Research Article

Application of metaheuristic control strategies to voltage regulation

Rajeshree Ramjug-Ballgobin[1] · Roopesh Fowdar[1]

Received: 4 October 2019 / Accepted: 26 November 2019 / Published online: 30 November 2019

Abstract
This paper proposes four controllers applied to two existing generator models (1st and 4th order) with a type 1 excitation system taking into account nonlinearities. Jaya, crow search, and invasive weed optimization algorithm based PID controllers as well as adaptive neuro-fuzzy interface system controller were used to regulate the generator output in case of sudden voltage fluctuations. The results obtained were found to be very promising since most of them improved the uncompensated systems response in terms of overshoot, steady-state error, settling time and rise time. The overall best controller was found to be the Jaya based PID due to its responses as well as its ability to quickly converge to the best fitness.

Keywords Automatic voltage regulator · PID · ANFIS · Jaya · ITAE · Crow search · Invasive weed optimization

1 Introduction

During recent years, computers had a great impact on the power systems in preserving the quality of the power supplied and its reliability. The increase in power demand, as well as the growth in size and complexity of electrical systems, has urged the emergence of intelligent systems that combine different techniques and methodologies [1] in order to regulate voltage and frequency. The intelligent systems possess human-like knowledge and adapt themselves whenever there are changes in situations. Instability in power systems occurs mainly due to load changes that occur randomly as per the need of consumers. Unstable voltage may cause damage to electrical appliances or malfunction. To tackle this problem and maintain the voltage within an acceptable limit, the reactive power can be easily and efficiently controlled using an automatic voltage regulator (AVR) for each power generating unit in the network. AVR consist of six main components which are shown in Fig. 1.

AVR uses potential transformer to measure the output voltage of a generator which is rectified into dc voltage and compared to a pre-set value. The error is fed to a controller which produces a control signal. The latter is amplified and used to control the field of an exciter which in turn will produce an excitation signal for the main rotor field winding and correct the voltage difference [2].

During the past years, many researches and simulations were carried out using different algorithms and controllers in order to have better control over sudden voltage fluctuations. The most common type of controller used is the Proportional Integral Derivative (PID) which offers the simplest and most efficient solution to many real-world control problems and is still used in more than 90% of the industrial controllers [3] but nowadays more and more research are being done using artificial neural network-based intelligent controllers. The hybrid ANFIS controller was first introduced in 1993 by Jang [4] by combining a fuzzy interface system (FIS) with the learning capability of artificial neural networks (ANN) [5]. ANFIS has previously been applied in many fields such as speed controller of electro-mechanical system [6], control of autonomous vehicles [7], Voltage and Frequency Stability in Islanded Microgrids [5] and also in Power Flow Analysis and Control [8, 9].

✉ Rajeshree Ramjug-Ballgobin, r.ramjug@uom.ac.mu | [1]Electrical and Electronic Engineering Department, Faculty of Engineering, University of Mauritius, Reduit, Mauritius.

