ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

DIFFERENTIAL EVOLUTION ALGORITHM FOR AN OPTIMAL TUNING OF PROPORTIONAL INTEGRAL DERIVATIVE CONTROLLER FOR AUTOMATIC VOLTAGE REGULATOR

G. R. Venkatakrishnan[1], R. Rengaraj[1], Brathindara S.[2], Kashyap R.[2] and Krithika K.[2]
[1]Department of Electrical and Electronics Engineering, SSN College of Engineering, Kalavakkam, Chennai, India
[2]Department of Electronics and Communication Engineering, SSN College of Engineering, Kalavakkam, Chennai, India
E-Mail: venkatakrishnangr@ssn.edu.in

ABSTRACT

An Automatic Voltage Regulator (AVR) system maintains a constant terminal voltage irrespective of the load. The output of an AVR system without a proper controller has undesirable time domain specifications when subjected to a particular input. A Proportional-Integral-Derivative (PID) controller cascaded with the AVR system enables a stable control method resulting in optimum time domain specifications. PID controller is one of the most popularly used control algorithm in industrial problems. The controller must be tuned to obtain the best possible values for the three gains namely proportional (P), integral (I), and derivative (D) in order to achieve the desired performance by meeting the design requirements. Automatic tuning of PID controller is one of the feasible option for real time application of AVRsystems.In general, nature inspired evolutionary algorithms are employed for optimal tuning of the PID controller for the given system. In this paper, Differential Evolution (DE) algorithm which is one the evolutionary algorithms is used for the optimal tuning of PID controller. The choice of the fitness function plays a crucial role in the tuning process. Different fitness functions are used for optimal tuning of PID controller and the obtained results are compared and presented for the given AVR system in this paper.

Keywords: differential evolution algorithm, automatic voltage regulator system, optimal tuning, proportional-integral-derivative controller, fitness function.

1. INTRODUCTION

The main function of an Automatic Voltage Regulation (AVR) system is to ensure that the voltage generated by a generator is smooth and constant and stabilizes the voltage value when there is a sudden change in load. AVR systems are used to control the output voltage of the generator depending upon the load connected, thus preventing voltage fluctuations and effectively protecting the generator from possible damages. The control of output voltage is achieved by varying the excitation current to the generator depending on the load. This form of control can be manual or automatic. Due to numerous disadvantages, manual control is usually not preferred. Automatic control is the most favoured form of controller. [1]

The most popular automatic controller used for this purpose is the Proportional-Integral-Derivative (PID) controller. A PID controller is a control loop feedback mechanism that is widely used in industrial control. They are easy to construct and simple to manipulate based on the problem statement. The main function of the PID controller is to improve the performance indices of the AVR system. The performance indices are a set of parameters that determine the general characteristics of any system. An AVR system without a PID controller has undesirable characteristics like high peak overshoot, more settling time etc. that might cause damage to the system and degrade its efficiency. On the other hand, an AVR system with PID controller reduces the peak overshoot, settling time and so on.

A set of parameters of the PID controller, viz-a-viz, proportional gain K_p, integral gain K_i and derivative gains K_d affect the performance of the system. The process of setting the optimal gains for P, I and D to get an ideal response from a control system is called tuning. [3] The optimum values are achieved by trial and error method and could be time consuming. Thus, tuning is usually done automatically with the help of different algorithms and the most commonly used ones are Evolutionary Algorithms (EA).

EAs are nature inspired algorithms which are stochastic in nature. It imitates the natural process of evolution where in the organisms adapt themselves to the environment for their survival. Similarly, evolutionary algorithms are population based algorithms. A set of possible candidate solutions serve as the initial population. Processes such as mutation and recombination bring about diversity in the population. On the other hand, a deterministic selection process is used to reduce the diversity. These processes are carried on repetitively so that the population adapts to the required set of conditions, also known as the fitness function. Over time the fitness of the population increases and the best candidate solution for the given problem is obtained. [2]

One such evolutionary algorithm is the Differential Evolution (DE) algorithm. It was introduced in the year 1996 by Storn and Price. It is used for optimising real valued functions subject to the given constraints, which are otherwise difficult to solve manually. This metaheuristic approach works towards finding the global maxima/minima by using mutation, crossover and selection processes.

