Design and Comparative Performance Analysis of PID Controlled Automatic Voltage Regulator tuned by Many Optimizing Liaisons

B. K. Sahu, P. K. Mohanty, S. Panda, S. K. Kar, N. Mishra

Abstract—**This paper deals with the design of Proportional, Integral, and Derivative (PID) controller to an Automatic Voltage Regulator (AVR) tuned by recently developed Simplified Particle Swarm Optimization algorithm so called, Many Optimizing Liaisons (MOL) algorithm. MOL simplifies the original PSO by randomly choosing the particle to update, instead of iterating over the entire swarm thus eliminating the particle's best known position and making it easier to tune the behavioural parameters. The proposed method is compared with the earlier used PSO algorithm. For performance studies; Transient response analysis, Bode plot analysis and Root locus analysis are explained in details. The robustness analysis is done by varying the time constants of amplifier, exciter, generator & sensor in the range of -50% to + 50% with a step size of 25% respectively. The results of these analyses using the MOL algorithm are found to be better with respect to the analysis of the PID controller using PSO algorithm.**

Index Terms—**Automatic Voltage Regulator, Many Optimizing Liaison, Proportional Integral and Derivative, Particle Swarm Optimization.**

I. INTRODUCTION

IN a power system network, all the equipments are designed for rated voltage. Any considerable changes or deviation in the voltage levels considerably reduces the durability of the equipment also hampers the dynamics of the system. Moreover, as known that the system voltage is affected by the reactive power flow this in turn is affected by the real line losses.

Manuscript received April 28, 2012.

Binod Kumar Sahu, Pradeep Kumar Mohanty, Sanjib Kumar Kar and Nandan Mishra are with Department of Electrical Engineering, ITER, S'O'A University, Bhubaneswar, odisha, India. (e-mail: binoda@rediffmail.com, +919937404248 pkmohanty68@rediffmail.com, skkariter@rediffmail.com, aksanebs@gmail.com)

Sidhartha Panda is with Department of Electrical and Electronics Engineering, VSSUT, Burla, Odisha, India (e-mail: sidharthapanda@rediffmail.com)

So, in order to minimize the real power losses, the system voltage level has to be controlled to a great extent. These problems can be solved by using an Automatic Voltage Regulator (AVR) system applied to the power generating units [1]. The terminal voltage and the reactive power of the alternator are controlled by the AVR which in turn, also manages the proper sharing of the reactive power among all the alternators, connected in parallel. Generally, the AVR is a closed loop control system which is compensated with a PID controller and maintains the terminal voltage at a desired level. PID controllers have a wide range of applications in the process control as well as instrumentation industries due to their easy installation, simplicity in structure and robustness in their performances. A PID controller consists of three control parameters i.e. proportional gain, integrating gain, derivative gain respectively. Till date, many researchers have suggested various tuning techniques to determine the PID parameters. Of them, the conventional methods are Zeigler/Nichols method minimum variance method, predictive method, Cohen/Coon pole placement method. But these conventional techniques have a lot of disadvantages: poor dynamics of the close loop systems, extensive rules to set gains, complexity of the control designs and difficulty to deal with non linear systems. Recently, many researchers have proposed different algorithms to tune the parameters of the PID controller. These are: simulated annealing [2], evolutionary algorithm [3], genetic algorithm [4], fuzzy systems [5], multi-objective optimization [6], artificial bee colony (ABC) [7], etc. But in recent times Particle Swarm Optimization (PSO) has come up in a more effective and better way than all other algorithms for optimization purposes. PSO is a population based stochastic optimization technique, inspired by the social behaviour of bird flocking or fish schooling [8]. PSO uses the optimization techniques, like: initialising the population of random solutions and searching for the optimal solution by updating the generations which is in turn, quite similar to that of the Genetic Algorithm(GA) but PSO does not employs evolution operators such as mutation and crossover. Along with it the PSO has less parameter, thus easy to install which can be said to have more merits over the GA. In PSO, the potential solutions called particles; fly through the problem space by following the current optimum particles. As suggested by various researchers, the PSO method can be made better by making the algorithm more complex and redundant to increase its applications to handle complex optimization problems. However, a simplified PSO called "social only" was suggested by Kennedy. Later, Pederson and

978-1-4673-2043-6/12/$31.00 ©2012 IEEE

Chipperfield [9] made extensive research and called it as Many Optimizing Liaisons (MOL) which resulted in a better and improved performance. Using this algorithm, the parameters of PID controller are tuned-in with a more improvised way. Thus, MOL can be compared to that of PSO as it is used in eliminating the particle's best known position and thus, converting the complex algorithm into a simpler and easier one.

