VOL. 10, NO. 1, JANUARY 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

SOFT COMPUTING BASED CONTROLLERS FOR ELECTRIC DRIVES - A COMPARATIVE APPROACH

M. Gowrisankar[1] and A. Nirmal Kumar[2]
[1]Department of CSE, Karpagam College of Engineering, Coimbatore, India
[2]Karpagam College of Engineering, Coimbatore, India
E-Mail: gowrisankarkce@gmail.com

ABSTRACT

PID controllers are widely used in industrial plants because it is simple and robust. Industrial processes are subjected to variation in parameters and parameter perturbations, which when significant makes the system unstable. The aim of this paper is to design a controller for applications of various electric drives in industry by selection of PID parameters using soft computing techniques. Performance of Z-N methods have been compared and analyzed with the intelligent tuning techniques like Genetic algorithm, Evolutionary Programming, Particle Swarm Optimization and Bacterial Foraging Optimization. Soft computing methods have proved their excellence in giving better results by improving the steady state characteristics and performance indices.

Keywords: Genetic Algorithm, Evolutionary Programming, Particle Swarm Optimization and Soft computing.

1. INTRODUCTION

Conventional proportional integral derivative controller is widely used in many industrial applications due to its simplicity in structure and ease to design [1]. However it is difficult to achieve the desired control performance. Tuning is important for the best performance of PID controllers. PID controllers can be tuned in a variety of ways including hand tuning, Ziegler Nichols tuning, Cohen-coon tuning and Z-N step response, but these methods have their own limitations [3]. Soft computing techniques like GA, PSO and EP methods have proved their excellence in giving better results by improving the steady state characteristics and performance indices.

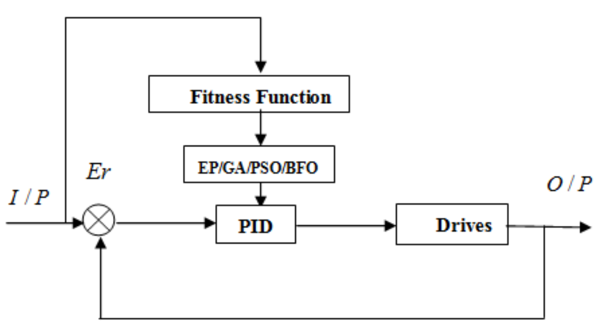

Figure-1. Block diagram of Intelligent PID controller.

1.1. Proportional integral derivative controller

The PID controller calculation involves three separate control parameters, i.e. proportional, integral and derivative values .The proportional value determines the reaction of the current error, the integral value determines the reaction based on the sum of recent errors and derivative value determines the reaction based on the rate at which the error has been changing and the weighted sum of these three actions is used to adjust the process via the final control element.

The block diagram of a control system with unity feedback employing Soft computing PID control action is shown in Figure-1 and the mathematical representation of PID control is given in (1).

$$Y(t) = [Kpe(t) + Kd\frac{d(e)}{d(t)} + Ki\int_0^t e(t)d(t)] \quad (1)$$

2. REASON FOR SELECTING SOFT COMPUTING TECHNIQUES

Optimization techniques like Genetic Algorithm (GA), Evolutionary Programming (EP), Particle Swarm Optimization (PSO) and Bacterial Foraging Optimization (BFO) belonging to the family of evolutionary computational algorithms have been widely used in many control engineering applications. These are powerful soft computing techniques which create a set of potential solutions called as populations. EP, GA, PSO and BFO found the optimal solution through cooperation and competition among potential solutions. These algorithms are highly relevant for industrial applications, because they are capable of handling problems with non linear constraints, multiple objectives and dynamic properties of the components that frequently appear in real-world problem.

The advantages of using heuristic techniques for PID are listed below [16].

a) Heuristic Techniques can be applied for higher order systems without model reduction [16].
b) These methods can also optimize the design criteria such as gain margin, Phase margin, closed loop bandwidth when the system is subjected to step and load change [16].

