Genetic Algorithm Based Adaptive PID Tuning of Time Delay Process

Alpana Barman[1], Souvik Dutta[1](✉), Kamalika Tiwari[2], Soumitra Roy[1], and Santigopal Pain[1]

[1] Haldia Institute of Technology, Haldia, West Bengal 721657, India
souvikdutta77@yahoo.in

[2] Dr. B. C. Roy Engineering College, Durgapur, West Bengal 713206, India

Abstract. Conventional Proportional Integral Derivative controllers are (CPID) unable to provide suitable performance because of large overshoots and oscillations for integrating and nonlinear systems. PID control is extensively used in different control systems, but due to analytically selected parameters K_P, K_I, K_D it is tough to attain parameter optimization. So, it is necessary to use adaptive PID controllers (APID) to get desired performance. Ziegler–Nichols(Z-N) method is a classical method which is used tune the PID parameters in conventional way. It is not easy to tune PID parameters using this method. Ziegler–Nichols tuned PID controller provides better response compare to conventional PID controller. But still there exist some large overshoot and takes more time to damp out oscillations. Natural evolution is represented by Genetic Algorithms (GAs) and for optimization this stochastic global search method is used in this problem. Here, the comparison has been made between PID controller tuned with different standard classical methods along with fuzzy PID (FPID) and proposed genetic algorithm based adaptive PID control (GAPID) technique. It has been found that better response is obtained by adaptive GAPID compare to classical methods and FPID. It is observed that the 2nd order time delay system having delay time L = 0.2 s and L = 0.3 s, GAPID gives minimum rise time and settling time compare to other methods.

Keywords: Time delay process · Adaptive PID · Genetic Algorithm

1 Introduction

The time gap between the starting point of an event and its output in another point is characterized as delay in a control system [1]. It is important to evaluate the performance as well as stability of the system with delay which renowned as transport lag, dead time, time lag etc. Mostly PI and PID controllers are used to get better performance due to their simple construction and less maintenance [2, 3]. It has been observed that in absence of derivative action creates a PID controller simple and less responsive to noise. For more settings of PI controller, usually Ziegler–Nichols proportional Integral Control (ZNPIC) technique is applicable for first and second order system. Sometimes ZNPIC technique

A. A. Sk et al. (Eds.): ISAI 2022, CCIS 1695, pp. 64–75, 2022.
https://doi.org/10.1007/978-3-031-22485-0_7

is not preferable due to unnecessary oscillation and overshoot [4–6]. PID controllers are widely used in industrial closed loop control now-a-days and ZN tuning is one of the most common approaches for obtaining realistic initial settings for PID controllers. It has been observed that Ziegler-Nichols Proportional Integral Derivative (ZNPID) technique is acceptable for first order system but not able to get suitable performance for high order system [6]. So far several classical methods are implemented to find specific parameters of PI or PID controllers to get desirable output in physical system. It has been seen that Panda et al. [7] proposed a PI controller in which gain values can be scheduled. Here the values of proportional and integral gain can be updated depending on the process error. Weng Khuen Ho et al. (1996) focused on the efficiency and resilience of well-known PID formulas (Ziegler-Nichols, Cohen-Coon, and tuning formulas) for the processes having deadtime between 0.1 and 1 [8]. In 2011, C. Dey et al. reported an auto tuning PID controller All of the PID controller's components are separately updated online by a nonlinear updating factor. In this paper performance and stability robustness of APID are implemented by introducing some disturbances in model and controller parameters [9]. Rajani K. Mudi et al. proposed an augmented Ziegler–Nichols tuned PI controller (AZNPIC) to get better values of system parameters. Because of poor performance of Ziegler–Nichols tuned PI and PID controllers for high order and non-linear system, this AZNPIC is introduced [10]. It is also observed that the performance of AZNPIC is far better than ZNPIC as well as Refined Ziegler–Nichols tuned PI controller (RZNPIC). To resolve this above said issue an improved auto-tuning scheme is implemented for Ziegler–Nichols (ZN) tuned PID controllers (ZNPIDs). To overcome maximum overshoot in high order non-linear system ZNPIDs are upgraded and implemented [11]. In this paper, a comparative study is being observed of augmented Ziegler–Nichols tuned PID controller (AZNPID) for high order linear and non-linear dead-time processes over ZNPID and RZNPID. In spite of quite acceptable output using these classical methods, it will be very difficult to tune for a high order system. It is very complicated to design a suitable PID controller due to large size, more space and huge maintenance for high order system. The complexity of the system is increased due to large structure; nonlinearities present in the system. All the above mentioned conventional classical methodologies are designed for specific type of disturbances and low order system. To overcome above said problems different adaptive control techniques are introduced in days to get satisfactory performance in real life system along with high order system i.e. pure integrating processes with delay (IPD) [12]. Parikshit Kr Paul et al. proposed IMC-PID controller implemented by fuzzy for pure integrating process with delay. Here, the tuning parameters are selected using fuzzy logic to get better performance of a system. It is found that Internal Model Control-PID (IMC-PID) controller delivers an overall enriched performance along with sufficient robustness in its behavior [13]. Solihin, Mahmud et al. [14] employed particle swarm optimization (PSO) technique to enrich the capability of traditional techniques. In this paper, a comparative study has been shown of PSO-PID based technique over ZN-PID. In tune with the above reported PID settings, here a new optimization technique is proposed which will give better result compare to the old techniques of parameter optimization of PID controllers for a linear time delay second order system. These time delay models are often encountered in everyday real life and industrial application especially in process control and others that use a PID controller to control the output. The objective of this work is to show that by employing the GA based PID tuning for a second order time delay system, the best solution can be attained. This is verified by comparing the GAPID optimized plant's outcome to the classically adjusted plant's output like ZNPID, RZNPID, Luyben PID controller (LPID), AZNPI, AZNPID and FPID. It is observed that the adaptive GAPID optimization process gives minimum value of percentage overshoot along with lowest value of settling time having a time delay of L = 0.2 s. Also, by increasing time delay L = 0.3 s, the system performance has been analyzed.

