



# African Buffalo Optimization Algorithm for PID parameters tuning of Automatic Voltage Regulators

**Julius Beneoluchi Odili, Mohd Nizam Mohmad Kahar,  
A.Noraziah**

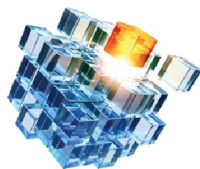
Fakulti Sistem Komputer & Kejuruteraan Perisian, Universiti Malaysia Pahang, Lebuhraya Tun Razak, 26300 Gambang, Pahang, MALAYSIA  
odili\_julest@yahoo.com, mnizam@ump.edu.my, norazia@ump.edu.my

**Highlights:** To obtain stability of electric power generating system, there is the need for an Automatic Voltage Regulator (AVR) component. The effectiveness of the AVR is traceable to the proper tuning of the Proportional, Integral and Derivative (PID) parameters within the AVR. Our innovation is the successful use of the African Buffalo Optimization Algorithm to tune PID parameters for optimum efficiency and effectiveness.

**Key words:** AVR, African Buffalo Optimization, PID, Parameters, Tuning, Stability

## Introduction

The primary benefit of an Automatic Voltage Regulator (AVR) is to ensure a constant voltage level (Widyan, 2015). Regulating a voltage may require a feed-forward system as in 'open loop' mechanisms or a closed loop system as in 'feedback' mechanisms. The AVR, which uses feedback model may employ an electronic mechanism or an electromechanical component in its operation and has the capacity to regulate one or more DC or AC voltages (Bharothu & Venkatesh, 2014). A basic AVR is made up of the amplifier, exciter, generator and sensor, all working in harmony to ensure the output voltage of the power system is at a specified range. It is to be noted that an increase in the generator reactive



power load usually leads to a drop in the terminal voltage. A PID controller is, therefore, handy in minimizing the resulting error and ensuring improved dynamic response. As a result of this, the stability or otherwise of the AVR system seriously affects the stability of the power output (Sambariya & Prasad, 2015).

### **ABO-PID Tuning Process**

The need for electric power voltage stability cannot be over-emphasized. Power fluctuation is a threat to electrical and electronic gadgets all over the world. The need for optimal tuning arose out of the observable needs for improvements in rise time, settling time, systems gain overshoot, steady state errors of existing systems such as Genetic Algorithm PID (GA-PID) (Neath, Swain, Madawala, & Thrimawithana, 2014), Particle-Swarm Optimization PID (PSO-PID) (Solihin, Tack, & Kean, 2011), Ant Colony Optimization PID (ACO-PID) (Ünal, Ak, Topuz, & Erdal, 2012), PID-Tuner (Aravind, Valluvan, & Ranganathan, 2013), Bacteria-Foraging Optimization PID (BFO-PID) (Manuaba, Abdillah, Priyadi, & Purnomo, 2015) etc.

The searching mechanism of the ABO-PID controller is itemized below.

Step 1: Initialize the buffalos on the search space in sets of three buffalos per set. That is to say that if there are a population of N buffalos, they will consist of N/3 components. Set  $s$  which represents the step function as 2

Step 2: Calculate the evaluation value of each individual in the population using the democratic equation 1 and 2 respectively



$$m_k' = m_k + lp1(bg - w_k) + lp2(bp.k - w_k) \quad (1)$$

$$w_k' = \frac{(w_1 + m_1)}{\lambda} \quad (2)$$

Step 3: Determine Gp Gi and Gd for each set of buffalos

Step 4: Plot the Gp Gi and Gd into the PID benchmark transfer function represented by Equation

$$Gp(s) + \frac{G_i}{s} + Gd(s) \quad (3)$$

Determine the buffalo set with the best performance and set as bg

Step 5: Set the values of x/y. If the output is 1 which represents the steady state, terminate the run, else return to Step 2

The Block diagram of an ABO-PID is presented in Figure 1:

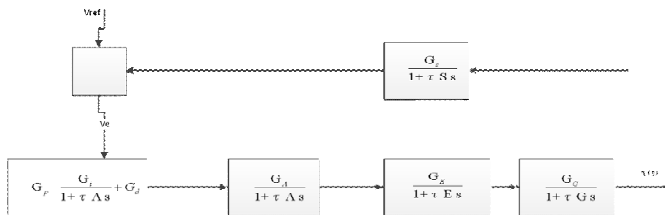
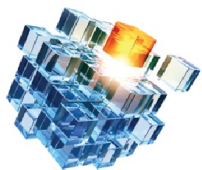


Figure 1: Block diagram of AVR system with a PID



### Implementation

The ABO-PID was implemented using MATLAB on a desktop computer: Intel Duo Core™ i7-3770 CPU, 3.40 GHz with 4GB RAM. The simulation results are presented in Table 1. The comparative results are from the existing models mentioned above