ARPN Journal of Engineering and Applied Sciences

Depending on the fitness function chosen, different optimal values of K_p,K_i&K_d are obtained. As expected, the performance of the system differs with each set of optimal values. This paper explores the optimal values obtained with different fitness functions and compares the results, in an effort to identify the most suitable values for the system.

The rest of the paper is organized as follows: Section 2 is allotted for problem description of the AVR system. The detailed description of DE algorithm is explained in Section 3. The results obtained by implementing the DE algorithm to AVR system are presented in Section 4 and finally conclusion is drawn in Section 5.

2. PROBLEM DESCRIPTION

The block diagram [5] shown in Figure-1 is the general representation of an AVR system with no control element.

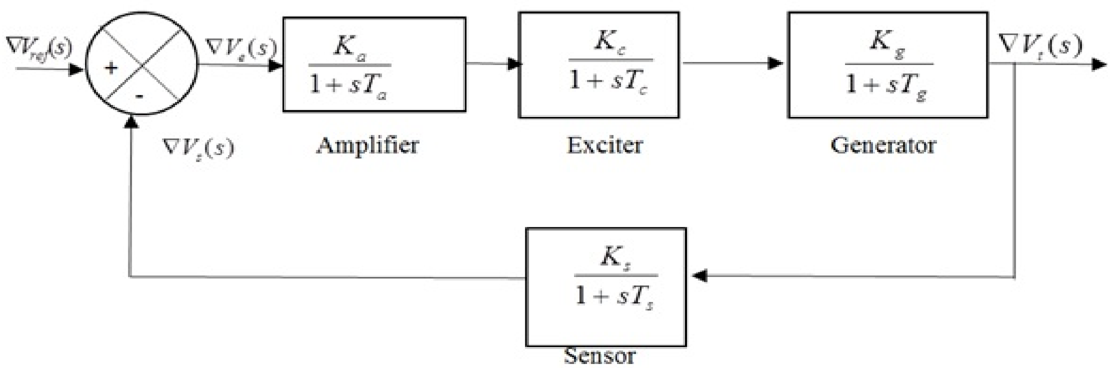

Figure-1. Block diagram of AVR system.

It consists of an amplifier stage, which increases the magnitude of the input quantity for further manipulation. It is followed by the exciter stage which controls the current to the excitation field and directly influences the speed of the generator, which is the next stage of the block diagram. The sensor stage, which is an integral part of the feedback loop monitors the control variable and directs some portion of it to the summing block.

The feedback signal, which is a portion of the generated output voltage, is compared with the reference voltage set by the user. Any mismatch between the two is amplified and fed to the exciter which in turn controls the output voltage of the generator. The feedback employed is degenerative in nature.

The transfer function of the system without any controller is given by Equation. (1).

$$\frac{\Delta V_t(s)}{\Delta V_{ref}(s)} = \frac{K_a K_e K_g (1+sT_s)}{(1+sT_a)(1+sT_e)(1+sT_g)(1+sT_s)+K_a K_e K_g K_s} \quad (1)$$

where T_a,T_e, T_g and $T_s$are the time constants offered by the amplifier, exciter, generator and sensor blocks respectively, K_a ,K_e, K_g and K_s are the gain offered by the amplifier, exciter, generator and the sensor stages respectively, $\Delta V_t(s)$is the Laplace transform of output voltage of the AVR system and $\Delta V_{ref}(s)$is the Laplace transform of input voltage applied to AVR system.

The block diagram [5] of the AVR system coupled with a PID controller is represented in Figure-2.

