Automatic Tuning of Proportional–Integral–Derivative Controller using Genetic Algorithm

Joseph Stephen Bassi, Ph.D. and Emmanuel Gbenga Dada, Ph.D.

Department of Computer Engineering, University of Maiduguri, Maiduguri, Borno State, Nigeria.

Email: sjbassi74@unimaid.edu.ng*

ABSTRACT

The application of intelligent approaches for tuning the gains of Proportional-Integral-Derivative (PID) controller parameters has been growing recently. The flexibility ability of evolutionary procedures have elevated its acceptability for adjusting the gains PID controllers. This work presents an automatic strategy for adjusting the gains of a PID controller parameters of systems with scarce initial information and integrative and unstable dynamics, using evolutionary Genetic Algorithm (GA), an Evolutionary Computation (EC) technique strategy. The advantages of the proposed approach were highlighted through the comparison with classical Ziegler-Nichols closed loop approach. Experiments with different processes indicate that the gains obtained through genetic algorithms may provide better responses than those obtained by the classical Ziegler -Nichols approach in terms of time domain specification and performance indices.

(Keywords: PID Tuning, genetic algorithm, evolutional computation, Siegler-Nichols method, optimization)

INTRODUCTION

Model-based control techniques are usually implemented under the assumption of good understanding of process dynamics and their operational environment. These techniques, however, cannot provide satisfactory results when applied to poorly modelled processes, which can operate in ill-defined environments. This is often the case when dealing with complex dynamic systems for which the physical processes are either highly nonlinear or are not fully understood (Karray *et. al.,* 2002). PID controllers are the most popular industrial controllers today (Jaen-Cuellar *et.al,* 2013). The acceptance of PID controllers is due to their simplicity both from design and parameter tuning points of view.

Genetic algorithms (GA) are a modern meta-heuristic algorithm inspired in natural evolution and genetic recombination mechanisms. GA is optimization technique that is being researched by both academicians and researchers searching for optimal PID parameters. It is inspired by Darwin's theory of evolution which states that the survival of an organism is affected by rule "the strongest species that survives" (Hermawento, 2013; Yusuf *et.al.,* 2015). Genetic algorithm can provide solutions for highly complex search space and perform well approximately solution for all types of problems because they do not make any assumption about the underlying fitness landscape (Zvirgzdina and Tolujevs, 2013; Yusuf *et.al.,* 2015). This technique is basically a procedure of adaptive and parallel search for the solution of complex problems and can be used in conjunction with other intelligent techniques.

During the past decades, process control techniques in the industry have made great advances. Numerous control methods such as: adaptive control; neural control; and fuzzy control have been studied (Visioli, 2001; Seng *et al.,* 1999; Krohling and Rey, 2001; Mitsukura *et al.,* 1999). It is well known that most control problems can be adequately handled by the PID control strategy (Hugo,2002), in fact, many advanced control algorithms and strategies are based on a form of PID or the other moreover most industrial process control are handled by the standard PID controller (Nagaraj *et al.,* 2008) because of their simple structure and robustness (Hugo, 2010) and the principles involved can learnt very easily.

Despite this popularity, the tuning of a PID controller is a very subjective procedure which relies heavily on the knowledge and skill of the plant engineer or that of a process operator

(Rasmussen, 2002), moreover it is a tedious and a time-consuming procedure, which is compounded by the fact that most processes contain more than one control loop that require separate tuning (Hugo, 2010). This may account for the fact that more than 70% of industrial plants are poorly tuned and potentially account for - loss of revenue in terms of percentage of defective products and energy utilization (Hugo, 2010 and Rasmussen, 2002). Furthermore, plant parameters are subject to change as operating conditions change and as result of aging, which then requires the re-tuning of process controller(s).

The goal of this paper is to examine the application of GA algorithm for tuning the gains of PID controller parameters. The algorithm searches for controller gains *Kp* (proportional gain), *Ki* (integral gain) and *Kd* (derivative or differential gain) such that specifications for the closed-loop step response are fulfilled.

