Experimental Evaluation of Optimal Tuning for PID Parameters in an AVR System

F. Foroutan, and F. Tootoonchian*

Abstract—Automatic Voltage Regulator (AVR) is employed to stabilize the output voltage of the generators in the electric power plants. However, reliable performance of AVR depends on professional tuning of its PID controller’s parameters. Therefore, different optimization algorithms are used to determine those parameters. The objective of the optimization is defined as minimizing the characteristics of transient step response such as settling time, rise time, overshoot, and steady state error. Then, to verify the optimization results, a simulator is built experimentally for AVR and PID system that can also be used for other studies on AVR systems. The experimental results are compared with those of MATLAB and Pspace Software. Close agreement between the simulation and experimental results confirms the success of the optimization.

Keywords—Automatic Voltage Regulator (AVR), Proportional-Integral-Derivative (PID) controller, Population-Based Optimization Algorithms, Simulator, Transient Step Response

1. INTRODUCTION

Automatic Voltage Regulator (AVR) is essential equipment used in power systems. The main role of the AVR system is to control the output voltage of a synchronous generator in power plants by applying fast and transient changes to its exciter. However, the generator responses to these changes are usually slow because

Farshid Foroutan is with Railway Engineering Department, Iran University of Science and Technology (IUST), Tehran, Iran (Email: Farshidforoutan@alumni.iust.ac.ir)

*F. Tootoonchian (Corresponding author) is with Electrical Engineering Department, Iran University of Science and Technology (IUST), Tehran, Iran, (Email: tootoonchian@iust.ac.ir, +98-21-73225668, +98-9121792802).
it has high inductance and also its load varies fast [1]. In order to increase the AVR efficiency and improving its dynamic behavior, a Proportional-Integral-Derivative (PID) controller is added to the AVR system due to its easy implementation, robust performance and simple physical principle [2].

To achieve an appropriate closed-loop performance of the whole system, three parameters of the PID controller (K_p, K_i, and K_d) must be accurately tuned. Tuning methods of PID controller include trial-and-error, traditional and artificial methods. Trial-and-error and traditional approaches such as Zigeler and Nichols is not appropriate for tuning PID parameters due to its high overshoot and long oscillation in the step response [3]. Moreover, finding the best PID parameters using those two approaches is a time-consuming process due to massive calculations and the results are not always optimal [2]. To overcome the mentioned drawbacks, Artificial Intelligence (AI) methods are proposed. The objective function of such optimization process is usually defined as a combination of step response characteristics including minimizing the overshoot, rise time, settling time, and steady state error [4].

There are various AI methods for optimizing PID parameters in an AVR system and each one has its own merits and demerits [5]. Neural network, fuzzy system, and neural-fuzzy logic were three famous AI techniques [6–8]. These methods suffer from the problems of convergence time, training process and tuning the membership function [9]. Therefore, heuristic methods are welcomed to achieve higher performance. Among different heuristic algorithms, Genetic Algorithm (GA) [10] and Particle Swarm Optimization (PSO) [11] are widely used for tuning the parameters of PID controller. However, they have high computational burden [12], especially when there is a correlation between optimization parameters [13]. Therefore, many authors tried to modify these algorithms or combine them with other algorithms to improve their efficiency and obtain minimum step response characteristics (including overshoot, rise time, settling time, and steady state error) and convergence time. For example, authors of [14] combined Taguchi with PSO and they revealed that this new algorithm can tune PID parameters faster and better than PSO and Taguchi combined with GA in order to achieve the best AVR step response. Furthermore, a new Modified PSO (MPSO) algorithm was developed in [15] for a PID controller leading to better AVR step response in comparison with
conventional PSO algorithms in terms of improving computational efficiency and time complexity. In addition, multi-objective non-dominated shorting genetic algorithm [16,17], modified PSO algorithms so-called Velocity Update Relaxation Particle Swarm Optimization (VURPSO) and Craziness based Particle Swarm Optimization (CRPSO)[18], combined Taguchi with GA [19], Chaotic Particle Swarm Optimization (CPSO) [20], Particle Swarm Optimization with the Gravitational Search Algorithm (PSOGSA) [21], simplified Particle Swarm Optimization (PSO) also called Many Optimizing Liaisons (MOL) algorithm [22], and Adaptive PSO (APSO) [23] are other suggestions having faster and more efficient optimized AVR step response with respect to the conventional optimization algorithms.

