

A novel Implementation Technique for Genetic Algorithm based Auto-Tuning PID Controller

Student
school address

Antonio
UAC address

Co-Author
UAC address

S. K. Gadi
FIME, Universidad Autónoma de Coahuila,
Torreón, Coahuila, México.
Tel.: (+52 1) 871 757 0239
Fax: (+52 1) 871 757 0239
Email: research@skgadi.com

Abstract—The auto-tuning proportional-integral-derivative (PID) controllers based on genetic algorithms have many limitations in its implementation. This article presents such limitations and provides the possible solutions. An implementation technique is proposed based on these solutions. Numerical simulations are given to evaluate the proposed method applied to a system with variable parameters, a non-linear system and a time delayed system. This technique is used to control a DC servo motor. These results coincide with the obtained simulation results.

I. INTRODUCTION

Theoretically, a proportional-integral-derivative (PID) controller guarantees stability for lower order linear systems [Please correct and provide citation]. However, these are used in many industries to control nonlinear and delayed systems [1], [2]. A nonlinear system behaves like a linear system under certain operating conditions. Hence stability is achieved by a PID controller for a particular operating condition. The PID parameters must be updated when the operating conditions change for a stable operation. In the case of a delayed system, apart from the PID parameters the transportation delay also plays a significant role in its stability. In some cases, introducing an additional delay into the system stabilizes a system.

II. NOMENCLATURE

C	Population	$\mathbb{R}^{n \times 4}$
C_i	i^{th} chromosome of the population	\mathbb{R}^4
e	Error between reference and measured plant output	
K_D	Derivative constant of PID controller	
K_I	Integral constant of PID controller	
K_P	Proportional constant of PID controller	
N	Constant associated with the filter implementing the derivative term in a PID controller	
n	Population size	
r	Reference signal	
s	Laplace transform's complex variable	
T	Time interval for which the i^{th} chromosome is applied	
t	Time	
T_i^+	End time of the implementation of the i^{th} chromosome	
T_i^-	Start time of the implementation of the i^{th} chromosome	

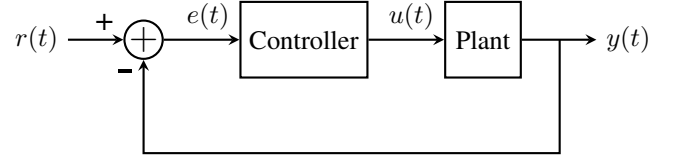


Fig. 1. Closed loop system with negative feedback

u Control signal
 y Output of the plant

III. PID CONTROLLER

Fig. 1 shows a typical closed loop system with negative feedback. The plant and controller blocks represent its dynamics. $r(t)$, $u(t)$, $y(t)$ and $e(t) = r(t) - y(t)$ are reference, control, output and error signals respectively. The objective of a controller is to achieve

$$\lim_{t \rightarrow \infty} e(t) = 0.$$

A PID controller is defined as

$$u(t) = K_P e(t) + K_I \int e(t) dt + K_D \frac{d}{dt}(e(t)), \quad (1)$$

where K_P , K_I , and K_D are nonnegative constant values. These parameters are tuned to achieve stability.

Usually, sensors are used to measure the output of the plant $y(t)$ which are prone to noise. The noise power is amplified when differentiation is performed [3]. Therefore instead of using the PID controller as it is the third term associated with the derivative is modified as a filter. Modified version of PID controller in frequency domain is

$$u(s) = K_P e(s) + \frac{K_I e(s)}{s} + \frac{K_D e(s)s}{s + N}, \quad (2)$$

where s is the Laplace transform's complex variable and $N \gg 1$ is a constant.

Tuning a PID controller is a task of selecting optimal values for the four parameters K_P , K_I , K_D , and N for which the plant is stable with required performance. The required performance can be ensuring settling time, providing damping or improving raise time, etc. In practice, N is kept constant.

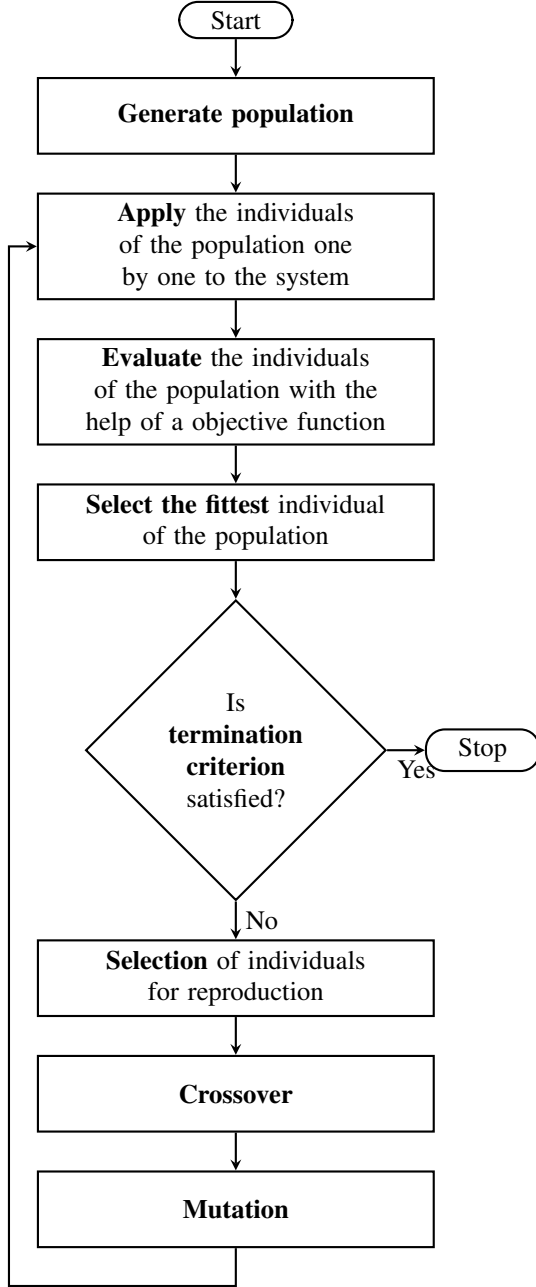


Fig. 2. Conventional genetic algorithm based PID tuning algorithm

However, in this article N is considered as one of the tunable parameters.