2 PID Controller

Industrial control systems are mainly associated with PID controller. The error value has been calculated by PID controller. The error value is the difference between the desired set point and a measured value and concerns the correction based on proportional, integral and derivative terms. Generally, PID controller is the combination of proportional action, Integral action and Derivative action. PID controller's algorithms are mostly used in feedback systems. The proportional controller compares the desired value with the real value or the value of the feedback process. Whereas, the output is the product of resulting error and proportional constant. It delivers stable operation of a system but always maintains the steady-state error. To eliminate the steady-state error of a system Integral controller is introduced. The error value is integrated over a period of time to make error value zero which introduces a lag in a system. Derivative controller is introduced to predict the future behavior of the error. By compensating lag initiated by P-action, this D-controller improves the stability of a given system.

The PID controller is formed by combining three types of controllers together having the transfer function:

$$C_{PID(S)} = K_D S + K_P + \frac{K_I}{S}. \tag{1}$$

Now the task is to find the optimum value of three parameters: K_P, K_I, K_D by suitable tuning procedure, to build a system which is capable of satisfying the desired performance criteria. This PID controller must be tuned to fit with dynamics of the process to be controlled before the working of the PID controller takes place. In order to achieve the desired output from the controller, many strategies are available such as trial and error, Zeigler-Nichols and several optimization techniques. In this paper, a comparative study is implemented with K_P, K_I and K_D values using different optimization techniques.

Mostly PID controllers are used to their minimal structure and tuning methods. But due to large overshoot and oscillation the conventional PID controllers normally fail to provide satisfactory performance in real life system [15]. To get a desired output of an actual system a number of nonlinear and adaptive PID controllers are being developed and tuning is the most important step for a successful controller design [10]. It has been observed that a good performance of a first order system can be achieved by Ziegler-Nichols tuned PI controller (ZNPIC) but it fails to deliver pleasing output for high order system. Sometimes this ZNPIC of a system is unsatisfactory because of large overshoot and oscillation [4–6]. It is also observed that Ziegler-Nichols tuned PID controller

(ZNPID) is not desirable in a system due to large overshoot [10]. Similarly, Augmented ZNPID (AZNPID) gives improved performance over ZNPID, refined ZNPID (RZNPID) when it is tested on a second order linear time-delay process.

In this paper the performance of Genetic Algorithm PID controller (GAPID) is experienced and compared with those of ZNPID, RZNPID, AZNPI, AZNPID, FPID and Luyben (LPID) in terms of a number of performance indices like rise time (t_r), percentage overshoot (%OS), Settling time (t_s) and Steady-State Error (SSE).

3 Plant Model

The Plant Transfer Function of time delay process is given as.

$$G_P(s) = e^{-Ls}/(1+s)^2 \tag{1}$$

where L = Delay Time.