Some other researchers proposed to use newly developed heuristic optimization methods to overcome the above-mentioned limitations of conventional methods. They claimed that they found a more optimum response by comparing the step response characteristics with those of GA and PSO in a faster optimization. Those algorithms include Monarch Butterfly Optimization Algorithm (MBO) [24], Taguchi method [25], Slap Swarm Algorithm (SSA) [26], Artificial Bee Colony (ABC) [27], Bacterial Foraging Technique (BFT) [28], Memetic Algorithm (MA) [29], Firefly Optimization Technique (FOT) [30], Shuffled Frog Leaping (SFL) [31], Continuous Action Reinforcement Learning Automata (CARLA) [32], Differential Evolution (DE) and Teaching-Learning-Based Optimization (TLBO) algorithms [33,34], Pattern Search Algorithm (PSA) [35], Simulated Annealing (SA) [36], Finite Gradient [37], Global Neighborhood Algorithm (GNA) [38], Imperialist Competitive Algorithm (ICA) [39], Gravitational Search Algorithm [40], Vector-Based Swarm Optimization (VBSO) [41], Continuous Human Learning Optimizer (CHLO) [42], Artificial Electric Field (AEF) [43], Whale Optimization Algorithm (WOA) [44], Cuckoo Search (CS) [45,46], Jaya Optimization Algorithm (JOA) [45], Ant Colony Optimization (ACO) [21], Chaotic Ant Swarm (CAS) algorithm [47], Chaotic optimization algorithm [48] and Grey Wolf Optimizer (GWO) [49].

Although there are a large number of researches on using different optimization methods for tuning parameters of PID controller for AVR system, it is still unknown that which of the proposed algorithms has the best performance considering the fast convergence time and the best tuning of PID parameters for AVR
step response. Also, based on our literature review, there are few articles, in which a number of optimization algorithms are compared to each other. They only chose one or two algorithms and made a comparison with one conventional method. However, in this paper, eight of the best population-based optimization methods are competing to establish the superiority of one over others. The studied optimization algorithms are Whale Optimization Algorithm (WOA), Ant Lion Optimizer (ALO), Slap Swarm Algorithm (SSA) and Dragonfly Algorithm (DA), and the results are compared with four conventional algorithms including Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Simulated Annealing (SA), and Artificial Bee Colony (ABC) that are used to improve the quality of the step response of the AVR system. Furthermore, in this paper, an electronic dual simulator is built for AVR system which was not done before. The prototype AVR simulator can be regarded as a preliminary linear modelling to analyze the dynamic behavior of a real AVR system. Finally, the electronic model of the PID controller is also built and the optimum parameters are experimentally tuned. The results of experimental test verify the success of the employed optimizations.

2. **System Description**

There are various AVR mathematical models and lately-developed nonlinear models are more accurate than linear models. However, due to this fact that this paper wants to take preliminary steps towards dual electrical modelling of the AVR, the basic linear model is considered. Fig. 1 shows the AVR system with its PID controller. As it can be seen, the AVR system has four main components including Amplifier, Exciter, Generator and Sensor. All components are modelled linearly with a gain of $K$ and a time constant of $\tau$. The amplifier is modeled with a gain of $K_A=10$ and a time constant of $\tau_A = 0.1$ sec. The linear model of the exciter includes a gain of $K_E =1$ and a time constant $\tau_E = 0.4$ sec. The generator gain could be changed from 0.7 to 1 and its time constant has a range between 1 to 2 seconds. In this paper, $K_G =0.7$ and $\tau_G =1$ sec. The last component of the AVR is sensor. The gain of the sensor model ($K_R$) is equal to 1 and its time constant ($\tau_R$) is assumed to be 0.01 seconds [2].

