Available online at www.sciencedirect.com

SciVerse ScienceDirect

Journal of the Franklin Institute 349 (2012) 2609–2625

Journal of The Franklin Institute

www.elsevier.com/locate/jfranklin

Design and performance analysis of PID controller for an automatic voltage regulator system using simplified particle swarm optimization

S. Panda[a,*], B.K. Sahu[b], P.K. Mohanty[b]

[a]Department of Electrical and Electronics Engineering, Veer Surendra Sai University of Technology (VSSUT), Burla 768018, Odisha, India

[b]Department of Electrical Engineering, Institute of Technical Education and Research (ITER), SOA University, Bhubaneswar 751030, Odisha, India

Received 18 January 2012; received in revised form 24 June 2012; accepted 26 June 2012
Available online 3 July 2012

Abstract

This paper presents the design and performance analysis of Proportional Integral Derivate (PID) controller for an Automatic Voltage Regulator (AVR) system using recently proposed simplified Particle Swarm Optimization (PSO) also 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 particles best known position and making it easier to tune the behavioral parameters. The design problem of the proposed PID controller is formulated as an optimization problem and MOL algorithm is employed to search for the optimal controller parameters. For the performance analysis, different analysis methods such as transient response analysis, root locus analysis and bode analysis are performed. The superiority of the proposed approach is shown by comparing the results with some recently published modern heuristic optimization algorithms such as Artificial Bee Colony (ABC) algorithm, Particle Swarm Optimization (PSO) algorithm and Differential Evolution (DE) algorithm. Further, robustness analysis of the AVR system tuned by MOL algorithm is performed by varying the time constants of amplifier, exciter, generator and sensor in the range of −50% to +50% in steps of 25%. The analysis results reveal that the proposed MOL based PID controller for the AVR system performs better than the other similar recently reported population based optimization algorithms.

*Corresponding author. Tel.: +91 9438251162.
E-mail addresses: panda_sidhartha@rediffmail.com, sidharthapandas@yahoo.com (S. Panda).

0016-0032/$32.00 © 2012 The Franklin Institute. Published by Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.jfranklin.2012.06.008

1. Introduction

All the equipments that are connected in a power system network are designed for nominal or rated voltage. Any deviation in voltage levels causes considerable changes in the system dynamics leading to decrease in the performance and life span of the connected equipments. Also the real line losses depend on the reactive power flow and the reactive power depends upon system voltage. Hence to minimize the real power losses the system voltage levels must be controlled. To overcome these problems, an Automatic Voltage Regulator (AVR) system is applied to power generation units [1]. The AVR of the alternator controls the terminal voltage and reactive power and also ensures proper sharing of the reactive power amongst all the generators connected in parallel. The AVR system is a closed loop control system compensated with a Proportional Integral Derivate (PID) controller that maintains the terminal voltage at the desired level. PID controllers are widely used in the process control and instrumentation industries because of its simple structure, easy implementation and providing robust performance in a wide range of operating conditions. PID controllers have three control parameters, i.e. proportional gain, integrating gain and derivative gain which are to be tuned properly. The conventional tuning techniques reported in literature includes, Ziegler/Nichols method, gain-phase margin method, Cohen/Coon pole placement, minimum variance, gain scheduling and predictive. However, the disadvantages of these tuning techniques are: extensive rules to set the gains, poor dynamics of closed loop responses, difficulty to deal with nonlinear systems and complexity of the control design. In recent years, many artificial intelligence algorithms are proposed to tune the controller parameters. These approaches include Simulated Annealing (SA) [2], Tabu Search Algorithm [3], Differential Evolution (DE) algorithm [4,5], evolutionary algorithms [6], Genetic Algorithm (GA) [7], fuzzy systems [8], Artificial Bee Colony (ABC) [9], Particle Swarm Optimization (PSO) [10] and multi-objective optimization [11,12]. Recently, to determine the PID parameters suitably, various tuning schemes have been reported by many researchers. Iruthayarajan and Baskar [13] proposed a novel covariance matrix adaptation evolution strategy algorithm to design a centralized multivariable PID controller for a binary distillation column plant described for two inputs–two outputs system and three inputs–three outputs system. Menhas et al. [14] analyzed the comparative performance of various binary coded Particle Swarm Optimization (PSO) algorithms on optimal proportional-integral (PI) and PID controller design for multiple inputs multiple outputs (MIMO) systems. Iruthayarajan and Baskar [15] have compared the performance of various evolutionary algorithms (EAs) such as real coded genetic algorithm (RGA), modified particle swarm optimization (MPSO), covariance matrix adaptation evolution strategy (CMAES) and differential evolution (DE) on optimal design of multivariable PID controller. Zhao et al. [16] have applied two local best multi-objectives PSO (2LB-MOPSO) to design multi-objective robust PID controllers for two MIMO systems. Sumar et al. [17] have designed a PID controller, based on the universal model of the plant, in which there is only one parameter to be tuned. Coelho and Pessoa [18] developed a technique for tuning of decoupled PI and PID multivariable controllers based on a chaotic differential evolution (DE) approach. Coelho and Bernert [19] have proposed the tuning of a PID controller using a modified Tribes optimization algorithm based on truncated chaotic Zaslavskii map for synchronization of two identical discrete chaotic systems subjected to different initial conditions. Coelho [20] has proposed a tuning method for determining the parameters of PID control for an automatic regulator voltage (AVR) system using a chaotic optimization approach based on Lozi map. Coelho and Grebogi [21] have presented the synchronization of two identical discrete chaotic systems subject to the different initial conditions by designing a Proportional-Integral-Derivative (PID) controller. Menhas et al. [22] have presented a new variant of binary particle swarm optimization (PSO) algorithm; called probability based binary PSO (PBPSO), to tune the parameters of a coordinated controller.