The most important aim and objective of this whole work is the designing and implementation of an efficient MOL algorithm based PID controller to control the voltage in an autonomous power generating station, in an intelligent and robust way. The proposed controller's performance has been analysed by employing various analysis techniques such as transient response analysis, bode plot analysis, root locus analysis and respectively. The supremacy of this algorithm is shown by comparing it with other published heuristic optimization algorithm, PSO algorithm. The robustness of the proposed AVR system is checked by varying the time constants of amplifier, exciter, generator and sensor.

II. DESCRIPTION AND MODELLING OF AN AVR SYSTEM

The basic and simple arrangement of AVR consists of: amplifier, sensor, exciter and the generator. The generator's terminal voltage is continuously being sensed by the voltage sensor .This signal, in turn is rectified and smoothened and thus, compared with the reference signal in the comparator. The amplifier amplifies the voltage from comparator and eventually feeds to the field windings of the exciter of the generator. The four components used are supposed to be linear for mathematical modelling of the transfer function. The transfer functions [10], the range of various gains and their actual values of the above components are given in table I.

TABLE I
TRANSFER FUNCTIONS AND PARAMETERS OF THE AVR SYSTEM [7]

Elements	Transfer function	Range of Gains	Range of time constants	Gain	Time Constants
Amplifier	$TF_A = \frac{K_A}{1+sT_A}$	10≤K_a≤40	0.02≤T_a≤0.1	K_a = 10	T_a = 0.1
Exciter	$TF_E = \frac{K_E}{1+sT_E}$	1≤K_e≤10	0.4≤T_e≤1.0	K_e = 1.0	T_e = 0.4
Generator	$TF_G = \frac{K_G}{1+sT_G}$	0.7≤K_g≤1.0	1.0≤T_g≤2	K_g = 1.0	T_g = 1.0
Sensor	$TF_S = \frac{K_S}{1+sT_S}$	K_s=1.0	0.001≤T_s≤0.06	K_s = 1.0	T_s = 0.01

The complete transfer function model of the AVR system without controller is shown in fig 1.

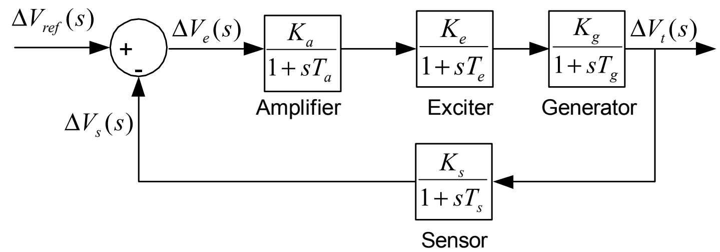

Fig.1. Transfer function model of AVR system without controller

Thus, the transfer function of the AVR system with above mentioned parameters is:

$$\frac{\Delta V_t(s)}{\Delta V_{ref}(s)} = \frac{0.1s+10}{0.0004s^4+0.0454s^3+0.555s^2+1.51s+11} \quad (1)$$

With the application of a step input signal to the above system, its response is shown in fig. 2. It is clear from fig.2 that, the system is oscillatory stable with two real poles located at $s = -100$ *and* $s = -12.5$ and two complex poles at $s = -0.52 \pm 4.66i$. The terminal voltage reaches a steady value of 0.91 p.u. at about 10 sec which is not acceptable.