SN Applied Sciences (2019) 1:1731 | https://doi.org/10.1007/s42452-019-1802-8

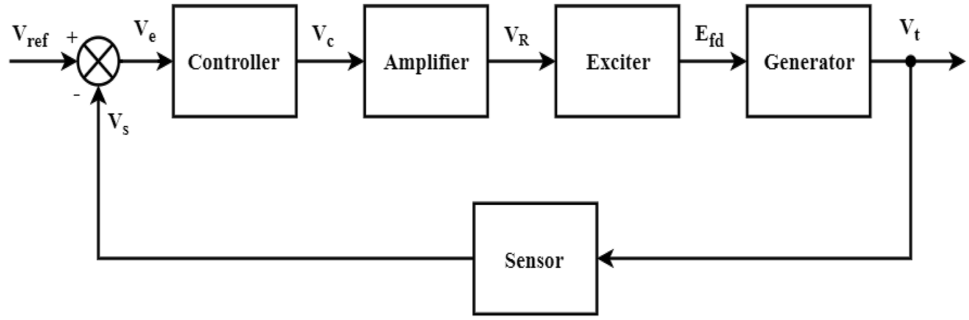


Fig. 1 Basic block diagram of an AVR system

In many research papers [1, 10–12], the most common algorithm used to tune PID gains for voltage regulation is the particle swarm optimization (PSO). In 2011, Gozde and Taplamacioğlu [12] implemented the Artificial Bee Colony (ABC) algorithm to optimize the control problem of an AVR system and compared its performance with the differential evolution (DE) algorithm and in 2014 they presented a new Chaotic PSO algorithm to solve the problem of voltage fluctuation as quickly as possible. Ramjug-Ballgobin and Calchand presented a paper [13], in which both unsaturated and saturated AVR systems were taken into consideration and PID, PI-Fuzzy and Fuzzy compensators were designed and implemented to optimize the response of the AVR in case of disturbances.

Invasive weed optimization (IWO) algorithm, first proposed by Mehrabian and Lucas [14], was used by many authors such as Oozeer and Ramjug-Ballgobin [15], Patel et al. [16] and also by Mishra et al. [17] in the field of load frequency control (LFC). Jaya algorithm(JA) developed by Rao [18] in 2016, have been used by Purey and Arya [19] to obtain optimum reactive power reserve by using shunt compensation to IEEE 14 bus and 30 bus systems taking into account constraints on the stability. Crow search algorithm (CSA) was implemented on IEEE 30, 57 and 118 bus systems in order to optimize reactive power dispatch system by Lakshmi and Ramesh in 2018 [20]. The objective was to enhance voltage stability and minimization of voltage deviation and real power losses.

In this research work, the concept of Neuro-fuzzy controller and PID controller tuned by IWO, JA, and CSA are used as compensator in an AVR and the responses were compared to each other to determine the best response. Despite the enormous advance in the field of control systems over the last few decades, the PID controller is still the most commonly used controller. This is the result of its better adaptability as well as the relative simplicity with which it can be implemented. The new aspect brought about by this work involves the optimisation of the Neuro-fuzzy controller and PID controller through metaheuristic control strategies. Our main purpose is to study and analyse three different PID tuning algorithms, namely, the invasive weed algorithm, the Jaya algorithm and the Crow search algorithm and their application to the problem of voltage control. Since, to the authors' best knowledge, these algorithms have never been applied to voltage control, the research aims at establishing the authority that these algorithms can demonstrate in this field. The superiority of the best algorithm with respect to the remaining two is also investigated.

2 System modelling

2.1 Generator

Two types of synchronous generator were used for simulation and comparison namely a first order and a fourth-order model which better describes a generator in terms of accuracy. The transfer function of the models is given by Eqs. (1) and (2) respectively.

$$G_{G1}(s) = \frac{V_t(s)}{V_F(s)} = \frac{K_{G1}}{1 + s\tau_{G1}} \tag{1}$$

$$G_{G4}(s) = \frac{V_t(s)}{V_F(s)} = \frac{K_{G4}\left(1 + sT_{z1}\right)\left(1 + sT_{z2}\right)\left(1 + sT_{z3}\right)\left(1 + sT_{z4}\right)}{\left(1 + sT_{p1}\right)\left(1 + sT_{p2}\right)\left(1 + sT_{p3}\right)\left(1 + sT_{p4}\right)} \tag{2}$$

The gain, K_{G1} and time constant, τ_{G1} are chosen to be 1 and 1.5 respectively while the 4th order generator gains were based on the papers presented by Walton [21], Law [22] and by Enzhe et al. [23]. $V_t(s)$ is the terminal voltage of the stator windings of the machine and $V_F(s)$ is the field voltage produced by the exciter output E_{fd} to be applied to the field winding of the synchronous generator.

2.2 Exciter

The Type 1 excitation system presented on the IEEE committee report 1968 [24] was taken into consideration and which consists of the amplifier, the saturation functions, the Power system Stabilizer (PSS) and the exciter (Table 1).

