مجلة العلوم الهندسية
FES Journal of Engineering Sciences
http://journal.oiu.edu.sd/ojs/index.php/fjes/index

Optimizing the PID Controller by Using the Genetic Algorithm

Muhammad A. H. Abdurrahman[1], Moamer A. O. Nourain[1], Mohammed O. A[1], Omer H. Alnour[1], and Manareldeen Ahmed[1]*.

[1] *Electrical and Electronic Engineering Department, Omdurman Islamic University, Omdurman, Sudan*
* *Corresponding author: Manareldeen Ahmed (manareldeen@oiu.edu.sd).*

Article history: Received 03 February 2022, Received in revised form 26 February 2022, Accepted 02 March 2022
Digital Object Identifier (doi): https://doi.org/10.52981/fjes.v%vi%i.1929

ABSTRACT This paper aims to develop a Proportional Integral Derivative (PID) controller for the Automatic Voltage Regulator (AVR) of the Synchronous Generator, Automatic Voltage Regulator is responsible for keeping the output voltage of the Generator constant and stable. The Automatic Voltage Regulator system can be represented by a fourth order Transfer Function. The parameters of the controller are evaluated by using the Genetic Algorithm tuning method, for this method the Genetic Algorithm Optimization toolbox in MATLAB is used. A PID controller and a PID Filter controller are developed in order to come up with the optimum controller design for the system. The performance of the Genetic Algorithm based controller has been compared with the classical Ziegler-Nichols based controller. The results show that the Genetic Algorithm based controller outperforms the Ziegler-Nichols based controller, the optimum controller design is achieved by the GA based PID filter controller.

Keywords*:* AVR system, Genetic Algorithm, PID controller, PID Optimization.

1. INTRODUCTION

Synchronous Generators require control systems in order to deliver a quality electrical current, these control systems are the Automatic Voltage Regulator (AVR), and the Automatic Load Frequency Control (LFC). The main role of an AVR is to maintain the terminal voltage of the synchronous generator at a precise value by controlling the Exciter using a proper controller [1-3]. this controller can be, for example, a PID controller. Since the stability of the generator output voltage would affect the stability of the power system, the designed voltage controller should be robust to disturbance.

In this paper we will design the PID controller of the AVR by using a stochastic search optimization technique called Genetic Algorithm (GA), GA uses the concept of evolution, which is based on the rule, "survival for fittest". The new classes of living things come into existence through the processes of Reproduction, Crossover, and Mutation among the previous class [4]. GA optimization toolbox in MATLAB will be used for finding the optimum tuning parameters of the PID controller [5].

Several publications have adopted the same approach to solve similar problems. [6] presents a new approach in PID tuning methodology for a processing plant using GA, was a good reference. [7] proposed a FPIDF controller for the AVR, it was a helpful resource; it introduced a new point of view in designing a controller for a system, by creating initial solution using the TLBO and blogging them into the FPIDF controller. [8] Introduced a Genetic Algorithm based PID controller for a speed control system of a DC motor, gave a taste of what GAs can do to a control system. [9] Introduces a GA based PID for a turbine speed control system. It used a

MATLAB M-file based approach for applying the GA. The main contributions of this paper are:

A. Optimizing the PID controller of the AVR system.
B. Improving the transient response and small signal response of the controller by adding a low-pass filter.
C. Studying the change of search boundaries of the Genetic Algorithm in order to find the global optima.
D. Comparing the optimized controller with the conventionally designed controller Ziegler-Nichols's controller.

2. PROBLEM STATEMENT

Synchronous Generator has two main control systems, LFC which maintains the angular speed of the prime mover that drives the synchronous generator by controlling a solenoid valve or a hydraulic servo motor. And the AVR which is responsible for regulating the terminal voltage by controlling the exciter that feeds the field windings of the generator with a help of a feedback from a sensor which basically measures the terminal voltage. Figure 1 shows the control systems of the synchronous generator.

When the load in the supply system changes, the Generator output voltage changes as well, thus the output voltage can be controlled by changing the field current in respect to the change in load. This is known as Excitation Control Method. The AVR senses the generator output voltage and adjusts the amplifier input, which in turn controls the exciter in such a way that maintains a constant terminal voltage.

