Hindawi Publishing Corporation
Advances in Artificial Neural Systems
Volume 2014, Article ID 126317, 8 pages
http://dx.doi.org/10.1155/2014/126317

Research Article

Optimal Design of PID Controller for the Speed Control of DC Motor by Using Metaheuristic Techniques

Mirza Muhammad Sabir[1] **and Junaid Ali Khan**[2]

[1] *University Institute of Information Technology, PMAS-Arid Agriculture University, Rawalpindi 46000, Pakistan*
[2] *Faculty of Engineering and Technology, HUIC, Islamabad 44000, Pakistan*

Correspondence should be addressed to Junaid Ali Khan; junaid.ali@ciit-attock.edu.pk

Received 22 July 2014; Accepted 19 November 2014; Published 10 December 2014

Academic Editor: Paolo Gastaldo

DC motors are used in numerous industrial applications like servo systems and speed control applications. For such systems, the Proportional+Integral+Derivative (PID) controller is usually the controller of choice due to its ease of implementation, ruggedness, and easy tuning. All the classical methods for PID controller design and tuning provide initial workable values for K_p, K_i, and K_d which are further manually fine-tuned for achieving desired performance. The manual fine tuning of the PID controller parameters is an arduous job which demands expertise and comprehensive knowledge of the domain. In this research work, some metaheuristic algorithms are explored for designing PID controller and a comprehensive comparison is made between these algorithms and classical techniques as well for the purpose of selecting the best technique for PID controller design and parameters tuning.

1. Introduction

In this modern industrial age, there is hardly any industrial application in which DC motors are not being used [1, 2]. This is so because of ease of control, low cost maintenance especially of brushless DC motor type, low price, and ruggedness of DC motor over a wide range of applications. Some industrial applications, which are worth mentioning, in which DC motors are being used widely are machine tools, paper mills, textile industry, electric traction, and robotics. The flexibility in controller design of DC motors is due to the fact that armature winding and field winding could be controlled separately [3]. In most of the applications of speed control of DC motors, the current in field winding is kept constant and the current in armature winding is varied or vice versa which gives excellent speed control performance over a wide range of desired values. In these applications, the purpose is to track the speed command by keeping output speed at desired level and to achieve desire speed or position control in minimum time without having large overshoots and settling times [4, 5].

There are different types of controllers like lead, lag, LQR (linear quadratic regulator), PID, and sliding-mode control that could be incorporated in control applications [6–8]. Among the few mentioned types of controllers, PID controller is one of the earliest and best understood controllers which is incorporated in almost every industrial control application due to its efficiency and ease of implementation [9]. Although there are many classical techniques for designing and tuning PID controller parameters (K_p, K_i, K_d) which are widely understood and easily applied, one of the main disadvantages of these classical techniques is that, for tuning PID controller through these techniques, expertise and experience are required. This is so because these methods provide a starting point and achieving desired performance fine tuning of parameters through hit-and-trial method is required. However, metaheuristic techniques may be a good choice to its dynamic nature.

Over the years, many metaheuristic and stochastic optimization techniques have been developed which are being applied in every discipline of life [10–12]. These techniques are nature inspired depending upon the swarm intelligence, evolutionary, or foraging behavior of different species. Some of the widely used techniques are genetic algorithm (GA), particle swarm optimization (PSO), and simulated annealing (SA). These metaheuristic algorithms have been successfully

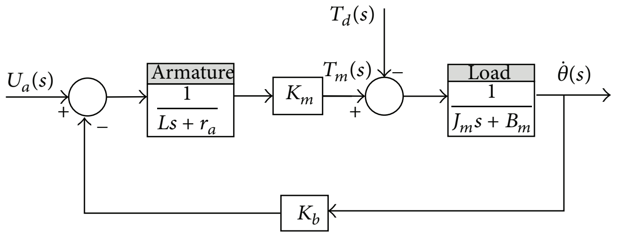


Figure 1: Block diagram of DC motor.

Table 1: DC motor parameters.

Motor parameters	Value
K_m	0.023 N·m/A
K_t	0.023 N·m/A
r_a	1.0 Ω
L	0.5 H
J_m	0.01 Kg·m^2/rad
B_m	0.00003 N·m

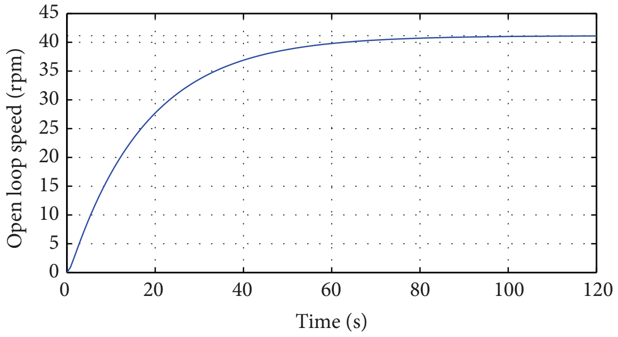


Figure 2: Open loop step response of DC motor.

applied also in various fields of control systems and results obtained through these techniques have proven their supremacy over the classical techniques [13–18].