Table 1 Excitation system components with their transfer function and value

Component	Transfer function	Parameter value
Amplifier	$G_A(s) = \frac{K_A}{1+s\tau_A}$	Gain, $K_A = 10$ Time constant, $\tau_A = 0.1$
Exciter	$G_E(s) = \frac{K_E}{1+\tau_E s}$	Gain, $K_E = 1$ Time constant, $\tau_E = 0.5$

However, the effect of the PSS was ignored in this research. Sample data from [24] was used to determine the saturation functions and the values used for each component and their transfer function are given below:

2.3 Sensor

The terminal voltage was measured and the signal was rectified and filtered using a potential transformer modelled as a first-order transfer function shown below.

$$G_s(s) = \frac{K_s}{1 + s\tau_s} \tag{3}$$

where K_s is the sensor gain having a value of 1 and τ_s is the time constant which has a very small value between 0.001 and 0.06 representing the filtering and rectification of the output voltage. A value of 0.05 was chosen in this work.

3 Optimization techniques

3.1 ANFIS controller

The principle of operation of an ANFIS is based on extracting fuzzy rules at each level of a neural network [7]. The overall behaviour of the process is determined by the rules obtained through training of a Sugeno type FIS from sample data. The functioning of the ANFIS is explained in detail in [4] with its five layers architecture. The fuzzy logic toolbox in MATLAB which is dedicated for Fuzzy Logic and Neuro-Fuzzy Controllers was used to design the ANFIS controller which was provided with two inputs, the Error and the derivative of the Error in order to have appropriate rule base.

Sample data were collected by simulating the systems with a PID Controller tuned by the inbuilt PID tuner. The data was fed to the neuro-fuzzy designer window which was used to train the ANFIS controller using 2 membership functions, 50 epochs and an error tolerance of 0.001. A FIS file is obtained upon training the controller which needs to be assigned to the ANFIS Controller. A gain of 0.96 and 0.89 were added at the output of the ANFIS for 1st order generator (Fig. 2) and 4th order generator respectively to reduce the steady-state error of the output.

3.2 Algorithm based PID controller

Every optimization technique requires an objective function and stopping criterion in order to evaluate the best solution for the parameters used in the controller (P, I & D). The ITAE performance index is chosen for the fitness function (Fig. 3) due to its sensitivity and its ability to produce lesser settling time and overshoot [25]. The equation for ITAE is given in Eq. (4).

$$ITAE = \int_0^T t|e(t)|dt \tag{4}$$

where T settling time for the system, t time, $e(t)$ error between the input and output.

The stopping criterion used to end the algorithm, is the number of iteration/generations that is the amount of time to run the algorithm so that the result converges to the best value.

The PID Block in MATLAB is given by Eq. (5) where P, I and D needs to be optimized by the algorithms used.

$$G_{PID}(s) = P + I\frac{1}{s} + D\frac{N}{1 + N\frac{1}{s}} \tag{5}$$

3.2.1 Invasive weed optimization (IWO)

IWO is based on the weed behaviour of colonization. It can be broken down into 4 parts which are the initialization of a first population, reproduction, seed dispersal, and competitive elimination. The procedure to optimized the PID parameters in the AVR are as follows:

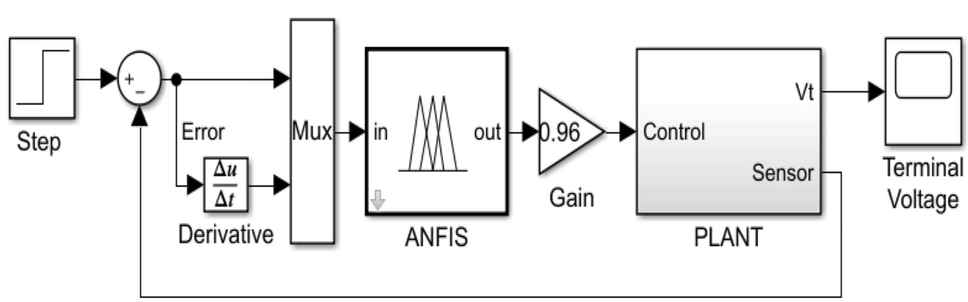


Fig. 2 Implementation of ANFIS Controller for 1st order generator

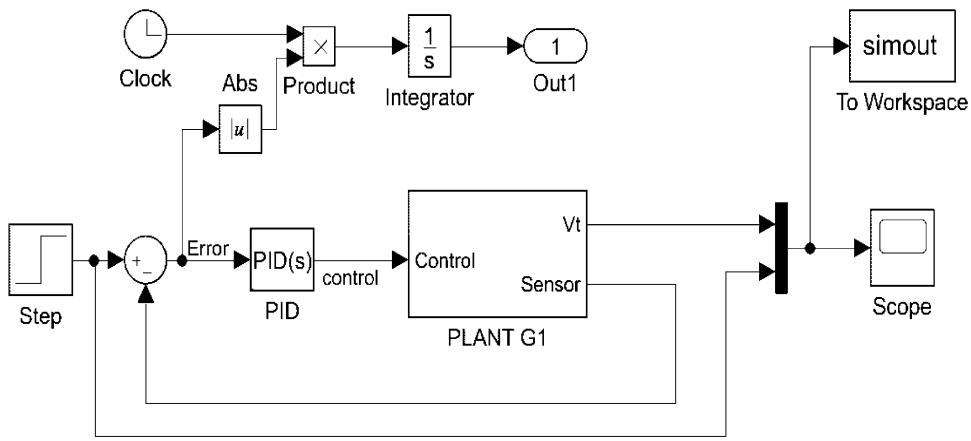


Fig. 3 Implementation of algorithm based PID with ITAE

Step 1: 3 variables with limit 0–10, a population size of 100, an initial population of 10 weed, a minimum and maximum of 0 and 5 seeds and 125 iterations are first initialized and an array is created to store the random initial values of the decision variables for the initial population together with their corresponding fitness value. The initial and final standard deviation used are 0.5 and 0.001 respectively.

Step 2: The standard deviation is calculated for the current iteration and the best and worst fitness is found using the min and max command. For each individual, the number of seeds produced is calculated using Eq. (6) and each seed are spread randomly according to the standard deviation (Eq. (8).

$$S = s_{\min} + \frac{\left(s_{\max} - s_{\min}\right)\left(f_i - f_{worst}\right)}{\left(f_{best} - f_{worst}\right)} \tag{6}$$

where S is the number of seed produce, $s_{max/min}$ are the maximum and minimum seed respectively and f_i is the fitness of the weed and $f_{best/worst}$ is the overall best and worst fitness.

$$\sigma_{\text{iter}} = \left(\frac{iter_{\max} - iter}{iter_{\max} - 1}\right)^n \left(\sigma_{\text{initial}} - \sigma_{\text{final}}\right) + \sigma_{\text{final}} \tag{7}$$

$$newposition = currentposition + \sigma_{\text{iter}} \times randn(0, 1) \tag{8}$$

where σ_{iter}, $\sigma_{initial}$, σ_{final} current, initial and final standard deviation respectively, $iter_{max}$, iter maximum and current iteration, n nonlinear modulation index (n = 2).

Step 3: The fitness of the all the new offspring is evaluated and the population of the parent and offspring are merged together in a matrix and are sorted so that the best solution is on the top and worst solution at the end.

Step 4: If the population size exceeds the maximum population size, the excess least fit weeds are eliminated and the best value found is displayed along with its iteration number.

Step 5: Steps 2 to 3 are repeated until the maximum iteration is reached.

3.2.2 Jaya algorithm (JA)

Rao [18] developed Jaya algorithm in 2016 inspired by the Teaching Learning Based Optimization (TLBO) which gained popularity amongst researchers. JA is simpler and more powerful than TLBO and can be easily implemented for a constraint or unconstrained problem. Both of these algorithms require only common parameters. The basic concept of JA is to optimize the objective function by moving towards Jaya (victory in Sanskrit) and avoiding the worst solution to do so, JA uses Eq. (9).

$$X'_{j,k,i} = X_{j,k,i} + r_{1,i,j}\left(X_{j,best,i} - \left|X_{j,k,i}\right|\right) - r_{2,i,j}\left(X_{j,worst,i} - \left|X_{j,k,i}\right|\right) \tag{9}$$

where $X'_{j,k,i}$ New solution, $X_{j,k,i}$ Current solution, $X_{j,best,i}$, $X_{j,worst,i}$ best and worst individual in the population respectively, $r_{1,i,j}$ and $r_{2,I,j}$ random number between 0 and 1.

