iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL
FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 8 Issue: VI Month of publication: June 2020

DOI: http://doi.org/10.22214/ijraset.2020.6009

A Multiobjective Optimisation based Approach of PID Controller Tuning in an AVR System

Saransh Chourey[1], Yash Agrawal[2], Sanket Bose[3], Minal Khewarkar[4], Rupendra Kumar[5], Naushin Anjum[6]
[1, 2, 3, 4, 5, 6]Department of Electrical and Electronics Engineering, Bhilai Institute of Technology Durg

Abstract: We all know very well that there is need of PID controller tuning in an AVR system to maintain stability in transient conditions. Many soft computing techniques such as Artificial Neural Network, Fuzzy logics, Optimisation and many more can be used to achieve the purpose.

There can be many parameters for selecting an appropriate method such as Robustness, Accuracy, Efficiency, Time required, complexity etc. In this paper we have selected Optimisation technique just because of its less complexity and many options available within it.

We also have many wide options in the form of a vast range of different algorithms. We have taken Multiobjective Optimisation in our study, its basic working is same as conventional optimisation techniques and difference is it allows us to achieve more than one objectives in an optimisation problem.

Keywords: Proportional Integral Derivative, Automatic Voltage Regulator.

I. INTRODUCTION

As we know in a practical power system there are many disturbances faced by different power system components applied on the grid due to various reasons such as sudden load fluctuations, faults, generation loss and sudden load losses etc.

So in order to counter these disturbances we use a PID controller to maintain the stability of an AVR alternator. And the determination of this PID controller parameters is done through various soft computing techniques from which we have used Multiobjective optimisation which is described in this paper [9] [3]. The best way of transient stability analysis is by taking the step response of the system which is what has been done in this paper.

There are works of PID tuning by using single objective optimisation algorithms, in which they have taken overshoot as the objective function and try to minimise it by using optimisation algorithm for which they get the corresponding best values of PID controller gains [1].

But in those case we are guaranteed to have minimum steady state error (Ess) i.e. overshoot error, but other step response parameters like settling time and rise time are not guaranteed to be minimum, which are also important from the stability point of view as the system should come out of the transient condition as soon as possible so they are also need be minimum to get an overall improvement in the step response [4] – [8].

Therefore, keeping in mind above additions, in this paper we have tried to achieve these improvements by using Multiobjective Optimisation [3]. As it's very clear from the name itself that this method is used when we require to achieve more than one objectives so in our study we have taken three different types of errors (which are described in detail in next section) as our objective our functions to get more improved step response. As in case of single objective optimisation we have many algorithms in Multiobjective Optimisation as well out of which we have used Genetic Algorithm [10].

To understand the problem the step response of an AVR model is given in Fig 2, whose block diagram is shown in Fig 1. AVR consist of three block i.e. Amplifier, Exciter and Generator in the forward path and sensor in the feedback path.

The time response specification of the step response of an AVR gives results showing a very high instability like Overshoot Error (Mp) = 65.7223%, Rise Time(Tr) = 0.2607 sec, Peak Time(Tp) = 0.7522 sec and Settling Time(Ts) = 6.9865 sec. All these specifications are significantly improved in this paper.

There are certain sets of value range of parameters of PID controller and the AVR for this particular application. Those are shown below in table 1 for AVR block parameters and table 2 for PID controller parameter ranges. [1]- [2]

II. OBJECTIVE FUNCTIONS

There are three objective functions which we are using in our Multiobjective Optimisation approach using Genetic Algorithm, all three are different types of error signals which we try to reduce using optimisation to achieve an improved step response. These are integral errors which allows us to have the values in frequency domain. [2] [12]-[15]