Using Padhe's approximation we can write

$$e^{-Ls} = \left(1 - \frac{Ls}{2}\right)\Big/\left(1 + \frac{Ls}{2}\right). \tag{2}$$

and

$$G_P(s) = \left(1 - \frac{Ls}{2}\right)\Big/(1+s)^2\left(1 + \frac{Ls}{2}\right). \tag{3}$$

Here the block diagram of this system with unity negative feedback has been drawn (Fig. 1):

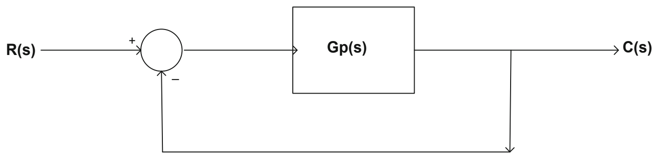

Fig. 1. Block diagram of the plant model with unity negative feedback

The Closed-loop Transfer function by taking L = 0.2 s will be:

$$\frac{C(s)}{R(s)} = \frac{-0.1s + 1}{0.1s^3 + 1.2s^2 + 2s + 2} \tag{4}$$

4 Objective Function or Fitness Function

The objective function has been used for measurement of how individuals have shown performance in the problem domain. When a problem is to be minimized, the fit individuals will have the objective function of minimum numerical value [16]. This raw value of fitness that has been measured is generally used for finding the relative effectiveness of particulars in GA. The objective function value that is transformed into a measure of relative fitness by another function called Fitness Function. In this problem, the maximum fitness values of GA have been calculated for finding the fitter chromosomes. But when it is required to go for the minimization form rather than finding the maximum value, there is need of transformation, a maximization problem to a minimization problem. Here the objective function is inverse of Integral Absolute Error (IAE). The objective function is denoted in Eq. (5). The primary task is to minimize IAE to get optimum solution.

$$F = \frac{1}{\int_0^\infty |e(t)|dt} \tag{5}$$

where, F is fitness function and *e(t)* is error.

5 Proposed Approach

Genetic Algorithms (GAs) are a stochastic global search method that follows the procedure of natural evolution and this method is used for optimization [16]. This method was introduced by John Holland [16]. The genetic algorithm had no starting knowledge of the accurate solution and completely depends on responses from its environment and evolution operators such as selection, crossover and mutation to reach at the exact result. When the algorithm starts at numerous independent points and continue searches in parallel, it doesn't face local minima and sometimes converges to sub optimal solutions [17, 18]. GA have the capability of finding the high-performance areas in complex domains without facing any problem. GAs don't just deal with one possible solution to a problem; they deal with a population of possible alternatives. Chromosomes are the probable solution in the population. Whatever the parameter solution found, these chromosomes are encoded from there. There is a comparison between each and every chromosome in the population and honored by their fitness rating that shows how successful this chromosome to the later [16]. Now to generate new chromosome from the old ones GA will go for crossover and mutation. This is done by either mixing already existed chromosome in the population or by adjusting the current chromosome. The better solution will be achieved when the parent chromosome fitness is taken into account by selection mechanism. It will contribute positive vibes to their offspring. Random population comprising of 20–100 individuals has been considered in GA. Here each chromosome has been represented by a binary string. Every individual has been allotted a corresponding number by the objective function called its fitness. After analyzing the fitness of each and every chromosome, survival of the fittest policy is applied. Here the magnitude of the error is considered to measure fitness of each chromosome. The three important parts of GA are Selection, Crossover and Mutation.

Defining population size is one of the major issues in GA. Sometimes, the population size is decided based on trial and error [19]. The literature survey reveals that in many papers 80 to 100 has been taken as the population size. Here at starting, a population of 100 is considered and the outcome is taken for 50 generation. But, satisfactory result was not found. So, repetition of the process is done with another 50 generations.

Selection: The fitness value is evaluated for each chromosome in selection phase. This value has been applied in the selection process for the individuals to bias which are considered to be fit. As similar to natural evolution, there is a high chance for a fit chromosome to be selected. As a result, the likelihood of an individual being chosen is linked to their fitness, guaranteeing that fitter individuals are more likely to have offspring [20]. Here Stochastic Universal Sampling (SUS) method has been used because of its simplicity. Stochastic Universal Sampling means where individuals are selected randomly from an entire population. Stochastic Universal Sampling method is generally based on Roulette Wheel selection method. Here in SUS multiple pointers has been used for selection. So, by one rotation individuals can be selected.