There have been previously many attempts for the nature inspired PID controller tuning; however, according to the best of our knowledge there has been a little work done in exploiting the power of hybrid techniques for the PID controller tuning. In this work, a PID controller design for speed control of DC motor is presented. First, the design through classical techniques like Zeigler-Nichols and Cohen-Coon methods is presented for establishing a base line. Then, six metaheuristic optimization algorithms are used to find the best possible parameters of PID controller subjected to minimization of a cost function and among these three of the hybrid techniques are used to establish the superiority of hybrid metaheuristic techniques over the others. A comprehensive comparison is made between the classical techniques and the metaheuristic techniques to show the strength, stability, and efficiency of these methods over the classical techniques. Although some stochastic algorithms have been used [13, 16] previously for design of PID controller for different applications, this study presents some hybrid techniques by combining the global and local search techniques of swarm intelligence and evolutionary algorithms for PID controller design and it could provide a framework of PID controller tuning by considering the hybrid nature of metaheuristic techniques.

The organization of the rest of the paper is as follows. In Section 2, mathematical model of DC motor is derived. In Section 3, overview of classical techniques for PID tuning is presented. In Section 4, an overview of GA, PSO, SA, and Nelder-Mead (NM) is presented. In Section 5, results and simulations are discussed. In Section 6, conclusions about the present study are presented.

2. Mathematical Model of DC Motor

The dynamic behavior of DC motor is given by following set of relations [19] and its block diagram is shown in Figure 1. A simplified linear model is presented for this work ignoring the nonlinearities like the backlash and dead zones to simplify the application of metaheuristic techniques. Consider

$$\begin{gathered}
T_m(s) = K_m I_a(s), \\
U_a(s) = (r_a + Ls) I_a(s) + U_b(s), \\
U_b(s) = K_b \dot{\theta}(s), \\
I_a(s) = \frac{U_a(s) - K_b \dot{\theta}(s)}{r_a + Ls}, \\
T_l(s) = J_m s^2 \theta(s) + B_m s \theta(s), \\
T_l(s) = T_m(s) - T_d(s),
\end{gathered} \tag{1}$$

where U_a is armature applied voltage, U_b is back-emf, K_m is motor constant, K_b is back emf constant, J_m is inertia of rotor, B_m is viscous damping, T_m is developed motor torque, T_l is torque delivered to load, T_d is disturbance torque (we neglect it), r_a is armature resistance, L is armature inductance, I_a is armature current, and s is s-plane.

By using (1) the transfer function of DC motor is

$$\frac{\dot{\theta}(s)}{U_a(s)} = \frac{K_m}{(r_a + Ls)(J_m s + B_m s) + K_m K_b}. \tag{2}$$

The parameters selected for DC motor simulation are listed in Table 1 [20].

The open loop step response of DC motor without PID controller is shown in Figure 2 and output values of response are provided in Table 2.

3. Classical Techniques

The response of DC motor is calculated by Ziegler-Nichols (Z-N) and Cohen-Coon (C-C) methods, respectively, in order to validate the proposed scheme.

3.1. Ziegler-Nichols (Z-N) Method. Ziegler and Nichols proposed a rule for design and tuning of PID controller. From the

Table 2: DC motor output response.

Response	Value
Settling time, t_s	68.6029
Rise time, t_r	38.2458
Maximum percent overshoot, M_p	0
Steady state error, e_{ss}	0.0237

open loop step response of system one can find the following set of points (t_a, y_a) and (t_b, y_b). The first point corresponds to output of step response at a value of 35.3% and the second point corresponds to output of step response at a value of 85.3% [20]. By using this method, we get open loop step response values of DC motor given as follows:

$$\begin{gathered} t_a = 8\text{ sec}, \qquad y_a = 14.52, \\ t_b = 37.7\text{ sec}, \qquad y_b = 35.09, \end{gathered} \tag{3}$$

and dc gain is

$$\beta = 41.1449. \tag{4}$$

According to Ziegler-Nichols rule the following relations could be used for determining PID controller parameters:

$$K_p = 1.2\frac{T}{\beta\alpha}, \tag{5}$$

$$T_i = 2\alpha, \tag{6}$$

$$T_d = 0.5\alpha, \tag{7}$$

$$K_i = \frac{K_P}{T_i}, \tag{8}$$

$$K_d = T_d K_P, \tag{9}$$

where

$$T = 0.67\left(t_b - t_a\right), \tag{10}$$

$$\alpha = 1.3t_a - 0.29t_b. \tag{11}$$

By using (5) to (11) we get the values of K_p, K_i, and K_d listed in Table 3.

3.2. Cohen-Coon (C-C) Method. Cohen and Coon also proposed a method for design and tuning of PID controller. Based on this method, PID controller parameters could be determined as follows [20]:

$$\begin{gathered} K_p = \frac{T}{\alpha\beta}\left[\frac{4}{3} + \frac{\alpha}{4T}\right], \\ T_i = \alpha\left[\frac{32 + 6\alpha/T}{13 + 8\alpha/T}\right], \\ T_d = \alpha\left[\frac{4}{11 + 2\alpha/T}\right]. \end{gathered} \tag{12}$$

By using (12) and also from (8) to (11) the following values of K_p, K_i, and K_d have been obtained and listed in Table 4.