2.1. GA based tuning of the controller

The optimal value of the PID controller parameters K_p, K_i, K_d are to be found using GA. All possible sets of controller parameter values are particles whose values are adjusted to minimize the objective function, which in this case is the error criterion, and it is discussed in detail. For the PID controller design, it is ensured that the controller settings will provide the estimated results in a stable closed-loop system [1]. This is the most challenging part of creating a genetic algorithm in writing the objective function. In this project, the objective function is required to evaluate the best PID controller for the system. An objective function could be created to find a PID controller that gives the smallest overshoot, fastest rise time or quickest settling time. However in order to combine all of these objectives it was decided to design an objective function that will minimize the performance indices of the controlled system instead [2]. Each chromosome in the population is passed into the objective function one at a time. The chromosome is then evaluated and assigned a number to represent its fitness, the bigger its number the better its fitness [3]. The genetic algorithm uses the chromosomes fitness value to create a new population consisting of the fittest members. Each chromosome consists of three separate strings constituting a P, I and D term, as defined by the 3-row bounds declaration when creating the population [3]. When the chromosome enters the evaluation function, it is split up into its three terms. The newly formed PID controller is placed in a unity feedback loop with the system transfer function. This will result in a reduction of the compilation time of the program. The system transfer function is defined in another file and imported as a global variable. The controlled system is then given a step input and the error is assessed using an error performance criterion such as Integral Square Error (ISE).

$$\text{ISE}=\int_0^\infty e^2(t)dt$$

The chromosome is assigned an overall fitness value according to the magnitude of the error, smaller the error larger the fitness value. The flowchart of the GA control system is shown in Figure-2.

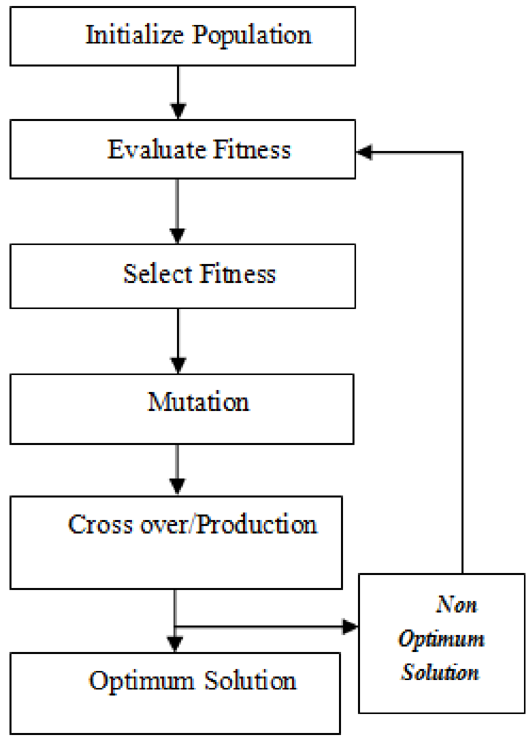

Figure-2. Flowchart of GA.

2.2. EP based tuning of controller

There are two important ways in which EP differs from GA. First, there is no constraint on the representation. The typical GA approach involves encoding the problem solutions as a string of representative tokens. The steps involved in creating and implementing evolutionary programming are as follows:

a) Generate an initial, random population of individuals for a fixed size (according to conventional methods K_p, K_i, K_d ranges declared).
b) Evaluate their fitness (to minimize integral square error ISE)
c) Select the fittest members of the population.
d) Execute mutation operation with low probability.
e) Select the best chromosome using competition and selection.
f) If the termination criteria reached (fitness function) then the process ends. If the termination criteria not reached search for another best chromosome. The flowchart of the EP control system is shown in Figure-3.

VOL. 10, NO. 1, JANUARY 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

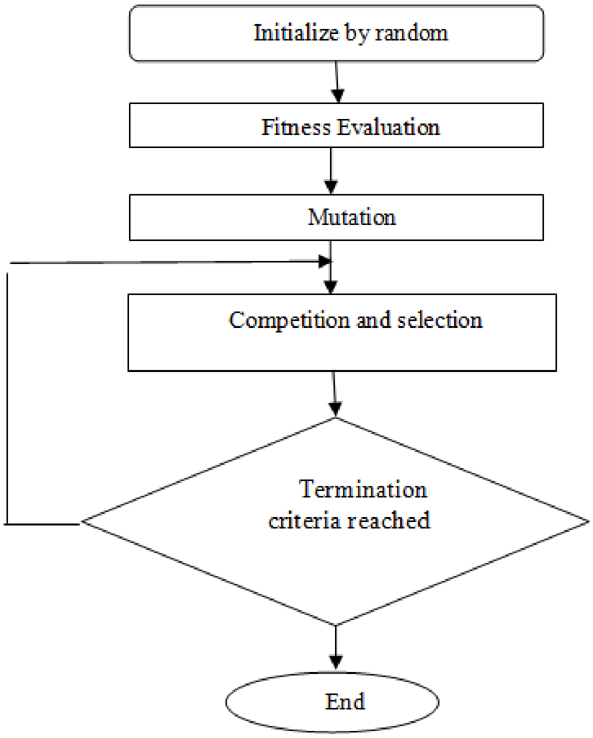

Figure-3. Flow Chart of EP.