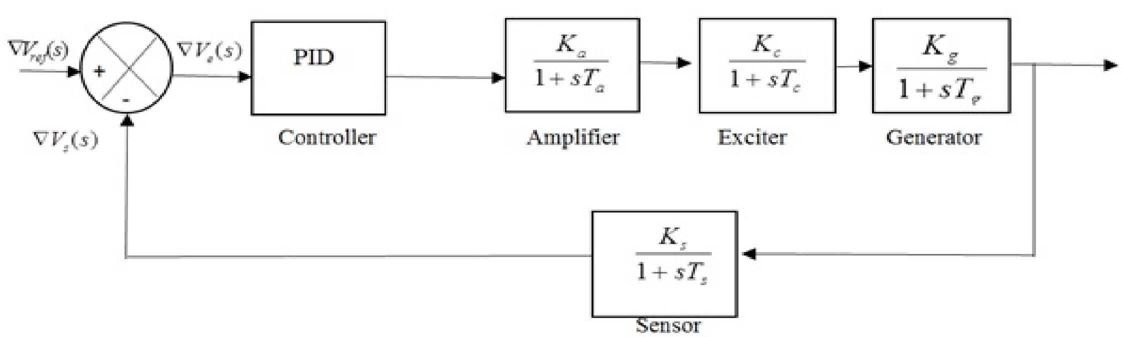

Figure-2. Block diagram of AVR system with PID controller.

ARPN Journal of Engineering and Applied Sciences

Equation 2 describes the transfer function of the AVR system with PID controller

$$\frac{\Delta V_t(s)}{\Delta V_{ref}(s)} = \frac{K_a K_e K_g (1+sT_s)(K_p s + K_i + K_d s^2)}{(1+sT_a)(1+sT_e)(1+sT_g)(1+sT_s)s + K_a K_e K_g K_s (K_p s + K_i + K_d s^2)} \tag{2}$$

Where K_p,K_i and K_d are the gains offered by the proportional, integral and derivative components of a PID Controller respectively.

In this case, the mismatch between the feedback signal and reference voltage is given to a PID controller whose output is amplified and fed to the exciter. Depending on the weightage given to each set of time domain specifications, different values of PID gains are obtained. These gains change the transfer function of the system and affect its performance. In the following few paragraphs, we explore the values of α and βused for tuning, followed by the results section which compares the simulation results corresponding to different α and βvalues.

The performance of the system is altered depending on the values of Kp, Ki and Kd. The process of tuning is done by minimising the fitness function described by Equation. (3) [6].

$$f(s) = \alpha(M_p + ess) + \beta(T_r + T_s) \tag{3}$$

where Mp is the maximum peak overshoot which denotes the maximum value of the response, rise time tr is the time required for the response to reach its maximum value, t_s is the time required for the system response to settle down to a steady value and ess is the steady state error.

The optimal solution of the above fitness function is obtained through software simulation by giving different values of α and β . With each set of αandβ, the performance indices of the system changes. For example, by choosing a value of α greater than β , we obtain optimal gains which prominently reduce the value of peak overshoot with little or no improvement in settling time. Often, the value of α and β are chosen depending on the problem statement, with a trade-off between peak overshoot and the settling time. In this paper, we compare the optimal values of gains obtained through software simulation for different values of α and β.

The fitness function described by Equation. (3) is minimized by satisfying the following inequality constraints described in Equation. (4)

$$K_{p\min} < K_p < K_{p\max},\ K_{i\min} < K_i < K_{i\max} \text{ and } K_{d\min} < K_d < K_{d\max} \tag{4}$$

where $K_{p\min}$, $K_{i\min}$ and $K_{d\min}$ are the minimum limits of proportional, integral and derivative gains respectively and K_{pmax} , $K_{i\max}$ and $K_{d\max}$ are the minimum limits of proportional, integral and derivative gains respectively.

3. DIFFERENTIAL EVOLUTION (DE) ALGORITHM

The first article on Differential Evolution (DE) algorithm was prepared by R.Storn and K.V.Price in the year 1995. The algorithm derives its name from the fact that it involves differential operation in its mutation stage. More than a decade ago, the DE algorithm, a population based stochastic, fast, simple and parallel search technique emerged as a very competitive form of evolutionary computing. [7]-[9]. Its ability to solve real life engineering and scientific problems made it popular. [10][11].