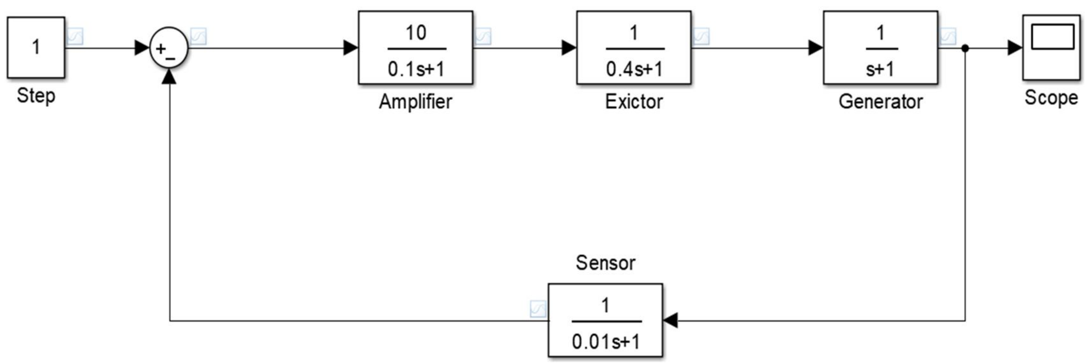

Fig 1 Block diagram of an AVR to view its step response

These errors are as follows:-

1) *Integral Absolute Error (IAE)*:- In this error first the absolute value of the error signal is taken and then it is taken through the integrator to generate signal IAE. MATLAB Simulink model is shown in fig 3.
2) *Integral Square Error (ISE):*- This error is generated by multiplying the error signal by a unit square math function and then taking that through the integrator thus we get the signal ISE. MATLAB Simulink model is shown in fig 4.
3) *Integral Time Absolute Error (ITAE):*- This error is generated by first determining its absolute value and then that signal is connected to the one input of the product block. The other input in the product block is given to clock input. Now this whole signal is then taken through the integrator to get the signal ITAE. MATLAB Simulink model is shown in fig 4.

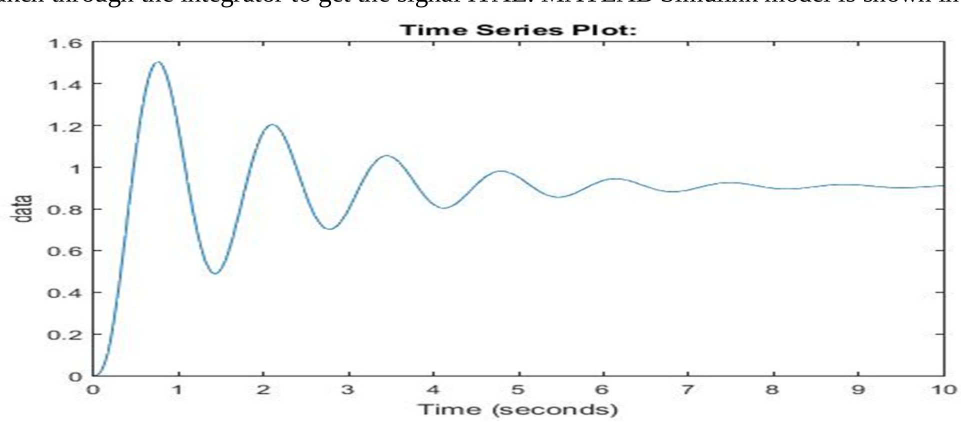

Fig 2 Step Response of an AVR [2]

TABLE 1
Range Of Avr Blocks Parameters For Application Across A Generator [2]

Block/parameters	Gain (Kx)		Time Constant (Tx)	
	Min.	Max.	Min.	Max.
Amplifier (Ka, Ta)	10	400	0.02	0.1
Exciter (Ke, Te)	10	400	0.5	1
Generator (Kg, Tg)	0.7	1	1	2
Sensor (Ks, Ts)	1	2	0.001	0.06

TABLE 2
PID Controller Parameters Range For Its Application In Avr Control [2]

S. NO.	Parameter	Minimum value	Maximum value
1.	Kp	0	1.5
2.	Ki	0	1
3.	Kd	0	1

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 [2].