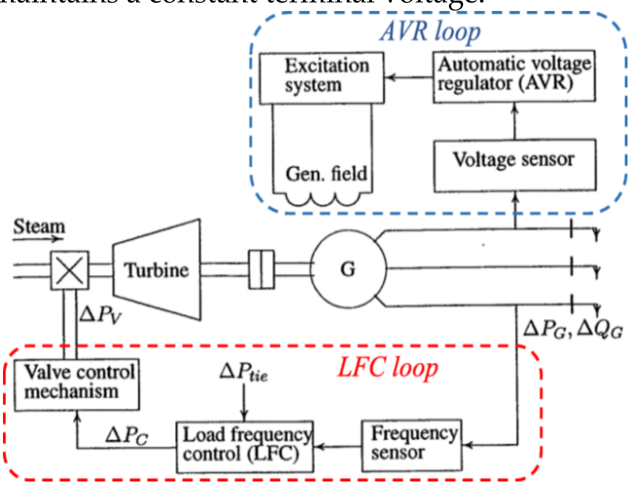

Figure 1: control systems of synchronous generator

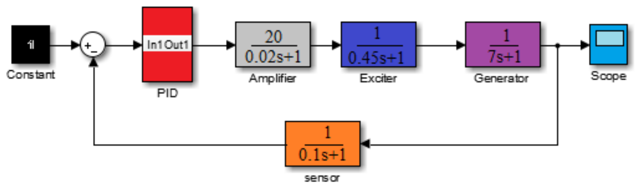

Figure 2: system model

Figure 2 shows the AVR control system. The comparator compares reference input and the actual voltage to generate a voltage error, which is fed to the amplifier. Due to the delay in the response of the amplifier, it can be described by a first order Transfer Function with a gain equal to 20 and a time constant equal to 20 milliseconds. The output voltage of the exciter in non-linear function of the field voltage because of saturation, which means there is no simple relationship Exciter input and output. A linearized model of exciter which considers the time constant and ignores the saturation and other nonlinearities is used in this paper with time constant of 450 milliseconds. The generated EMF of the generator is a function of the magnetization curve, it can be simulated by a first order Transfer Function with a time constant of 7 seconds. the sensor measures voltage through a potential transformer connected to a rectifier, its response is faster than the other components so the time constant of it is assumed to be 10 milliseconds.

3. ZIEGLER-NICHOLS TUNING METHOD

In this method the effect of integral and derivative terms of the controller are canceled, the proportional gain is increased to a critical value called Kcr, it is the gain value at which the system exhibits sustained oscillations. This gain value can be obtained by using Routh's array, or by trial and error in MATLAB. The second method is used in this paper, the obtained critical gain is 9.49999999, it gave the response shown in figure 3.

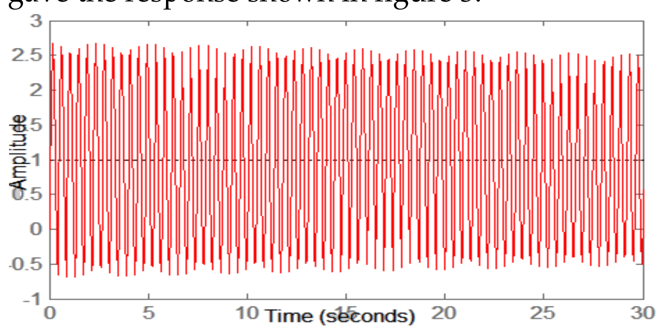

Figure 3: sustained oscillations

To proceed with Z-N method, we need to find the critical period (Pcr) corresponding to the evaluated critical gain, first we need to evaluate the angular speed by putting S equal to j in the characteristic equation and solving for

$$0.0063(j\omega)^4 + 0.3929(j\omega)^3 + 4.046(j\omega)^2 + 7.57j\omega + 76.2628 = 0$$

$$\omega = 3.73285$$

$$Pcr = \frac{2\pi}{\omega} = \frac{2\pi}{3.73285} = 0.719467$$

Table 1 presents the Z-N rules, Z-N rules are used to obtain the tuning parameters of the controller. According to table 1, the controller parameters can be given by table 2.

As shown in figure 4, the maximum overshoot of the Z-N based controller is 82.9% at 0.207 seconds, the settling time is 1.57 seconds, the rise time is 0.0629 seconds. the designed system has a fast response in terms of settling time and rise time, but the overshoot is unacceptable. in an online implementation, such a big overshoot can lead into serious damage.