Crossover: Crossover algorithm will start after selection process is over. In crossover operation there is a swapping of certain parts of two particular strings in a bid to consider better parts of old chromosomes and generate good new ones. Genetic operators directly adjust the behaviour of a chromosome, using the consideration that fitter individuals on average can be produced by certain individual gene codes. The crossover probability shows how frequently crossover operation is executed. A likelihood of 0% indicates that that the offspring will be identical copies of their parents and a probability of 100% implies that each generation will be made up completely of new offspring [16].

Mutation: Selection and crossover will produce a huge quantity of dissimilar strings. Though, two main problems are associated with this:

a) There may not be enough variation in the initial strings to confirm that the GA explores the whole problem space, based on the beginning population chosen. [16].
b) The GA may coincide on sub-optimum strings because of poor starting population selection [16].

So, rectification of these problems may be done by the inclusion of a mutation operator into the GA. In case of a string mutation is the infrequent random change of a value. It is regarded as a contextual operator in the GA. The mutation probability is generally low due to fit strings may be destroyed by a high mutation rate and degrade GA into a random search.

5.1 Genetic Algorithm Summary by Flow Chart

Here the process of GA based adaptive PID tuning method is explained through a flowchart (Fig. 2).

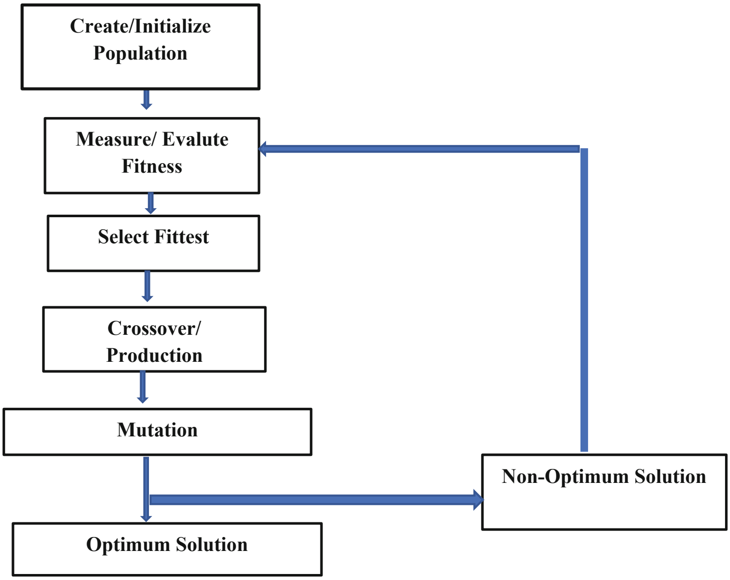

Fig. 2. Genetic Algorithm process flowchart

The steps that are involved in generating and executing the control gains of PID using GA are:

I. Generate a fixed size initial and arbitrary population of individuals.
II. Assess their fitness.
III. Members that are fittest, selected from the population.
IV. Regenerate chromosomes using a probabilistic method that is roulette wheel, SUS etc.
V. Apply crossover procedure for the regeneration of chromosomes
VI. Perform mutation procedure having less possibility.
VII. Start repeating from step II until a prespecified convergence criterion is achieved.

GA having the stopping criterion, is a manipulator stated criteria. The optimum solution is obtained when the string fitness value exceeds a specified threshold or when the maximum number of generations is reached.

Elitism: In case of crossover and mutation process, there are huge chances that the best solution can't be achieved. The fittest string will be preserved by these operators, there is no assurance about that. The elitism models are preferred to avoid this. Here, the best individual is saved from a population before taking place of these operations. When a new population is designed and assessed, this model will go for examining to check if this best structure has been well-kept-up and if it is not satisfied the kept copy is reinjected into the population. The Genetic Algorithm will then resume as usual.

6 Result Analysis

In this program the population size of 100 is formed by creating a 2D matrix, taking 100 rows and 24 columns (8 bits for each of the three parameters K_P, K_I, K_D). Actually, this matrix is a decoded version of the real values of K_P, K_I, K_D as Canonical Genetic Algorithm (CGA) is used. Here Strings or chromosomes, which were initially intended as binary illustrations of solution vectors, make up the population members. So, there is a need to make a subroutine which will convert the real set of values into binary digits and vice versa. Here 2^{nd} order linear process having some time delay $L = 0.2$ s has been considered. The required subroutines are called on for the entire population of 100 rows. This procedure is continued for 50 Generations. Since best solution is still not achieved the whole procedure is repeated for further 50 Generation. Here 2-point crossover with probability of 0.7, and mutation with a probability of 0.01 has been used. Here the plant has high rise time and steady state error.