Step 1: 100 number of individuals, 125 iterations and 3 variables with limit 0–10 are initialized. A matrix is created that can hold the random initial values of the variables for each individual.

Step 2: The Fitness of each individual is determined using the objective function.

Step 3: The best fitness and the worst fitness is found and a new solution is calculated using Eq. (9). The new solution is limited between its limits if ever it is exceeded.

Step 4: The fitness of the new individual is calculated and if the latter is better than the current one, the old solution is replaced and the final population is displayed.

Step 5: If the maximum iteration is not reached, steps 3, 4 and 5 are repeated.

3.2.3 Crow search algorithm (CSA)

CSA is a novel metaheuristic optimization algorithm, first introduced by A. Askarzadeh. It is based on the cleverness behaviour of crows, whereby crows store their excess food in hiding places and retrieve the food when it is needed by trying to fool other crows who want to steal their food sources [26].

Step 1: 3 decision variables, a flock size of 100, flight length of 2, awareness probability of 0.1, lower and upper bound of 0 and 10 and 125 iterations are initialized.

Step 2: A population of randomly generated individuals across the search space is created and the fitness value of each one is evaluated using the ITAE criterion.

Step 3: Random candidate crows are generated to be followed by the thief and its new position is evaluated by Eq. (10).

$$X_i^{new} = \begin{cases} X_i^{old} + r_i * flight\ length * \left(memory_i - X_i^{old}\right), & r_j \geq AP \\ random\ position, & otherwise \end{cases} \tag{10}$$

$r_i = r_j$ = random number with uniform distribution between 0 and 1 and AP = awareness probability of crow j.

Step 4: For the new position of each crow, the fitness values are computed and the memory is updated if the fitness is better than the previous one as long as the solution is within limits.

Step 5: The best solution is displayed and as long as the maximum iteration is not reached, steps 2–5 are repeated with the new positions of crows produced.

4 Result and discussion

All the optimization techniques were implemented on MATLAB/Simulink 2017b software and the following results were obtained. The PID parameters found by each algorithm run for 125 iterations are shown in Table 2.

The different algorithms are compared using several parameters, namely, the fitness value, the maximum overshoot, the settling time, the rise time and the steady state error.

For both the 1st order and 4th order generator, the Jaya algorithm reached the best fitness value compare with IWO and CSA. IWO had the worst fitness for both generators. The fitness value gives an indication of the suitability of the particular technique in solving the control problem. Therefore, based on this criterion, it can be inferred that the Jaya algorithm is most suitable.

The step response for each case is shown in Fig. 4a, b. The maximum overshoot is defined as the maximum deviation of the response from its reference value and the settling time is the time taken to reach within a given percent of the final value. The aim of the controller is to ensure that the response reaches the final value in the minimum amount of time and with minimum overshoot. Also, the rise time, that is, the time taken to reach 95% of the final value and the steady state error which is discrepancy between desired final value and actual final value are considered. Significant improvement can be observed from the step responses with almost no steady-state error and minimum overshoot except for the IWO based PID controller compared to the