MATERIALS AND METHODS

The arrangement for closed-loop PID controlled system is shown in Figure 1. Where the values of k_p, k_i and k_d are the PID parameters to be tuned. The PID controller generates the control effort *u* using the current gain values.

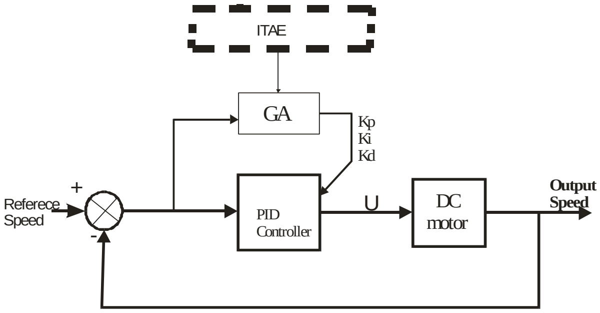

Figure 1: The Block Diagram of Proposed PID Controller with Genetic Algorithms.

Performance Index

The performance index is defined as a quantitative measure to depict the system performance of the designed PID controller. Using this technique an 'optimum system' can often be designed and a set of PID parameters in the system can be adjusted to meet the required specification. For a PID- controlled system, there are often four indices to depict the system performance: ISE, IAE, ITAE and ITSE. This Paper will use ITAE as the performance index which is defined as:

$$J_{ITAE} = \int_0^\infty t|e(t)|dt \qquad \ldots 1$$

Plant Equations

The equations describing the dynamic behavior of the DC motor are given by Equations 2-4:

$$\frac{\theta}{V} = \frac{K}{(Js+b)(Ls+R)+K^2} \qquad \ldots 2$$

$$\frac{d}{dt}\begin{bmatrix}\theta \\ i\end{bmatrix} = \begin{bmatrix} -\frac{b}{J} & \frac{K}{J} \\ -\frac{K}{L} & -\frac{R}{L}\end{bmatrix}\begin{bmatrix}\dot{\theta} \\ i\end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L}\end{bmatrix} v$$

$$\dot{\theta} = [10]\begin{bmatrix}\dot{\theta} \\ i\end{bmatrix} \qquad \ldots 3$$

Where J=moment of inertia of motor and load in Kg-m2/rad, b= Damping ratio of the mechanical system, L= Electric Inductance in Henri, R= Armature resistance in Ohm, K= constant in N-m/Ampere, V= Armature voltage in volts, $\theta =$ rotational speed in radians and i=Armature current in ampere.

System Parameters

L=0.5 H, J=0.01 Kgm^2, R=1Ω, b=0.1 Nms, K=0.01 Nm/A. The overall transfer function of the system is given by Equation 4.

$$\frac{\theta(s)}{V(s)} = \frac{1}{s(s+1)(s+5)} \qquad \ldots 4$$

PID Controller

The PID controller transfer function is given by Equation 5.

$$U(t) = K_P e(t) + K_I \int e(t)dt + K_D \frac{de(t)}{dt} \quad \text{...5}$$

Taking the Laplace transform of Equation 5.

$$G_C(s) = K_P \left(1 + \frac{1}{T_I s} + T_D s \right) \quad \text{...6}$$

GENETIC ALGORITHM

GA's are a stochastic global search method that mimics the process of natural evolution. The genetic algorithm starts with no knowledge of the correct solution and depends entirely on responses from its environment and evolution operators (i.e., reproduction, crossover and mutation) to arrive at the best solution. By starting at several independent points and searching in parallel, the algorithm avoids local minima and converging to sub optimal solutions. In this way, GAs have been shown to be capable of locating high performance areas in complex domains without experiencing the difficulties associated with high dimensionality, as may occur with gradient decent techniques or methods that rely on derivative information (Houck *et al.,* 1996).