2.3. PSO based tuning of controller

The algorithm proposed by Eberhart and Kennedy (1995) uses 1-D approach for searching within the solution space. For this study the PSO algorithm will be applied to 2-D or 3-D solution space in search of optimal tuning parameters for PI, PD and PID control. The flowchart of the PSO - PID control system [21] is shown in Figure-4. Consider position $X_{i,m}$ of the i-th particle as it traverses a n-dimensional search space: The previous best position for this i-th particle is recorded and represented as pbest $_{I,n}$. The best performing particle among the swarm population is denoted as gbest $_{I,n}$ and the velocity of each particle within the n-dimension is represented as $V_{I,n}$. The new velocity and position for each particle can be calculated from its current velocity and distance respectively [18]. The velocity of each particle, adjusted accordingly to its own flying experience and the other particles flying experience [7]. In PSO method each particle contains three members P, I and D. It means that the search space has three dimensions and particles must 'fly' in a three dimensional space.

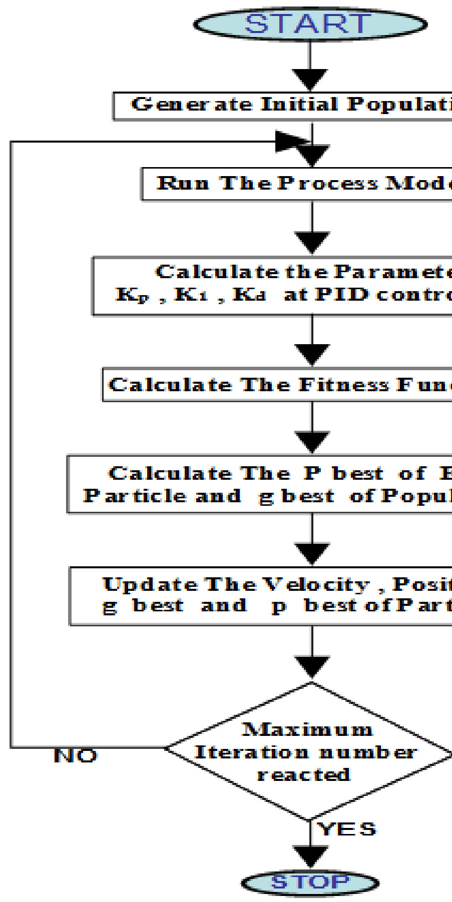

Figure-4. Flowchart of PSO.

2.4. Bacterial Foraging Optimization

The survival of species in any natural evolutionary process depends upon their fitness criteria, which relies upon their food searching and motile behavior. The law of evolution supports those species who have better food searching ability and either eliminate or reshape those with poor search ability. The genes of those species who are stronger gets propagated in the evolution chain since they posses ability to reproduce even better species in future generations. So a clear understanding and modeling of foraging behavior in any of the evolutionary species, leads to its application in any nonlinear system optimization algorithm. The foraging strategy of Escherichia coli bacteria present in human intestine can be explained by four processes, namely chemotaxis, swarming, reproduction and elimination dispersal [7].

2.4.1. Chemotaxis

The characteristics of movement of bacteria in search of food can be defined in two ways, i.e. swimming and tumbling together known as chemotaxis. A bacterium is said to be 'swimming' if it moves in a predefined direction, and 'tumbling' if moving in a random direction. Mathematically, tumble of any bacterium can be represented by a unit length of random direction φ (j) multiplied by step length of that bacterium C(i). In case of swimming, this random length is predefined.

2.4.2. Swarming

For the bacteria to reach at the richest food location, it is desired that the optimum bacterium till a point of time in the search period should try to attract other bacteria so that together they conquer the desired location more rapidly. To achieve this, a penalty function based upon the relative distances of each bacterium from the fittest bacterium till that search duration, is added to the original cost function. Finally, when all the bacteria have merged into the solution point, this penalty function becomes zero. The effect of swarming is to make the bacteria congregate into groups and move as concentric patterns with high bacterial density.

2.4.3. Reproduction

The original set of bacteria, after getting evolved through several chemotaxis stages reaches the reproduction stage. Here, best set of bacteria gets divided into two groups. The healthier half replaces with the other half of bacteria, which gets eliminated, owing to their poorer foraging abilities. This makes the population of bacteria constant in the evolution process [11].

2.4.4. Elimination and dispersal

In the evolution process, a sudden unforeseen event can occur, which may drastically alter the smooth process of evolution and cause the elimination of the set of bacteria and/or disperse them to a new environment. Most ironically, instead of disturbing the usual chemo tactic growth of the set of bacteria, this unknown event may place a newer set of bacteria nearer to the food location. From a broad perspective, elimination and dispersal are parts of the population level long distance motile behavior. In its application to optimization, it helps in reducing the behavior of stagnation often seen in such parallel search algorithms. The flow chart of BFO control system is shown in Figure-5.