The step response of the mentioned AVR system is given in Fig. 2. As it can be seen in Fig. 2, it has large
amplitude oscillations with high overshoot. Such a response is not suitable for AVR’s step response. The characteristics of the studied step response shows the performance of AVR system is not suitable. It means that the amplitude of the overshoot is 50.46%, the rise time is 0.3174 seconds, the settling time is 4.90 seconds, and the steady state value is 0.873. Therefore, a PID controller must be added to improve the dynamic response of the AVR and decrease its steady state error.

The model of PID controller can be defined as:

\[ G_{PID}(s) = K_p + \frac{K_i}{s} + K_ds \]  

where \(K_p, K_i,\) and \(K_d\) denote the coefficients for the proportional, integral, and derivative terms respectively and must be tuned simultaneously to improve both transient and steady state response.

3. OPTIMIZATION

In this section the objective function of the optimization, the studied optimization techniques, and the optimization results are discussed.

3.1. Problem Formulation

\(K_p, K_i,\) and \(K_d\) are three parameters of PID controller that need to be tuned. On the other hand, an ideal step response has no overshoot, no steady state error, fast rise time and fast settling time. Therefore, the proposed objective function should include rise time, settling time, overshoot, and steady state error and the variables of the optimization are the parameters of the PID controller.

Integrated Absolute Error (IAE), the Integral of Squared-Error (ISE), or the Integrated of Time-Weighted-Squared-error (ITSE) are the usual forms of the objective function for tuning the PID parameters. However, they have disadvantages of being time-consuming and inaccurate in co-improvement of settling time and rise time. Therefore, a multi-objective optimization problem is defined as [50]:

\[ F_{obj} = (1 - e^{-\beta}) \times (M_p + ESS) + e^{-\beta} \times (t_s - t_r) \]
where $\text{ESS}$ is steady-state error, $M_p$ is amplitude of overshoot, $t_s$ is settling time, $t_r$ is rise time, and $\beta$ is a weighting factor which reflects the optimization performance. If you choose $\beta$ more than 0.7, you can reduce overshoot and steady state error. However, when $\beta$ is set under 0.7, you will be able to reduce the rise time and settling time. Thus, according to the results achieved by the reference paper [50], the range of $\beta$ is between 0.8 and 1.5. Since the objective function of (2) is able to consider all the required factors of ideal step response, it is recommended for the optimization in this paper.

3.2. Optimization Methods

As mentioned earlier eight different optimization methods are competing in this paper to minimize the objective function of (2). The studied optimization algorithms are Whale Optimization Algorithm (WOA) [51], Ant Lion Optimizer (ALO) [52], Slap Swarm Algorithm (SSA) [53,54], Dragonfly Algorithm (DA) [55], Genetic Algorithm (GA) [56], Particle Swarm Optimization (PSO) [11,57], Simulated Annealing (SA) [36], and Artificial Bee Colony (ABC) [27,58]. It is worth mentioning that the optimization variables are the parameters of PID controller ($K_p$, $K_i$, and $K_d$).

3.3. Optimization Results

Fig. 3 shows the variation of objective function versus different iterations using the studied optimization algorithms. The maximum iteration is set to 200 for all algorithms except GA and SA. GA termination criteria was different from other algorithms based on MATLAB Optimtool and it could achieve its optimum response at maximum 60 iterations. Moreover, SA could not optimized the objective function in less than four digits after the decimal point after 457 iterations, but we set the maximum iterations at 2000 to evaluate its behavior. Among the studied algorithms, PSO and WOA are faster than other algorithms in getting the final optimum point. However, ALO has the lowest value of objective function ($F_{\text{obj}}=0.0806$) by tuning $K_p$, $K_i$, and $K_d$ equal to 0.8593, 0.6076, and 0.2919, respectively and it is regarded as the best response for tuning PID controller parameters. Although ALO has the optimum value of objective function, the lowest amount of settling time, 0.4962 sec, lies with ABC and the lowest amount of rise time, 0.3113 sec are devoted to the results of GA
optimization method.