PSO is a population based stochastic optimization technique, inspired by social behavior of bird flocking or fish schooling [23]. PSO shares many similarities with GA optimization technique; like initialization of population of random solutions and search for the optimal by updating generations. However, unlike GA, PSO has no evolution operators such as crossover and mutation. One of the most promising advantages of PSO over GA is its algorithmic simplicity as it uses a few parameters and easy to implement. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles. Various studies exist on how to make the PSO method perform better by making the algorithm more complex to increase its application to complex optimization problems [24,25]. However, a simplified PSO called ''social only'' was suggested by Kennedy [26]. Later, Pedersen and Chipperfield [27] made extensive studies and named it Many Optimizing Liaisons (MOL) which was found to give somewhat improved performance and also makes it easier to tune the behavioral parameters. MOL differs from PSO in that it eliminates the particle's best known position thus making the algorithm simpler.

The main objective of this work is to design and implement an efficient MOL-PID controller for intelligent control of voltage in an autonomous power generating system. In the proposed approach, the design problem is formulated as an optimization problem control and MOL is employed to search for optimal controller parameters. For artificial intelligence algorithms based controller design, selection of suitable objective function is very important. In this regard, four different objective functions namely; the integral of time multiplied by absolute value of error (ITAE), integral of absolute value of error (IAE), integral of time multiplied by squared error (ITSE) and integral of squared error (ISE) are considered. The effect of these objective functions on the performance of the system in terms of maximum overshoot, settling time, rise time and peak time are analyzed. The performance of the proposed controller has also been done employing various analysis methods such as transient response analysis, root locus analysis and bode analysis. Further, the analysis results are compared with some recently published modern heuristic optimization algorithms such as Artificial Bee Colony (ABC) algorithm, Particle Swarm Optimization (PSO) algorithm and Differential Evolution (DE) algorithm to show the superiority of proposed technique and objective function. Also, robustness analysis of the proposed AVR system is performed by varying the time constants of amplifier, exciter, generator and sensor.

2. Description and modeling of an AVR system

Excitation control of synchronous alternator is one of the most important factors to improve power system stability and quality of electrical power. The function of AVR is to maintain the magnitude of terminal voltage of a synchronous alternator at its nominal value. A simple AVR system comprises four main components, namely sensor, amplifier, exciter and generator. The terminal voltage of generator is continuously sensed by a voltage sensor. This signal is rectified, smoothed and compared with a reference signal in the comparator. The error voltage obtained from the comparator is amplified and is used to control the field windings of the generator by means of the exciter. For mathematical modeling transfer function of the above four components are supposed to be linear, which takes into relation the major time constant and ignores the saturation or other non-linearities. The transfer functions of the above components are given below [1,28].

2.1. Amplifier model

Transfer function of an exciter is modeled by a gain and a time constant given by

$$TF_A = \frac{K_a}{1 + sT_a} \quad (1)$$

where K_a and T_a are the gain and time constant of the amplifier system. Usual values of K_a are in the range of 10–40 and the amplifier time constant T_a is very small ranging from 0.02 to 0.1 s.

2.2. Exciter model

Transfer function of an exciter modeled by a gain of K_e and a time constant of T_e is given by

$$TF_E = \frac{K_e}{1 + sT_e} \quad (2)$$

Typical values of K_e are in the range of 1–10 and the time constant T_e is in the range of 0.4–1.0 s.

2.3. Generator model

The generator is represented by a transfer function given by

$$TF_G = \frac{K_g}{1 + sT_g} \quad (3)$$

The generator gain K_g and time constant T_g are load dependent. K_g varies between 0.7 and 1.0, and T_g between 1.0 and 2.0 s from full load to no load.

2.4. Sensor model

A sensor may be represented by a simple first-order transfer function with a gain K_s and time constant T_s and is given by

$$TF_S = \frac{K_s}{1 + sT_s} \quad (4)$$

Normally T_s is very small, ranging from 0.001 to 0.06 s and K_s is around 1.0.

The complete transfer function model of an AVR system (without controller) is shown in Fig. 1.

In Gozde and Taplamacioglu [9], the parameters of the AVR system are taken as: K_a=10.0, T_a=0.1, K_e=1.0, T_e=0.4, K_g=1.0, T_g=1.0, K_s=1.0 and T_s=0.01. The same parameters have been taken in this work.

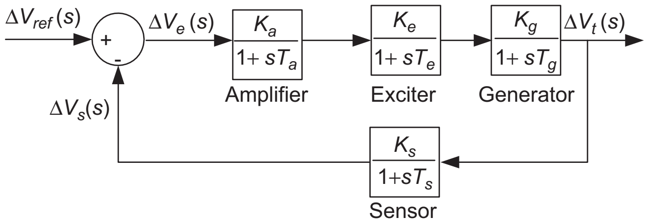

Fig. 1. Transfer function model of the AVR system without controller.

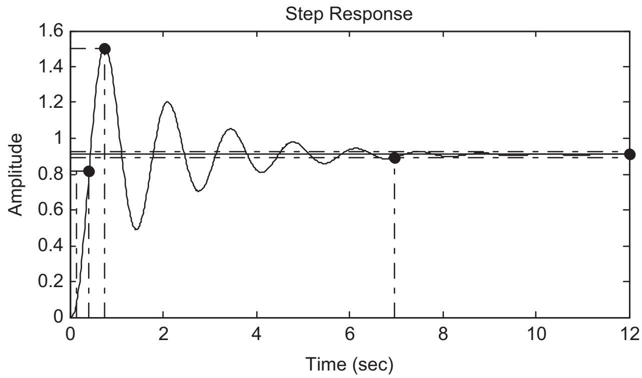

Fig. 2. Response of the AVR system without PID controller.