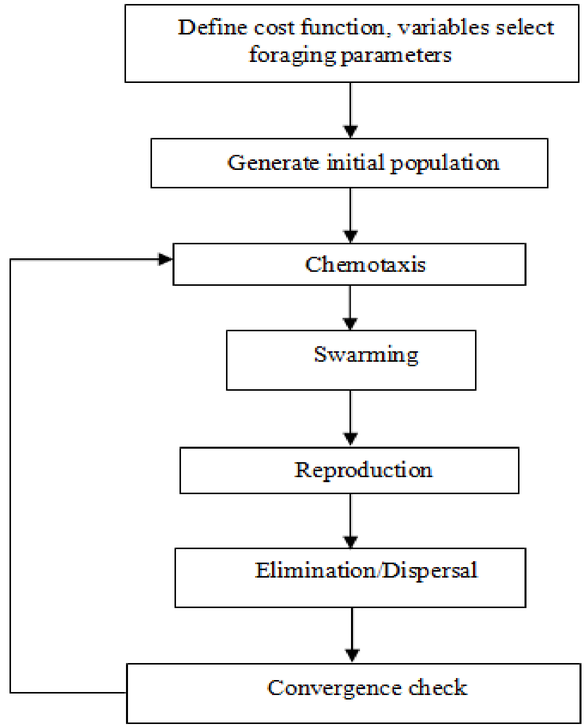

Figure-5. Flow chart of foraging process.

3. RESULTS AND DISCUSSIONS

This paper discusses about the implementation of soft computing based controller tuning for the following process models which are taken from various literatures.

- Speed control of DC motor [23]
- High performance drilling process [11]
- Servo control system [24]

3.1. Speed control of DC motor

In this experiment the Z-N closed loop tuning will be compared to the Soft Computing Methodology for a Speed control of DC motor process model.

The transfer function of DC motor has been taken to analyze the performance of various evolutionary algorithms. The transfer function of DC motor model used in the experiment is given in (2).

From the characteristic equations of the motor, the transfer function is obtained.

$$G_p(s) = \frac{0.1433}{5.2\exp^{-0.007} s^2 + 0.000217\, s + 2.265} \quad (2)$$

Based on Z-N closed loop tuning the values of designed PID controller are K_p=9.3883, K_i=0.2574 and K_d= 5.6274. The Table-1 summarise the initializing parameters value of soft computing methods.

VOL. 10, NO. 1, JANUARY 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

Table-1. PSO, GA, EP and BFO parameters.

PSO Parameters	GA Parameters	EP Parameters	BFO Parameters
Population size:100	Population size:100	Population size:100	Number of bacterium =5
Wmax=0.6/ Wmin=0.1	Mutation rate:0.1	Normal distribution	Number of iteration in a Chemotactic loop *(N_c)* =10
C1 = C2 = 1.5	Arithmetic Crossover	Mutation rate:0.01	Number of reproduction *(N_{re})* =15 Number of Parameters *(P)* =3
Iteration:100	Iteration:100	Iteration:100	$W_{attract}$ =0.04 $D_{attract}$ =0.01
Fitness function:ISE	Fitness function:ISE	Fitness function:ISE	$H_{repellent}$ =0.01 $W_{repellent}$=10 Fitness functions :ISE

The PID tuning parameters and the dynamic closed-loop performance specifications are shown in Table-2. The closed loop responses for model-1 are shown in Figure-6. The Z-N provides more rise time and settling time. Soft computing tuned controller provides an improved response when compared to the Z-N methods. From Table-2, it is observed that the PSO method yields a system having rapid settling time, smaller rise time and improved performance index over the other methods.

Table-2. PID Parameters and closed - loop response specifications for model -1.