Comparative study of the newly formed system, PID controller tuned using GA with the system formed earlier by using PID controller tuned with Ziegler Nichol's method is given bellow (Fig. 3, Table 1)).

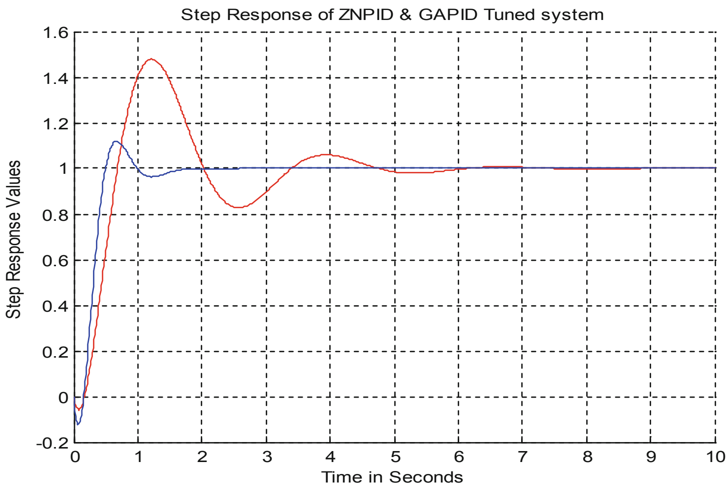

Fig. 3. Comparison of the unit step responses of two systems for $L = 0.2$ s

Table 1. Different parameters of the plant where PID controller tuned with Z-N method and PID controller tuned with GA method

Specifications	PID tuned with Ziegler Nichol's	PID tuned with GA
Rise time (sec)	0.6900	0.5
% Overshoot	47.97	11.97
Settling time (sec)	5.41	1.47

So, after comparing the performance of the two systems, GA based optimization of PID gives more satisfactory result with respect to PID tuned with Ziegler Nichol's method. There is a decrease in rise time, percentage overshoot, settling time in case of GA-PID (Fig. 4, Table 2).

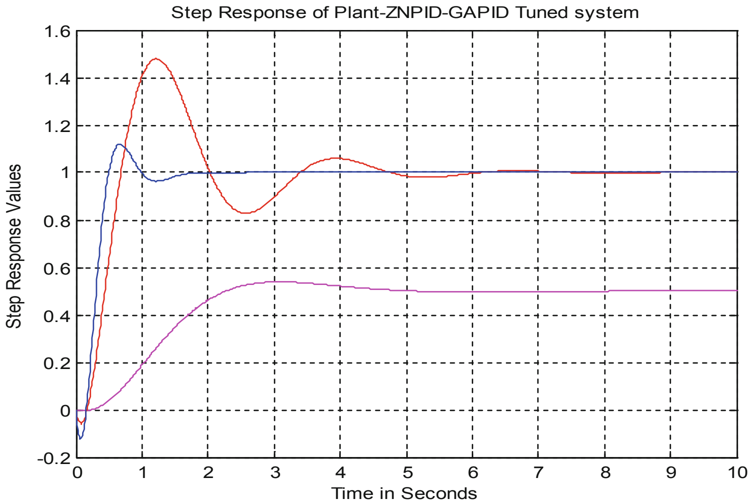

Fig. 4. Comparison of the unit step responses of three systems for L = 0.2 s

Table 2. Different parameters of the plant where PID controller tuned with Z-N classical methods, Fuzzy PID and PID controller tuned with GA method

Specifications	Plant	ZNPID	RZNPID [11]	LPID [11]	AZNPI [11]	AZNPID [11]	FPID [21]	GAPID
Rise time (sec)	2.28	0.69	1.50	1.60	2.30	1.30	2.4	0.5
% Overshoot	7.77	47.97	11.40	0.00	4.04	0.56	30.3	11.97
Settling time (sec)	4.48	5.41	3.90	10.90	2.60	1.60	13.1	1.47
SSE	0.5	0	0	0	0	0	0	0