TABLE I: Z-N TUNING RULES

Controller type	Kp	Ti	Td
P	0.5Kcr	∞	0
PI	0.45Kcr	$\frac{1}{1.2} * Pcr$	0
PID	0.6Kcr	0.5 Pcr	0.125Pcr

TABLEII: Z-N PARAMETERS

	Kp	Ti	Td
Gain	5.699999	0.3597336	0.0899334

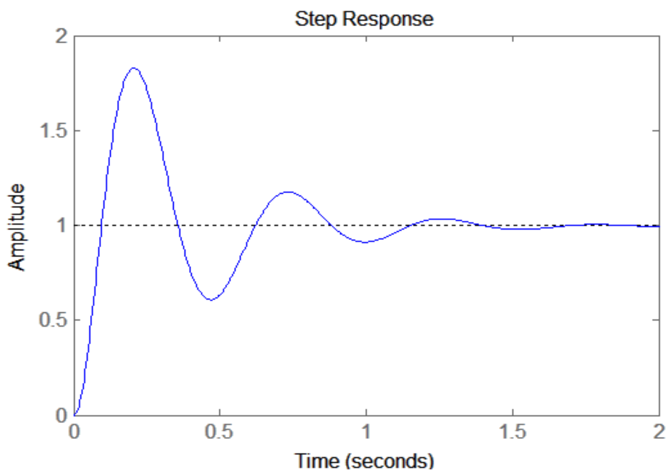

Figure 4: Z-N controller response

4. MATH GENETIC ALGORITHM METHOD

For the GA method, GA toolbox in MATLAB will be used. GA mimics the biological principle of reproduction, the best parents are selected for reproduction, the selection happens for the fittest parents by the so-called Selection. the newly generated off-springs gain their characteristics from their parents by what called Crossover, the last phase of the process is about giving a chance for Mutation to occur, but mutation must not exceed a certain value such that it destroys the new generation. Figure 5 describes the flowchart of the Genetic Algorithm method.