A genetic algorithm is typically initialized with a random population consisting of between 20-100 individuals (Griffin, 2003). This population (mating pool) is usually represented by a real-valued number or a binary string called a chromosome. In this case the controller gains (K_d ,K_p, & K_i) forms the chromosomes (Griffin, 2003). How well an individual performs a task is measured and assessed by the objective function. The objective function assigns each individual a corresponding number called its fitness. The fitness of each chromosome is assessed, and a survival of the fittest strategy is applied. In this paper, the magnitude of the error will be used to assess the fitness of each chromosome. There are three main stages of a genetic algorithm; these are known as reproduction, crossover and mutation (Griffin, 2003).

Reproduction

During the reproduction phase the fitness value of each chromosome is assessed. This value is used in the selection process to provide bias towards fitter individuals. Just like in natural evolution, a fit chromosome has a higher probability of being selected for reproduction. Four common methods for selection are:

- Roulette Wheel selection
- Stochastic Universal sampling
- Normalized geometric selection
- Tournament selection

Crossover

Once the selection process is complete, the crossover algorithm is initiated. The crossover operations swap certain parts of the two selected strings in a bid to capture the good parts of old chromosomes and create better new ones. Genetic operators manipulate the characters of a chromosome directly, using the assumption that certain individual's gene codes, on average, produce fitter individuals. The crossover probability indicates how often crossover is performed. A probability of 0% means that the 'offspring' will be exact replicas of their 'parents' and a probability of 100% means that each generation will be composed of entirely new offspring. The simplest crossover technique is the Single Point Crossover.

Mutation

Using *selection* and *crossover* on their own will generate a large amount of different strings. However, there are two main problems with this:

1. Depending on the initial population chosen, there may not be enough diversity in the initial strings to ensure the GA searches the entire problem space.

2. The GA may converge on sub-optimum strings due to a bad choice of initial population.

These problems may be overcome by the introduction of a mutation operator into the GA. Mutation is the occasional random alteration of a value of a string position. It is considered a background operator in the genetic algorithm.

The probability of mutation is normally low because a high mutation rate would destroy fit strings and degenerate the genetic algorithm into a random search. Mutation probability values of around 0.1% or 0.01% are common, these values represent the probability that a certain string will be selected for mutation i.e. for a probability of 0.1%; one string in one thousand will be selected for mutation. Once a string is selected for mutation, a randomly chosen element of the string is changed or 'mutated'.

The steps involved in creating and implementing a genetic algorithm are as follows:

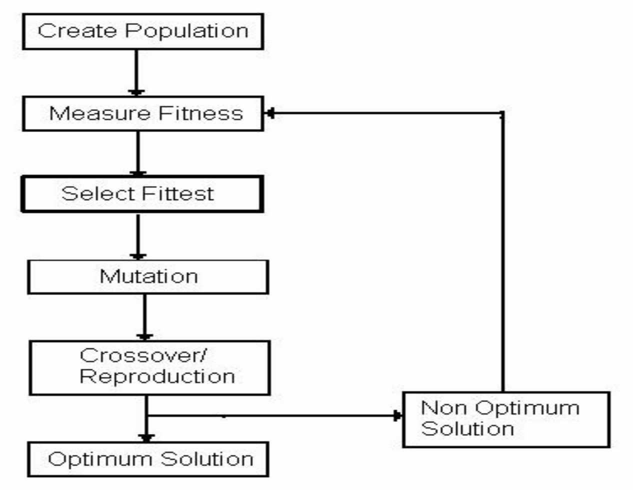

Figure 2: Graphical Illustration the Genetic Algorithm Outline.

TUNING OF PID CONTROLLER USING HEURISTIC APPROACH OF ZIEGLER NICHOLS

The control system performs poor in characteristics and even it becomes unstable, if improper values of the controller tuning constants are used. So it becomes necessary to tune the controller parameters to achieve good control performance with the proper choice of tuning constants. Controller tuning involves the selection of the best values of kc, Ti and TD (if a PID algorithm is being used). This is often a subjective procedure and is certainly process dependent. It is widely accepted method for tuning the PID controller.