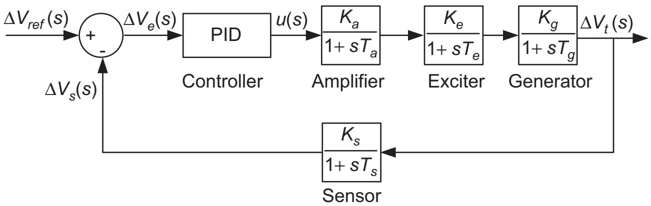

Fig. 3. Transfer function model of the AVR system with PID controller.

The transformer function of the AVR system with above parameters is

$$\frac{\Delta V_t(s)}{\Delta V_{ref}(s)} = \frac{0.1s+10}{0.0004s^4+0.045s^3+0.555s^2+1.51s+11} \tag{5}$$

The terminal step voltage response of the above system is shown in Fig. 2 from which it is clear that the system is oscillatory stable with two real poles at $s=100$ and -12.5 and two complex poles at $s=-0.52\pm4.66i$. The system has a rise time of 0.261 s, peak amplitude of 1.5 (65.4% overshoot) at $t=0.755$ s, settling time of 6.97 s and steady state value of 0.909.

To improve the dynamic response of the AVR system and to maintain the terminal voltage at 1.0 pu a PID controller is included as shown in Fig. 3. With PID controller the transfer function of the AVR system of Fig. 3 becomes

$$\frac{\Delta V_t(s)}{\Delta V_{ref}(s)} = \frac{0.1K_d s^2 + (0.1K_p + 10K_d)s^2 + (0.1K_i + 10K_p)s + 10K_i}{0.0004s^5 + 0.045s^4 + 0.555s^3 + (1.51 + 10K_d)s^2 + (1 + 10K_p)s + 10K_i} \tag{6}$$

3. Proportional Integral Derivate (PID) controller

The Proportional Integral Derivate (PID) controller is the most popular feedback controller used in the process industries. It has been successfully used for over 50 years. It is a robust, easily understood controller that can provide excellent control performance despite the varied dynamic characteristics of process plant. As the name suggests, the PID algorithm consists of three basic modes, the proportional mode, the integral and the derivative modes. A proportional controller has the effect of reducing the rise time, but never eliminates the steady-state error. If the proportional gain is too high, the system can become unstable whereas a small gain results in a small output response to a large input error, and a less responsive or less sensitive controller. An integral controller has the effect of eliminating the steady-state error, but it may make the transient response worse. High integral gain can cause overshoot and low value will make the system sluggish. A derivative controller has the effect of increasing the stability of the system, reducing the overshoot, and improving the transient response [29,30]. If the derivative gain is sufficiently large it can cause a process to become unstable. The conventional fixed gain PID controller is a well known technique for industrial control process. The design of this controller requires the three main parameters, proportional gain (K_p), integral gain (K_i) and derivative gain (K_d). The gains of the controller are tuned by the trial and error method based on the experience and plant behavior. The block diagram of PID controller for a closed loop system is shown in Fig. 4. The transfer function of PID controller Laplace domain is represented by

$$TF_{PID} = K_p + \frac{K_i}{s} + K_d s \tag{7}$$

The output of PID controller in time domain is given by

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(\tau)}{d\tau} \tag{8}$$

where $u(t)$ is the control signal and $e(t)$ is the error signal.

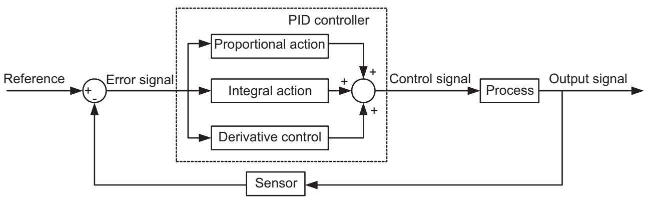

Fig. 4. Block diagram of PID controller for a closed loop system.

For proper design of PID controller the technical expert must select a suitable tuning method of the PID parameters to improve the transient response and minimize the steady-state error.

4. Overview of Many Optimizing Liaisons (MOL) algorithm

Many Optimizing Liaisons (MOL) algorithm is the simplified form of particle swarm optimization (PSO) algorithm. PSO algorithm was first developed in 1995 by Kennedy and Eberhart [23]. Initially the PSO algorithm was introduced for simulating the behavior of bird flock. Latter the PSO algorithm was simplified and applied to the individual particles (bird) which were actually involved in performing the optimization. In PSO algorithm, all the particles are placed at random position and are supposed to move randomly in a defined direction in the search space. Each particle's direction is then changed gradually to insist to move along the direction of its best previous positions of and its peers, searching in their locality to discover even a new better position with respect to some fitness measures $f : \Re^n \to \Re$.

Let $\vec{X} \in \Re^n$ be the position of a particle and $\vec{V}$ be its velocity. Both the initial velocity and position of the particle are chosen randomly and updated iteratively. The formula for updating the velocity of the particle is given by Shi and Eberhart [31]:

$$\vec{V} \leftarrow w\vec{V} + \Phi_P R_P(\vec{P} - \vec{X}) + \Phi_G R_G(\vec{G} - \vec{X}) \tag{9}$$

Φ_P and Φ_G are the positive pre-defined constants called cognitive and social parameters, respectively. Usually $\Phi_P + \Phi_G \geq 4.0$. R_P and R_G are the random numbers uniformly generated in the range (0,1). The inertia weight 'w' helps in convergence of PSO. The inertia weight 'w' is used to control the effect of the previous velocities on the current velocity. The inertia weight compromises between global and local exploration abilities of the swarm. As the search progresses, inertia weight 'w' decreases linearly as per the equation given below:

$$w = w_{max} - \left(\frac{w_{max} - w_{min}}{k_{max}}\right)k \tag{10}$$

where w_{max} and w_{min} are the maximum and minimum values of inertia weight respectively. k_{max} is the maximum number of iterations and k is the current iteration number.