The optimal value of PID parameters, optimal value of objective function, and the characteristics of the step response are presented in Table 1, for different optimization methods. According to Table 1, GA and PSO are the worst optimization techniques with highest value of the objective function. It means the recent population-based optimization algorithms are more efficient than conventional ones.

Furthermore, Table 1 shows that all algorithms could remove the maximum overshoot from the step response of the AVR system. It means the competition for best results is referred to lowest rise time and settling time and fast convergence of the method.

The step Response of the AVR system with tuned parameters of PID controller based on ALO results is shown in Fig. 4.

4. Experimental Evaluation

Experimental measurement is always the best verification method. However, the AVR system of power plants is not easily available for such measurements. Therefore, in this section a simulator is developed for AVR and its PID controller. The proposed simulator is designed and simulated using ORCAD Family software and then experimentally built.

4.1. Developing a Simulator

In order to develop a simulator for AVR system and its PID controller, it is required to find the best electronic circuit of them. Regarding the transfer function of the AVR system in Fig. 1, it is obvious that this function is the product of multiplying several first-degree Resistor-Capacitor (RC) circuits. The product of RCs represents time constant (τ) for each component and all gains can be produced by a multiplier comprises of an operational amplifier (OP AMP).

The proposed electronic model of the AVR with PID controller is given in Fig. 5. The employed OP AMP is LM324 that is easy-to-access in the electronic market.

The transfer function of PID controller, as the ratio of its output voltage to input voltage is defined as:
\[
\frac{V_o(s)}{V_i(s)} = \frac{R_2}{R_1} \frac{(R_1 C_1 s + 1)(R_2 C_2 s + 1)}{R_2 C_2 s}
\]  
(3)

where \( V_o(s) \) is the output voltage of the PID controller, \( V_i(s) \) is the input voltage of the PID controller, \( R_1 / C_1 \) is the resistor/capacitor of the derivative term and \( R_2 / C_2 \) is the resistor/capacitor of the integral term.

Since the transfer function of PID controller includes a derivative operator, an integral operator and a sign inverter, \( K_p \), \( K_i \) and \( K_d \) could be derived from (3) as:

\[
K_p = \frac{R_2}{R_1} \left( \frac{R C_1}{R_2 C_2} + 1 \right)
\]

(4)

\[
K_i = \frac{1}{R C_2}
\]

(5)

\[
K_d = R_2 C_1
\]

(6)

Substituting the optimal value of \( K_p \), \( K_i \) and \( K_d \) from Table 1 into (4)-(6), and supposing \( C_1=10 \mu F \), \( R_2 \) is determined equal to 29.19 kΩ. Then, considering the known parameters and the fact that \( \Delta = K_p^2 - 4 \times K_i \times K_d \) must be positive to get real numbers, \( R_1 \) and \( C_2 \) can be determined. It should be mentioned that there are two acceptable values for \( R_1 \) and \( C_2 \) because they are the result of solving a second-degree equation which has two valid answers. Considering \( R_1 = 84.71 kΩ \), \( C_2 \) will be calculated equal to 19.42 \( \mu F \) and for \( R_1 = 56.71 kΩ \), \( C_2 = 29.02 \mu F \).

The PSpice step response of the proposed AVR system with and without PID controller is presented in Figs. 6 and 7, respectively.

4.2. Experimental Implementation of Simulator

The developed simulator is experimentally built as shown in Fig. 8. A digital oscilloscope is used to capture and save the step response of the simulator. Considering a step DC input voltage of 1 V, the measured step response of the simulator without and with PID controller is given in Figs. 9 and 10, respectively.
Comparing step response characteristics of the experimental results with those of PSpice and MATLAB are shown in Table 2. According to Table 2, the developed simulator is a suitable representative for an AVR with PID controller in power plants. Before using a PID controller the maximum peak/ steady state value of the step response is measured equal to 1.32 V /0.8 V that are equal to the predictions of simulation results. After adding PID controller, the rise time and the settling time of the measured response are in agreement with simulation results. The difference between the results of PSpice and those of experimental test and MATLAB is referred to the tolerance of the employed capacitors and resistances.