The method is straight forward. First, set the controller to P mode only. Next, set the gain of the controller (kc) to a small value. Make a small set point (or load) change and observe the response of the controlled variable. If kc is low the response should be sluggish. Increase kc by a factor of two and make another small change in the set point or the load. Keep increasing kc (by a factor of two) until the response becomes oscillatory. Finally, adjust kc until a response is obtained that produces continuous oscillations. This is known as the ultimate gain (ku). Note the period of the oscillations (Pu). The steps required for the method are given below. We have to set the integral and derivative coefficients are zero. Gradually increase the proportional coefficient from 0 to until the system just begins to oscillate continuously.

The proportional coefficient at this point is called the ultimate gain Ku. And the period of oscillation at this point is called ultimate period Pu. The Ku=gain margin of the system and the Pu=(2*pi)/wcg. Where, the wcg. is the gain cross over frequency. Gain margin is the reverse of amplitude ratio. The control law settings are then obtained from Table 1 and also the PID gain values after simulation is given in Table 2.

Table 1: Control Law Settings.

Controller	Kp	Ti	Td
P	Ku/2		
PI	Ku/2.2	Pu/1.2	
PID	Ku/1.7	Pu/2	Pu/8

Table 2: Z-N PID Controller Gain Values.

Gain Coeff.	P	I	D
Values	39.42	3.077	0.7692

From the above algorithm the step response of the system with conventionally tuned PID controller is shown in Figure 3.

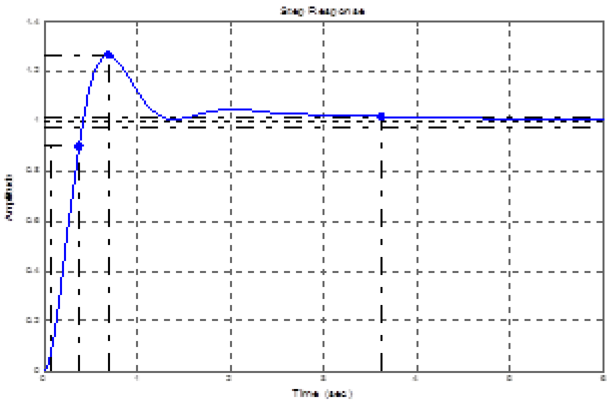

Figure 3: Ziegler Nichol Controlled System.

TUNING OF PID CONTROLLER USING GENETIC ALGORITHM APPROACH

The gains of the PID controller are designed optimally using GA. To optimize the performance of the system, the PID gains are adjusted to minimize a performance index. The PID gain values after simulation are given in Table 3.

Table 3: GA- PID Controller Gain Values.

Gain Coeff.	P	I	D
Values	19.9512	0.1143	19.9641

The parameter gain convergences, of the GA tuned parameters are shown in Figure 4.

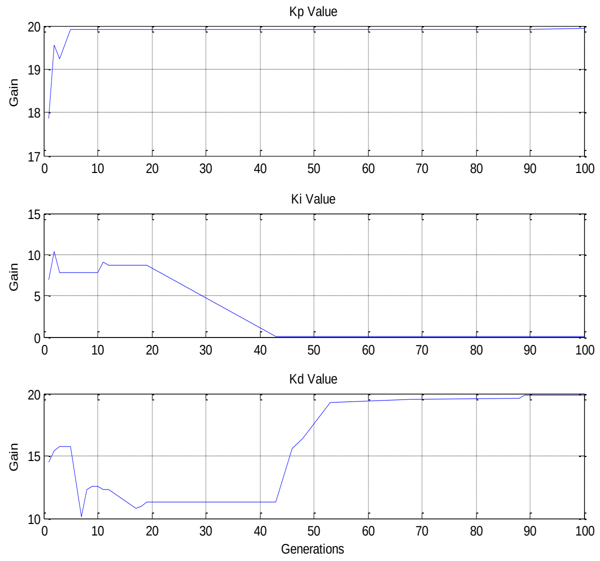

Figure 4: Genetic Algorithm Converging through Generations.

RESEARCH FINDING

The best population is plotted to give an insight into how the Genetic Algorithm converged to its final values as illustrated in Figure 4. The GA-based PID controller is initialized with a population size of 50, 70 and 100 and the responses to a step input signal 100 population size shown in Figure 5 gives the best result. The responses are analyzed for the smallest overshoot, peak amplitude and the fastest settling time and the summary shown in Table 4.