Table 3: K_p, K_i, and K_d values by using Z-N method.

Method	K_p	K_i	K_d
Ziegler-Nichols	0.8000	0.6380	0.2508

Table 4: K_p, K_i, and K_d values by using C-C method.

Method	K_p	K_i	K_d
Cohen-Coon	0.3522	0.2317	0.0798

4. Metaheuristic Algorithms: An Overview

4.1. Genetic Algorithm (GA). The GA was invented by John Holland by inspecting adaptation in natural and artificial systems in 1975 [21]. Holland proposed GA as a heuristic tool which is based on the method of "survival of the fittest." The principle of GA is simple: *imitate genetic and natural selection in the form of computer program*. The parameters of the problem are coded as DNA-like linear data structure, a vector or a string. GA differs from conventional optimization techniques in a way that

(i) GA works with coding the solution set and the solution is not itself used;

(ii) GA uses population of solution for finding the optimum solution;

(iii) it uses the fitness function rather than derivatives for evaluation purpose;

(iv) it uses the probabilistic transition operators whereas other techniques use deterministic transition operators.

The general steps in GA algorithm are as follows.

Step 1 (create a random initial population). An initial population is created from a random selection of solution.

Step 2 (evaluate fitness). Each candidate solution is evaluated against a fitness function.

Step 3 (reproduction). The chromosomes with higher value of fitness value are more likely to reproduce offspring.

Step 4 (next generation). If the new generation contains desired solution, then problem has been solved, and if it is not the case, then the new population will go through the same process as described above.

Step 5. Termination based on number of iterations or required fitness value was achieved.

In our design method PID controller parameters (K_p, K_i, K_d) are viewed as individuals and each parameter value is coded by a real number.

4.2. Particle Swarm Optimization (PSO). The particle swarm optimization algorithm was proposed by Kennedy and Eberhart in 1995 [22]. This algorithm is based on the social behavior of swarm such as bird flocking and fish schooling. Here, each individual bird is called the particle, and in our case each individual particle has three attributes which are K_p, K_i, and K_d.

Each particle in PSO flies in the whole search space with a certain velocity which is dynamically adjusted according to its own flying experience and the knowledge gain by the velocities of its peers. Each particle is treated as particle in d-dimensional search space.

The change in velocity and position of each particle is governed by the following relations:

$$v(\text{new}) = \delta v(\text{old}) + a_1 \left[p_{\text{best}} - x(\text{old})\right] + a_2 \left[g_{\text{best}} - x(\text{old})\right]$$
$$x(\text{new}) = x(\text{old}) + v(\text{new}), \qquad (13)$$

where δ is inertia weight factor, p_{best} is local best, g_{best} is global best, v is velocity of the particle, x is the position of the particle, $a_1[p_{\text{best}} - x(\text{old})]$ is individual intelligence, and $a_2[g_{\text{best}} - x(\text{old})]$ is collective intelligence.

4.3. Simulated Annealing (SA). The simulated annealing was originally inspired by the crystal formation process when solids are cooled down from some high temperature [21]. The slower the cooling is, the more perfect crystal is formed. In cooling process, we ultimately converge towards minimal and stable energy state. The system movement towards stability is random, but the probability to stay in any particular configuration directly depends on the energy of the system and on its temperature. The said probability is given by the following Gibbs relation:

$$p = e^{E/kT}, \qquad (14)$$

where E is the energy in different energy states, k is the Boltzmann constant, and T is the temperature.

In 1970, Kirkpatrick formed the simulated annealing algorithm by using the above said analogy. The SA behaves like hill climbing algorithm, but it avoids the local minimum and it usually gives good results. As in stochastic hill climbing algorithm, in each iteration of SA a new solution in the neighborhood of actual solution is randomly chosen. If the fitness value of the new solution is better than the previous value, then the new solution is accepted as the new current solution. If the fitness function is not improved, then the new solution is retained with a probability

$$p = e^{-[f(y)-f(x)]/kT}, \qquad (15)$$

where $f(y) - f(x)$ is the difference of fitness function between the new and the old solutions.

4.4. Nelder-Mead (NM) Method. Nelder-Mead method is a simplex method which is used to find the local minimum of a function of several variables [23]. For the two variables function simplex is a triangle of three vertices and the method is a pattern search. At the worst vertex the function value is the largest and this worst vertex is replaced with new vertex and the search is continued by forming the triangular patterns. The function values at vertices get smaller and smaller and the triangle size reduces which ultimately results in finding the local minimum.

Table 5: Parameters of PID through Z-N and C-C methods.

Method	K_p	K_i	K_d
Ziegler-Nichols	0.8000	0.6380	0.2508
Cohen-Coon	0.3522	0.2317	0.0798

Table 6: Response for Z-N and C-C methods.

Method	t_s	t_r	M_p	e_{ss}
Ziegler-Nichols	6.2222	0.8058	29.2329	0.5
Cohen-Coon	12.6787	1.3370	36.8658	0.5

5. Results and Discussions

Using the classical methods of PID tuning, Ziegler-Nichols and Cohen-Coon, PID controller was tuned by keeping the objective that settling time t_s and rise time t_r should be minimum and there should be no overshoots; that is, maximum percent over shoot M_p should be zero. The PID controller parameters obtained by these classical methods are given in Table 5.