Gain overshoot (%)	Type of controller	PID parameters			Rise time	Settling time	Steady State Error
		Gp	Gi	Gd	(secs)	(secs)	
0	ABO-PID	3.007	1.073 4	0.430 4	1.77	2.85	0
5.54	PID-PSO	0.612 5	0.419 7	0.201 3	4.8	>10.00	0.06
2.44	LQR-PID	1.010 0	0.500 0	0.100 0	3.96	> 10.00	0.02
0	GA-PID	3.156 3	0.946 3	0.493 0	1.83	4.87	0.005
0.487	ACO-PID	2.991 7	1.105 3	0.308 5	1.72	2.66	0
0	PSO-PID	3.317 2	0.899 3	0.281 4	1.75	>10.00	0.008
0.288	BFO-PID	3.072 5	1.105 4	0.260 1	1.69	2.59	0

Table 1: Simulation results

### Discussion of Results

Combining the gain overshoot and the steady state error as measuring indices, the exceptional performance of the ABO-PID becomes clearer. It was only the ABO of all the algorithms under investigation in this study that was able to obtain zero score on both counts. That means that not only was ABO able to maintain no overshoot, it was also able to ensure that the system ran smoothly at all times without any error. That means that that the output voltage is the same as the reference voltage:




$$V_t(s) = V_{ref}(s) \quad (4)$$

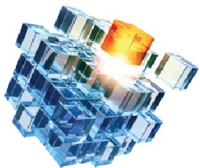
However, in terms of rise time and settling time, there is still room for improvement of the ABO-PID. As can be observed the BFO-PID, ACO-PID and PSO-PID rose faster than the ABO-PID. Similarly, BFO-PID and ACO-PID settled before the ABO-PID, thus emphasizing the No Free Lunch theorem (Xu, Caramanis, & Mannor, 2012).

### **Conclusion**

Effective tuning process like that of the ABO-PID is of immense benefit to the computing, electrical/electronic manufacturers and end users. It is our belief that this research effort will spur the computing research community to investigate further the systems control engineering fields. This project is of commercial benefit to virtually all electric and electronic companies as it ensures the stability of the voltage flow in and out of their systems. ABO-PID could be used in regulating AVRs in fuzzy washing machines, fuzzy ovens, pressing irons, microwaves etc.

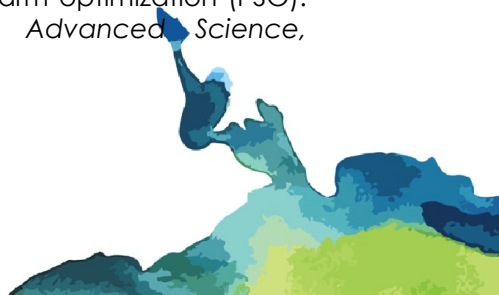
### **Acknowledgement**

The team is grateful to the Faculty of Computer Systems and Software Engineering, Universiti Malaysia Pahang, Kuantan 26300 for funding this project under Grant GRS 1403118. Also, our appreciation to the Ministry of Higher Education, Malaysia for addition funding under the Exploratory Research Grant RDU120608.



## References

- Aravind, P., Valluvan, M., & Ranganathan, S. (2013). Modelling and Simulation of Non Linear Tank. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, 2(2), 842-849.
- Bharothu, J. N., & Venkatesh, G. (2014). Reduction of damping oscillations of power systems with combinational operation of LFC & AVR in single area. *American Journal of Electrical and Electronic Engineering*, 2(2), 27-39.
- Manuaba, I. B., Abdillah, M., Priyadi, A., & Purnomo, M. H. (2015). COORDINATED TUNING OF PID-BASED PSS AND AVR USING BACTERIAL FORAGING-PSOTVAC-DE ALGORITHM. *Control and Intelligent Systems*, 43(3).
- Neath, M. J., Swain, A. K., Madawala, U. K., & Thrimawithana, D. J. (2014). An optimal PID controller for a bidirectional inductive power transfer system using multiobjective genetic algorithm. *Power Electronics, IEEE Transactions on*, 29(3), 1523-1531.
- Sambariya, D., & Prasad, R. (2015). Optimal tuning of fuzzy logic power system stabilizer using harmony search algorithm. *International Journal of Fuzzy Systems*, 17(3), 457-470.
- Solihin, M. I., Tack, L. F., & Kean, M. L. (2011). Tuning of PID controller using particle swarm optimization (PSO). *International Journal on Advanced Science*,





*Engineering and Information Technology*, 1(4), 458-461.

Ünal, M., Ak, A., Topuz, V., & Erdal, H. (2012). *Optimization of PID controllers using ant colony and genetic algorithms* (Vol. 449): Springer.

Widyan, M. S. (2015). Operational characteristics of hybrid-powered three-phase induction motor via synchronous and photovoltaic generators with MPPT utilised by synchronous generator automatic voltage regulator. *IET Renewable Power Generation*, 9(6), 638-646.

Xu, H., Caramanis, C., & Mannor, S. (2012). Sparse algorithms are not stable: A no-free-lunch theorem. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(1), 187-193.

