Design, Analysis and Tuning of an Optimal Controller for an AVR: A Simulation Approach

Dr. Satyabrata Sahoo and Dr. V. Mahesh

Department of Electrical and Electronics Engineering, Nalla Malla Reddy Engineering College Hyderabad-500088, India

E-mail: sahoo.eee@nmrec.edu.in

Keywords: AVR system, Bacteria Foraging, Particle Swarm Optimization, Bacterial Foraging adapted Particle swarm Optimization, Optimal Control, PID Controller

Abstract. Terminal voltage of the generator keeps changing when there is a mismatch between generator supply and load demand. Particularly when load increases terminal voltage of the generator decreases. This problem can be mitigated by the use of reactive power but it leads to the damage of stator winding. To overcome this Automatic voltage regulator (AVR) is used. But it is not robust. To enhance the stability and working of an AVR system, an optimal PID controller is used. In this paper, a combination of Bacteria Foraging (BF) and Particle Swarm Optimization (PSO) is used for regulating the strictures of PID controller of an AVR to get the better optimization parameter values. The performance of the algorithm is studied using MATLAB and is equated with the Bacterial Foraging algorithm and Particle swarm Optimization. Simulation outcome evidently clarify that the suggested approach is very effective and efficient as compared to further population centered global optimization algorithms.

1. Introduction

In an electrical power station, a number of alternators are associated to a shared bus bar and every one of these alternators has an automatic voltage regulator (AVR). The foremost target is to regulate the alternator terminal voltage [1]. Many control methods are used in the AVR such as neural control, fuzzy control and adaptive control. Among these, the best proven is the PID regulator, which is commonly utilized in the industry since of its simple construction & robust performance in extensive range of working condition. But, it has been quite tough to tune properly the gains of PID controllers [2-6]. Due to scheme disruptions the electrical oscillations can happen for a long time and might end in system instability. Thus to overcome these problems, operative control algorithms are essential.

To regulate a PID controller in an AVR, a hybrid Genetic Algorithm and Bacterial Foraging Algorithm is utilized in [8]. In [7] a compensator of lag-lead type is used in an AVR based power system stabilizer and its strictures are tuned with a multi-objective optimization approach. Craziness centered PSO is found in [9] to adjust the PID parameters for a Power System Stabilizer (PSS) regulated AVR scheme. Chaotic Ant swarm based optimization technique is recommended for tuning
the AVR parameters in [10]. A PSO based fuzzy logic controller is developed in [11] for two area automatic generation control restructured power system.

In this work, for tuning the parameters of PID controller of an AVR system, an amalgamation of new type of Particle Swarm Optimization (PSO) algorithm is used. This paper structured as follows: In section II, mathematical model of an AVR system is described first. Then motivation of combining BF and PSO for PID controller tuning is discussed in section III. In section IV, V and VI, basics of PSO, BF and BF oriented PSO is reviewed. Simulation results of different optimization techniques used for AVR system control is discussed in section VII, followed by the conclusions in section VIII.

2. Mathematical model of an AVR system

2.1. Automatic Voltage Regulator system modeling

The terminal voltage of an alternator should keep within a prescribed limit in spite of the variation of load. This can be possible by the use of an automatic voltage regulator system. A simplified AVR system block diagram with PID controller is shown in Fig. 1. It comprises of mainly four parts i.e. sensor, amplifier, exciter & generator. Each of these parts are represented by linear transfer function models and they are

Model of an Amplifier using transfer function

\[
\frac{V_a}{V_d} = \frac{\zeta_a}{1 + \tau_a s}
\]  

(1)

Model of an Exciter using transfer function

\[
\frac{V_e}{V_a} = \frac{\zeta_e}{1 + \tau_e s}
\]  

(2)

Model of a Generator using transfer function

\[
\frac{V_g}{V_e} = \frac{\zeta_g}{1 + \tau_g s}
\]  

(3)

Model of a Sensor using transfer function

\[
\frac{V_s}{V_g} = \frac{\zeta_s}{1 + \tau_s s}
\]  

(4)

The detailed description about above equation symbols and values are given in Table. I

| Parameters   | Name of Parameter                          | Range                       |
|--------------|--------------------------------------------|-----------------------------|
| \(\xi_a, \xi_e, \xi_g \& \xi_s\) | Gain of the Amplifier, Exciter, Generator and Sensor | 10 – 40, 01-10, 0.7-01 and 01- 02. |
| \(\tau_a, \tau_e, \tau_g \& \tau_s\) | Amplifier, Exciter, Generator and Sensor time constant | 0.02-1s, 0.4-1s, 1-2s and 0.001-0.06 |