Here, from the above table it has been observed that the response of the system is improved with the addition of a PID controller tuned with Ziegler Nichol's classical methods, with respect to rise time of the overall system performance (except AZNPI) when subjected to a unit step input. However, this addition of PID controller has introduced high oscillations and instability in the system, which is extremely undesirable. To compensate this or to improve the system performance, the PID controller is tuned with a new technique using Genetic Algorithm. This newly tuned PID controller when added to the existing system shows remarkable improvements in the overall system performance and the drawback of the earlier system i.e. a high percentage overshoot is reduced to a great extent. The settling time is also reduced by a considerable amount. Finally, the efficiency of the system is also improved. In the field of time-delay system the faster the response to reach stability, the better is the design for the plant. Here in case of ZNPID optimum results have been found with control gains $K_p = 6.6$, $K_i = 6.6438$ and $K_d = 1.6401$ and in case of GAPID best result has been found when the value of $K_p = 6.6447$, $K_i = 3.3247$ and $K_d = 3.7153$.

The performance of adaptive GAPID is also studied by taking time delay L = 0.3 s. The result is shown below (Fig. 5, Table 3).

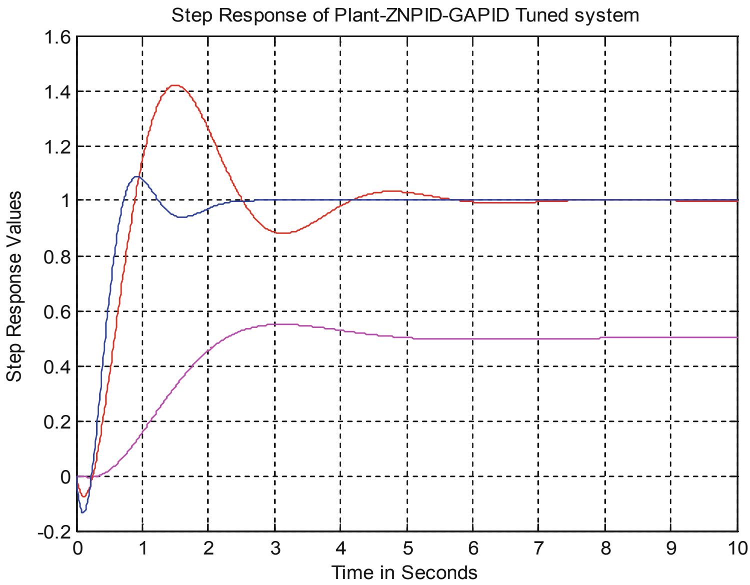

Fig. 5. Comparison of the unit step responses of three systems for L = 0.3 s

Table 3. Different parameters of the plant where PID controller tuned with Z-N classical methods, fuzzy PID and PID controller tuned with GA method

Specifications	Plant	ZNPID	RZNPID [11]	LPID [11]	AZNPI [11]	AZNPID [11]	FPID [22]	GAPID
Rise time (sec)	2.28	0.89	1.30	1.30	2.40	1.10	2.3	0.72
% Overshoot	10.01	42.02	26.40	6.60	5.98	12.27	32.7	8.73
Settling time (sec)	4.59	5.29	5.60	6.60	5.90	4.20	17.2	2.1
SSE	0.5	0	0	0	0	0	0	0

From the above table it has been observed that as the time delay increases from 0.2 to 0.3, there is an increase of percentage overshoot and settling time of the plant. These can be handled sensibly by incorporating PID controller tuned with Ziegler Nichol's classical methods but still the percentage of overshoot and the value of settling time is high. So, system performance has been improved remarkably by incorporating evolutionary adaptive control algorithm that is GAPID. It has been observed from the table that the rise time, maximum overshoot and settling time obtained using GAPID are considerably low compare to different classical ZNPID tuning techniques and FPID. It is also been observed from the response characteristics that the system stability is also improved using GAPID technique.

7 Conclusion

The results reveal that the proposed PID with GA responds substantially faster than the traditional approaches (classical methods). The classical methods are good for determining the PID parameters. Though, the approach for finding the PID values utilizing the classical methods is painful. There are many cumbersome steps need to follow to get the exact PID control gain. In order to see and evaluate how the system responsiveness improves, an optimal method is used to modify the PID parameters. This is accomplished by putting the GA into action. This GA designed PID is much better in terms of the rise time, the settling time and percentage overshoot compared to other methods. Servo-based position control is a common issue in a wide range of industrial operations. This GAPID technique can be implemented on a servo-based position control system to get better performance. This method can also be used in many other time delay processes. The limitation of this algorithm is that the exact values of different controlling parameters may not found. Only optimal solutions are achieved which are good for smooth functioning of the system.