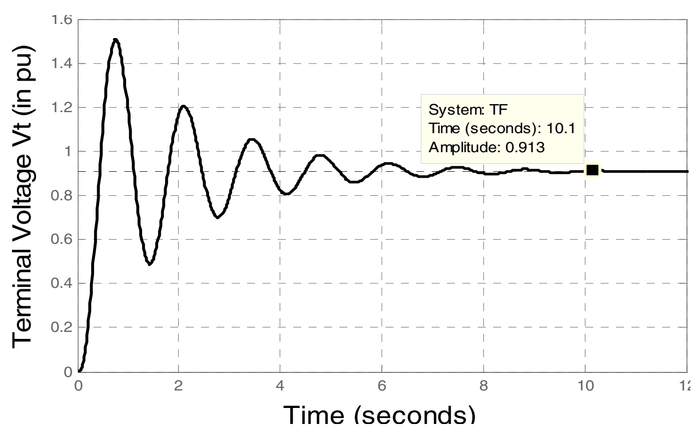

Fig.2. Response of AVR system without PID controller

The dynamic response of AVR system can be improved and the terminal voltage can be maintained at 1.0 pu (per unit) by using PID controller as shown in fig.3.With the use of PID controller, the transfer function of the AVR system as shown in fig.4 becomes

$$\frac{\Delta V_t(s)}{\Delta V_{ref}(s)} = \frac{0.1K_d s^3+(0.1K_p+10K_d)s^2+(0.1K_i+10K_p)s+10K_i}{0.0004s^5+0.0454s^4+0.555s^3+(1.51+10K_d)s^2+(1+10K_p)s+10K_i} \quad (2)$$

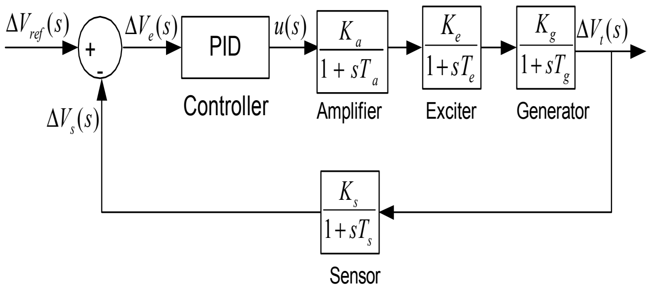

Fig.3. Transfer function model of the AVR system with PID controller.

III. PROPORTIONAL INTEGRAL DERIVATIVE (PID) CONTROLLER

Till date from half of this century, it is the most popular feedback controller which is being used in the process industries. This can be considered as an excellent controller that can help in providing excellent performance of the process plant. The PID controller consists of three basic modes: proportional, integral, derivative modes respectively. A proportional controller gain (K_p) reduces the rise time but does not eliminate the steady-state error, integral gain (K_i) eliminates the steady state error but resulting a worse transient response, derivative gain (K_d) increases the stability of the system and improves the transient response and reduces overshoot. These values of above gains are obtained by hit and trial method based on the plant behaviour and experiences. The equation below represents the transfer function of PID controller:

$$TF_{PID} == K_p + \frac{K_i}{s} + K_d s \qquad (3)$$

IV. MANY OPTIMIZING LIAISONS (MOL) ALGORITHM USED FOR AVR PARAMETER TUNING

Many optimizing liaisons (MOL) algorithm can be said a simple form of Particle Swarm Optimization (PSO) algorithm.PSO algorithm was developed by Kennedy and Eberhart in 1995 [8]. In this algorithm, all particles are placed at random positions and are supposed to move in random directions in search space. Then, each particle is instigated to change its direction and move along a path of the best previous positions of itself and its neighbours, searching for a better position with respect to fitness measures$f: \Re^n \rightarrow \Re$.