2.2. PID Controller

For the adequate operation of a system, accurate selection of PID controller parameters is necessary. In fact it is used to remove the steady state error and to improve the dynamic output of the system. It also calculates the error value by utilizing the difference between a required set point and a measured process variable. The transfer function of a PID controller is given as

\[
G(s) = G_p + \frac{K_i}{s} + K_d
\]  

(5)
where $K_d$, $K_i$ and $K_p$ are the derivative, integral and proportional gain of the controller. By proper tuning of these gains, the rise time, the error at steady state and stability limit of the scheme is controlled.

An AVR scheme transfer function is displayed in figure 1 and is written by

$$\frac{V_g(S)}{V_{Ref}(S)} = \frac{\zeta_a\zeta_e\zeta_g(1+\tau_p S)}{(1+\tau_a S)(1+\tau_e S)(1+\tau_g S)+\zeta_a\zeta_e\zeta_g\zeta_s}$$

(6)

**Figure 1.** Block figure of PID controlled AVR

3. Motivation of combining BF & PSO for PID tuning

The motivation reasons are

- Evolutionary Computation procedure is a population centered search method. This works thru the population of strings which signify diverse potential solutions.
- When this algorithm is applied to any complex optimization problems, it enhances its searching ability and locates the optimal value quickly.
- Herein, the Bacterial Foraging adapted particle swarm Optimization pattern is utilized for determining the unidentified parameters of PID regulator through minimization of cost function of the scheme.
- Here the optimized output values of the unknown parameters based on BF-PSO takes less simulation time as compared to BF & PSO, without decreasing efficacy of the system.

4. Basics of Particle Swarm Optimization

The model of a Particle Swarm Optimization comprises a swarm of particles. Thru a population of arbitrary candidate solutions, these particles are initialized.

To find the new solutions, they travel and search iteratively in the space dimension. A position vector $X^i_k$ is designated for each particle to represent its position (where the particle index is $i$) and a velocity denoted thru a velocity vector $U^i_k$. Every particle recollect their own best position $P^i_{local}$. The finest position of the vector amongst the swarm then kept in a new vector $P^i_{global}$. Throughout this time of iteration $k$, the revise of the velocity from the preceding velocity to the fresh velocity is governed thru

$$U^i_{k+1} = U^i_k + C_1 R_1 (P^i_{local} - X^i_k) + C_2 R_2 (P^i_{global} - X^i_k)$$

(7)

Particles latest location is analyzed by comprising the preceding location and the latest velocity

$$X^i_{k+1} = X^i_k + U^i_{k+1}$$

(8)

where $R_1$ and $R_2$ represent any random numbers. These are determines the velocity and location of the particle. Based on the most successful particle’s own experience and best past position memory, a particle chooses where to travel next.
5. Basics of Bacterial Foraging Optimization

To eradicate poor foraging approaches and to develop it to effective foraging strategies, the selection behavior of bacteria plays a crucial role. After several productions a foraging bacteria yields actions to exploit the energy acquired during time utilized in foraging

Due to this action of foraging, investigators utilized it in their process of optimization. The *Ecoli* bacterium has a property of control system that allows it to find their food and try to duck the noxious ingredients. The following four stages are describe the bacteria distributed motion model:

5.1. *Swarming and Tumbling* ($M_s$)

The flagellum design is a left handed helix. It is organized in such a way that it rotate in counter clockwise and its base is attached to the cell, as given in figure 2-a.

By observing near the cell to the open end of the flagellum, it generates a power opposite the bacterium thrusting the cell. This type of movement is named as swimming. Based on the nutrition concentration and environment situation, bacterium swims for highest amount of steps $M_s$ or less. Every flagellum drags on the cell when it rotates in clockwise as shown in figure. 2-b. Then the net result is that every flagellum works quite easily of the others and so the bacterium ‘tumble’. Tumbling type reveals a variation in the upcoming swim track.