The following section explains the various steps of DE algorithm:

a) Initialization of parameter vectors

The DE algorithm starts with initialization of parameter vectors or target vector of decision variables D based on their upper and lower bounds. The population size N of the parameter vectors varies depending upon the problem. The parameter vector in DE algorithm is represented as $X_{i,G} = (x^1_{i,G}, x^2_{i,G}, ..., x^D_{i,G})$, where $i = 1,2,...,N$ and G represents the current generation. These parameter vectors are generated randomly between predefined lower and upper bounds $[X_{min}, X_{max}]$ where $X_{min} = (x^1_{min}, x^2_{min}, ..., x^D_{min})$ and $X_{max} = (x^1_{max}, x^2_{max}, ..., x^D_{max})$and is initialized using Equation.(5).

$$x^i_{j,G} = x^i_{\min} + rand^i_j (x^i_{\max} - x^i_{\min}) \tag{5}$$

Where $rand^i_j$ is a uniformly distributed random number lying between 0 and 1.

b) Mutation

The mutation strategies which is implemented in this paper is given by Equation. (6).

DE/best/1:

$$V_{i,G} = X_{best,G} + F(X_{r_1,G} - X_{r_2,G}) \tag{6}$$

Where $r_{1,G}$ and $r_{2,G}$ are mutually exclusive integers which are randomly generated within [12] which also differs from index i . The scaling factor or mutation

ARPN Journal of Engineering and Applied Sciences

factor F is a positive control parameter for scaling the difference vector is in the range $0 < F \leq 1.2 \cdot X_{best,G}$ the best individual target vector with the best fitness in the population of the current generation G and the mutant vector $v_{i,G} = \left(v_{i,G}^1, v_{i,G}^2, ..., v_{i,G}^D\right)$ where v_i^j is the mutant vector of decision variable j in vector i.

Crossover

Generally in an evolutionary algorithm a crossover step is introduced so as to increase the diversity of the parameter vector In DE, the trial vector is produced by applying this operation to each pair of target vector and its corresponding mutant vector [5][10]. There are two major kinds of crossover techniques. They are exponential crossover and binomial crossover techniques [5]. The binomial crossover function is predominantly used in these real world problems is represented in Equation. (7)

$$u_{i,G}^j = \begin{cases} v_{i,G}^j & \text{if} \quad rand_i^j \leq CR \;\; or \;\; j = j_{rand} \\ x_{i,G}^j & \text{otherwise} \end{cases} \tag{7}$$

Where CR is the crossover rate which is a user-specified constant within the range of 0 and 1, which controls the fraction of variables to be copied from the mutant vector, $rand_i^j$ is the randomly generated number between 0 and 1 and $_j rand$ is the randomly chosen integer in the range $[1, D]$ which ensures that trial vector $U_{i,G}$ differs from its target or parameter vector.

c) Selection

Selection is the last process in DE which decides whether or not the trial vector $U_{i,G}$ becomes the member of the parameter vector in the next generation. Based on the selection scheme. The one-to-one greedy scheme based on the fitness value of the trial and parameter vector for minimization problem can be formulated as in equation (8):

$$X_{i,G+1} = \begin{cases} U_{i,G} & \text{if} \quad f(U_{i,G}) \leq f(X_{i,G}) \\ X_{i,G} & \text{otherwise} \end{cases} \tag{8}$$

Where $f(U_{i,G})$ and $F(X_{i,G})$ are the fitness values of $U_{i,G}$ and $X_{i,G}$.

The above mentioned steps are repeated until the best fitness of the population does not show much changes over successive changes or the iterative process is carried on for a fixed number of generations.[6].

4. RESULTS

In this section, the results obtained by implementing DE algorithm to the AVR system is presented and discussed. The parameter values for each variable existing in transfer function described in Equation. (1) is obtained from [11]. The step response of the AVR system without PID controller is shown in Figure-3.