Let $\vec{X} \in \Re^n$ be the position of a particle and the $\vec{V}$ be the velocity of the particle respectively. The position of the particle and the initial velocity of the particle are chosen randomly and get updated each time by iteration. The velocity of the particle can be updated by a formula given by Shi and Eberhart [11];

$$\vec{V} \leftarrow \omega\vec{V} + \emptyset_P R_P(\vec{R}_P - \vec{X}) + \emptyset_G R_G(\vec{G} - \vec{X}) \qquad (4)$$

In the formula; $w \in \Re$ can be said as a user-defined behavioural parameter which can be termed as inertia weight which controls the number of repetition in the velocity of particles.$\vec{P}$ and $\vec{G}$ are the best positions of the particles and swarm respectively and are measured by stochastic variables $R_P, R_G \sim U(0,1)$. ϕ_P, ϕ_G are the behavioural parameters designed by the user. The velocity is summed up with the current position of particle to move to a newer position in search space

$$\vec{X} \leftarrow \vec{X} + \vec{V} \qquad (5)$$

After the particle's position update, limitations get imposed on the particle which can now only move from one search space to another in a single step.

The steps of a PSO algorithm are:

a) Randomly, initialize the positions and velocities of each particle.
b) Update the position and velocity of each particle.
c) Update the personal and global best.
d) Find the velocity of a new particle using (4).
e) Using (5) move the particle to a new position.
f) Enforce search-space boundaries.
g) Update the particle's best position ,if $f(\vec{X}) < f(\vec{P})$
h) The above steps are repeated for swarm's best position $(\vec{G})$.

By setting the swarm's best position as ϕ_P=0 in PSO algorithm, the best position $(\vec{G})$ is made to eliminate. ω (the inertia weight) is initialized as 0 to eliminate the recurrence in (4), thus, making the particle not to have any persistence in the previously followed path.

As the value of ϕ_P is set as zero in MOL algorithm, this helps in eliminating the previous best position $(\vec{P})$ values of the particle. Hence, the velocity is updated and (4), turns into:

$$\vec{V} = w\vec{V} + \emptyset_G R_G (\vec{G} - \overrightarrow{X}) \qquad (6)$$

where $\vec{X}$ is the current position of the particle which gets updated as similar to that of PSO method and $\vec{G}$ is the best position of the entire swarm.

Thus, summing up it can be said that MOL algorithm is quite similar to that of PSO algorithm but PSO algorithm gets updated with each iteration over the whole swarm rather MOL algorithm updates the particle randomly. So, MOL algorithm can be called as a "simplified PSO" or "Social Only PSO".

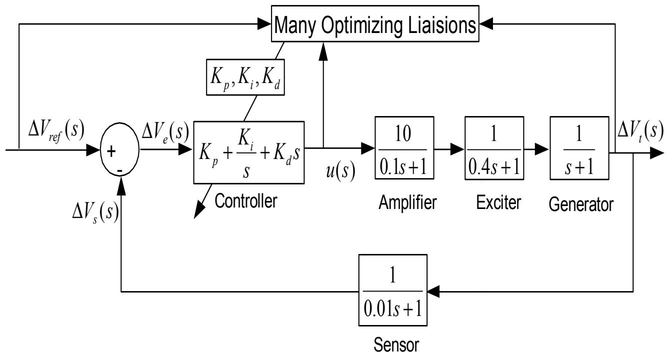

Fig.4. Transfer function model of AVR system with PID controller tuned by MOL algorithm.

V. SIMULATION RESULTS AND DISCUSSION

MOL algorithm is used to tune the parameters of PID controller. The lower and upper limits of the parameters are taken as [0.01, 2].The objective function '*OF*' used to determine the optimum values of PID controller parameters is given in as

$$OF = \alpha_1 \times ITAE + \alpha_2 \times T_S + \alpha_3 \times M_P \quad (7)$$

Where ITAE is the Integral Time Absolute Error.

$$ITAE = \int |V_r - V_t|\, tdt \quad (8)$$

Here V_r is the reference voltage and V_t is the terminal voltage. T_S is the settling time(2% band) and M_P is the peak overshoot. α_1, $\alpha_2$and α_3 are various weight factors having values 0.5,0.25and 0.25 respectively. The optimum values of PID controller parameters are obtained by running the simulations for 20 times and the parameters are given in Table II.