Tuning method	PID parameters			Dynamic performance specifications			Performance index
	K_p	K_i	K_d	$T_{r(sec)}$	$T_{s(sec)}$	M_p (%)	*ISE*
ZN	9.3883	0.2574	5.6274	1.27	2.235	45.3	2.2926
EP	10	0.1	1	0. 468	0.877	0. 0	1.334
GA	3	0.0333	0.003	0. 444	0.781	0. 0	1.4406
PSO	1.5	0.003	0.03	0. 0761	0.13	0. 0	1.0024
BFO	10.521	0.0043	3.4195	0. 105	0.429	0. 0	1.454

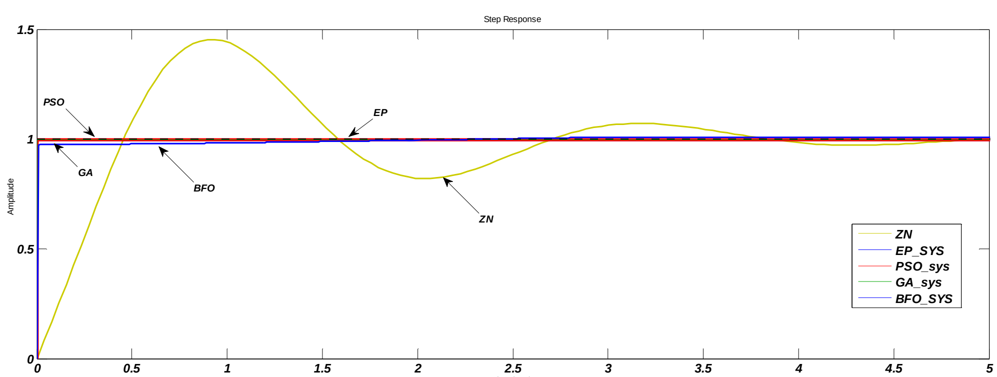

Figure-6. System response specifications for model-1.

$$\left[G_p(s) = \frac{0.1433}{5.2\exp^{-0.007} s^2 + 0.000217\, s + 2.265} \right]$$

3.2. High performance drilling process

The objective of this experiment is to compare the performance of the Soft Computing tuning methodology to that of the Z-N method for a high performance drilling process model. The third order model (high performance drilling process) [11] used in the experiment is given in (3).

The modeling of a high performance drilling process [11] includes the modeling of the feed drive system, the spindle system and the cutting process. In this study, the overall plant model is obtained by experimental identification using different step shaped disturbances in

VOL. 10, NO. 1, JANUARY 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

the command feed. The drilling force, F is proportional to the machining feed and the corresponding gain varies according to the work piece and drill diameter. The overall system of the feed drive, cutting process and dynamometric platform was modeled as a third-order system and the experimental identification procedure yielded the transfer function as:

$$G(s) = \frac{1958}{s^3 + 17.89s^2 + 103.3s + 190.8} \tag{3}$$

Where s is the Laplace operator, f is the command feed, and F is the cutting force. The model does have certain limits in representing the complexity and uncertainty of the drilling process. However, it provides a rough description of the process behavior that is essential for designing a network - based PID control system.

The PID tuning parameter and closed loop dynamic performance specifications of the system are shown in Table-3.

Table-3. PID Parameters and closed -loop response specifications for model- 2.

Tuning Method	PID Parameters			Dynamic performance specifications			Performance Index
	K_p	K_i	K_d	$T_{r(sec)}$	$T_{s(sec)}$	M_p (%)	ISE
ZN	0.4979	1.6108	0.0385	0.15	1.62	42	1.8876
EP	0.8123	1.0000	0.2502	0.068	I.18	22.7	1.5208
GA	0.6	1.4	0.1134	0.105	0.537	15	1.1649
PSO	0.5452	1.2502	0.1000	0.115	0.5	12.9	1.2038
BFO	0.324	1.8	0.095	0.136	1.84	11	1.0014