By using the above values of K_p, K_i, and K_d, PID controller was designed and the closed loop response of this controller with the plant is shown in Figure 3 and also the output values of t_s, t_r, M_p and steady state error e_{ss} are shown in Table 6.

It is apparent from Figure 3 and Table 6 that the settling time and maximum percent overshoot are very large and PID controller designed by these methods will not provide desired performance. So these techniques are not helpful in tuning the PID controllers for the systems in which the parameters of t_s, t_r, and M_p are critical.

The same problem is also solved by six metaheuristic algorithms, namely, GA, PSO, SA, GA-NM, PSO-NM, and SA-NM, for designing and tuning of PID controller. Not only the global search but also the hybrid searching techniques using the Nelder-Mead algorithm as the local minimization search technique are also applied. The purpose is to study the optimization capabilities of these metaheuristic algorithms and to demonstrate that by using these techniques PID controller design and tuning give the more accurate and better results. There might be many possible objective functions like integral of time-absolute-error (ITAE), integral of absolute-error (IAE), integral of time-weighted-squared-error (ITSE), and integral of squared-error (ISE), but the objective function used in these algorithms for the minimization is defined as follows [24]:

$$\min J = \left(1 - e^{-\xi}\right)\left(M_p + e_{ss}\right) + e^{-\xi}\left(t_s - t_r\right). \qquad (16)$$

One can change the value of ξ, which is the weighting factor, in three steps, that is, $\xi = 0.5$, $\xi = 1.0$, and $\xi = 1.5$,

TABLE 7: Parameters range.

Parameter	Lower bound	Upper bound
K_p	0.0001	1.5000
K_d	0.0001	1.0000
K_i	0.0001	1.0000

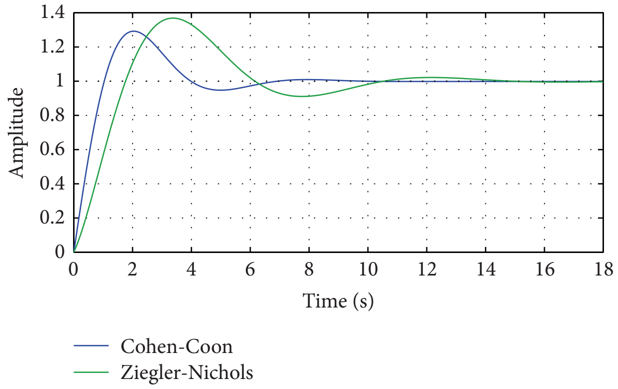


FIGURE 3: Step response of system tuned by ZN and CC methods.

and see the effect of change of ξ on t_s, t_r, and M_p. The lower and upper bounds on the values of K_p, K_i, and K_d are given in Table 7 [25], while the parameters settings or values used in the simulations of various solvers are provided in Table 8.

The fitness function J values obtained by using these algorithms are shown in Tables 9–11.

As shown in Tables 9–11 and also in Figures 4, 5, and 6 the minimum objective function fitness value is found by SA-NM. The objective function contains the exponential factor and also the SA algorithm is governed by the Boltzmann distribution of the energies in different energies states which is also in the form of exponential function. So it is the reason that, for this particular problem, when we combined the SA with the local search method of NM it gave the best results even better than PSO and GA alone. So if the problem landscape and the algorithm which is being used match, then best results would be obtained. Also from Tables 8–10, it could be seen that the standard deviation for the SA-NM is small, which is a measure of convergence speed, and it could be inferred that SA-NM converges fast as compared to other algorithms used in this research work.

The step response for the mean values of K_p, K_i, and K_d tuned by SA-NM is shown in Figure 7 and also values of t_s, t_r, and M_p are given in Table 12. The step response in Figure 7 shows marked improvement over the previous step response of Figure 3 when PID controller was tuned by using classical techniques.

A multiple setpoint command is applied to the system tuned by SA-NM, Z-N, and C-C. The response to multiple setpoint is also shown in Figures 8 and 9. It is quite obvious from Figure 9 that the multiple setpoint tracking is excellent when PID controller is tuned using SA-NM. In this work, a simplified linear model of the DC motor is considered neglecting the nonlinearities like backlash, dead zone, and effects of load torque changes.

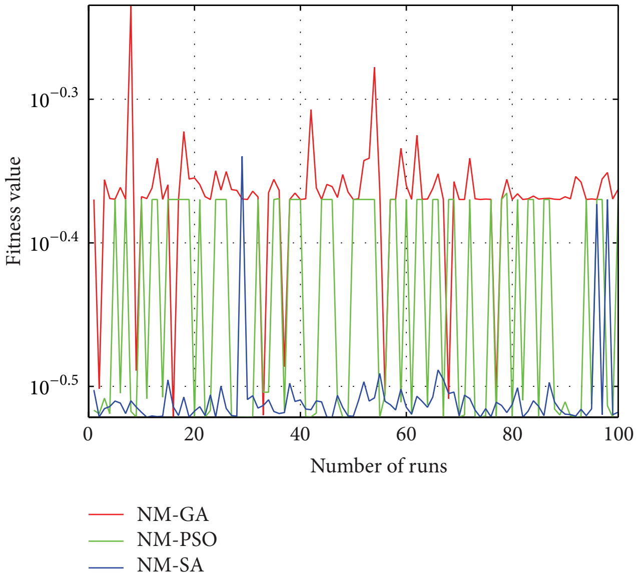


FIGURE 4: Fitness (J) for GA-NM, PSO-NM, and SA-NM, ξ = 0.5.