TABLE II
PARAMETERS OF PID CONTROLLER WITH THE OBJECTIVE FUNCTIONS GIVEN IN (7) USING MOL ALGORITHM

K_p	K_i	K_d
0.5473	0.3556	0.1668

The transfer function of the system with the above parameters is:

$$\frac{\Delta V_t(s)}{\Delta V_{ref}(s)} = \frac{0.01668\, s^3 + 1.723\, s^2 + 5.509s + 3.556}{0.0004s^5 + 0.0454s^4 + 0.555s^3 + 3.178s^2 + 6.473\, s + 3.556} \quad (9)$$

The performance of AVR system with PID controller tuned by MOL algorithm is analysed using transient response analysis, bode analysis and root locus analysis. The robustness of the system is also studied by varying the different time constants from -50% to +50% in steps of 25%.

A. Transient Response Analysis

The figure 5 shows the transient response analysis of AVR system tuned by MOL algorithm with different weightage factors and PSO algorithm [7].

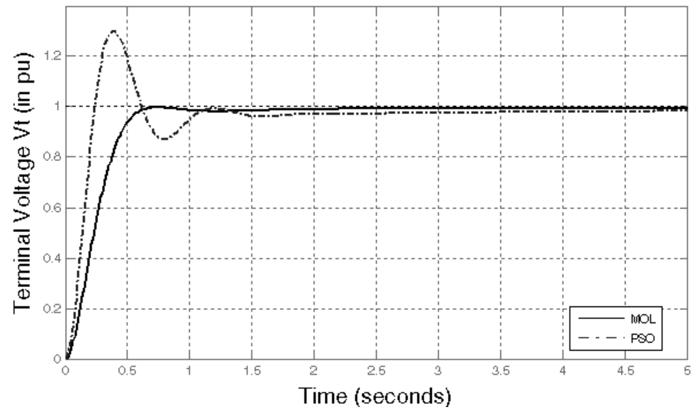

Fig.5. Transient response analysis of AVR system tuned by MOL algorithm with different weightage and PSO algorithm

It has been observed that the peak overshoot in MOL algorithm is 30 % less than that of PSO algorithm and the settling time (2 % band) in MOL algorithm is 478% less than that of PSO algorithm. The results of transient response analysis are depicted in Table III. The maximum overshoot and settling time for MOL algorithm are better than PSO algorithm which are the major factors for comparing stability analysis of a system [12].

TABLE III
TRANSIENT RESPONSE ANALYSIS

Different Algorithms	**Peak overshoot (in pu)**	**Settling time (in sec)**	**Rise time (in sec)**	**Peak time (in sec)**
MOL	**0.9997**	**0.5864**	0.3744	0.7451
PSO [7]	1.3005	3.3890	**0.1609**	**0.3908**

B. Bode Analysis

The Magnitude and Phase plot of the AVR system is shown in fig.6. The peak gain, phase margin, delay margin and the bandwidth obtained from Bode Plot are given in Table IV. It has been observed that the best frequency response is obtained by using MOL algorithm.

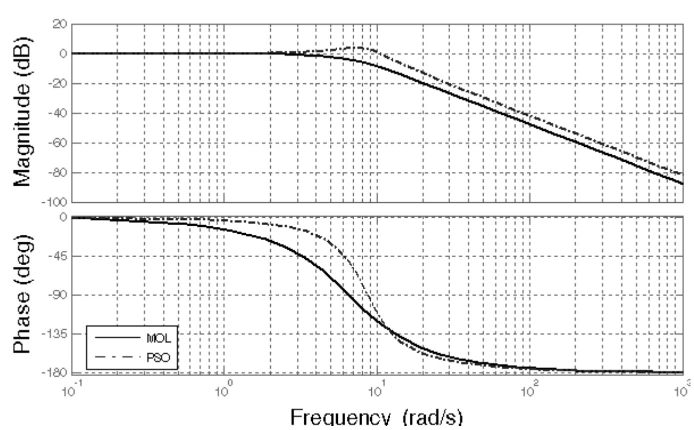

Fig.6. Bode Plot Analysis of AVR system tuned by MOL algorithm with different weightage and PSO algorithm