![Figure 2. Swimming or tumbling of an *Ecoli* bacteria](image)

5.2. *Chemotaxis* ($M_c$)

A number of significance swim steps followed by tumble constitute a chemotactic step. The highest of swim step through a chemotetic step is designated as $M_c$. Actually the environment decides the real number of swim steps. In the direction of the swim, if the atmosphere indicates suitable nutrients attention then the *Ecoli* bacteria swims further step. The finish of the chemotactic step is governed by either getting a poor environment or by getting the highest number of steps $M_c$. A tumble action takes place when the swim stages are closed

An arbitrary unit length vector through the route Delta ($n, i$) is created for representing a tumble. Here index of bacterium is $i$, which has the highest quantity of bacteria $\sigma$ and $j$ is the index for the chemotactic step. After a tumble, this vector is utilized to outline the way of movement. Let $M_c$ is the bacteria lifetime length which is computed by the amount of chemotaxis step taken thru their lifetime. Let $C_i>0, i = 1, 2, \ldots, \sigma$ represent the chemotetic step magnitude which defines the length of the step through runs. Here the magnitude of the step is presumed to be constant. Every bacterium location is indicated by $P(n, i, j, k, dlll)$. Here the length of explore space is $n$, the reproduction index step is $k$
and the index of abolition dispersal events is $dll$. The position of fresh bacterium just after tumbling is specified as

$$p_{n,i+1,k,dll} = p_{n,j,k,dll} + \text{Delta} \ast C^l \quad (9)$$

5.3. Reproduction ($M_{re}$)

A reproduction step is chosen after $N_c$ chemotactic steps. Here the amount of reproduction steps is chosen as $M_{re}$. Presume that $\sigma$ is a +ve even integer for ease and the population of members in number is $\sigma_r = \sigma/2$. The members have adequate nutrients so that they will replicate with no mutations. For regeneration, the population is organized in order of rising accrued value (superior accrued value depicts that it is not able to acquire as much nutrients through its foraging period. Hence it is not 'healthy' and subsequently it will not reproduce properly). During this reproduction minimal healthy bacterium dies are $\sigma_r$ and at the same location the healthiest bacteria $\sigma_r$ split in to two bacteria for each member

5.4. Elimination and dispersal ($M_{ed}$)

Elimination process is takes place here also. For instance, substantial rise in heat eradicate a population of bacteria which are presently through an area of high intensity of nutrients. Disperse of bacteria from one region to another takes place with an abrupt flow of water. The chemotactic progress is destroyed by the consequence of eradication and dispersal events. But it has an assisting result in Chemotaxis, because dispersal can place bacteria close to sources of good food. For control applications, the bacteria foraging algorithm is used in adaptive control [2] and estimation of harmonics in a signal distorted thru additive noise [3]. In [4] to regulate a PID controller of an automatic voltage regulator, the mixture of genetic algorithm and bacteria foraging is utilized. In this work an E. coli is utilized for regulating a PID controller of the plant transfer function. Its solutions are stated.

6. Bacterial Foraging Optimization oriented by PSO

The BF-PSO unites both the algorithms i.e. BF and PSO. In this combination, BF is able to find a fresh outcome by eradication and dispersal, while PSO is able to exchange the social information.

For initialization, the user selects $\sigma$, $M_s$, $M_c$, $M_{re}$, $M$, $P_{ed}$, $C_1$, $C_2$, $R_1$, $R_2$ and $c(i), i = 1, 2 \ldots \sigma$. Also initialize the Position $p_{n,1,1,1,i} = 1, 2 \ldots \sigma$ and Velocity. They are initialized with an arbitrary way. The BF-PSO replicas with bacteria Population Chemotaxis, swarming, regeneration, abolition and dispersal tuned by PSO is placed under (Initially, $j = k = dll = 0$). For the ease of analysis dropping of implicit subscrbe is considered.

6.1. Initialize parameters $n, \sigma, M_s, M_c, M_{re}, M, P_{ed}, c(i), i = 1, 2 \ldots \sigma, \Delta, C_1, C_2, R_1, R_2$.