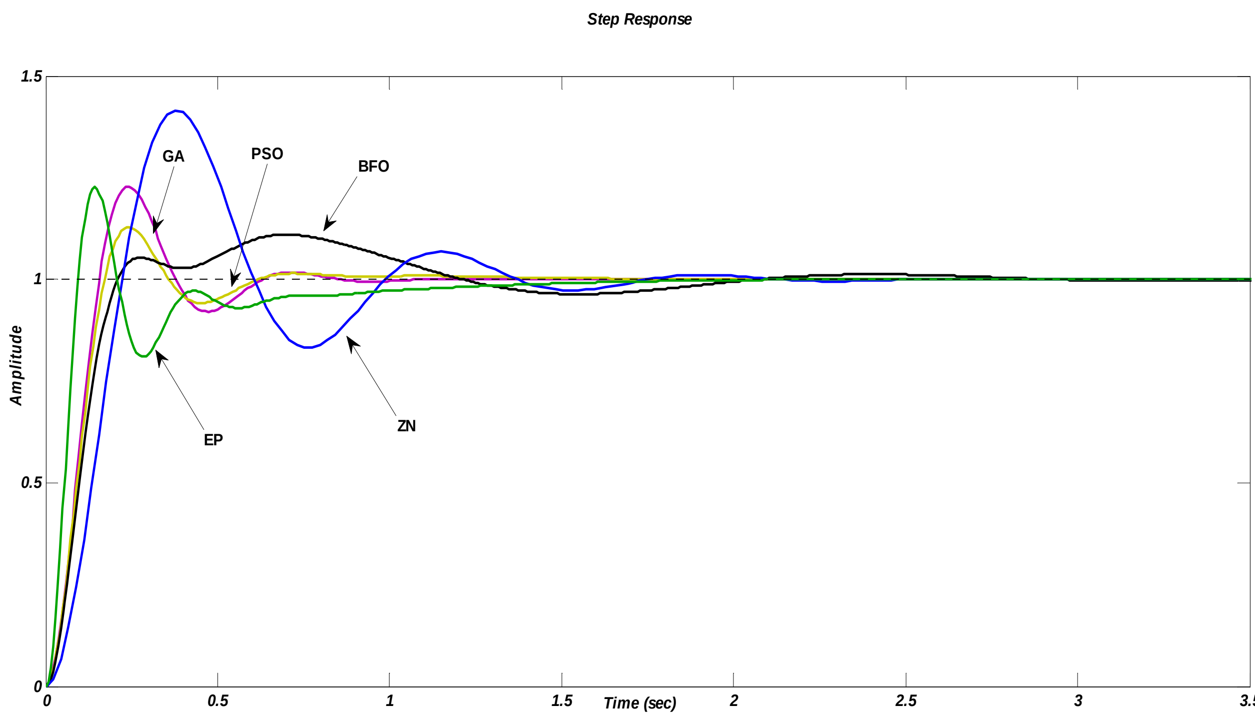

Figure-7. System responses for Model-2.

$$\left[G_p(s) = \frac{1958}{s^3 + 17.89s^2 + 103.3s + 190.8} \right]$$

The PID tuning parameters and closed - loop dynamic performance specification are shown in Table-3 and Figure-7 respectively. The Z-N tuning method delivers a response that has higher overshoot, longer settling time and larger rise time than that of the soft

ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

computing techniques. The EP and GA methods produce a slightly oscillatory system with smaller overshoots and undershoot. On the other hand the PSO and BFO tuned PID provides a closed loop system with performance improvements in overshoot, settling time and rise time.

3.3. POSITIONING SERVO - SYSTEM CONTROL LOOP

The positioning system is actuated by means of an armature controlled DC motor with gear speed reduction. The transfer function of DC motor model used in the experiment is

$$\left[G_p(s) = \frac{0.62\exp(-0.1s)}{(0.5s+1)} \right] \tag{4}$$

From Figure-8 and Table-4, it is observed that the Z-N method yields a system with higher overshoot, longer settling and rise time in comparison to other methods. EP method yields a system with very smaller overshoot but longer settling time. The PSO and BFO method delivers superior control performance with improved dynamic performance specifications over the other tuning methods.

Table-4. PID Parameter and closed loop response specification for model-3.

Tuning Method	PID Parameters			Dynamic performance specifications			Performance Index
	K_p	K_i	K_d	$T_{r(sec)}$	$T_{s(sec)}$	M_p (%)	ISE
Z-N	3	6	0	3.66	3.68	89%	11.8514
EP	5	40	0	0.456	6.67	0.2%	0.451
GA	2.4658	4.97	0	0.971	1.67	0. 0	0.613
PSO	0.5190	0.9531	0	0.325	0.89	0. 0	0.00281
BFO	1.2989	2.8817	0	0.156	0.638	0	0.00111