$\vec{P}$ and $\vec{G}$ are the best positions of particle and swarm respectively. Velocity is added with the current position of the particle to move to another new position in the search space.

$$\vec{X} \leftarrow \vec{X} + \vec{V} \tag{11}$$

After updating a particle's position, limitations are imposed on the distance covered by the particle in a single step so that the particle can move from one search space to another in a single step. The steps involved in PSO algorithm are as follows [23].

(a) Initialize randomly 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 Eq. (9).

(e) Using Eq. (11) move the particle to a new position.
(f) Enforce search-space boundaries.
(g) Update the particle's best position, if $f(\overrightarrow{X})<f(\overrightarrow{P})$
(h) The above steps are repeated for the swarm's best position ($\overrightarrow{G}$).

The MOL algorithm is similar to PSO algorithm but the difference is that, in MOL algorithm the particle is updated randomly where as in PSO algorithm the particle is updated iteratively over the entire swarm. This simplified version of PSO is also known as Social Only PSO. In the MOL algorithm the swarm's best position $\overrightarrow{P}$ is eliminated by setting $\Phi_P=0$ and the velocity update formula becomes

$$\overrightarrow{V} \leftarrow w\overrightarrow{V} + \Phi_G R_G(\overrightarrow{G}-\overrightarrow{X}) \tag{12}$$

where 'w' is inertia weight and R_G is the random number uniformly generated in the range [0,1] as explained earlier. The particles current position is denoted by $\overrightarrow{X}$ and updated using Eq. (11) as before. $\overrightarrow{G}$ represents entire swarm's best known position.

5. Results and discussion

5.1. Application of MOL algorithm

An AVR system with PID controller tuned by MOL algorithm is shown in Fig. 5. The gains of the PID controller are tuned by MOL algorithm. If the proportional gain is too high, the system can become unstable whereas a small gain results in a small output response to a large input error, and a less responsive or less sensitive controller. High integral gain can cause overshoot and low value will make the system sluggish. If the derivative gain is sufficiently large it can cause a process to become unstable. These gains depend on the process to be controlled. For the very first execution of the program, a wider solution space can be given and after getting the solution one can shorten the solution space nearer to the values obtained in the previous iteration. For an AVR system the typical ranges used in literature are (0.0–1.5) and (0.2–2) in Refs. [9,20] respectively. To increase the search space and to have better optimized gains, the lower and upper bounds of the gains had been chosen as (0.01,2). After running the optimization algorithm for several times it is found that the minimum and maximum values of gains for different objective functions lie in the chosen range. In Ref. [27] a set of behavioral parameters are suggested for use in the MOL method. After some initial trial and error testing $S=100$, $w=-0.2723$ and $\Phi_G=3.8283$ are chosen in the present study. Optimization is terminated by the pre-specified number of generation which is set to 100.

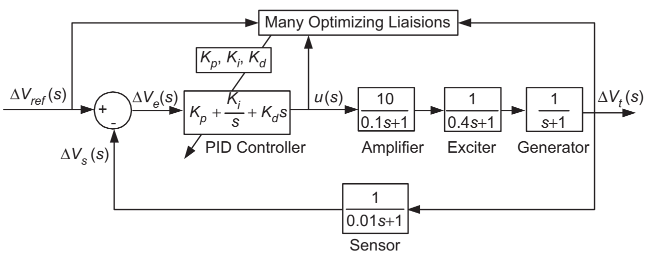

Fig. 5. Transfer function model of an AVR system with PID controller tuned by MOL algorithm.

For determining the optimum values of the gains of the controller, the integral of time multiplied by absolute value of error (ITAE), integral of absolute value of error (IAE), integral of time multiplied by squared error (ITSE) and integral of squared error (ISE) are taken as objective function. The expression for these error functions are given below

$$ITAE = \int_0^t t|V_r - V_t|dt \tag{13}$$

$$IAE = \int_0^t |V_r - V_t|dt \tag{14}$$

$$ITSE = \int_0^t t(V_r - V_t)^2 dt \tag{15}$$

$$ISE = \int_0^t (V_r - V_t)^2 dt \tag{16}$$

where V_t is the terminal voltage, V_r is the reference voltage and t is the time range of simulation.

It is well known that stochastic techniques depend on an initial population which is generated randomly. So in each run, different initial populations will be generated hence different evolutions and different final solutions will be achieved. If one carried out just one run, the result might just be a lucky run or an unlucky one, so the conclusions might be strongly biased by that run, and no general conclusion could be drawn about the actual usefulness of these techniques. This is why in stochastic techniques reported in literature to face a given problem; authors always report result of more runs, usually at least 20 [7,20]. In view of the above, the optimization process is run 20 times in the present paper. After running the optimization technique for 20 times it is seen that the values of objective function are lying within a narrow range for most of the optimization run. The best PID controller parameters corresponding to the minimum fitness value among 20 runs are given in Table 1 for each objective functions. Fig. 6 shows the typical convergence curves for different objective functions.

5.2. Effect of objective function

In order to compare the performance of PID controller optimized using various objective functions, simulation studies are carried out. Fig. 7 shows the terminal voltage step response with PID controller optimized using four objective functions. Maximum overshoot, rise time, peak time and settling time for each case are shown in Table 2.

Table 1
Parameters of PID controller with different objective functions.