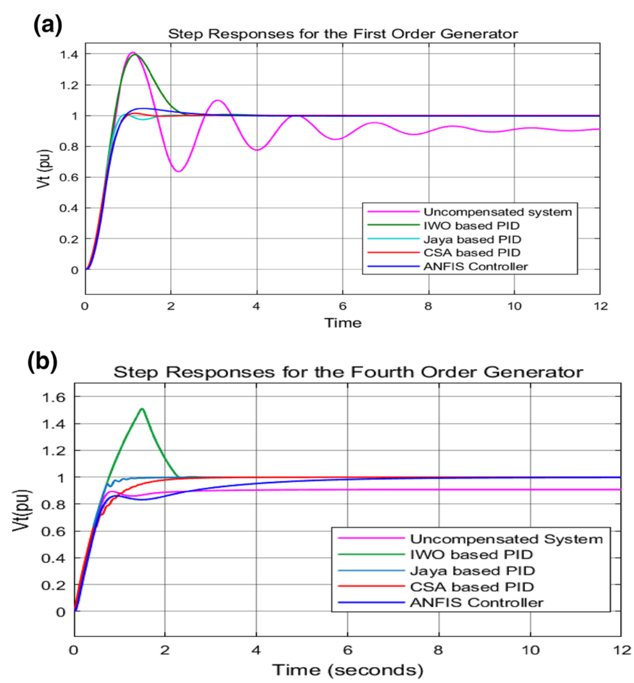


Fig. 4 Step responses for the 1st and 4th order generator

Table 2 PID coefficients and best fitness value computed by the algorithms

	1st order generator			4th order generator		
	IWO	JA	CSA	IWO	JA	CSA
K_P	0.8551	1.2010	0.7491	3.4393	10	6.6107
K_I	0.8491	0.2052	0.2021	7.2360	0.2410	0.1860
K_D	0.2037	0.3670	0.2367	0.2719	1.7215	2.911
Best fitness	0.6056	0.1554	0.1620	0.6888	0.1324	0.2840

Table 3 Analysis of step responses

Controllers	Percentage overshoot (%)	Settling time (s)	Rise time (s)	Steady state error (pu)	Final value (pu)
1st order generator					
Uncompensated	55	8.683	0.42	−0.091	0.909
ANFIS	4.91	2.200	0.549	−0.003	0.997
JA	0.7	1.501	0.514	0	1
$IWO_{125\ iteration}$	39.5	2.212	0.459	0	1
$IWO_{200\ iteration}$	1.6	0.802	0.519	0	1
CSA	1.4	0.904	0.580	0	1
4th order generator					
Uncompensated	0	2.127	0.535	−0.092	0.908
ANFIS	0	5.654	2.438	0.002	0.998
JA	0	1.035	0.602	0	1
$IWO_{125\ iteration}$	51	2.233	0.602	0	1
$IWO_{200\ iteration}$	3.4	0.984	0.602	0	1
CSA	0	2.006	1.049	0	1

uncompensated system. Table 3 summarises the analysis of the step responses.

It could be observed that among all the controllers, the IWO based PID had the worst response. However, the IWO algorithm was run a second time with 200 iterations which drastically improved the response. The PID values obtained for the 1st order generator was 1.090, 0.2069 and 0.3296 respectively and for the 4th order generator: 10, 0.2500 and 0.9236.

From Table 3 and Fig. 4 we can observe the elimination of the steady-state error, the decrease in settling time and rise time upon using the controllers. For the 1st order generator, all the compensators including the ANFIS resulted in promising results. Unfortunately, IWO could not remove the overshoot. For the 4th order generator, the latter deteriorated the response by adding an overshoot of 51%.