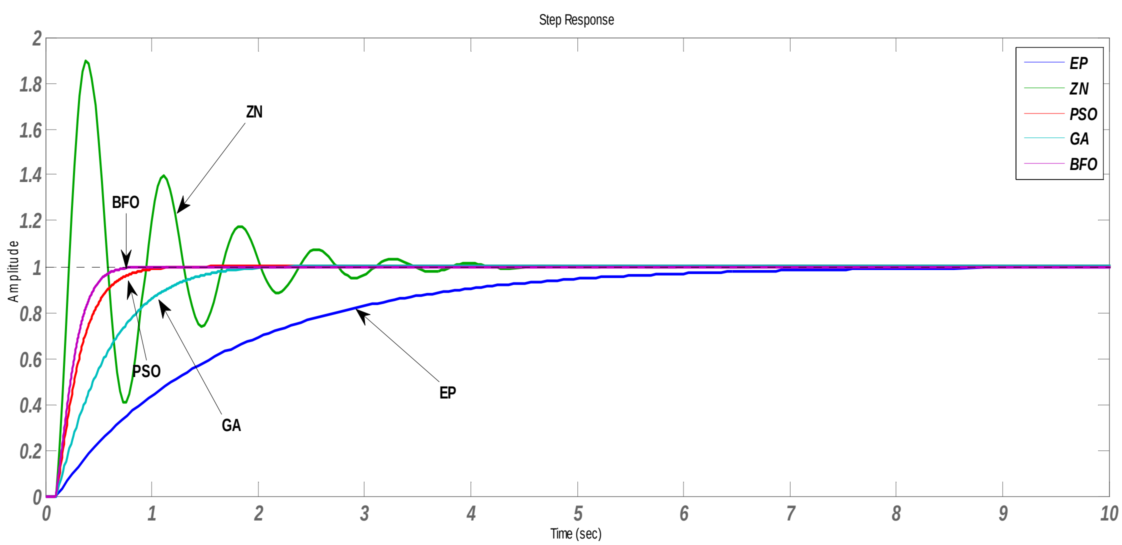

Figure-8. System responses for Model - 3 $\left[G_p(s) = \frac{0.62\exp(-0.1s)}{(0.5s+1)} \right]$.

4. CONCLUSIONS

The optimal parameters of the PID controller for various processes were computed by using GA, EP, BFO and PSO. From the simulation, it is found that the soft computing tuned controller has minimum settling time and minimum ISE compared to the Z-N method tuned controller. It is also evident that the PSO tuned system gives a better closed loop performance than the BFO, GA, EP and Z-N by achieving smaller ISE criterion.

The dynamic performance of the soft computing tuned system out performs that of the same system tuned with ZN for the following reasons. The Z-N method provides only initial tuning parameters. Fine tuning for an improved response depends on the experience and

VOL. 10, NO. 1, JANUARY 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

intuition of the control practitioner. The PSO and BFO methods do not suffer from premature convergence. This is not true for the GA and EP. Improvements in tuning performance can be achieved if the GA and EP are run for a greater number of iterations. This comes at the cost of increased computational burden and process delays. The GA and EP depend on genetic operators. This implies that even weak solutions could contribute to the composition of future candidate solutions. GA and EP operate according to a sharing mechanism during their evolutionary process whereby the previous solutions are potentially lost while the PSO relies on a memory based progression. This ability to 'remember' its previous best solution means that the PSO can converge much faster than the GA and EP on an optimal solution.

REFERENCES

[1] Andrey Popov, Adel Parag and Herbert Werner. 2005. Tuning of a PID controller using a Multi-objective Optimization Technique Applied to A Neutralization Plant. Proceeding of the 44th IEEE Conference on Decision and Control, and the European Conference 2005, Seville, Spain, December 12-15, pp. 7139-7143.

[2] Astrom K.J. and T. Hagglund. 1995. PID Controllers: Theory, Design, and Tuning. Triangle Park, NC: ISA.

[3] Boumediene Allaoua, Brahim Gasbaoui, Brahim Mebarki. 2009. Setting Up PID DC Motor speed Control Alteration Parameters Using Particle Swarm Optimization Strategy. Leonardo Electronic Journal of Practices and Technologies ISSN1583-1078, (14): 19-32.

[4] Cohen G.H. and Coon G.A. 1953. Theoretical consideration of retarded control. Trans. ASME. 75: 827-834.

[5] Dong Hwa Kim, Ajith Abraham and Jae Hoon Cho. 2007. A hybrid genetic algorithm and bacterial foraging approach for global optimization. Information Sciences. 177, pp. 3918-3937.

[6] Eberhart R.C. and Shi Y. 1998. Comparison between genetic algorithms and particle swarm optimization. IEEE Int. Conf. Evol. Comput. Anchorage. pp. 611-616.

[7] Ellis G, Control systems design guide. Academic Press, London. 1991 Gaing Z.L. 2004. A Particle Swarm Optimization approach for Optimum Design of PID controller in AVR System. IEEE Transactions on Energy Conversion. 19(2): 284-291.

[8] Giriraj Kumar S.M., R. Sivasankar, T.K. Radhakrishnan, V. Dharmalingam and N. Anantharaman. 2008. Particle Swarm Optimization Technique Based Design of Pi Controller for a Real-Time Non-Linear Process. Instrumentation Science and Technology. 36: 525-542.

[9] Giriraj Kumar S.S, R.Jain, N.Anantharaman, V. Dharmalingam, K.M.M. Sheriffa Begum. 2008. Genetic Algorithm Based PID Controller Tuning for a Model Bioreactor. Indian Chemical Engineer Journal. 50(3): 214-226.