III. METHODOLOGY

A. We have used MATLAB 2015a software for implementation of the optimisation process.
B. Before starting the actual optimisation, first we have to generate the model of the system known as plant model.
C. The whole process will be in two major steps, first the plant model is simulated to get an output which is further used in second step as an input for optimisation simulation.
D. Plant model is designed in the Simulink app of the MATLAB where the entire system is designed which is explained later.
E. After plant model design the next step is optimisation, when the plant model is ready then we are ready with our objective function which is provided to the Optimisation algorithm.
F. Now the code for the Multiobjective optimisation using Genetic Algorithm technique is written and saved as a function file in the same folder where the plant model is saved.
G. Supporting M – file if required, should also be saved on same location as function file and Simulink file.
H. Please make sure the path location chosen in the MATLAB home window is same as the location where we have saved our files, otherwise execution of simulation will show error.
I. Now after all these things run the function file of the optimisation code and observe the simulation.
J. The simulation is generally time taking as the plant model is as complex as its overall transfer function is of order 11 and also we have three unknown variables which make the entire process vary complex one and therefore it require large time in its execution.
K. However the exact time depends on the no. of population selected and also on the processor availability of RAM clock speed etc. are some other factors.
L. After simulation ends the value of three unknown variables are now given by the algorithm which we then enter to the Kp, Ki and Kd fields of PID controller employed in the plant model.
M. After loading controller value we run the plant model simulation and observe the step response and compare it with the step response of AVR only or with step response of the AVR- PID system with some random values of PID controller parameters.

TABLE 3
Results Of The Simulation At Different Populations

S.N O.	No. Of populat ion (n)	Kp	Ki	Kd	Overshoot (Mp)		Rise Time (Tr)		Settling Time (Ts)	
					Value (%)	% Improvement	Value (sec)	% Improvement	Value (sec)	% Improvemen t
1.	50	0.1239	0.6290	0.6605	11.6123	82.23 %	0.1358	47.91 %	18.6983	-
2.	100	0.4819	0.2963	0.7348	10.7063	83.71 %	0.1170	55.12 %	3.7738	45.98 %
3.	150	0.2768	0.2963	0.4381	4.6722	92.89 %	0.2159	17.18 %	6.7068	04.00 %
4.	200	0.9596	0.2275	0.4878	9.2001	86.00 %	0.1514	41.93 %	5.8387	16.43 %

N. We can also view time response specification data i.e. for example Rise Time, Peak Time etc. by using 'step' command in the command window of the MATLAB and again can compare it.
O. Actual picture of the plant model designed in the MATLAB Simulink is shown in fig 6.
P. Various blocks of the AVR-PID model can be designed by taking transfer function block from Simulink library for each parameter.
Q. Where the typical values can be entered by clicking on these blocks.
R. The PID controller can be designed by taking differential (s) and integral (1/s) and gain (k) blocks and cascading or serially connecting them to form PID controller block.
S. But complete PID controller block is directly available in the Simulink library which we have used directly and entered the values of Kp, Ki and Kd as unknown variables as this is what we have to find.
T. We have to declare Kp, Ki and Kd as unknown variables by going in the settings and provide their initial values which are 0, 0 and 1 for Kp, Ki and Kd respectively.
U. Then as we have to find the step response, we will select the step block in the sources section of the Simulink library.
V. Then to observe the step response we will require a scope to see the graph which can be chosen from Simulink library.
W. To export the plot to workspace in the MATLAB we will use block ''to workspace'' from the Simulink library.
X. Then we generate our main objective function for optimisation i.e. error signal which is generated by comparing the output of the system with the input.