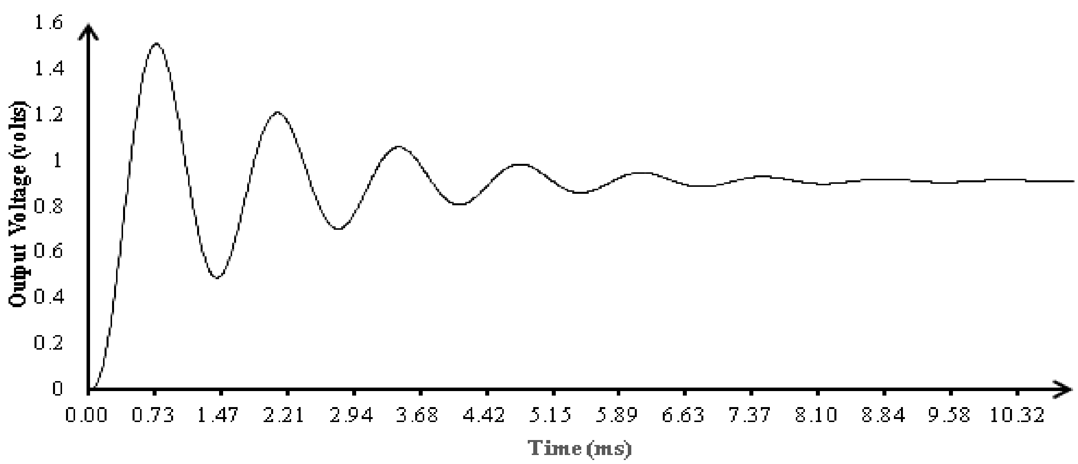

Figure-3. Step response of AVR system without PID controller.

The time domain specification of the AVR system without PID controller are: $M_p = 65.72\%$, $t_s = 6.9865s$, t_p= 0.7522 s and ess= 0.093826.Based on these results, it is evident that the system has undesirable characteristics such as a large overshoot, more settling time etc. which might cause harm to the AVR system. Hence, to improve its time domain specifications a conventional PID controller with DE algorithm is incorporated to the system. The fitness function to be minimized is explained in Equation. (3).

This section is fully dedicated for comparison of simulation results for each set of values for α and β.The flow of this section is as follows: first, the convergence graphs for all three cases is plotted. This is followed by the comparison of time domain specifications obtained for the

VOL. 12, NO. 12, JUNE 2017 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

different sets of values of α and β which is listed in Table-1.

Table-1. Different sets of weightage.

	Value of α	Value of β
Case 1	0.5	0.5
Case 2	0.75	0.25
Case 3	0.25	0.75

The maximum and minimum limits of K_p, $K_i$andK_d is listed in Table 2[12]. The DE algorithm for the givenAVR system has been implemented in MATLAB 2013b on Intel (R) Core (TM) i3 - 2100 CPU 3.10 GHz with 4G-RAM.

Table-2. Limits of PID controller gains.

	Maximum	Minimum
Proportional gain, K_p	1.5	0
Integral gain, K_i	1	0
Derivative gain, K_d	1	0

In this paper, the parameters for the DE algorithm for the chosen system are $N = 100, G_{\max} = 100$, $CR = 0.5$ and $F = 0.8$. The convergence characteristics obtained by implementing DE algorithm for different weightage is shown in Figure-3.

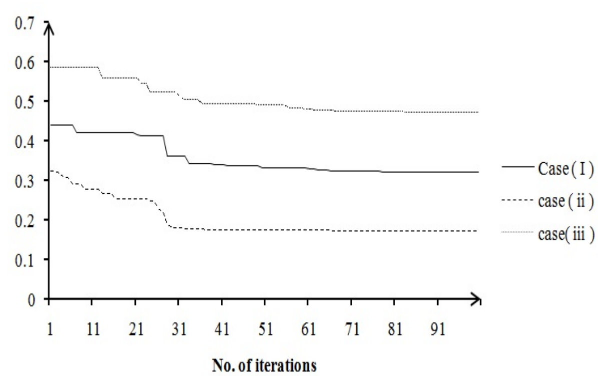

Figure-4. Convergence curve for different weightage.

The optimal PID controller gain values obtained for different fitness function is listed in Table-3.

Table-3. PID controller gains for different fitness functions.