Parameters/objective function	ITAE	IAE	ITSE	ISE
K_p	0.5857	0.9931	0.9877	0.9544
K_i	0.4189	0.7461	0.7780	0.9434
K_d	0.1772	0.4249	0.5014	0.9909

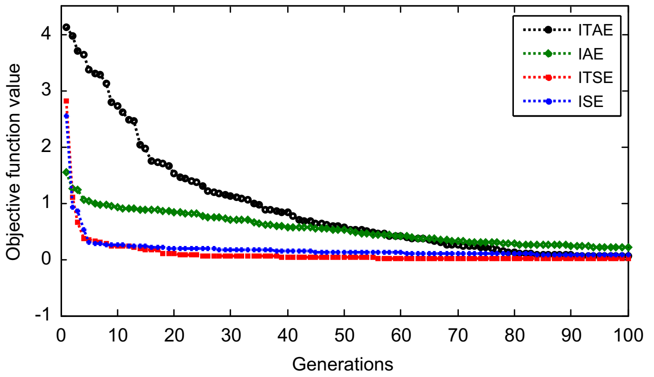

Fig. 6. Convergence curves for different objective functions.

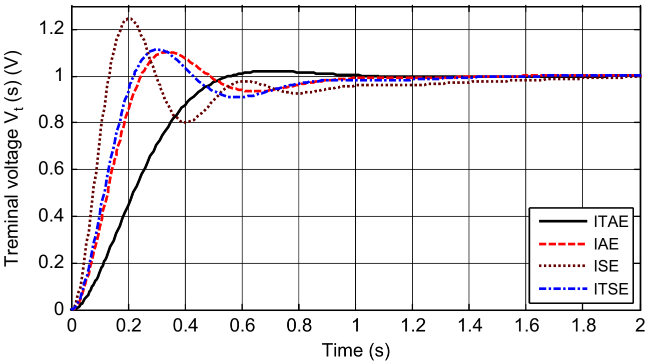

Fig. 7. Terminal voltage for different objective functions.

Table 2
Result of the transient response for different error functions.

Techniques/error functions	Maximum overshoot (pu)	Settling time (2% bant) (s)	Rise time (s)	Peak time (s)
MOL/ITAE	**1.0195**	**0.5155**	0.3433	0.7036
MOL/IAE	1.1040	0.8814	0.1644	0.3408
MOL/ITSE	1.1143	0.8769	0.1468	0.3056
MOL/ISE	1.2481	1.5324	**0.0895**	**0.2013**
ABC/ITSE [9]	1.25	3.19	0.156	0.3628
PSO/ITSE [9]	1.3005	3.5	0.1609	0.3908
DE/ITSE [9]	1.3281	2.77	0.1513	0.3636

For comparison, Table 3 also shows the corresponding values with other recently published modern heuristic optimization techniques such as Artificial Bee Colony (ABC) algorithm, Particle Swarm Optimization (PSO) and Differential Evolution (DE) algorithm [9]. In Ref. [9] an integral of time weighted squared error (ITSE) criterion has been used to

Table 3
Closed loop poles and damping ratio of the AVR system tuned by MOL, ABC, PSO and DE algorithm.

MOL		**ABC** [9]		**PSO** [9]		**DE** [9]	
Closed loop poles	**Damping ratio**	**Closed loop poles**	**Damping ratio**	**Closed loop poles**	**Damping ratio**	**Closed loop poles**	**Damping ratio**
−100	1	−100.98	1	−101	1	−101	1
−2.11	1	−4.74	1	−6.26	1	−6.3	1
−1.06	1	−0.25	1	−0.215	1	−0.228	1
−4.92+4.72i	**0.721**	−3.75+8.4i	0.4	−3.09+7.8i	0.368	−3.03+8.1i	0.347
−4.92−4.72i	**0.721**	−3.75−8.4i	0.4	−3.09−7.8i	0.368	−3.03−8.1i	0.347

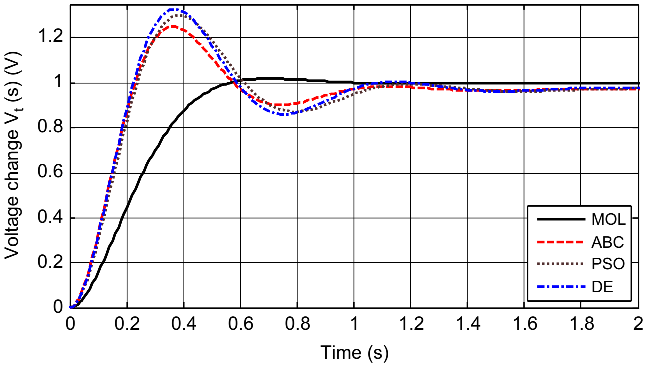

Fig. 8. Terminal voltage curves of the AVR system for different algorithms.

optimize the PID controller parameters. It is clear from Table 3 that with proposed MOL optimized PID controller, better performance is obtained compared to ABC, PSO and DE algorithm in terms of maximum overshoot, settling time, peak time and rise time when ITSE and ISE are used as objective functions. It is also clear from Fig. 7 and Table 2 that minimum rise time and settling time are obtained when ISE is used as an objective function. However when ITAE is used as an objective function maximum over shoot and settling times are reduced which are the major factors for comparing stability analysis of a system. Hence these controller parameters are chosen for further analysis. Transfer functions of the system according to the above gains using MOL algorithm is given by

$$\frac{\Delta V_t(s)}{\Delta V_{ref}(s)} = \frac{0.01772s^3 + 1.831s^2 + 5.899s + 4.189}{0.0004s^5 + 0.0454s^4 + 0.555s^3 + 3.282s^2 + 6.857s + 4.189} \tag{17}$$

5.3. Transient response analysis

Transient and steady state behavior of the system can be analyzed from the transient response analysis of MOL optimized PID controller in AVR system as shown in Fig. 8. For comparison, the responses with ABC algorithm, PSO algorithm and DE algorithm [9] are also shown in Fig. 8. It is observed that the maximum overshoot in MOL algorithm is 22.61% lesser than that of ABC algorithm, 27.51% lesser than that of PSO algorithm and 30.46% lesser than that of DE algorithm. The peak time in MOL algorithm is more than that of ABC algorithm, PSO algorithm and DE algorithm. Whereas the settling time in MOL algorithm is 518.82% less than that of ABC algorithm, 578.95% less than that of PSO algorithm and 437.34% less than that of DE algorithm. The maximum overshoot and settling time for MOL algorithm are better than ABC, PSO and DE algorithms which are the major factors for comparing stability analysis of a system.