References

1. O' Dwyer, A.: The estimation and compensation of processes with time delays. Ph.D., Thesis, School of Electronic Engineering, Dublin City University (1996)
2. Shinsky, F.G.: Process Control Systems Application, Design, and Tuning. McGraw-Hill, New York (1998)
3. Bissel, C.: Control Engineering. Chapman and Hall, New York (1994)
4. Astrom, K.J., Hang, C.C., Person, P., Ho, W.K. : Towards intelligent PID control. Automatica **28**(1), 1–9 (1992)
5. Basilio, J.C., Matos, S.R.: Design of PI and PID controllers with transient performance specification. IEEE Trans. Educ. **45**(4), 364–370 (2002)
6. Hang, C.C., Astrom, K.J., Ho, W.K.: Refinements of Ziegler-Nichols tuning formula. IEEE Proc-D **138**(2), 111–118 (1991)
7. Panda, S.K., Lim, J.M.S., Dash, P.K., Lock, K.S..: Gain scheduled PI speed controller for PMSM Drive. In: Proc. IEEE industrial electronics society international conference IECON'97, vol. 2, pp. 925–30 (1997)
8. Ho, W.K., Gan, O.P., Tay, E.B., Ang, E.L.: Performance and gain and phase margins of well-known PID tuning formulas. IEEE Trans. Contr. Syst. Technol. **4**(4), 473–477 (1996)
9. Dey, C., Mudi, R.K., Simhachalam, D.: An auto-tuning PID controller for integrating plus dead-time processes. Adv. Mater. Res. **403–408**, 4934–4943 (2011)

10. Mudi, R.K., Dey, C., Lee, T.T.: An improved auto-tuning scheme for PI controllers. ISA Trans. **47**(1), 45–52 (2008)
11. Dey, C., Mudi, R.K.: An improved auto-tuning scheme for PID controllers. ISA Trans. **48**(4), 396–409 (2009)
12. Chidambaram, M., Padma Sree, R.: A simple method of tuning PID controllers for integrator dead-time processes. Comput. Chem. Eng. **27**(2), 211–215 (2003)
13. Paul, P.K., Dey, C., Mudi, R.K.: Design of fuzzy based IMC-PID controller for IPD process. In: International Symposium on Computational and Business Intelligence (2013)
14. Solihin, M., Tack, L., Moey, L.K.: Tuning of PID controller using particle swarm optimization (PSO). In: Proceeding of the International Conference on Advanced Science, Engineering and Information Technology, 1. https://doi.org/10.18517/ijaseit.1.4.93 (2011)
15. Verma, S., Mudi, R.K.: Genetic algorithm-based adaptive PID CONTROLLER. In: Jain, L.C., Patnaik, S., Ichalkaranje, N. (eds.) Intelligent Computing, Communication and Devices. AISC, vol. 308, pp. 57–64. Springer, New Delhi (2015). https://doi.org/10.1007/978-81-322-2012-1_7
16. Meena, D.C., Devanshu, A.: Genetic algorithm tuned PID controller for process control. In: International Conference on Inventive Systems and Control (ICISC-2017)
17. Wang, Q., Spronck, P., Tracht, R..: An overview of genetic algorithms applied to control engineering problems. In: Proceedings of the Second Conference on Machine Learning and Cybernetics (2003)
18. Skogestad, S.: Probably the best simple PID tuning rules in the world. J. Process Control (2001)
19. Luke, S., Balan, G.C., Panait, L.: Population implosion in genetic programming. Department of Computer George Mason University (2003)
20. Giriraj Kumar, S.M., Ravishankar, R., Radha Krishnan, T.K., Dharmalingam, V., Ananthara-man, N.: Genetic algorithms for level control in a real time process, Oct. (2008)
21. Mitra, P., Dey, C., Mudi, R.K.: Dynamic set-point weighted fuzzy PID controller. In: International Symposium on Computational and Business Intelligence (2013)
22. Mitra, P., Dey, C., Mudi, R.K.: Fuzzy rule-based set point weighting for fuzzy PID controller. SN Appl. Sci. **3**(6), 1–34 (2021). https://doi.org/10.1007/s42452-021-04626-0