where the change in membership functions for both the error, and the change of error are well demonstrated and compared before and after tuning. The optimum fitness is achieved after around 60 generations. The enhancements of the transient responses have obviously appeared in Figure 19 which illustrates the terminal voltages transient responses of the fourth order

TABLE 3. ISE reduction % of the terminal voltage response for all orders of SG model

Generator Order	Controller Type	ISE	ISE reduction%
First Order	Without	25.86	Reference
	GFC	1.057	95.91%
	GFPIDC-1	1.044	95.96%
	GFPIDC-2	1.002	96.12%
Second Order	Without	25.98	Reference
	GFC	1.663	93.6%
	GFPIDC-1	1.645	93.67%
	GFPIDC-2	1.049	95.96%
Third Order	Without	26.11	Reference
	GFC	2.038	92.2%
	GFPIDC-1	2.02	92.26%
	GFPIDC-2	1.075	95.88%
Fourth Order	Without	27.24	Reference
	GFC	2.046	92.49%
	GFPIDC-1	2.035	92.53%
	GFPIDC-2	1.076	96.05%

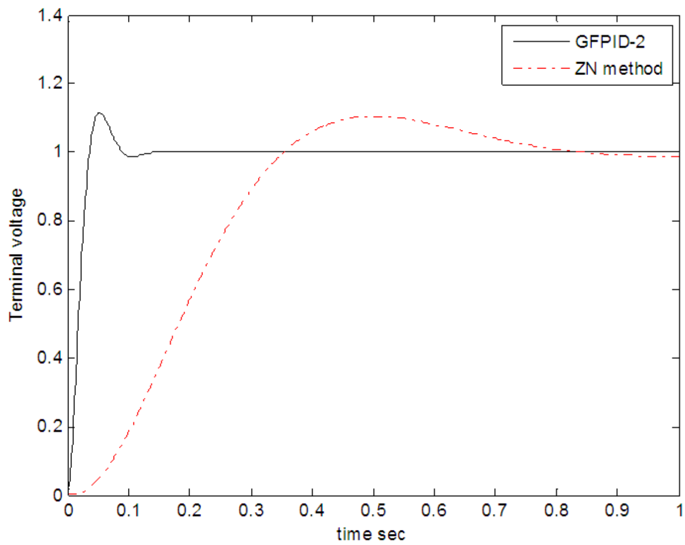

FIGURE 20. Step responses of the 4th order SG model: response with GFPIDC-2 compared to Ziegler-Nichols tuning method

SG model with GFPIDC-2, the performance indexes for this model is illustrated in Table 3. From the comparison, one can deduce that the new proposed intelligent controller, type GFPIDC-2 with the genetically tuned membership's functions gives the best transient response than the other types of controllers, where the ISE has been reduced from 27.24 without controller to 1.076 with the new controller or ISE % reduction of 96.05%. The comparison of proposed method with ZN tuning method shows clearly that settling time of proposed controller is relatively very short which confirms the efficacy of the proposed method.

REFERENCES

[1] Y. L. Karnavas and D. P. Papadopoulos, Excitation control of a synchronous machine using polynomial neural networks, *Journal of Electrical Engineering*, vol.55, no.7-8, pp.169-179, 2004.
[2] P. M. Anderson and A. A. Fouad, *Power System Control and Stability*, IEEE Press, USA, NY, 1994.
[3] Y. Wang, D. J. Hill, R. H. Middleton and L. Gao, Transient stability enhancement and voltage regulation of power systems, *IEEE Transactions on Power Systems*, vol.8, no.2, pp.620-627, 1993.
[4] K. Ogata, *Modern Control Engineering*, Prentice-Hall International, Inc. USA, 1970.
[5] B. C. Kuo, *Automatic Control Systems*, 5th Edition, Prentice Hall, Englewood Cliffs, 1987.
[6] K. J. Astrom, *Control System Design*, 2002.
[7] A. P. Loh, W. W. Tan and V. U. Vasnmu, Relay feedback of multivariable systems and its use for auto-tuning of multi-loop PI controllers, *Control '94 (IEE), Publication*, no.389, pp.1049-1054, 1994.
[8] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley Publishing Company, 1989.
[9] K. M. Passino and S. Yorkovich, *Fuzzy Control*, Addison Wesley Longman, Inc., Menlo Park, CA, USA, 1998.
[10] A. Ilyas, S. Jahan and M. Ayyub, Tuning of conventional PID and fuzzy logic controller using different defuzzification techniques, *International Journal of Scientific & Technology Research*, vol.2, no.1, 2013.
[11] P. Venugopal, A. Ganguly and P. Singh, Design of tuning methods of PID controller using fuzzy logic, *International Journal of Emerging Trends in Engineering and Development*, vol.5, no.3, 2013.
[12] W. Z. Qiao and M. Mizumoto, PID type fuzzy controller and parameters adaptive method, *Fuzzy Sets and Systems*, vol.78, pp.23-35, 1996.
[13] M. G. Uzelkaya, I. Eksin and E. Yesil, Self-tuning of PID-type fuzzy logic controller coefficients via relative rate observer, *Engineering Applications of Artificial Intelligence*, vol.16, pp.227-236, 2003.
[14] M. Dergisi, Optimal-tuning of PID controller gains using genetic algorithms, *Journal of Eng. Sciences*, vol.11, no.1, pp.131-135, 2005.
[15] Y. L. Abdel-Magid and M. M. Dawoud, Genetic algorithms applications in load frequency control, *Conference of Genetic Algorithms in Engineering Systems: Innovations and Applications*, pp.207-213, 1995.
[16] A. Mittal, A. Kapoor and T. K. Saxena, Genetic algorithm based incremental PID temperature controller for long dead time nonlinear bath, *Int. Journal of Engineering and Science Research*, vol.2, no.6, pp.449-467, 2012.
[17] W. M. Qi, W. Y. Cai, Q. L. Ji and Y. C. Cheng, A design of nonlinear adaptive PID controller based on genetic algorithm, *Proc. of Chinese Control Conference*, pp.175-178, 2006.
[18] P. Cominos and N. Munro, PID controllers: Recent tuning methods and design to specification, *Control Theory Appl.*, vol.149, no.1, pp.46-53, 2002.
[19] L. Davis, *Handbook of Genetic Algorithms*, Van Nostrand Reinholld, New York, 1991.
[20] L. X. Wang, *A Course in Fuzzy Systems and Control*, Prentice Hall, International, Inc., 1997.
[21] G. Cheng, *Genetic Algorithms & Engineering Design*, New York, Wiley, 1997.
[22] J. Zhang, J. Zhuang, H. Du and S. Wang, Self-organizing genetic algorithm based tuning of PID controllers, *Information Sciences*, vol.17, pp.1007-1018, 2009.
[23] F. Liu and M. J. Er, Design for auto-tuning PID controller based on genetic algorithms, *Proc. of IEEE ICIEA*, pp.1924-1928, 2009.
[24] Y. Yu, Y. Huang, M. Wang, B. Zeng and G. Zhong, Fuzzy neural PID controller and tuning its weight factors using genetic algorithm based on different location crossover, *Proc. of IEEE International Conference on Systems, Man and Cybernetics*, pp.3709-3713, 2004.
[25] G. Tan, B. Jiang and L. Yang, A novel immune genetic algorithm-based PID controller design and its application to CIP-intelligent leg, *The 3rd International Conference on Natural Computation*, pp.282-286, 2007.
[26] C. Lu and J. Zhang, Design and simulation of a fuzzy-PID composite parameters' controller with MATLAB, *Proc. of International Conference on Computer Design and Applications*, 2010.