TABLE IV
BODE ANALYSIS

Different Algorithm	Peak Gain (in dB)	Phase Margin (in deg)	Delay Margin (in sec)	Bandwidth (in Hz)
MOL	**0.0**	**-180**	**Inf.**	**5.9023**
PSO [7]	3.75	62.1906	0.1034	12.1825

C. Root Locus Analysis

The Root Locus curves for MOL algorithm and PSO algorithm is shown in fig. 7. It is seen that in the closed loop poles for both the algorithms lies to the left of the s-plane. Hence, AVR systems tuned by both the algorithms are stable. The closed loop poles and the damping ratios of the system tuned by both the algorithms are given in table V. The conjugate poles for MOL algorithm are located away from the imaginary axis as compared to PSO algorithm which increases the stability of the system tuned by MOL algorithm.

The damping ratio of the AVR system tuned by MOL algorithm is 50.4% more than that of the system tuned by PSO algorithm.

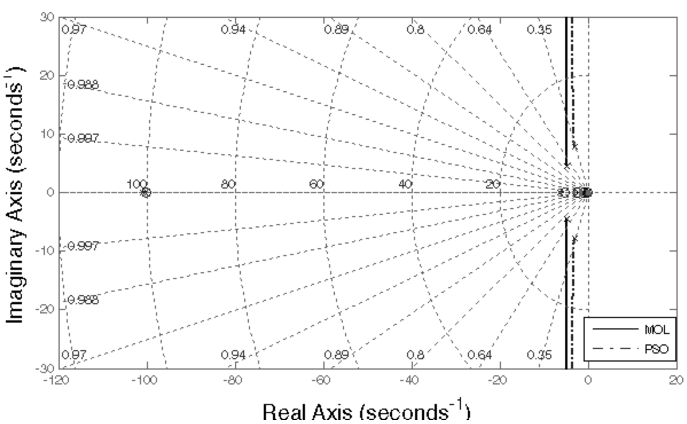

Fig.7. Root Locus Analysis of AVR system tuned by MOL algorithm with different weightage and PSO algorithm

TABLE V
POSITION OF CLOSED LOOP POLES AND DAMPING RATIO OF THE AVR SYSTEM TUNED BY MOL & PSO ALGORITHM

MOL		PSO [7]	
Closed Loop Poles	***Damping Ratio***	***Closed Loop Poles***	***Damping Ratio***
-100	1	-101	1
-2.32	1	-6.26	1
-0.865	1	-0.215	1
-4.93+4.45i	0.742	-3.09+7.8i	0.368
-4.93-4.45i	0.742	-3.09-7.8i	0.368

D. Robustness Analysis

Robustness of the AVR system tuned by MOL algorithm is examined by varying the time constants of amplifier, exciter, generator and sensor in the range of -50% to +50% in step of 25%.The results obtained from robustness analysis are depicted in table VI. The responses of AVR system tuned by MOL algorithm by varying the time constants are shown in fig.8, fig.9, fig.10, & fig.11. The range of deviations in rise time, settling time, peak value and peak time is being depicted in table.VII. The average deviation of peak value is 3.81 % and for settling time (2 % band), rise time and peak time are 49 %, 15.91 % and 81.3 % respectively. The ranges of total deviation are within limit, and therefore, AVR system with PID controller tuned by MOL algorithm is robust.

TABLE VI
RESULTS OF ROBUSTNESS ANALYSIS OF PID CONTROLLER TUNED BY MOL ALGORITHM

Parameter	Rate of Change (in %)	Rise Time (in sec)	Settling Time(2% band) (in sec)	Peak Value (in pu)	Peak Time (in s)
T_a	-50	0.4417	1.1398	0.9989	4.1358
	-25	0.3936	0.9173	0.9993	4.6641
	+25	0.3706	0.8483	1.0276	0.7394
	+50	0.3733	1.0047	1.0552	0.7551
T_e	-50	0.2833	1.7639	0.9962	3.2274
	-25	0.3291	1.5125	0.9982	3.7786
	+25	0.4161	1.1332	1.0330	0.8875
	+50	0.4545	1.5592	1.0621	1.0001
T_g	-50	0.2127	2.4560	1.0160	0.3928
	-25	0.2942	1.9153	0.9978	4.5855
	+25	0.4505	0.6919	1.0159	1.0329
	+50	0.5213	2.2457	1.0371	1.2960
T_s	-50	0.3845	0.6146	0.9976	3.2104
	-25	0.3794	0.6001	0.9977	3.2485
	+25	0.3695	0.5736	1.0022	0.7335
	+50	0.3647	0.5615	1.0049	0.7218