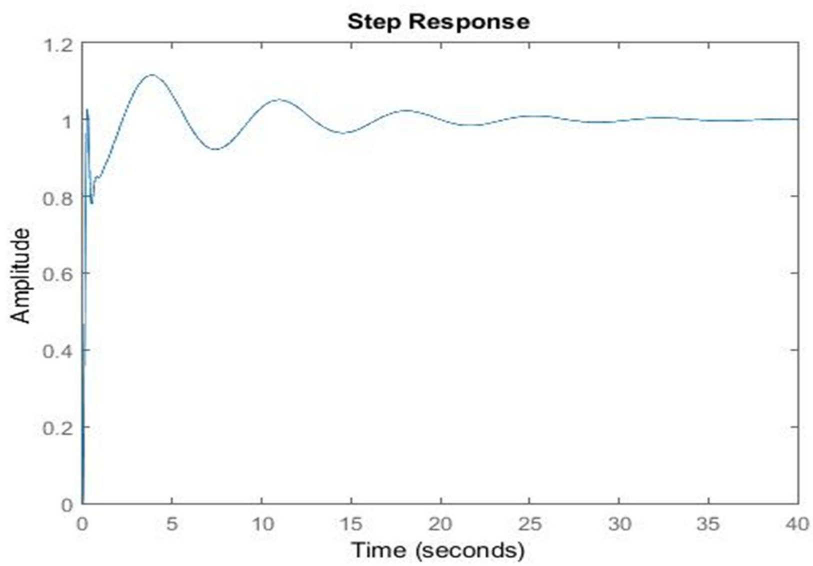

Fig 3 Result for n = 50

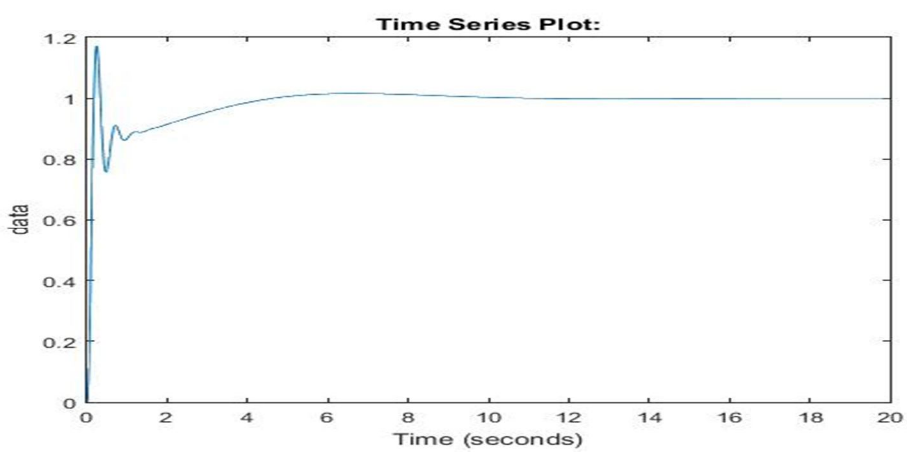

Fig 4 Result for n=100

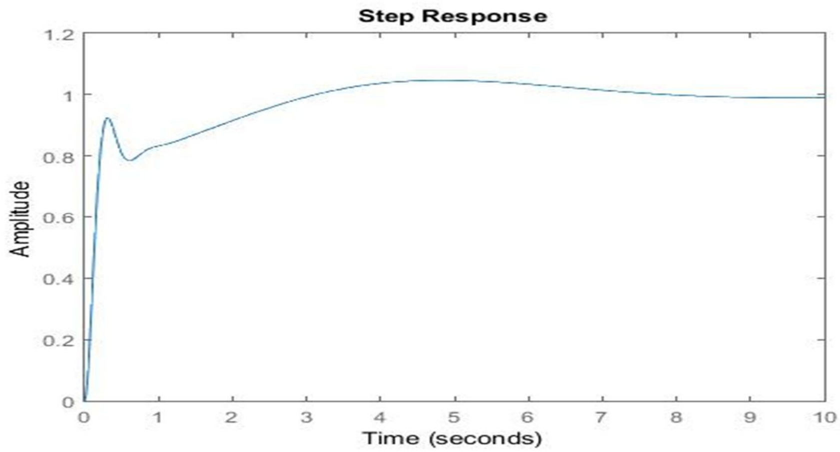

Fig 5 Result for n = 150

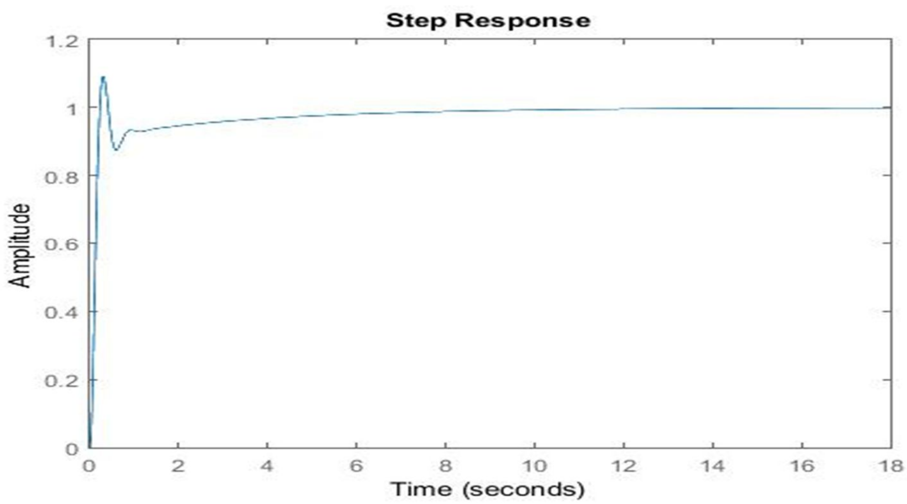

Fig 6 Result for n = 200

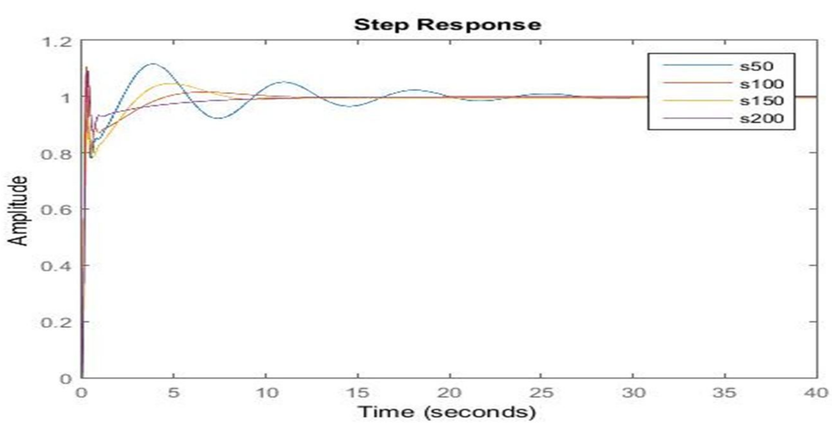

Fig 7 Comparison of different Results

Y. This is done by connecting output as negative feedback and the step input as a positive feedback to a comparator or a summing block and then by applying it into a system consist of integrator, square and absolute blocks together or with selected combination which can again be chosen from Simulink library.

IV. RESULTS

The result of the simulation is shown in the tabular form in table 3, with percentage improvement of the parameters from the original step response along with determined controller parameters and new time response specifications.

Each of the step response plot is also shown in various Figures (Fig 3 – Fig 6). Also the single plot comparing different results is also shown in Fig 7 to get an idea of the time response specifications in a single glimpse.