5.4. Root locus analysis

Fig. 9 shows the root locus curve for MOL algorithm. The closed loop poles and damping ratios of the system tuned by MOL, ABC, PSO and DE algorithms are given in Table 3. It can be seen in Table 3 that all closed loop poles are lying to the left of the *s*-plane for all the algorithms indicating that, the AVR system tuned by all algorithms are stable. The conjugate poles obtained with MOL algorithm are located away from the imaginary axis as compared to ABC, PSO and DE algorithm. Damping ratio of AVR system tuned by MOL algorithm is 44.52% more than ABC, 48.96% more than PSO and 51.87% more than DE algorithm.

5.5. Bode analysis

Frequency response of the control is analyzed through Bode plot. The magnitude and phase plot of the AVR system tuned by MOL algorithm is shown in Fig. 10. The peak gain, phase margin, delay margin and bandwidth obtained from Bode plots are depicted in Table 4 and compared with ABC, PSO and DE. The minimum peak gain and minimum bandwidth along with maximum phase margin and maximum delay margin are obtained from MOL algorithm. It is observed that best frequency response is obtained by using MOL.

5.6. Robustness analysis

Robustness analysis of the AVR system tuned by MOL algorithm is performed by varying the time constants of amplifier, excitor, generator and sensor in the range of −50% to +50% in steps of 25%. The results obtained are depicted in Table 5. The responses of

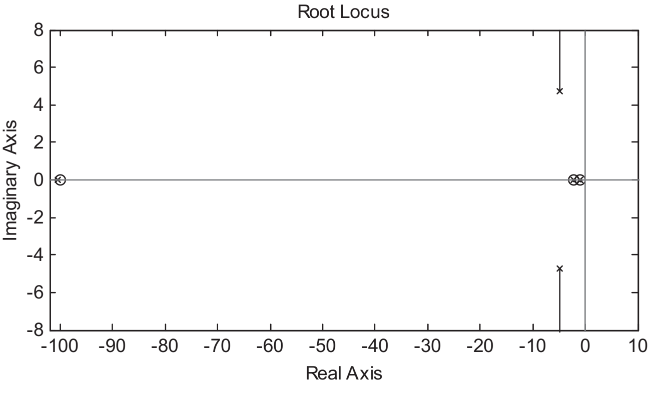

Fig. 9. Root locus curve of the system tuned by MOL algorithm.

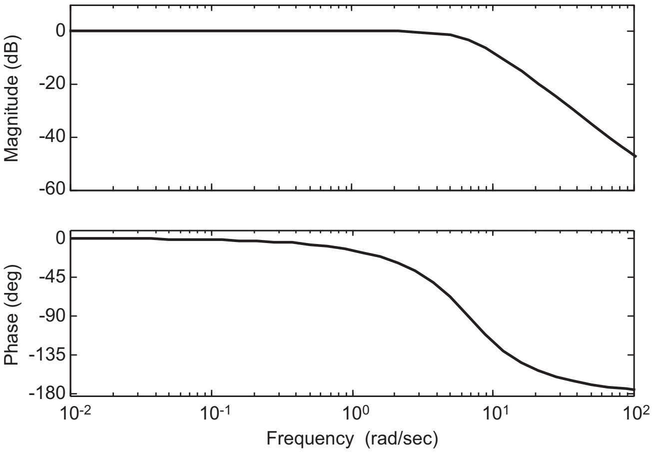

Fig. 10. Bode plots of the system tuned by MOL algorithm.

Table 4
Bode analysis.

Different algorithm	Peak gain (dB)	Phase margin (deg.)	Delay margin (s)	Bandwidth
MOL	**0.0**	**180**	**Inf.**	6.3373
ABC [9]	2.87 (7.51 Hz)	69.4	0.1109	**12.8791**
PSO [9]	3.75 (1.14 Hz)	62.2	0.103	12.182
DE [9]	4.20 (1.21 Hz)	58.4	0.092	12.8

Table 5
Results of robustness analysis of PID controller tuned by MOL algorithm.

Parameter	Rate of change (%)	Peak value (pu)	Settling time (2% bant) (s)	Rise t ime (s)	Peak time (s)
T_a	−50	1.0004	0.7872	0.3926	1.8995
	−25	1.0003	0.5916	0.3553	1.7669
	+25	1.0503	0.9493	0.3431	0.7010
	+50	1.0799	1.0590	0.3478	0.7254
T_e	−50	1.0002	1.4001	0.2494	3.7656
	−25	0.9989	1.1781	0.2986	2.2362
	+25	1.0533	1.3005	0.3838	0.8287
	+50	1.0833	1.6677	0.4210	0.9500
T_g	−50	1.0389	1.9792	0.198	0.3751
	−25	1.019	1.3652	0.2713	0.5244
	+25	1.0343	1.5761	0.4120	0.9453
	+50	1.0551	2.4591	0.4765	1.1826
T_s	−50	1.0142	0.5345	0.3523	0.7169
	−25	1.0168	0.5247	0.3477	0.7049
	+25	1.0225	0.7565	0.3389	0.6920
	+50	1.0258	0.7824	0.3347	0.6803

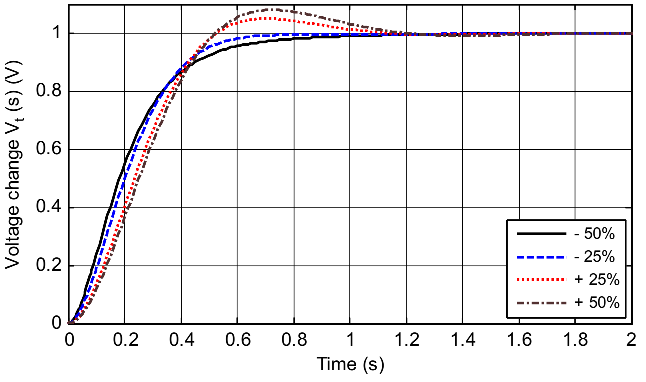

Fig. 11. Voltage change curves ranging from −50% to +50% for T_a.