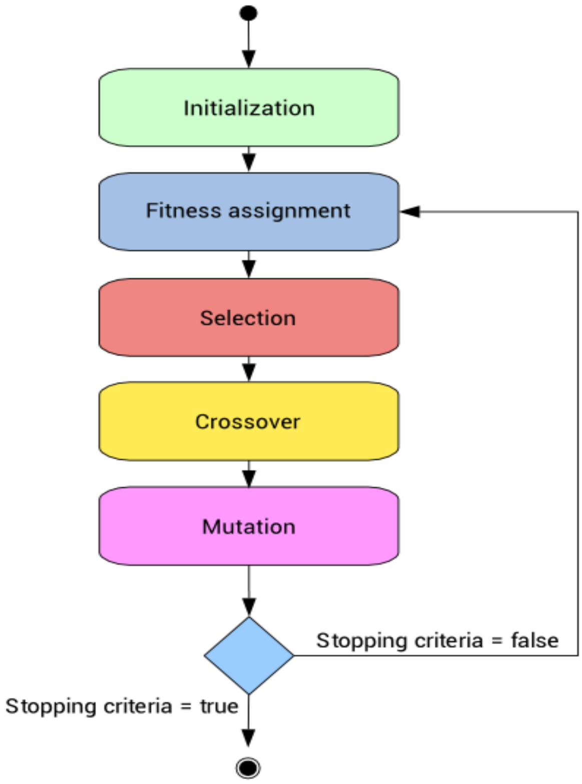

Figure 5: Genetic Algorithm flowchart

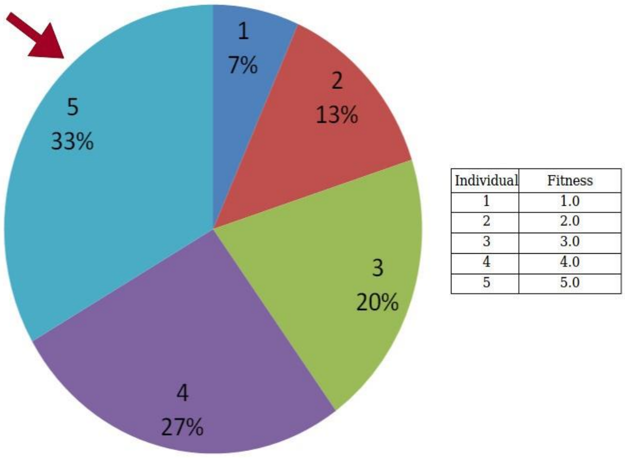

Figure 6: Roulette Wheel selection

GA is initialized by creating the first population randomly, the initial population will be pushed into the fitness function one by one in order to obtain the fitness of each individual for Selection. In selection phase the population is sorted in ascending order. this will be done by performing Roulette Wheel Selection, each individual of the population is allocated a section of a roulette wheel, and the size of each section is proportional to its fitness. A pointer is spun and the individual to whom it points is selected; this stops when the selection criterion is met. Figure 6 describes the idea of roulette wheel selection.

Once the selection process is completed, the crossover algorithm is initiated. Crossover algorithm swaps certain parts of two selected parents in order to capture the good parts of them and create better new ones. There are some different types of crossovers:

1. Single point crossover
2. Multipoint crossover
3. Uniform crossover

Multipoint crossover is preferred in this paper. In multipoint crossover there are three main stages involved:

[1] Members of the reproduced strings are mated randomly

[2] Multiple positions are selected randomly

[3] The bits between successive crossover points are exchanged to produce the new off-springs.

Multipoint crossover is shown in figure 7.

Using only Selection and Crossover comes up with some problems:

- There may not be enough diversity in the initial population, we need to make sure that GA researches the entire search space.
- GA may converge on some local optima due to a bad selection decision.

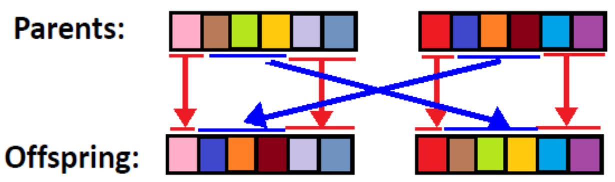

Figure 7: Multipoint Crossover

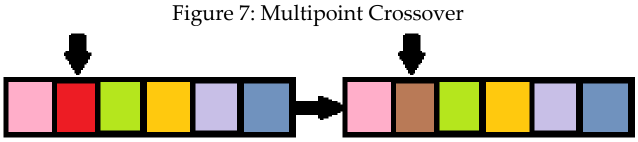

Figure 8: Mutation

In order to solve these two problems, Mutation has been introduced. Mutation is the occasional random alternator of a value of a string position. The probability of mutation must be low because a high mutation rate destroys fit strings, thus the optimum solution is lost. Mutation is illustrated in figure 8.

Choosing the fitness function is the most challenging part of optimizing by using GA. In this paper the fitness function is required to evaluate the PID parameters, it could be created to find the parameters that gives the smallest overshoot, the smallest rise time, or the smallest settling time, however the performance index of the GA must combine all of these objectives, so we will design a fitness function that minimizes the error of the system, thus overshoot, rise time, and settling time will all be minimized. There are several methods to calculate the error signal for the fitness function, we preferred to choose Integral Absolute Error (IAE), as given by the following equation.

$$IAE = \int_0^\infty |e(t)|\, dt$$

By running the optimization process in MATLAB, we obtain the optimized tuning parameters shown in table 3.

By blogging these gain values into the controller, we will get the response of the optimized controller. When comparing it to the Z-N tuning method, the comparison is shown in figure 9. the overshoot of the GA based controller is 0.413% compared to 82.9% for the Z-N controller which is about 200 times better.

TABLEIII: GA PARAMETERS

	Kp	Ki	Kd
Gain	1	0.15	0.05

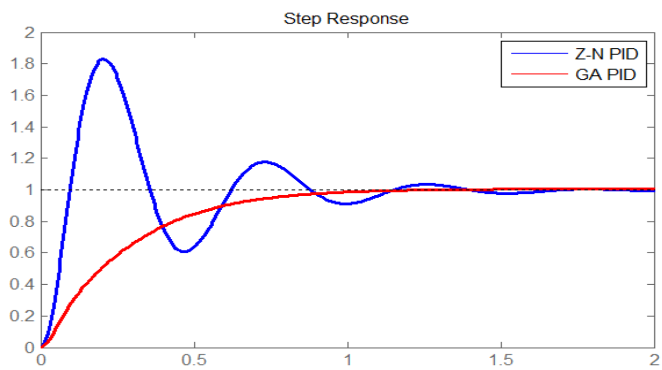

Figure 9: Z-N PID versus GA PID

The rise time is 0.467 seconds for the GA controller compared to 0.0629 seconds for the Z-N which is slower. The settling time is 1.23 seconds for the GA controller compared to 1.57 seconds for the Z-N controller. The GA based controller has much better response.