V. CONCLUSION

As described above the transient stability of an AVR can be significantly improved when we use an optimised PID controller parameters. The three controller parameters can have many no. of combinations even in the range of their bounded values, so it is almost impossible to use trial and error method for predicting the controller parameter to be used. Therefore we have to have apply some soft computing technique which are advanced, current edge, robust, accurate and time saving. In this study we have used optimisation as soft computing technique. In the Multiobjective Optimisation technique which is a new conceptual technique try to minimise all the three errors at the same time hence it takes a lot of time in execution. There are three objective function hence it will require a lot more population for better result otherwise the result is not so good. So with greater no. of population and greater time taken for execution it provide good result.

REFRENCES

[1] ‘’Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems ‘’. By Yamille del Valle, Student Member, IEEE, Ganesh Kumar Venayagamoorthy, Senior Member, IEEE, Salman Mohagheghi, Student Member, IEEE, Jean-Carlos Hernandez, Student Member, IEEE, and Ronald G. Harley, Fellow, IEEE.

[2] ‘’A Particle Swarm Optimization Approach for Optimum Design of PID Controller in AVR System’’. By Zwe-Lee Gaing, Member, IEEE.

[3] ‘’Multiobjective Optimization: An Educational Opportunity in Power Engineering’’. By Gerald T. Heydt, Fellow, IEEE.

[4] T. Okada, T. Watanabe, and K. Yasuda, “Parameter tuning of fixed structure controller for power system stability enhancement,” in Proc. IEEE-PES Transmission and Distrib. Conf. Exhibition 2002: Asia Pacific, Oct. 2002, vol. 1, pp. 162–167.

[5] W. Qiao, G. K. Venayagamoorthy, and R. Harley, “Design of optimal PI controllers for doubly fed induction generators driven by wind turbines using particle swarm optimization,” in Proc. IEEE World Congr. Comput. Intell., Jul. 2006.

[6] Y. Mitsukura, T. Yamamoto, and M. Kaneda, “A design of self-tuning PID controllers using a genetic algorithm,” in Proc. Amer. Contr. Conf., San Diego, CA, June 1999, pp. 1361–1365.

[7] T. Ota and S. Omatu, “Tuning of the PID control gains by GA,” in Proc. IEEE Conf. Emerging Technol. Factory Automation, Kauai, HI, Nov. 1996, pp. 272–274.

[8] R. C. Eberhart and Y. Shi, “Comparison between genetic algorithms and particle swarm optimization,” in Proc. IEEE Int. Conf. Evol. Comput., Anchorage, AK, May 1998, pp. 611–616.

[9] H. Saadat, Power System Analysis. New York: McGraw-Hill, 1999.

[10] E. Zitzler and L. Thiele, “Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach,” IEEE Trans. Evol. Comput., vol. 3, no. 4, pp. 257–271, Nov. 1999.

[11] D. Van Veldhuizen and G. Lamont, “Multiobjective evolutionary algorithms: Analyzing the state of the art,” IEEE Trans. Evol. Comput., vol. 8, no. 3, pp. 125–148, 2000.

[12] R. A. Krohling and J. P. Rey, “Design of optimal disturbance rejection PID controllers using genetic algorithm,” IEEE Trans. Evol. Comput., vol. 5, pp. 78–82, Feb. 2001.

[13] Y. Mitsukura, T. Yamamoto, and M. Kaneda, “A design of self-tuning PID controllers using a genetic algorithm,” in Proc. Amer. Contr. Conf., San Diego, CA, June 1999, pp. 1361–1365.

[14] T. Kawabe and T. Tagami, “A real coded genetic algorithm for matrix inequality design approach of robust PID controller with two degrees of freedom,” in Proc. 12th IEEE Int. Symp. Intell. Contr., Istanbul, Turkey, July 1997, pp. 119–124.

[15] R. A. Krohling, H. Jaschek, and J. P. Rey, “Designing PI/PID controller for a motion control system based on genetic algorithm,” in Proc. 12th IEEE Int. Symp. Intell. Contr., Istanbul, Turkey, July 1997, pp. 125–130.