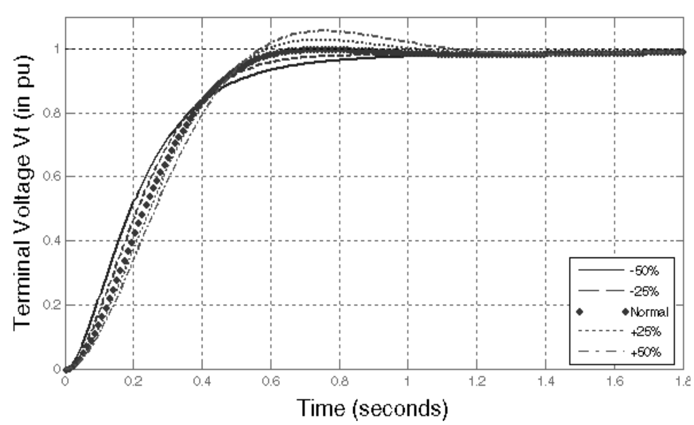

Fig.8. Voltage change curves ranging from -50% to +50% for T_a

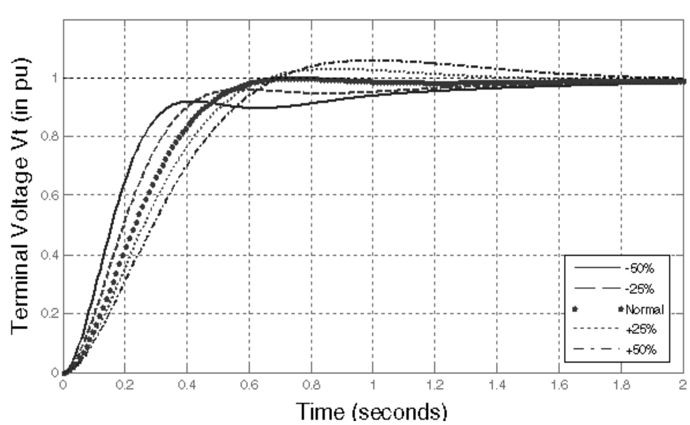

Fig 9.Voltage change curves ranging from -50% to +50% for T_e

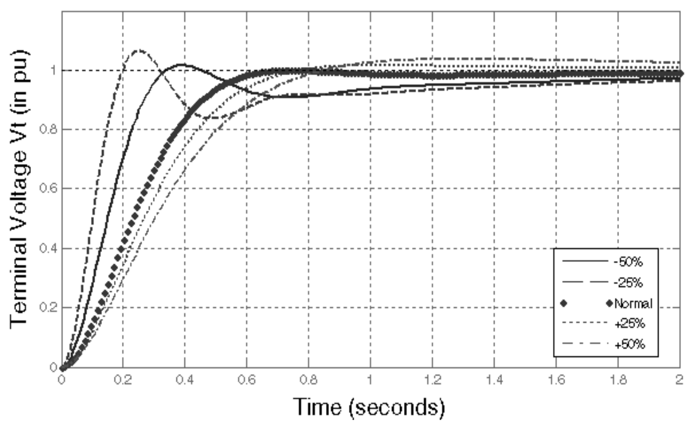

Fig.10. Voltage change curves ranging from -50% to +50% for T_g

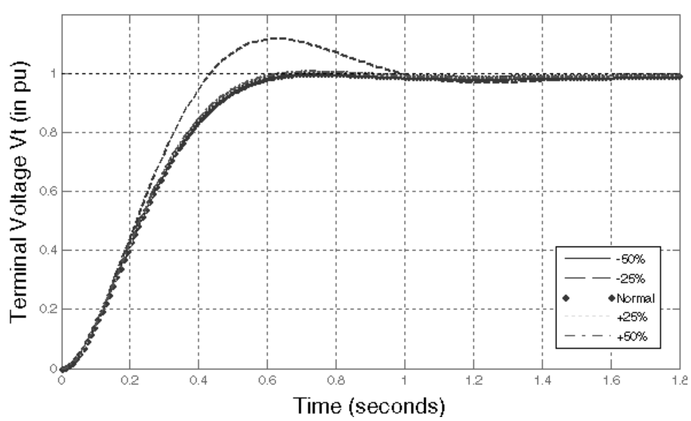

Fig.11. Voltage change curves ranging from -50% to +50% for T_s

TABLE VII
RANGE OF TOTAL DEVIATIONS AND PERCENTAGE OF MAXIMUM DEVIATION OF THE SYSTEM TUNED BY MOL ALGORITHM

Time Constants	Parameter	Range of Total Deviation	Percentage of Total Deviation (in %)
T_a	Rise Time(sec)	0.0711	15.23
	Settling Time(sec)	0.2915	48.55
	Peak (pu)	0.0563	5.26
	Peak time (sec)	3.9247	84.02
T_e	Rise Time(sec)	0.1712	17.62
	Settling Time(sec)	0.6307	66.75
	Peak (pu)	0.0659	5.87
	Peak time (sec)	2.8911	80.28
T_g	Rise Time(sec)	0.3086	28.18
	Settling Time(sec)	1.7641	76.12
	Peak (pu)	0.0393	3.6
	Peak time (sec)	4.1927	83.75
T_s	Rise Time(sec)	0.0198	2.62
	Settling Time(sec)	0.0531	4.6
	Peak (pu)	0.0073	0.52
	Peak time (sec)	2.5267	77

VI. CONCLUSION

The objective of this paper is to obtain the optimum values of PID controller parameters using MOL algorithm for an AVR system. The performance of the proposed algorithm is compared with PSO algorithm. Considering different testing parameters we can say that, PID controlled AVR system tuned by MOL algorithm can satisfactorily and successfully provides better transient response as well as quite robust. The above studies of AVR system validities the proposed MOL algorithm.