The next section presents a typical method for tuning these parameters with the help of a genetic algorithm.

IV. CONVENTIONAL GENETIC ALGORITHM BASED PID TUNING

Fig. 2 shows a standard genetic algorithm based PID tuning algorithm [4], [5]. In this context, the word population is used to represent a fixed set of individuals or chromosomes; and an individual or a chromosome represents an array of values for the parameters to be tuned. In this case the parameters are

K_P , K_I , K_D and N . Each chromosome is a candidate for the optimal solution. The genetic algorithm mimics the evolution process to generate optimal values for the parameters. It starts with a generation of a random population; it undergoes reproduction and mutation process to produce a next generation population with values much closer to the optimal values. This algorithm includes the following operations.

Generate population: A predetermined number of random chromosomes, $C \in \mathbb{R}^{n \times 4}$, are created to group them as a first generation population. The constant n represents the population size. The random numbers are scaled to match the upper and lower limits of the parameters. In this case, the random numbers must be greater than zero.

Apply: The values for the tunable parameters of the closed-loop system, K_P , K_I , K_D and N are replaced by each of the present generation's chromosomes, $C_i \mathbb{R}^4$, one after another for the time-period $[T_i^-, T_i^+)$. This process repeats n times. Here C_i , $i \in \{1, 2, 3, \dots, n\}$ represents the i^{th} chromosome of the population, T_i^- and T_i^+ start and end time of the implementation of the i^{th} chromosome. Since one chromosome is applied after another $T_i^+ = T_{i+1}^-$.

Evaluate: The objective function or the cost function determines how suitable is a chromosome for the closed-loop system. The scale ranges from 0 to ∞ , where zero associated with the optimal value. In the literature, a variety of cost functions such as mean square error (MSE), integration of time multiplied by absolute error (ITAE), integral of the absolute magnitude of the error (IAE), integral of the squared error (ISE) and root mean square error (RMSE) are utilized [6], [4]. In this article, the objective function uses RMSE defined as

$$RMSE = \frac{1}{T} \int_{T_i^-}^{T_i^+} \sqrt{(e(t))^2} dt,$$

where $T = T_i^+ - T_i^-$ is the time interval for which the i^{th} chromosome is applied.

Select the fittest: The fittest chromosome in the present generation of the population is the one which produced minimum value for the cost function.

Termination criterion: In the practical implementation, the objective function never reaches to a zero value. Hence, the chromosome which offers the objective function below a certain predetermined value is considered optimal. If all of the chromosomes fail to provide optimal results, a new generation is created and evaluated.

Selection: This is the first stage of the reproduction process. In this juncture, some chromosomes from the present population are selected to involve in the reproduction process. Roulette wheel selection, stochastic universal sampling, normalized geometric selection, and tournament selection are the most commonly used selection process methods [7], [4]. This article uses roulette wheel selection method.

Crossover: This is another stage of the reproduction process, where the selected chromosomes undergo the crossing.

It is carried out by 1) splitting the selected or parent chromosomes array at a random point and then 2.) the children chromosomes form by joining one parent's front end of separated array with the rear end of another.

Mutation: In this stage of the reproduction process, at randomly locations the values are changed. The mutation rate determines the number of mutations. Usually, it is less than 1%.

Since this algorithm uses random values for many of its operations, the optimum values may vary every time we restart the process. The next section presents the problems associated with the implementation of this algorithm.

V. PROBLEMS IN THE IMPLEMENTATION

A. *Previous chromosome's error*

B. *Tuning may be required in the future*

C. *Certain chromosomes may cause instabilities*

VI. PROPOSED ALGORITHM

VII. SIMULATION RESULTS

VIII. EXPERIMENTAL RESULTS

IX. CONCLUSIONS

REFERENCES

- [1] K. H. Ang, G. Chong, and Y. Li, "Pid control system analysis, design, and technology," *IEEE transactions on control systems technology*, vol. 13, no. 4, pp. 559–576, 2005.
- [2] A. O'Dwyer, "Pid compensation of time delayed processes 1998-2002: a survey," in *Proceedings of the American Control Conference*. Dublin Institute of Technology, 2003, pp. 1494–1499.
- [3] Mathworks. (2017, jul) Take derivatives of a signal. [Online]. Available: <http://www.mathworks.com/help/signal/ug/take-derivatives-of-a-signal.html>
- [4] J.-S. Kim, J.-H. Kim, J.-M. Park, S.-M. Park, W.-Y. Choe, and H. Heo, "Auto tuning pid controller based on improved genetic algorithm for reverse osmosis plant," *World Academy of Science, Engineering and Technology*, vol. 47, no. 2, pp. 384–389, 2008.
- [5] A. Zemmit, S. Messalti, and A. Harrag, "A new improved DTC of doubly fed induction machine using GA-based PI controller," *Ain Shams Engineering Journal*, 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S2090447917300205>
- [6] 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, June 2004.
- [7] M. Gen and R. Cheng, *Genetic algorithms and engineering optimization*. John Wiley & Sons, 2000, vol. 7.