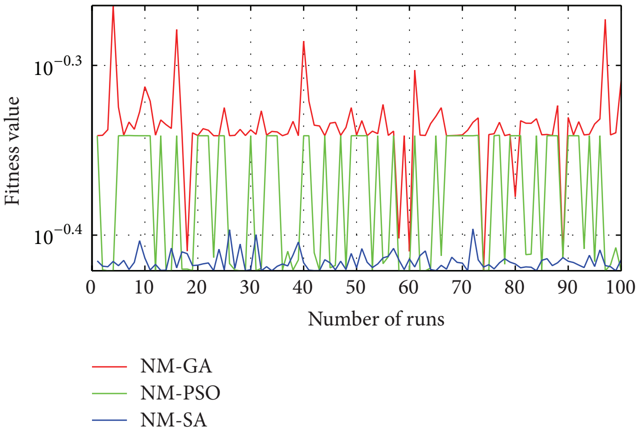


FIGURE 5: Fitness (J) for GA-NM, PSO-NM, and SA-NM, ξ = 1.0.

6. Conclusion

On the basis of results obtained in the previous section the following conclusions could be drawn.

The problem under discussion could be solved using metaheuristic techniques which give a substantial amount of improvement in terms of t_s, t_r, and M_p as can be seen from Tables 6 and 12. Using metaheuristic techniques t_s decreased by 83.93%, t_r decreased by 17.58%, and M_p decreased by 98.71% as compared to Z-N method. Also the decrease in value for t_s, t_r, and M_p was 92.11%, 50.32%, and 98.97%, respectively, as compared to C-C method. So the objective

Table 8: Parameters setting and fitness function.

Fitness function: $\min J = (1 - e^{-\xi})(M_p + e_{ss}) + e^{-\xi}(t_s - t_r)$			
GA parameters	PSO parameters	SA parameters	NM parameters
Population type: double vector	Population type: double vector	Max. evaluations: 3000	Max. iterations: 200
Population size: 20	Population size: 40	Initial temperature: 100	Max. function evaluations: 600
Selection function: tournament	Cognitive attraction: 0.5	Temperature update function: exponential	Start point = Best parameters of SA, GA, PSO
Crossover fraction: 0.8	Social attraction: 1.25	Reannealing interval: 100	
Mutation function: Gaussian	Stall generations: 50	Annealing function: Boltzmann	Others = Default
Stall generations: 50	Function tolerance: $1e - 6$	Stall iterations: 500	X tolerance: $1e - 4$
Function tolerance: $1e - 6$	Number of epochs: 100	Function tolerance: $1e - 6$	Function tolerance: $1e - 6$
Number of epochs: 100	Others = Default	Number of epochs: 100	Number of epochs: 100

Table 9: Fitness (J) case 1, $\xi = 0.5$.

Algo.	Min.	Max.	Mean	Var.	Std.	Runs
GA	0.4303	0.6804	0.4693	0.0019	0.0433	100
PSO	0.3010	0.4402	0.3665	0.0037	0.0612	100
SA	0.3040	15.9880	1.1158	6.7519	2.5984	100
NM-GA	0.3011	0.5827	0.4272	0.0016	0.0398	100
NM-PSO	0.3010	0.4311	0.3655	0.0038	0.0618	100
NM-SA	0.3010	0.4573	**0.3108**	0.0005	0.0229	100

Table 10: Fitness (J) case 2, $\xi = 1.0$.

Algo.	Min.	Max.	Mean	Var.	Std.	Runs
GA	0.4563	14.5462	0.6789	2.2793	1.5097	100
PSO	0.3793	0.4566	0.4206	0.0014	0.0368	100
SA	0.3803	18.3989	NaN	NaN	NaN	100
NM-GA	0.3795	0.5436	0.4604	0.0005	0.0233	100
NM-PSO	0.3793	0.4559	0.4203	0.0014	0.0371	100
NM-SA	0.3793	0.4015	**0.3839**	0	0.0046	100

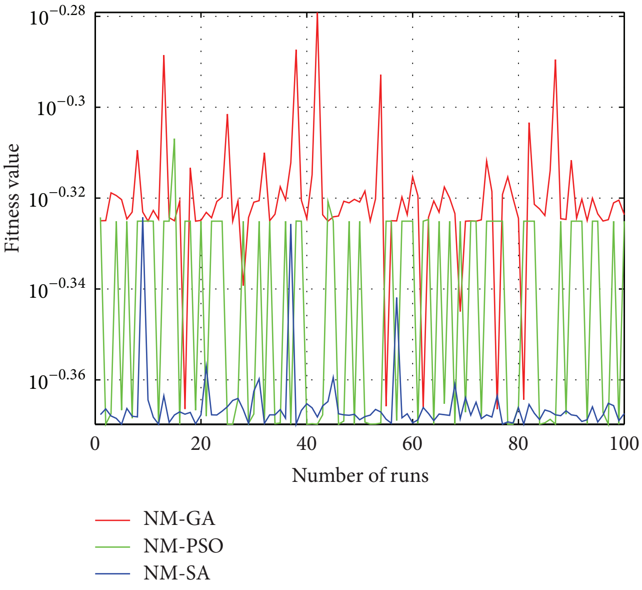


Figure 6: Fitness (J) for GA-NM, PSO-NM, and SA-NM, $\xi = 1.5$.