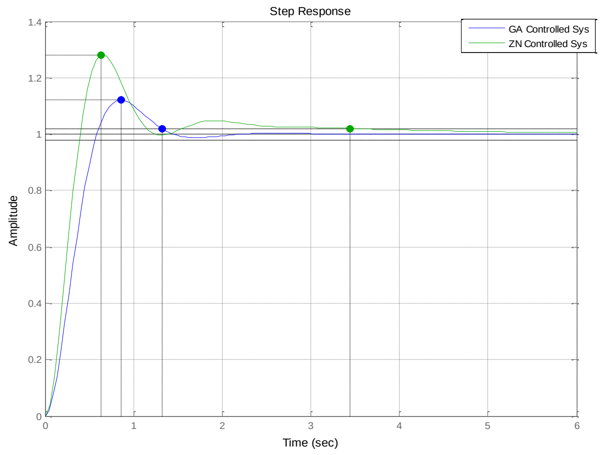

Figure 5: PID Response at Population Size Of 100.

Table 4: Summary of Results at Final Population Size of 100.

Measuring Factor	ZN Controller	GA Controller	Percentage Improvement
Peak Amplitude	1.28	1.12	12.5
Maximum Overshoot (%)	28	11	60.71
Settling Time (Sec)	3.44	1.32	61.63

In the conventionally Z-N tuned PID controller, the plant response produces high overshoot, but a better performance obtained with the implementation of GA-based PID controller tuning as seen in Table 4.

CONCLUSION

In this paper a new design method to determine optimal PID controller parameters using the GA method is presented. The speed of a DC Motor drive is controlled by PID-GA controller. Obtained through simulation of DC motor; the results show that the proposed controller can perform an efficient search for the optimal PID controller. By comparison with the classical method of Ziegler-Nichols PID controller tuning strategy, it shows that this method can improve the dynamic performance of the system in a better way. The PID-GA controller is the best which presented satisfactory performances and possesses good robustness.

REFERENCES

1. Clerc, M. 1999. “The Swarm and the Queen: Towards a Deterministic and Adaptive Particle Swarm Optimization”. *Proceedings of the Conference on Evolutionary Computation*. 1951-1957.

2. Eberhart, R.C. and Y.H. Shi. 2000. “Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization”. *Proc. The 2000 Congress on Evolutionary Computation*. 1:84-88.

3. Griffin, I. 2003. “On-line PID Controller Tuning using Genetic Algorithms”. Unpublished Masters degree thesis. Washington, DC.

4. Houck, C.R., J. Joines, and M.A. Kay. 1996. “Genetic Algorithm for Function Optimization: A Matlab Implementation”. *ACM Transactions on Mathematical Software*. http://www.eos.ncsu.edu/eos/service/ie/research/kay_res/GAToolBo/gaot

5. Hugo, J.A. 2002. “Process Controller Performance Monitoring and Assessment”. http/www.controlartsinc.com/support/articles/PerformanceAssessment.PDF. Accessed 31/6/10

6. Jaen-Cuellar, A.Y., R. de J. Romero-Troncoso, L. Morales-Velazquez, and R.A. Osornio-Rios. 2013. “PID-Controller Tuning Optimization with Genetic Algorithms in Servo Systems”. *International Journal of Advanced Robotic Systems*. 10(9):324.

7. Kennedy, J. and R.C. Eberhart. 1995. “Particle Swarm Optimization”. Proc. IEEE International Conference on Neural Networks: Perth, Australia. IEEE Service Center: Piscataway, NJ. IV: 1942-1948.

8. Krohling, R.A. and J.P. Rey. 2001. “Design of Optimal Disturbance Rejection PID Controllers using Genetic Algorithm”. *IEEE Trans Evol Comput*; 5:78–82.

9. Karray, F, W. Gueaieb. and S. Al-Sharhan, 2002. “The Hierarchical Expert Tuning of PID Controllers using Tools of Soft Computing”. *IEEE Transactions on Systems, Man, and Cybernetics: Part B: Cybernetics*. 32(1):77-90.