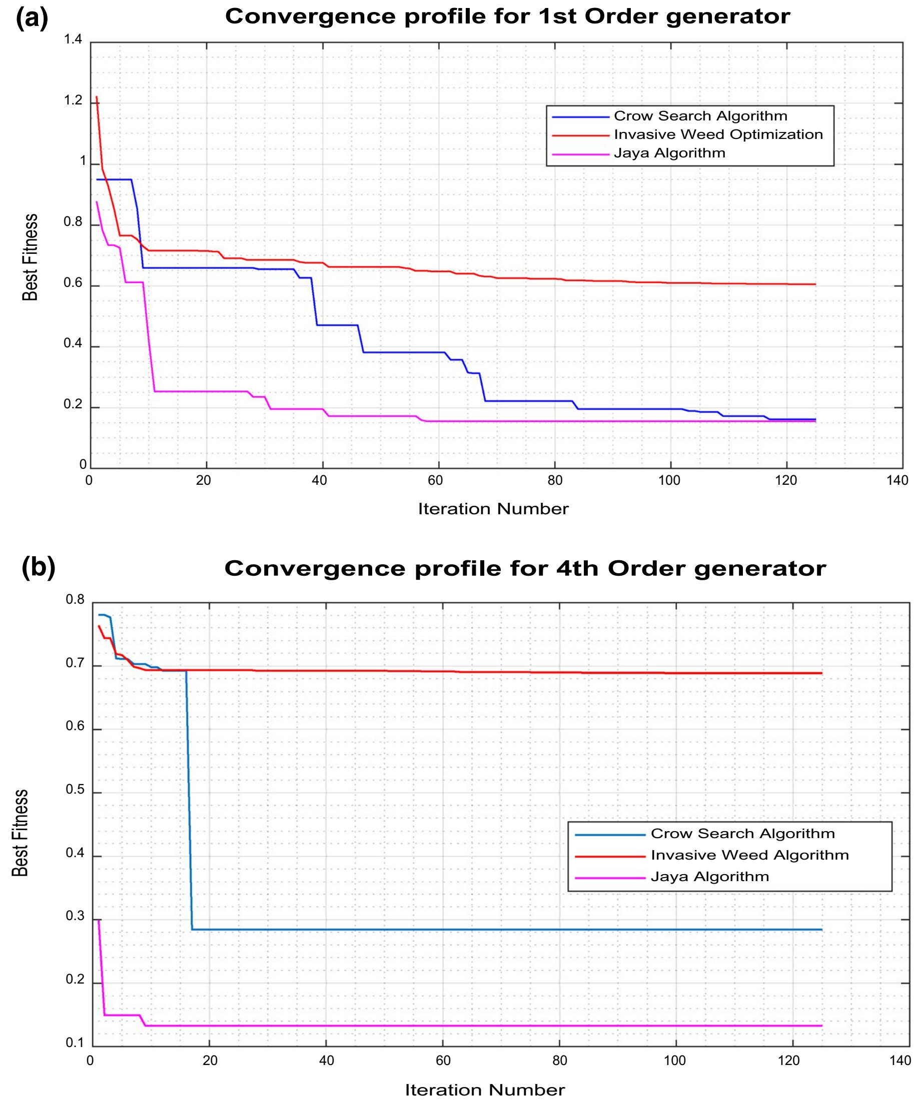
Figure 5a, b shows the convergence profile for IWO, JA and CSA for the 1st and 4th order generator. For both generators, IWO seems to converge to a value which is far from that of JA and CSA. JA proved to converge more rapidly and to the best fitness for both cases followed by CSA. However, for 200 iterations, IWO converged to a fitness value better than JA. This implies that JA is able to provide adequate performance for less iteration that IWO but if enough iterations are provided then IWO can ultimately attain better performances. Each algorithm follows a different principle and gives satisfactory results. This is supported by the fact the objective function used is the ITAE which specifies the quality of the whole response.

5 Conclusion

Two types of generator models, a first-order and a fourth-order, were used for the study of automatic voltage regulation using a type 1 excitation system with saturation functions. Four compensators were implemented in each model whereby three of them used optimization technique to find the parameters of a PID controller and one was based on artificial intelligence namely the neuro-fuzzy controller. The optimization techniques used were the JA, IWO, and CSA. The aim was to obtain a response with less overshoot, steady-state error and settling time by minimizing the ITAE objective function and the latter was achieved by all the controllers but with different degrees as some controllers performed better than others.

With a maximum iteration of 125, the controller that achieved the best step response was CSA based PID for the first order generator and JA based PID for the 4th order generator due to their overall performance.