[10] Girirajkumar S.M., Deepak Jayaraj, and Anoop.R. Kishan. 2010. PSO based Tuning of a PID Controller for a High Performance Drilling Machine. International Journal of Computer Applications. 1(19): 12-18.

[11] Govender P. 2003. Nonlinear Predictors in PID control: Controlling Processes having Dead-Time. PhD Thesis - Department of Electrical Engineering (Light Current) at Durban Institute of Technology.

[12] Hang C.C., Astrom K. and Ho W.K. 1991. Refinements of the Ziegler-Nichols tuning formula. IEEE Proceedings-D. 138(2): 111-118.

[13] Haung H.P. and Chen C.C. 1996. Control-system synthesis for open-loop unstable process with time delay. IEE Proc.-Control Theory Appl. 144(4): 334-345.

[14] Jin-Sung Kim, Jin-Hwan Kim, Ji-Mo Park, Sung-Man Park, Won-Yong Choe. 2008. Auto Tuning PID controller based on Improved Genetic Algorithm for Reverse Osmosis Plant. World Academy of Science, Engineering and Technology. 47, pp. 384-389.

[15] Jukka Lieslehto. 1995. PID Controller Tuning Using Evolutionary Programming. Proceedings of the American Control Conference Arlington, VA June 25-27, 2001. pp. 2828-2833. ennedy J. and Eberhart R. Particle swarm optimization. Proc. IEEE Int. Conf. Neural Networks. Vol. 4, Perth, Australia. pp. 1942-1948.

[16] Lelic M. 2000. A Reference Guide to PID Controllers in the Nineties, In Proceedings of IFAC Workshop: Past, Present and Future of PID Control.

[17] Liptak. 1995. Process Control. Chilton Book Company.

[18] Liu G.P. and Daley S. 2001. Optimal-tuning PID control for industrial systems. Control Engineering Practice. 9: 1185-1194.

[19] Mehdi Nasri, Hossein Nezamabad-pour and Malihe Maghfoori. 2007. A PSO-Based Optimum Design of PID Controller for a Linear Brushless DC Motor. World Academy of Science, Engineering and Technology. 26, p. 211.

[20] Mohammed EI-Said EI-Telbany. 2007. Employing Particle Swarm Optimizer and Genetic Algorithms for Optimal Tuning of PID Controllers: A Comparative Study. ICGST-ACSE Journal. 7(2): 49 -54.

[21] Neenu Thomas, Dr. P. Poongodi. 2009. Position Control of Motor Using Genetic Algorithm Based PID Controller. Proceeding of the World Congress on Engineering 2009 Vol. II WCE 2009, July 1- 3, London, U.K.

[22] Nelendran Pillay. Light Current, Durban University of Technology Department of Electronic Engineering. A Particle Swarm Optimization approach Tuning of SISO PID control loops. pp. 1-152.

[23] Ogata K. 1970. Modern Control Engineering. Englewood Cliffs, N.J., Prentice- Hall.

[24] Panda C.P., Yu C.C. and Huang H.P. 2004. PID tuning rules for SOPDT systems: Review and some new results. ISA Transactions. 43: 283-295.

[25] Pomerleau A., Poulin E., Desbines A. and Hodouin D. 1996. Development and Evaluation of an Auto-tuning and Adaptive PID Controller. Automatica. 32: 71-82.

[26] Salim Ali E. and S.M. Abad-Elazin. 2010. Optimal PID Tuning for Load Frequency Control Using Bacteria Foraging Optimization Algorithm. Conference Proceedings (MEPCON'10), Cairo University, Egypt. pp. 19-21.

[27] Seema Nara, Pooja Khatri, Jatin Garg. 2010. Proportional - Integral - Deriative Controller Tuning of Temperature Control System Using Genetic Algorithm. International Journal of Electronics, Information and System. 12: 37-42.

[28] Tushar Jain and M.J. Nigam. 2008. Optimization of PD-PI Controller Using Swarm Intelligence. Journal of Theoretical and Applied Information Technology. pp. 1013-1018.

[29] Valarmathi K., Devaraj D. and Radha Krishnan T.K. Enhanced Genetic Algorithm based proportional Integral Controller Tuning for pH process. Instrumentation science and Technology, Taylor and Francis Publishers. 35: 619-635.

[30] Venkatashankar V. and Chidambaram M. 1994. Design of P and PI controllers for unstable process with time delay. Int. J. Control. 60: 1367-144.

[31] Zhuy. 1999. Parametric Wiener model identification for control. Process. IFAC world congress. pp. 37-42.

[32] Ziegler, J.G. and Nichols, N.B. 1942. Optimum settings for Automatic Controllers. Trans. ASME. 64: 759-768.