10. Mitsukura, Y., T. Yamamoto, and M. Kaneda. 1999. “A Design of Self-Tuning PID Controllers using a Genetic Algorithm”. In: *Proc Am Contr Conf*: San Diego, CA, June 1999. 1361–5.

11. Nagaraj, B., S. Subba, and B. Rampriya. 2005. “Tuning Algorithm for PID Using Soft-Computing Techniques”. *International Journal of Computer Science and Network Security*. 8(4):278 – 289.

12. Ou, C. and W. Lin. 2006. “Comparison between PSO and GA for Parameters Optimization of PID Controller”. *Proc. IEEE International Conference on Mechatronics and Automation*. Luoyang, China.

13. Qu S., R. Li and P. an Zhang. 2003. “Stable and Optimal Adaptive Fuzzy Control of Complex Systems using Fuzzy Dynamic Model”. *Fuzzy Sets and Systems*. 133:1 – 17.

14. Rasmussen, H. 2002. “Automatic Tuning of PID Regulator”. http://www.volgas.dk/nores/auto.pdf downloaded 30/6/10.

15. Schei, T.S. 1994. “Automatic Tuning of PID Controllers Based on Transfer Function Estimation”. *Automatica*. 1983 – 1989.

16. Seng, T.L., M.B. Khalid, and R. Yusof. 1999. “Tuning of a Neuro-Fuzzy Controller by Genetic Algorithm”. *IEEE Trans Syst Man Cybern*. B;29:226–36.

17. Shi, Y.H. and R.C. Eberhart. 2001. “Fuzzy Adaptive Particle Swarm Optimization”. *Proc. Congress on Evolutionary Computation*. Seoul,Korea.

18. Visioli, A. 2001. “Tuning of PID Controllers with Fuzzy Logic”. *Proc Inst Elect Eng Contr Theory Appl*. 148(1):1–8.

19. Yusuf, T.A., M.F. Atan, N.A. Rahman, S.F. Salleh, and N.A. Wahab. 2015. “Optimization of PID Tuning using Genetic Algorithm”. *Journal of Applied Science & Process Engineering*. 2(2).

20. Zhong, J. 2006. “PID Controller Tuning: A Short Tutorial” http://saba.kntu.ac.ir/eecd/pd/download/PIDtutorial.pdf downloaded 1/7/2010

21. Shi, Y.H. and R.C. Eberhart. 1998. “A Modified Particle Swarm Optimizer”. *IEEE International Conference on Evolutionary Computation*. Anchorage, Alaska.

ABOUT THE AUTHORS

Dr. Joseph Stephen Bassi, received his Ph.D. degree in Electrical Engineering from the Universiti Teknologi Malaysia, in 2017, M.Eng. degree in Electrical & Electronics Engineering (Electronics) from University of Maiduguri, Nigeria in 2012, and B.Tech. degree in Computer Science & Mathematics from Federal University of Technology Minna, Nigeria in 2000. He is currently a Lecturer with the Department of Computer Engineering, Faculty of Engineering, University of Maiduguri, Nigeria. His research interests are in network algorithmic, data mining, artificial intelligence and optimization techniques and computer communication networks.

Dr. Emmanuel Gbenga Dada, holds a Ph.D. in Computer Science from the Department of Artificial Intelligence, Faculty of Computer Science and Information Technology at University of Malaya, Malaysia. He is a senior lecturer in the Department of Computer Engineering, University of Maiduguri, Nigeria. His current research interests include soft computing algorithms (particle swarm optimization, fuzzy logic, neural computing, evolutionary computation, machine learning, and probabilistic reasoning), swarm robotics, optimization techniques, and information security.

SUGGESTED CITATION

Bassi, J.S. and E.G. Dada. 2018. **“**Automatic Tuning of Proportional–Integral–Derivative Controller using Genetic Algorithm”. *Pacific Journal of Science and Technology*. 19(2):51-57.

Pacific Journal of Science and Technology