Table 11: Fitness (J) case 3, $\xi = 1.5$.

Algo.	Min.	Max.	Mean	Var.	Std.	Runs
GA	0.4739	0.5564	0.4907	0.0003	0.0177	100
PSO	0.4268	0.5093	0.4524	0.0006	0.0236	100
SA	0.4282	43.1661	1.8708	40.1765	6.3385	100
NM-GA	0.4301	0.5259	0.4767	0.0002	0.0149	100
NM-PSO	0.4268	0.4933	0.4520	0.0005	0.0231	100
NM-SA	0.4268	0.4740	**0.4306**	0	0.0070	100

Table 12: Step response values for SA-NM.

Method	t_s	t_r	M_p	e_{ss}
SA-NM	0.9999	0.6641	0.3773	0.5

of minimizing t_s, t_r, and M_p is better achieved through metaheuristic algorithms, which validates the capabilities of these algorithms of best optimization algorithms for complex problems like PID controller tuning.

Among all metaheuristic techniques for optimization the hybrid approach of SA-NM showed the best results and through this we obtained the PID controller parameters which showed the excellent results in terms of t_s, t_r, and M_p. Hence, metaheuristic techniques are far better than the classical techniques, and among these hybrid approach is preferable. However, in future work, these effects could be considered to form a comprehensive nonlinear model of DC motor and solve the problem of PID tuning based on the mentioned metaheuristic techniques because according to the best of our knowledge there has been a lot of research potential for the PID controller tuning by considering the nonlinearities of DC motor model. Some other techniques like ant colony optimization (ACO), bacterial foraging algorithm (BFA), and differential evolution (DE) could also be considered by making the hybrid pairs of these algorithms with some local search techniques.

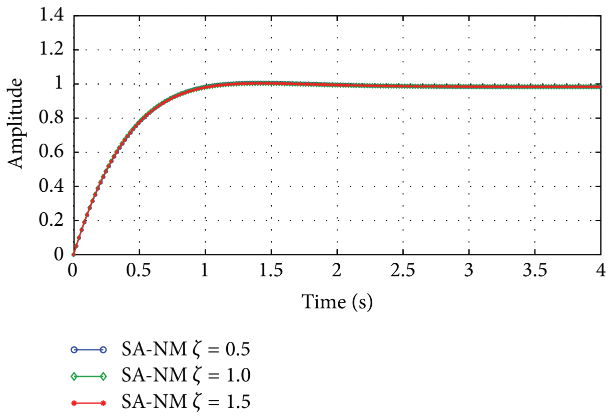


FIGURE 7: Step response for SA-NM.

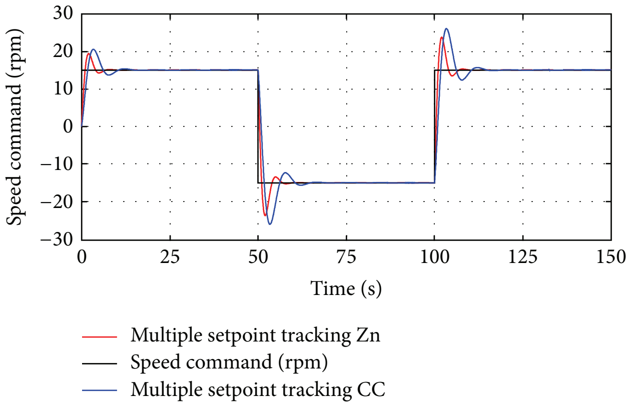


FIGURE 8: Multiple setpoint tracking for Z-N and C-C tuned PID.

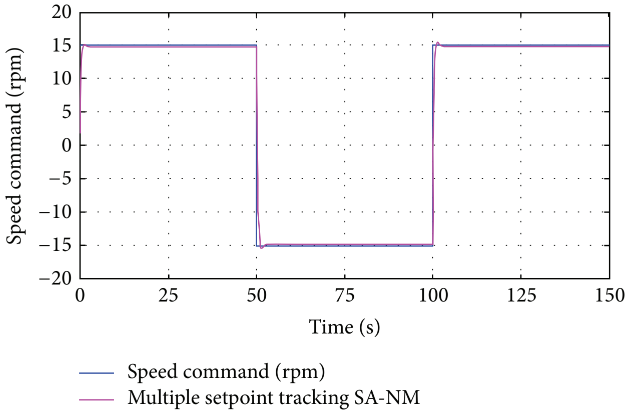


FIGURE 9: Multiple setpoint tracking for SA-NM tuned PID.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] W. Cui, Y. Gong, and M. H. Xu, "A permanent magnet brushless DC motor with bifilar winding for automotive engine cooling application," *IEEE Transactions on Magnetics*, vol. 48, no. 11, pp. 3348–3351, 2012.