- $n$: Search space dimension,
- $\sigma$: The population size of bacteria in integer,
- $\sigma_r$: 50% of all the bacteria available,
- $M_s$: Swim length largest number,
- $M_c$: Steps of chemotactic,
- $M_{re}$: Reproduction steps in number,
- $M_{ed}$: Events of elimination and dispersal,
- $P_{ed}$: Probability of elimination and dispersal,
- $c(i)$: Size of the step chosen for the random direction,
- $C_1, C_2$: Random parameters of PSO,
- $R_1, R_2$: Random parameters of PSO
6.2. Create an arbitrary path i.e., Delta \((n, i)\) and position

\[
\text{For (dll=1 to Med)} \\
\text{For (k=1 to Mre)} \\
\text{For (j=1 to Mc)} \\
\text{For (i=1 to } \sigma) \\
\text{Calculate the objective functions} \\
\Gamma(i, j) = \text{Func}(P(i, j)) \\
\text{Depot the finest objective functions in } \Gamma_{\text{last}} \\
\Gamma_{\text{last}} = \Gamma(i, j) \\
\text{For each bacteria the finest object will be chosen as the local best } \Gamma_{\text{local}} \\
\Gamma_{\text{local}}(i, j) = \Gamma_{\text{last}}(i, j) \\
\text{Upgrade the location and objective functions} \\
P(i, j + 1) = P(i, j) + C(i) \cdot \text{Delta}(n, i) \\
\Gamma(i, j + 1) = \text{Func}(P(i, j + 1)) \\
\text{while } (m < M_s) \\
\text{If } \Gamma(i, j + 1) < \Gamma_{\text{last}} \text{ then} \\
\Gamma_{\text{last}} = \Gamma(i, j + 1) \\
\text{Upgrade the location and objective functions} \\
P(i, j + 1) = P(i, j + 1) + C(i) \cdot \text{Delta}(n, i) \\
\Gamma(i, j + 1) = \text{Func}(P(i, j + 1)) \\
\text{For each bacteria estimate the present location and local objective function} \\
P_{\text{current}}(i, j + 1) = P(i, j + 1) \\
\Gamma_{\text{local}}(i, j + 1) = \Gamma_{\text{last}}(i, j + 1) \\
\text{else} \\
P_{\text{current}}(i, j + 1) = P(i, j + 1) \\
\Gamma_{\text{local}}(i, j + 1) = \Gamma_{\text{last}}(i, j + 1) \\
\text{end if} \\
m = m + 1 \\
\text{end while} \\
\text{next i (next bacteria)} \\
\text{For each bacteria estimate the local finest solution } (P_{\text{best}}) \text{ and global best position } (P_{\text{gbest}}). \\
\text{For every bacteria estimate the new direction} \\
U = \omega \cdot U + C_1 \cdot R_1(P_{\text{best}} - P_{\text{current}}) + C_2 \cdot R_2(P_{\text{gbest}} - P_{\text{current}}) \\
\text{Delta} = U \\
\text{Next j (next chemotactic)} \\
\text{For (i = 1 to } \sigma) \\
\Gamma_{\text{health}}(i, j, k, dll) = \sum_{j=1}^{M_c+1} (i, j, k, dll) \\
\text{End} \\
The } \sigma \text{ bacteria with the highest } \Gamma \text{ health remove and the other } \sigma \text{ bacteria with the finest values copies.} \\
\text{next k (next reproduction)} \\
\text{with Probability } P_e, \text{ eliminates and disperse every bacterium.} \\
\text{next dll (next elimination)} \\

7. Simulation results

To run the SIMULINK, the following parameters of an AVR are to be assumed. They are \(\zeta_a = 10, \tau_a = 0.1, \zeta_e = 1, \tau_e = 0.4, \zeta_{\epsilon} = 1, \tau_{\epsilon} = 1, \zeta_5 = 1, \tau_5 = 0.05\).

The root locus plot is shown in figure 3. From this figure we found the system is marginally stable for \(\zeta_a = 10\). Thus the amplifier gain is selected to 10 within the constraints of 10 to 40. From the figure
4, we found the system response is extremely oscillatory thru a very high overshoot and a lengthy settling time. Moreover the steady state error is also high.

Table II. Parameter values of BF-PSO

| Sl. No | Parameters                             | Values |
|-------|----------------------------------------|--------|
| 01    | No. of Bacteria($\sigma$)              | 10     |
| 02    | No. of Chematactic Steps ($M_C$)       | 03     |
| 03    | Length of Swim ($M_s$)                 | 04     |
| 04    | Reproduction Steps in No. ($M_{re}$)   | 04     |
| 05    | Elimination and Dispersal Events in No. ($M_{ed}$) | 02     |
| 06    | No. of Bacteria Reproduction (splits) per generation ($\sigma_r$) | $\sigma/2$ |

Figure 5 represents the Step output of an AVR terminal voltage with PID controller where every PID parameters are presumed to be 1.