5. THE IMPACT OF CHANGING BOUNDARIES

GA has been tested against different sizes of population between 50 and 200, and also number of iterations has been changed between 50 and 200 in order to find the global optima that we are looking for. As shown in table 4, the optimum tuning parameters are found when the population size was 200 and the number of iterations was 200. A bigger population size gives a more diverse population. The initial population is created randomly. It is obvious why a bigger population size gave us the best results. Same for the maximum number of iterations, the bigger it is, the greater the chance for improving the final results.

6. DESIGN OF PID FILTER

Although the PID controller has a good response especially in the case of a constant desired value, it is weak when sudden change in the error signal occurs due to load change and disturbance, and this weakness may lead to voltage instability.

It is proposed to develop a PID low pass filter for filtering the high frequency change in the error signal and improving the transient response of the overall system.

$$\frac{U(s)}{E(s)} = Kp + Ki * \frac{1}{s} + Kd * \frac{N}{1 + \frac{N}{s}}$$

By running the simulation again we obtain the gain values and the filter coefficient, they are shown in table 5.

As shown in figure 10, the maximum overshoot of the GA based PIDF is 0.084% compared to 0.413% for the conventional GA based PID and 82.9% for the Z-N based PID. The rise time of the GA based PIDF is 0.136 seconds compared to 0.467 seconds for GA based PID and 0.0629 seconds for the Z-N based PID. The settling time for the PIDF is 0.325 seconds compared to 1.23 seconds for the GA based PID and 1.57seconds for the Z-N based PID.

7. CONCLUSION

The designed Genetic Algorithm based PID controller has a better and more stable response, the maximum overshoot is noticeably smaller than the Z-N based controller, the settling time is also smaller for the GA based PID, but the rise time of the GA based PID was greater. However, the best results are obtained by using the GA to design a PID Filter; the GA-PIDF has the best overall response.

TABLEIV: THE EFFECT OF CHANGING SEARCH BOUNDARIES

Iterations	Pop size	t(sec) rise	t(sec) settling	t(sec) peak	MP %
50	50	0.563	4.4	1.29	26.7
	100	0.563	4.43	1.27	27.1
	200	0.134	7.3	0.271	8.35
100	50	0.46	0.936	2.94	1.7
	100	0.254	0.607	0.504	2.98
	200	0.244	4.55	2.24	4.61
150	50	1.32	11.6	4.18	12.6
	100	1.42	8.14	4.62	9.47
	200	1.45	2.98	1.16	1.1
200	50	0.474	1.83	>12	-.004
	100	1.46	2.99	5	1.17
	200	0.467	1.23	3.5	0.413

TABLEV: PIDF PARAMETERS

Gain	Kp	Ki	Kd	N
	1.9999	0.26007	0.16685	163.619

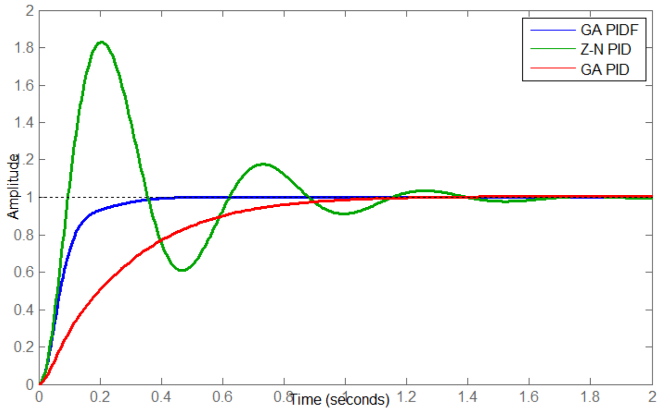

Figure 10: Z-N PID vs GA PID vs GA PIDF

APPENDIX

Fitness Function

```
function [j] = pid_optim(x)

s = tf('s');
%% system declaration
plant = 20 / (0.0063*s^3 + 0.299*s^2
+ 7.47*s + 1);
h = 1/(0.1*s + 1);
kp = x(1);
ki = x(2);
kd = x(3);
n = x(4);
cont = kp +ki/s + kd*(n/(1+n/s));
%% error calculation (IAE)
step(feedback(plant*cont,h));
dt = 0.01;
t = 0:dt:1;
e =
1-step(feedback(plant*cont,h),t);

j = sum(t'.*abs(e)*dt);
end
```