[2] A. Sathyan, M. Krishnamurthy, N. Milivojevic, and A. Emadi, "A low-cost digital control scheme for brushless DC motor drives in domestic applications," in *Proceedings of the 2009 IEEE International Electric Machines and Drives Conference*, pp. 76–82, May 2009.

[3] P. C. Krause, O. Wasynczuk, and S. D. Sudhoff, *Analysis of Electric Machinery and Drives*, John Wiley & Sons, 2013.

[4] A. Andrzejewski, "The time-minimal and without overshoot speed control of DC motor," in *Proceedings of the International Conference on Computer as a Tool (EUROCON '07)*, pp. 1792–1799, September 2007.

[5] S. Galijašević, Š. Mašić, S. Smaka, A. Akšamović, and D. Balić, "Parameter identification and digital control of speed of a permanent magnet DC motors," in *Proceedings of the 23rd International Symposium on Information, Communication and Automation Technologies*, October 2011.

[6] F. E. Hoyos, A. Rincón, J. A. Taborda, N. Toro, and F. Angulo, "Adaptive quasi-sliding mode control for permanent magnet DC motor," *Mathematical Problems in Engineering*, vol. 2013, Article ID 693685, 12 pages, 2013.

[7] H. Zhang, L. Ge, M. Shi, and Q. Yang, "Research of compound control for DC motor system based on global sliding mode disturbance observer," *Mathematical Problems in Engineering*, vol. 2014, Article ID 759147, 7 pages, 2014.

[8] J. Yao, G. Yang, Z. Jiao, and D. Ma, "Adaptive robust motion control of direct-drive DC motors with continuous friction compensation," *Abstract and Applied Analysis*, vol. 2013, Article ID 837548, 14 pages, 2013.

[9] R. Namba, T. Yamamoto, and M. Kaneda, "Robust PID controller and its application," in *Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 3636–3641, October 1997.

[10] Y. Harrath, B. Chebel-Morello, and N. Zerhouni, "A genetic algorithm and data mining based meta-heuristic for job shop scheduling problem," in *Proceedings of the 2002 IEEE International Conference on Systems, Man and Cybernetics*, pp. 280–285, October 2002.

[11] X. Li and M. Yin, "Self-adaptive constrained artificial bee colony for constrained numerical optimization," *Neural Computing and Applications*, vol. 24, no. 3-4, pp. 723–734, 2014.

[12] Z. Cui and X. Gao, "Theory and applications of swarm intelligence," *Neural Computing and Applications*, vol. 21, no. 2, pp. 205–206, 2012.

[13] R. G. Kanojiya and P. M. Meshram, "Optimal tuning of PI controller for speed control of DC motor drive using particle swarm optimization," in *Proceedings of the International Conference on Advances in Power Conversion and Energy Technologies (APCET '12)*, August 2012.

[14] Y. Wang, C. Xia, M. Zhang, and D. Liu, "Adaptive speed control for brushless DC motors based on genetic algorithm and RBF neural network," in *Proceedings of the IEEE International Conference on Control and Automation (ICCA '07)*, pp. 1219–1222, June 2007.

[15] H. M. Asifa and S. R. Vaishnav, "Particle swarm optimisation algorithm based PID controller," in *Proceedings of the 3rd International Conference on Emerging Trends in Engineering and Technology (ICETET '10)*, pp. 628–631, November 2010.

[16] C. Cao, X. Guo, and Y. Liu, "Research on ant colony neural network PID controller and application," in *Proceedings of*

the 8th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, pp. 253–258, August 2007.

[17] M. A. Sahib, B. S. Ahmed, and M. Y. Potrus, "Application of combinatorial interaction design for dc servomotor pid controller tuning," *Journal of Control Science and Engineering*, vol. 2014, Article ID 576868, 7 pages, 2014.

[18] Y.-C. Luo, Z.-S. Ke, and Y.-P. Kuo, "Sensorless rotor-field oriented controlled inductiion motor drive with particle swarm optimization algorithm speed controller design strategy ," *Controller Design Strategy*, vol. 2014, Article ID 861462, 13 pages, 2014.

[19] R. H. B. Richard and C. Dorf, *Modern Control Systems*, Pearson, 11th edition, 2008.

[20] A. N. A. K. Mishra, "Speed control of DC motor using particle swarm optimization technique," *International Journal of Engineering Research and Technology*, vol. 2, no. 6, pp. 1643–1649, 2013.

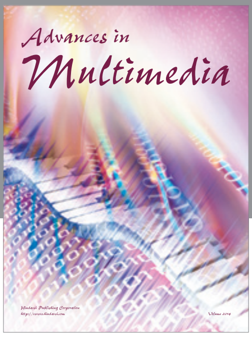
[21] S. D. S. N. Sivanandam, *Principles of Soft Computing*, John Wiley & Sons, New York, NY, USA, 2nd edition, 2013.