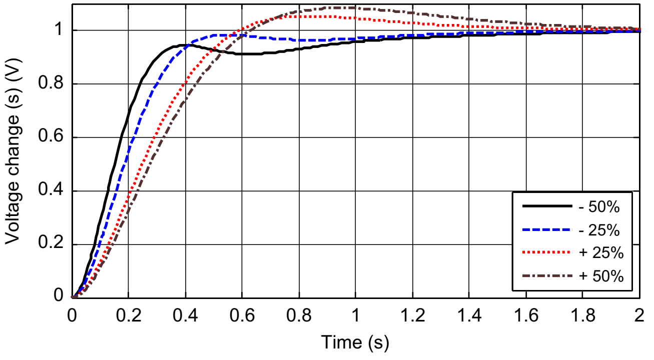

Fig. 12. Voltage change curves ranging from −50% to +50% for T_e.

the AVR system by varying the time constants T_a, T_e, T_g and T_s respectively are shown in Figs. 11–14. The range of deviations is provided in Table 6. It can be observed from the table that the deviation for the selected system parameters is in small range. The average deviation of maximum overshoot is 4.1% and for settling time (2% bant), rise time and peak time are 189.4%, 14.5% and 168.8% respectively. The ranges of total deviation are within limit. Therefore it can be concluded that the AVR system with PID controller tuned by MOL algorithm is robust.

6. Conclusion

There are several algorithms available for designing the parameters of the PID controllers. The aim of this paper is to determine the parameters of the PID controller for an AVR system using the simplified particle swarm optimization algorithm also known as Many Optimizing Liaisons (MOL). The proposed algorithm has been applied and the results are compared with some recently reported modern heuristic algorithms such as ABC, PSO and DE algorithms. The effect of selection of objective function on the performance has also been studied. It is observed that to reduce the maximum overshoot and settling time, the PID controller parameters optimized using ITAE gives better

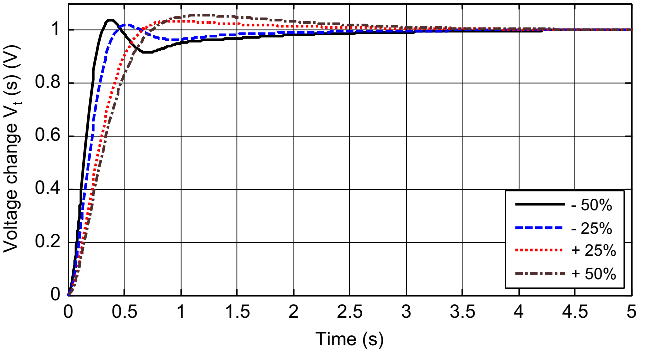

Fig. 13. Voltage change curves ranging from −50% to +50% for T_g.

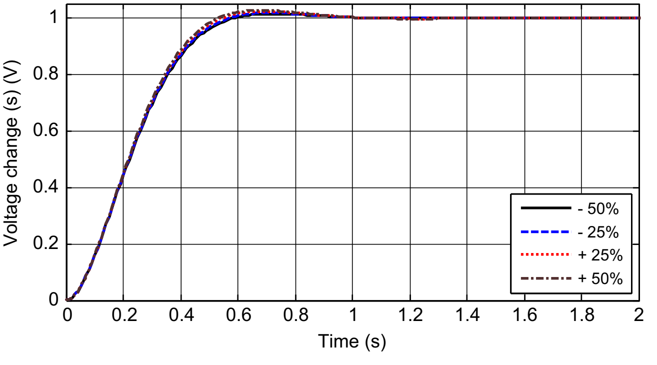

Fig. 14. Voltage change curves ranging from −50% to +50% for T_s.

Table 6
Range of total deviations and percentage of maximum deviation of the system.

Time constants	Parameter	Range of total deviations	Percentage of total deviation (%)
T_a	Peak (pu)	0.0796	5.9
	Settling time (s)	0.4674	105.43
	Rise time (s)	0.0495	14.4
	Peak time (s)	1.1985	170
T_e	Peak (pu)	0.0844	6.3
	Settling time (s)	0.3672	223.51
	Rise time (s)	0.1716	22.6
	Peak time (s)	2.9369	435.2
T_g	Peak (pu)	0.0361	3.5
	Settling time (s)	1.0939	377.03
	Rise time (s)	0.2785	13.3
	Peak time (s)	0.8075	68.1
T_s	Peak (pu)	0.0033	0.6
	Settling time (s)	0.2577	51.77
	Rise time (s)	0.0176	7.8
	Peak time (s)	0.0366	1.9

performance compared to ITSE, ISE and IAE objective functions. It is also seen that even for the same objective function (ITSE); proposed MOL algorithm outperforms ABC, PSO and DE algorithms. Further, transient response analysis, the root locus analysis and the bode analysis are applied and compared with recently published results to show the superiority of proposed MOL algorithm. The result also shows that, the change in the system parameters does not affect the quality of the proposed MOL algorithm.

References

[1] P. Kundur, Power System Stability and Control, TMH, 8th reprint, 2009.