REFERENCES

[1] P.Kundur, Power system Stability and Control, TMH,8^{th} reprint 2009.

[2] Ho SJ, Shu LS, Ho SY, "Optimizing fuzzy neural networks for tuning PID controllers using an orthogonal; simulated annealing algorithm OSA", IEEE Trans Fuzzy System 2006;14(3):421-34.

[3] Vlachos C, Williams D, Gomm JB, "Solution to the Shell Standard control problem using genetically Tuned PID controllers", Control Engg. Practr (2002;1092):151-63

[4] 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.

[5] A. Visioli, "Tuning of PID controllers with fuzzy logic," Proc. Inst.Elect. Eng. Contr. Theory Applicat., vol. 148, no. 1, pp. 1–8, Jan. 2001.

[6] Li B, Jiang W "Optimizing complex functions by Chaos Search" Cybernet System 1998;29(4):409-19.

[7] H.Gozde, M.C. Taplamacioglu. "Comparative Performance Analysis of Artificial Bee Colony Algorithm for Automatic Voltage Regulator (AVR) System", Journal of Franklin Institute 348 (2011), 1927-1946.

[8] J.Kennedy, R. C. Ebenhart, "Particle swarm optimization" in: Proceedings of IEEE Conference on Neural Networks, Perth, Australia, 1995, pp.1942-1948.

[9] M.E.H. Pedersen, A.J. Chipperfield, Simplifying Particle Swarm Optimization, Applied Soft Computing, 10(2010) 618-628.

[10] Hadi Sadat, Power System Analysis, TMH 17^{th} reprint 2009.

[11] Y. Shi, R. Eberhart, A Modified Particle Swarm Optimizer, IEEE International Conference on Evolutionary Computation, Anchorage, AK, USA, 1998.

[12] K.Ogata, Modern Control Engineering, Fourth Edition, Prentice-Hall Inc., USA, 2002.