ACKNOWLEDGMENT

We would like to acknowledge and give our warmest thanks to Dr. Khaled Eltag, without whom we would not have been able to complete this work.

REFERENCES

[1] P. KUNDUR "Power System Stability and Control", Edited by NealJ. Balu Mark. Lauby Power System Planning and Operations Program Electrical Systems Division Electric Power Research lnstioitute Palo Alto, California.J. U. Duncombe, "Infrared navigation—Part I: An assessment of feasibility," IEEE Trans. Electron Devices, vol. ED-11, pp. 34-39, Jan. 1959.

[2] K Ogata, "Modern Control Systems", University of Minnesota, Prentice Hall, 1987.

[3] Mehta, V. K., & Mehta, R. (2005). "Principles of Power System: Including Generation, Transmission, Distribution, Switchgear and Protection: for BE/B. Tech., AMIE and Other Engineering Examinations". S. Chand Publishing.

[4] Goldberg, D. E. (1989). "Genetic algorithms in search, optimization, and machine learning". Addison. Reading W.-K. Chen, Linear Networks and Systems, Belmont, CA: Wadsworth, 1993, pp. 123-135.

[5] Chipperfield, A. J., & Fleming, P. J. (1995). "The MATLAB genetic algorithm toolbox".

[6] ÙĐĐ, Á. "The PID controller based on Genetic Algorithm for voltage stability of the synchronous machine".

[7] Eltag, K., & Zhang, B. (2021). "Design Robust Self-tuning FPIDF Controller for AVR System". International Journal of Control, Automation and Systems, 19(2), 910-920.

[8] Elsrogy, W. M., Fkirin, M. A., & Hassan, M. M. (2013, May). "Speed control of DC motor using PID controller based on

[9] artificial intelligence techniques". In 2013 (CoDIT) (pp. 196-201). IEEE.

[10] Mohamed Ibrahim, S. B. (2005). "The PID controller design using genetic algorithm".

[11] Meena, D. C., & Devanshu, A. (2017, January). "Genetic algorithm tuned PID controller for process control". In 2017 International Conference on Inventive Systems and Control (ICISC) (pp. 1-6). IEEE.

[12] Wang, Q., Spronck, P., & Tracht, R. (2003, November). "An overview of genetic algorithms applied to control engineering problems". In Proceedings of the 2003 International Conference on Machine Learning and Cybernetics. IEEE.

Muhammad Abdulkhalig Husain Abdurrahman, was born in Sudan 1999, obtained B. SC in Electrical and Electronics Engineering from Omdurman Islamic University O.I.U on December 2021. He is currently a teaching assistant in Omdurman Islamic University faculty of engineering sciences.

Muhammad was a semifinalist of the 2021 Crossroads Emerging Leaders Program hosted by the Lakshmi mittal and family south Asia institute at Harvard University.

Moamer Abdullbagi Omer Nourain, was born in Sudan 1995, obtained B. SC in Electrical and Electronics Engineering from Omdurman Islamic University O.I.U on December 2021.

Mohammed Omer Abdulsalam Muhieeddeen, was born in Sudan 1998, obtained B. SC in Electrical and Electronics Engineering from Omdurman Islamic University O.I.U on December 2021.

Omer Hafiz ALnour Mahjoub, was born in Sudan 1996, obtained B. SC in Electrical and Electronics Engineering from Omdurman Islamic University O.I.U on December 2021.

Manareldeen Ahmed received the Ph.D. degree in electrical engineering from the Hebei University of Technology, Tianjin, China, in 2019. He is currently a Post-Doctoral Researcher with the Zhejiang University–University of Illinois at Urbana–Champaign Institute, Zhejiang University, Haining, China. His current research interests include neuromorphic computing, spiking neural networks, and AI chips.