Figure 3. Root-locus plot of an AVR system

Figure 4. Step output of an AVR terminal voltage without Controller
To simulate this case it is found the time domain performances are not satisfactory. To get satisfactory performance, the PID parameters must be tuned. In our work B-PSO is implemented and simulated for tuning of PID parameters.

**Figure 5.** Step output

**Figure 6.** Step output of an AVR terminal voltage using BF-PSO algorithm with different $\beta$ values

**Figure 7.** Step output of an AVR terminal voltage utilizing different controllers ($\beta=0.5$, generations=150)
Then we found BF-PSO based PID controller gives the better results. For the different values of beta (β=0.5,1,1.5) the optimized results using BF-PSO shown in the figure 6.

![Figure 8](image1)

**Figure 8.** Step output of an AVR terminal voltage using Different controllers (β=1.0, generations=150)

![Figure 9](image2)

**Figure 9.** Step output of an AVR terminal voltage using Different controllers (β=1.5, generations=150)

### Table III. BF-PSO tuned values for PID

| Sl. No | β   | K_p  | K_d  | K_i  |
|--------|-----|------|------|------|
| 01     | 0.5 | 0.8095 | 0.2425 | 0.2447 |
| 02     | 1.0 | 0.5463 | 0.6183 | 0.2130 |
| 03     | 1.5 | 0.7878 | 0.2622 | 0.1650 |

### Table IV. BF and PSO tuned values of PID

| Sr. No | β   | K_p  | K_d  | K_i  | K_p  | K_d  | K_i  |
|--------|-----|------|------|------|------|------|------|
| 01     | 0.5 | 0.9621 | 0.3148 | 0.9621 | 0.3148 | 0.9621 | 0.3148 |
| 02     | 1.0 | 1.0736 | 0.2941 | 1.0736 | 0.2941 | 1.0736 | 0.2941 |
| 03     | 1.5 | 1.0235 | 0.4556 | 1.0235 | 0.4556 | 1.0235 | 0.4556 |
In the above figure we found, BF-PSO gives the better results at $\beta=1$. For the different values of beta in figure 7, 8 & 9, the BF-PSO based PID controller gives better performance than BF & PSO while satisfying the time domain performance within the permissible limits.

8. Conclusion

This paper projected a novel amalgam approach consisting of BF (Bacterial Foraging) with Particle Swarm Optimization (PSO). In table 3 and 4 diverse values of beta are taken for simulation purpose. The results obtained through the proposed BF-PSO controller of an applied AVR system with an effective search for the optimal PID controller parameters are the best ones.

The output response overshoot magnitude is effectively reduced by utilizing the aforementioned two techniques. The PID controller response regulated through BF-PSO algorithm is robust and has faster response.

References

[1] Tang Y., Zhao L., Han Z. 2016 *Int. J. Mach. Learn. & Cyber.* 7 229–240
[2] Gizi A. J. H. 2019 *Soft Comput.* 23 8839–8853
[3] Gozde H., Taplamacioglu M.C. 2011 *Journal of the Franklin Institute* 348(8) 1927-1946
[4] Panda M. K., Pillai G. N., Kumar V. 2011 *Electric Power Components and Systems* 40(2) 219-235
[5] Gupta T., Sambariya D. K. 2017 *International Conference on Information, Communication, Instrumentation and Control (ICICIC) Indore India*
[6] Aguila N., Manuel C., Mermoudab D. 2013 *ISA Transactions* 52(6) 807-815
[7] Obaid Z. A., Cipcigana L. M., Muhssin M. T. 2017 *Renewable and Sustainable Energy Reviews* 79 839-849
[8] Kim D. H., Abraham A. 2007 *Studies in Computational Intelligence Springer Berlin Heidelberg*
[9] Chatterjee A., Mukherjee V., Ghoshal S. 2009 *International Journal of Electrical Power & Energy Systems*, 31(7) 323-333
[10] Pan I., Das S. 2012 *International Journal of Electrical Power & Energy Systems* 43(1) 393-407
[11] Muthuraman S., Ajitha Priyadarsini S., Arumugom S. 2016 *J Electr Electron Syst* 5(4) 1-7
[12] Korani W. 2008 *GECCO’08 Atlanta Georgia USA*