Gain	Case (i)	Case (ii)	Case (iii)
Proportional gain, K_p	0.6835	0.6838	0.6838
Integral gain, K_i	0.2721	0.2715	0.2715
Derivative gain, K_d	0.6334	0.6294	0.6294

The step response of the AVR system with PID controller for different fitness function is shown in Figures-4 to 6. The transfer function used for getting the step response is given in Equation. (2) where the PID controller gains for different cases is taken from Table-3.

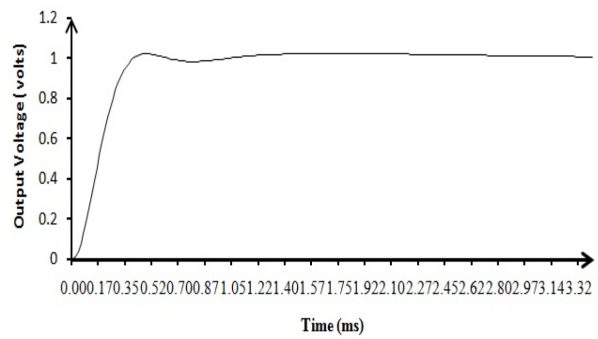

Figure-5. Step response of AVR system with PID controller for Case (i).

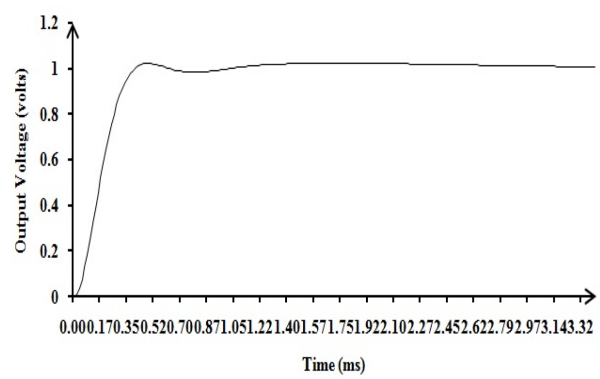

Figure-6. Step response of AVR system with PID controller for Case (ii).

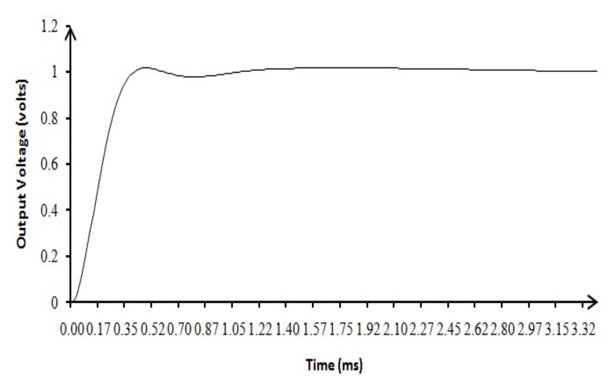

Figure-7. Step response of AVR system with PID controller for Case (iii).

The time domain specifications for different fitness function is listed in Table-4.

Table-4. Time domain specifications of different cases.

Time Domain Specification	Case (i)	Case (ii)	Case (iii)
Peak Overshoot (M_p) in %	1.998	1.9951	1.9951
Peak Time (t_p) in s	0.4892	0.4895	0.4895
Rise Time (t_r)in s	0.2500	0.2503	0.2503
Steady State Error (ess)	0	0	0

The effectiveness of the DE algorithm for the AVR system with PID controller is studied by conducting 50 test runs. The statistical results obtained from DE algorithm for 50 test runs is listed in Table-5 and is graphically represented in Figure-7.

Table-5. Statistical analysis of DE algorithm for AVR system with PID controller.

Statistical analysis	Case (i)	Case (ii)	Case (iii)
Minimum value	0.321343	0.170579	0.472586
Maximum value	0.50779	0.253236	0.658681
Mean	0.401739	0.175676	0.564477
Standard deviation	0.051652	0.013208	0.055349
Computational time	0.0065698 s	0.00633s	0.006988s

5. CONCLUSIONS

In this paper an AVR system with automatic PID controller was optimally tuned for proportional, integral and derivative gains using Differential Evolution (DE) algorithm. Different time domain specifications like maximum peak overshoot, settling time, rise time and so on is considered for this problem. In an effort to study the influence of the fitness function on the time domain specifications, different weightage is given to the fitness function. Software simulation of the system with different fitness function unveiled that the time domain specifications are almost similar for the three different sets of weights. However, the fitness function with weights α=0.75 and β=0.25 is found to yield better results for time domain specifications with lesser number of iterations to reach the minimum value. Statistical analysis of the simulation results reiterates the same fact.