[22] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the 1995 IEEE International Conference on Neural Networks*, pp. 1942–1948, Perth, Wash, USA, December 1995.

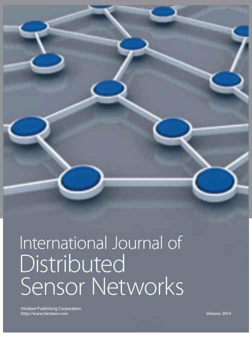
[23] J. H. Mathews and K. K. Fink, *Numerical Methods Using MATLAB*, Prentice-Hall, 4th edition, 2004.

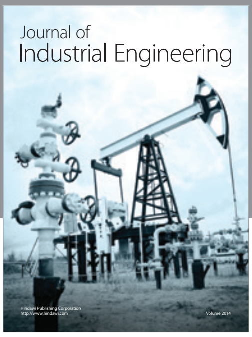
[24] Z.-L. Gaing, "A particle swarm optimization approach for optimum design of PID controller in AVR system," *IEEE Transactions on Energy Conversion*, vol. 19, no. 2, pp. 384–391, 2004.

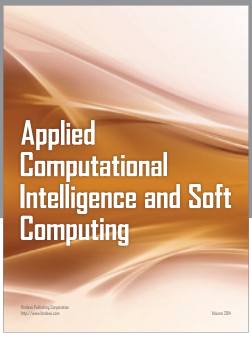
[25] C.-C. Wong, S.-A. Li, and H.-Y. Wang, "Optimal PID controller design for AVR system," *Tamkang Journal of Science and Engineering*, vol. 12, no. 3, pp. 259–270, 2009.

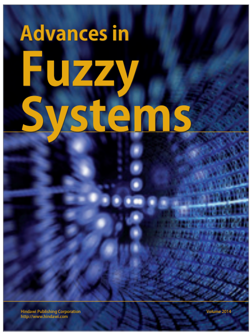

Advances in
Multimedia

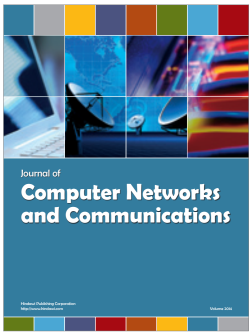
The Scientific
World Journal

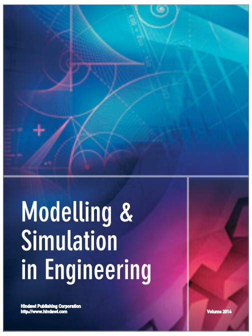

International Journal of
Distributed
Sensor Networks

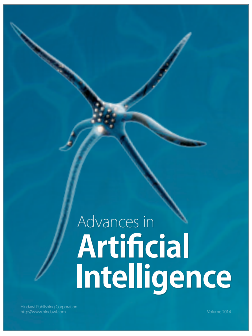

Journal of
Industrial Engineering

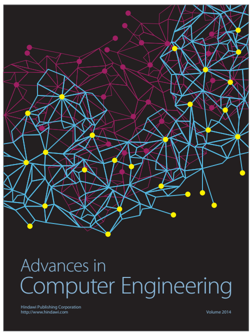

Applied
Computational
Intelligence and Soft
Computing


Advances in
Fuzzy
Systems

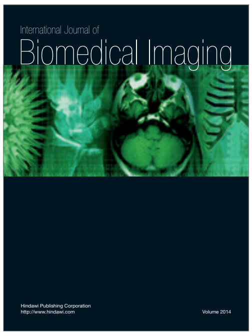

Journal of
Computer Networks
and Communications

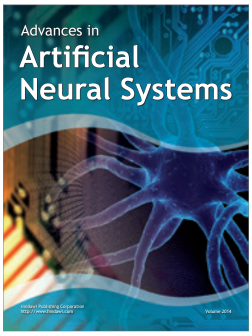

Modelling &
Simulation
in Engineering

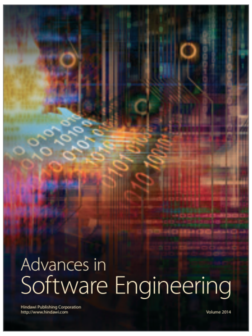

Advances in
Artificial
Intelligence


Advances in
Computer Engineering

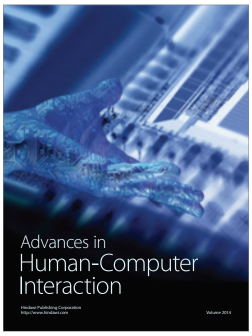
International Journal of
Computer Games
Technology

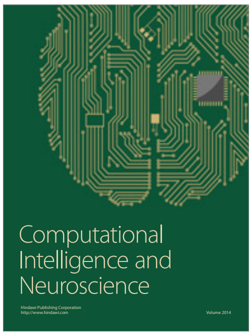

International Journal of
Biomedical Imaging


Advances in
Artificial
Neural Systems


Advances in
Software Engineering

Journal of
Robotics


Advances in
Human-Computer
Interaction


Computational
Intelligence and
Neuroscience

International Journal of
Reconfigurable
Computing

Journal of
Electrical and Computer
Engineering