[2] S.J. Ho, L.S. Shu, S.Y. Ho, Optimizing fuzzy neural networks for tuning PID controllers using an orthogonal simulated annealing algorithm (OSA), IEEE Transactions on Fuzzy Systems 14 (3) (2006) 421–434.

[3] A. Bagis, Tabu search algorithm based PID controller tuning for desired system specifications, Journal of the Franklin Institute 348 (2011) 2795–2812.

[4] S. Panda, Differential evolution algorithm for SSSC-based damping controller design considering time delay, Journal of the Franklin Institute 348 (8) (2011) 1903–1926.

[5] S. Panda, Robust coordinated design of multiple and multi-type damping controller using differential evolution algorithm, International Journal of Electrical Power and Energy Systems 33 (2011) 1018–1030.

[6] S. Panda, Multi-objective evolutionary algorithm for SSSC-based controller design, Electric Power Systems Research 79 (2009) 937–944.

[7] S. Panda, N.P. Padhy, Comparison of particle swarm optimization and genetic algorithm for FACTS-based controller design, Applied Soft Computing 8 (4) (2008) 1418–1427.

[8] K.A. El Metwally, A fuzzy logic based PID for power system stabilization, Electric Power Components and Systems 29 (2001) 659–669.

[9] H. Gozde, M.C. Taplamacioglu, Comparative performance analysis of artificial bee colony algorithm for automatic voltage regulator (AVR) system, Journal of the Franklin Institute 348 (2011) 1927–1946.

[10] S. Panda, N.P. Padhy, Optimal location and controller design of STATCOM for power system stability improvement using PSO, Journal of the Franklin Institute 345 (2) (2008) 166–181.

[11] S. Panda, Application of non-dominated sorting genetic algorithm-II technique for optimal FACTS-based controller design, Journal of the Franklin Institute 347 (7) (2010) 1047–1064.

[12] S. Panda, Multi-objective PID controller tuning for a FACTS-based damping stabilizer using non-dominated sorting genetic algorithm-II, International Journal of Electric Power and Energy Systems 33 (2011) 1296–1308.

[13] M.W. Iruthayarajan, S. Baskar, Covariance matrix adaptation evolution strategy based design of centralized PID controller, Expert Systems with Applications 37 (8) (2010) 5775–5781.

[14] M.I. Menhas, L. Wang, M. Fei, H. Pan, Comparative performance analysis of various binary coded PSO algorithms in multivariable PID controller design, Expert Systems with Applications 39 (4) (2012) 4390–4401.

[15] M.W. Iruthayarajan, S. Baskar, Evolutionary algorithms based design of multivariable PID controller, Expert Systems with Applications 36 (5) (2009) 9159–9167.

[16] S.Z. Zhao, M.W. Iruthayarajan, S. Baskar, P.N. Suganthan, Multi-objective robust PID controller tuning using two lbests multi-objective particle swarm optimization, Information Sciences 181 (16) (2011) 3323–3335.

[17] R.R. Sumar, A.A.R. Coelho, L.D.S. Coelho, Computational intelligence approach to PID controller design using the universal model, Information Sciences 180 (20) (2010) 3980–3991.

[18] L.D.S. Coelho, M.W. Pessoa, A tuning strategy for multivariable PI and PID controllers using differential evolution combined with chaotic Zaslavskii map, Expert Systems with Applications 38 (11) (2011) 13694–13701.

[19] L.D.S. Coelho, D.L.D.A. Bernert, PID control design for chaotic synchronization using a tribes optimization approach, Chaos, Solitons and Fractals 42 (1) (2009) 634–640.

[20] L.D.S. Coelho, Tuning of PID controller for an automatic regulator voltage system using chaotic optimization approach, Chaos, Solitons and Fractals 39 (4) (2009) 1504–1514.

[21] L.D.S. Coelho, R.B. Grebogi, Chaotic synchronization using PID control combined with population based incremental learning algorithm, Expert Systems with Applications 37 (7) (2010) 5347–5352.
[22] M.I. Menhas, L. Wang, M.R. Fei, C.X. Ma, Coordinated controller tuning of a boiler turbine unit with new binary particle swarm optimization algorithm, International Journal of Automation and Computing 8 (2) (2011) 185–192.
[23] J. Kennedy, R.C. Eberhart, Particles swarm optimization, in: Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, 1995, pp. 1942–1948.
[24] V. Mukherjee, S.P. Ghoshal, Comparison of intelligent fuzzy based AGC coordinated PID controlled and PSS controlled AVR system, Electric Power and Energy Systems 29 (2007) 679–689.
[25] H. Gozde, M.C. Taplamacioglu, Automatic generation control application with craziness based particle swarm optimization in a thermal power system, International Journal of Electric Power and Energy Systems 33 (2011) 8–16.
[26] J. Kennedy, The particle swarm: social adaptation of knowledge, in: Proceedings of the IEEE International Conference on Evolutionary Computation, Indianapolis, USA, 1997.
[27] M.E.H. Pedersen, A.J. Chipperfield, Simplifying particle swarm optimization, Applied Soft Computing 10 (2010) 618–628.
[28] O.I. Elgerd, Electric Energy Systems Theory. An Introduction, Tata McGraw-Hill, New Delhi, 1983.
[29] P. Cominos, N. Munro, PID controllers: recent tuning methods and design to specification, IEE Proceedings—Control Theory and Applications 149 (1) (2002) 46–53.
[30] K.J. Astrom, T. Hagglund, PID Controllers: Theory, Design and Tuning, Instrument Society of America, USA, 1995.
[31] Y. Shi, R. Eberhart, A modified particle swarm optimizer, in: Proceedings of the IEEE International Conference on Evolutionary Computation, Anchorage, AK, USA, 1998.