5. CONCLUSION
In this paper, parameters of Proportional-Integral-Derivative (PID) controller for an Automatic Voltage Regulator (AVR) were optimally tuned. The objective function of the optimization was defined in such a way to be consider all the performance characteristic of the step response to ensure, minimum rise and settling time, minimum overshoot amplitude and steady state error along with fast convergence time. The ability of well-known optimization methods includes Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) were compared with that of newly proposed evolutionary algorithms includes Whale Optimization Algorithm (WOA), Ant Lion Optimizer (ALO), Slap Swarm Algorithm (SSA), Dragonfly Algorithm (DA) Simulated Annealing (SA), and Artificial Bee Colony (ABC). Then, to experimentally evaluate the success of the best optimization results, a simulator was developed and experimentally built for AVR system and its PID controller. Comparing the experimental results with those of simulation by MATLAB and ORCAD Family software confirmed the success of the proposed optimization and developed simulator.

REFERENCES


**Farshid Foroutan** received his Master’s and Bachelor’s degree both from Iran University of Science and Technology (IUST) in Electrical Railway Engineering and Power Engineering, respectively. He entered his master’s program directly as a talented student in 2014 and he graduated with the highest Cumulative GPA among 13 students of the department in 2017. He has mainly worked in renewable energy and electricity market optimization for the recent years and also he has published a number of papers and books in this regard. Now, He is working as an Electricity Market Engineer at Clean Energy Development Company, which is the first and greatest Iran’s electricity retailer.

**Farid Tootoonchian** received the B.Sc. and M.Sc. degrees in electrical engineering from the Iran University of Sciences and Technology, Tehran, Iran, in 2000 and 2007, respectively, and the Ph.D. degree from the K. N. Toosi University of Technology, Tehran, in 2012. He is currently an Associate Professor with the Department of Electrical Engineering, Iran University of Sciences and Technology. His research interests include design, optimization, finite-element analysis, and prototyping of ultrahigh-speed electrical machines and ultrahigh-precision electromagnetic sensors.

Fig. 1. Block diagram of an AVR system with a PID controller

Fig. 2. The AVR step response without a PID controller

Fig. 3. The optimization trend of population-based optimization algorithms

Fig. 4. The step response of the AVR with optimum parameters of PID controller, based on the results of ALO algorithm

Fig. 5. Proposed electronic model of the AVR with PID controller

Fig. 6. Step response of the AVR electronic model in PSpice

Fig. 7. Step Response of the AVR with PID electronic model in PSpice

Fig. 8. The experimentally built simulator of AVR system and its PID controller

Fig. 9. The measured step response of AVR system’s simulator without PID controller

Fig. 10. The measured step response of AVR system’s simulator with PID controller

Table 1. PID parameters and step response characteristic for AVR system using different optimization techniques

Table 2. Comparing the step response of the AVR and PID controller between experiment, MATLAB and PSpice
Figures

Fig. 2. Block diagram of an AVR system with a PID controller

\[ V_{ref}(s) \rightarrow V_{e}(s) \rightarrow K_p + \frac{K_i}{s} + K_ds \rightarrow 10 \frac{1}{1 + 0.1s} \rightarrow 10 \frac{1}{1 + 0.04s} \rightarrow 0.7 \frac{1}{1 + s} \rightarrow V_t(s) \]

\[ V_s(s) \rightarrow \text{PID Controller} \rightarrow \text{Amplifier} \rightarrow \text{Exciter} \rightarrow \text{Generator} \rightarrow \text{Sensor} \]

Fig. 2. The AVR step response without a PID controller

Fig. 3. The optimization trend of population-based optimization algorithms
Fig. 4. The step response of the AVR with optimum parameters of PID controller, based on the results of ALO algorithm.