Fig. 5 Convergence profile of the optimization algorithms



Compliance with ethical standards

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Soundarrajan A, Sumathi S, Sivamurugan G (2011) Voltage and frequency control in power generating system using hybrid evolutionary algorithms. J Vib Control 18(2):214–227
2. Gozde H, Taplamacioglu MC, Ari M (2017) Simulation study for global neighborhood algorithm based optimal automatic voltage regulator (AVR) system. In: 2017 5th International Istanbul smart grid and cities congress and fair (ICSG), Istanbul
3. Kiam AH, Gregory C, Yun L (2005) PID control system analysis, design, and technology. IEEE Trans Control Syst Technol 13(4):559–575
4. Jang J-SR (1993) ANFIS: adaptive-network-based fuzzy inference system. IEEE Trans Syst Man Cybern 23(3):665–685
5. Shokoohi S, Bevrani H, Naghshbandy HA (2013) Application of neuro-fuzzy controller on voltage and frequency stability in islanded microgrids. In: 2012 Conference on smart electric grids technology, Tehran
6. Omar B, Haikal A, Areed F (2011) Design adaptive neuro-fuzzy speed controllerfor an electro-mechanical system. Ain Shams Eng J 2:99–107
7. Pérez J, Gajate A, Milanés V, Onieva E, Santos M (2010) Design and implementation of a neuro-fuzzy system for longitudinal control of autonomous vehicles. In: International conference on fuzzy systems, Barcelona
8. Abdellah D, Djamel L (2015) Power flow analysis using adaptive neuro-fuzzy inference systems. In: 3rd International renewable and sustainable energy conference (IRSEC), Morocco

9. Agus J, Ramadoni S (2013) UPFC based on adaptive neuro-fuzzy for power flow control of multimachine power systems. Int J Eng Sci Invent 2(10):5–14
10. Bhatt VK, Bhongade S (2013) Design of PID controller in automatic voltage regulator (AVR) system using PSO technique. Int J Eng Res Appl (IJERA) 3(4):1480–1485
11. Gozde H., Taplamacioğlu CM (2014) Automatic voltage regulator (AVR) design with chaotic particle swarm optimization. In: ECAI 2014—international conference, 6th edn, ROMÂNIA
12. Gozde H, Taplamacioglu CM (2011) Comparative performance analysis of artificial bee colony algorithm for automatic voltage regulator (AVR) system. J Franklin Inst 348:1927–1946
13. Ramjug-Ballgobin R, Calchand SG (2017) Voltage control of a power system with exciter and generator saturation. Springer, New York, pp 176–183
14. Mehrabian AR, Lucas C (2006) A novel numerical optimization algorithm inspired from weed colonization. Ecol Inform 1:355–366
15. Oozeer MY, Ramjug-Ballgobin R (2018) A Jaya-based invasive weed optimization technique. In: International conference on emerging trends in electrical, electronic and communications engineering
16. Patel NC, Debnath MK, Sahu BK, Dash SS, Bayindir R (2018) Multi-staged PID controller tuned by invasive weed optimization algorithm for LFC issues. In 7th International conference on renewable energy research and applications, Paris
17. Mishra S, Barisal AK, Babu BC (2019) Invasive weed optimization-based automatic generation control for multi-area power systems. Int J Model Simul 39(3):190–202
18. Rao RV (2016) Jaya: a simple and new optimization algorithm for solving constrained and unconstrained optimization problems. Int J Ind Eng Comput 7(1):19–34
19. Purey P, Arya R (2017) Application of Jaya algorithm for reactive power reserve optimization accounting constraints on voltage stability margin. Int J Eng Trends Technol 51(2):106–114
20. Lakxhmi M, Ramesh AK (2018) Optimal reactive power dispatch using crow search algorithm. Int J Electric Comput Eng 8(3):1423–1431
21. Walton A (2000) A systematic method for the determination of the parameters of synchronous machines from the results of frequency response tests. IEEE Trans Energy Convers 15(2):218–223
22. Law BE (2001) Simulation of the transient response of synchronous machines. School of Information Technology and Electrical Engineering, University of Queensland, Brisbane
23. Enzhe S, Noor SIA, Naeim FM (2014) Transient response enhancement of high order synchrounous machine based on evolutionary PID controllers. Int J Control Autom 7(12):383–398
24. Report IC (1968) Computer representation of excitation systems. IEEE Trans Power Appar Syst 87(6):1460–1464
25. Maiti D, Acharya A, Chakraborty M, Konar A, Janarthanan R (2008) Tuning PID and PI/λDδ controllers using the integral time absolute error criterion. In: 2008 4th International conference on information and automation for sustainability, Colombo
26. Askarzadeh A (2016) A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. Comput Struct 169:1–12