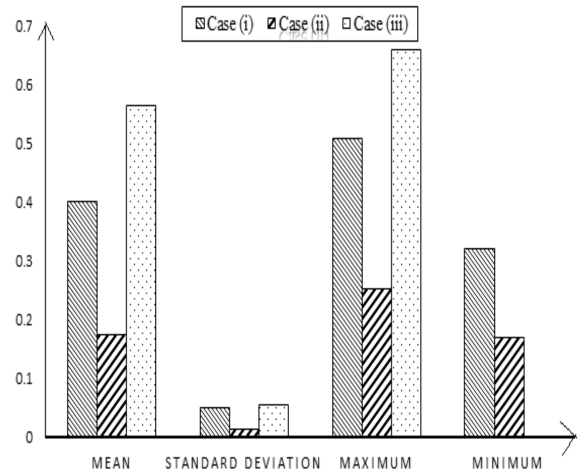

Figure-8. Statistical analysis of DE algorithm for AVR system with PID controller.

REFERENCES

[1] Qing Liu, Tarek Hassan Mohamed, ThongchartKerdphol, Yasunori Mitani4 - PID-MPC Based Automatic Voltage Regulator Design in Wide-Area Interconnect Power System, International Journal of Emerging Technology and Advanced Engineering, Volume 4, Issue 8, August 2014

[2] SwagatamDas,Ponnuthurai N. Suganthan-Differential Evolution: A Survey of the State-of-the-Art,4 IEEE Transactions on Evolutionary Computation, Vol. 15, no. 1, February 2011.

[3] http://www.ni.com/white-paper/3782/en/

[4] Pradeep K. Mohanty, Binod Kumar Sahu, Sidhartha Panda - Tuning and Assessment of Proportional–Integral–Derivative Controller for an Automatic Voltage Regulator System Employing Local Unimodal Sampling Algorithm, Electric Power Components and Systems, Vol. 42 May 2014.

[5] J. Rönkkönen, S. Kukkonen, K.V. Price, "Real-parameter optimization with differential evolution", IEEE Congress on Evolutionary Computation, 2005, pp. 506–513.

[6] Shuguang Zhao, Xuwang, Liang Chen, Wu Zhu, " A Novel self - adaptive differential evolution algorithm with population size adjustment scheme" Arabian Journal for Science and Engineering, August 2014, Vol. 39, Issue 8, pp: 6149-74

[7] Abhijit Chandra, Sudipta Chattopadhyay, " Role of mutation strategies of differential evolution algorithm in designing hardware efficient multiplier - less Low pass FIR filter", Journal of Multimedia, October 2012, Vol. 7, No. 5 pp: 353-63.

[8] R. Mallipeddi, P. N. Suganthan, Q. K. Pan, M. F. Tasgetiren, " Differential evolution algorithm with ensemble of parameters and mutation strategies", Applied Soft Computing, March 2011, Vol. 11, Issue 2, pp: 1679 – 96

[9] A.K. Qin, V.L. Huang, P.N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization", IEEE Transactions on Evolutionary Computation, April 2009, Vol. 13, No. 2 pp: 398–417.

[10] Storn R, Price K. "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces", International Journal of Global Optim 1997, Vol. 11, Issue. 4, pp: 341–59.

[11] Mohammad Sadegh Rahimian and Kaamran Raahemifar, "Optimal PID controller Design from AVR system using PSO Algorithm" IEEE CCECE 2011.

[12] Zwe-Lee Gaing, "A Particle Swarm Optimization Approach for Optimum Design of PID Controller in AVR System", IEEE Transactions on Energy Conversion, Vol. 19, No. 2, pp.384-391, June 2004.