Fig. 5. Proposed electronic model of the AVR with PID controller.
Fig. 6. Step response of the AVR electronic model in PSpice

Fig. 7. Step Response of the AVR with PID electronic model in PSpice
Fig. 8. The experimentally built simulator of AVR system and its PID controller

Fig. 9. The measured step response of AVR system’s simulator without PID controller
Fig. 10. The measured step response of AVR system’s simulator with PID controller

Table 1. PID parameters and step response characteristic for AVR system using different optimization techniques

<table>
<thead>
<tr>
<th></th>
<th>GA</th>
<th>SA</th>
<th>ABC</th>
<th>PSO</th>
<th>WOA</th>
<th>SSA</th>
<th>ALO</th>
<th>DA</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_p$</td>
<td>0.8679</td>
<td>0.8254</td>
<td>0.8611</td>
<td>0.8648</td>
<td>0.8258</td>
<td>0.8403</td>
<td>0.8593</td>
<td>0.7787</td>
</tr>
<tr>
<td>$K_i$</td>
<td>0.6055</td>
<td>0.5715</td>
<td>0.6014</td>
<td>0.6026</td>
<td>0.6325</td>
<td>0.5539</td>
<td>0.6076</td>
<td>0.5219</td>
</tr>
<tr>
<td>$K_d$</td>
<td>0.3050</td>
<td>0.2685</td>
<td>0.2916</td>
<td>0.2924</td>
<td>0.2811</td>
<td>0.2714</td>
<td>0.2919</td>
<td>0.2398</td>
</tr>
<tr>
<td>Objective function</td>
<td>0.3669</td>
<td>0.0869</td>
<td>0.0811</td>
<td>0.2206</td>
<td>0.0807</td>
<td>0.0871</td>
<td><strong>0.0806</strong></td>
<td>0.0963</td>
</tr>
<tr>
<td>Percentage overshoot (%)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Rise time (sec)</td>
<td><strong>0.3113</strong></td>
<td>0.3415</td>
<td>0.3119</td>
<td>0.3254</td>
<td>0.3326</td>
<td>0.3366</td>
<td>0.3193</td>
<td>0.3740</td>
</tr>
<tr>
<td>Settling time (sec)</td>
<td>1.1283</td>
<td>0.5330</td>
<td><strong>0.4962</strong></td>
<td>0.5095</td>
<td>0.5201</td>
<td>0.5236</td>
<td>0.4965</td>
<td>0.5863</td>
</tr>
<tr>
<td>Required iteration</td>
<td>55</td>
<td>457</td>
<td>79</td>
<td>26</td>
<td>136</td>
<td>166</td>
<td>151</td>
<td>142</td>
</tr>
</tbody>
</table>

**TABLES**
Table 2. Comparing the step response of the AVR and PID controller between experiment, MATLAB and PSpice

<table>
<thead>
<tr>
<th>Performance characteristics</th>
<th>Experimental test</th>
<th>Simulation results using MATLAB</th>
<th>Simulation results using PSpice</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AVR without PID</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>First peak Voltage (V)</td>
<td>1.32</td>
<td>1.32</td>
<td>1.32</td>
</tr>
<tr>
<td>First Peak time (sec)</td>
<td>0.84</td>
<td>0.877</td>
<td>0.926</td>
</tr>
<tr>
<td>First Valley Voltage (V)</td>
<td>0.64</td>
<td>0.638</td>
<td>0.644</td>
</tr>
<tr>
<td>First Valley time (sec)</td>
<td>1.56</td>
<td>1.64</td>
<td>1.62</td>
</tr>
<tr>
<td>Steady State Voltage (V)</td>
<td>0.89</td>
<td>0.875</td>
<td>0.880</td>
</tr>
<tr>
<td><strong>AVR with PID</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rise Time (sec)</td>
<td>0.36</td>
<td>0.3193</td>
<td>0.46</td>
</tr>
<tr>
<td>Settling Time (sec)</td>
<td>0.69</td>
<td>0.4965</td>
<td>1.5</td>
</tr>
<tr>
<td>Steady State Value (V)</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>