Online Tuning of PID Controller for Time Variant Systems using Genetic Algorithm

N. Selvaganesan[1] **Prabhu Jude Rajendran[2]** **S.Renganathan[3]**

[1] *Department of Instrumentation Engineering, Madras Institute of Technology, Anna University, Chennai 600 044.*
Tel: +91-94440-30231, Fax: +91-44-22232403, E-mail: n_selvag@yahoo.com

[2] *Infosys Technologies Limited, 138,Old Mahabalipuram Road, Sholinganallur, Chennai 600 119.*
Tel: +91-98414-42011, Fax: +91-44-24500390, E-mail: prabhu_jude@yahoo.com

[3] *Bharath Institute of Higher Education and Research, Chennai 600 077.*
Tel: +91-44-22303623, Fax: +91-44-22293886, E-mail: srenga@vsnl.net

Abstract

PID controllers are one of the most widely used controllers in many industrial sectors. However, in order to derive a good performance from the PID controllers, it is necessary to tune the controller parameters Kp, Ti, Td to appropriate values. Researchers, worldwide, have come out with numerous methods of tuning the parameters to derive an optimum performance from PID controllers. These controller settings are dependent on the system parameters, and hence the controller requires to be re-tuned whenever there is a change in the system parameters. This paper attempts to develop software to tune the PID controller parameters for processes with varying system parameters using Genetic Algorithm (GA). The program is designed to get the varying parameters of the system (assumed to be obtained using RLS or other well established system identification algorithm) during the runtime. This paper also analyses the performance of GA based tuning for adaptive controllers as compared with the Zeigler Nichols method of tuning for similar time varying systems.

Keywords:

Genetic Algorithm, Zeigler-Nichols Tuning, Adaptive Control, Time Variant Systems.

Introduction

In Process Control, more than 95% of the control loops are of PID type, most loops are actually PI control. The popularity of the PID controllers can be attributed partly to their robust nature and partly to the simplicity and flexibility in their design. In order to derive a good performance from the PID controllers, it is necessary to tune the controller parameters K_p (gain), T_i (integration time), T_d (derivation time) to appropriate values. The process of selecting the PID parameters to meet given performance specifications is termed as Optimal Tuning. Researchers, worldwide, have come out with numerous methods of Optimal Tuning to derive an optimum performance from PID controllers.

The controller settings are dependent on the properties of the system. Hence, a PID controller in the control loop of a time-variant system requires continuous tuning in order to perform at the optimum level.

Approach and Methods

Evolutionary Computation

Evolutionary computation is a field that is concerned with using the methodologies and optimization procedures that are derived by abstracting from the theory of biological evolution, to solve problems. Genetic algorithm (GA) is one of the very popular methods of Evolutionary Computation. GA is a global search method that performs a parallel, non-comprehensive search for the global maximum of a problem. It operates on a population of potential solutions applying the survival of the fittest to produce successively better approximations to a solution.

PID Controller

The transfer function of PID controller is given as

$$G_c\,(s) = K_p\,(1+i/T_i\,s+ T_d\,s) \quad (1)$$

If e (t) is the input to the PID controller, the output, u (t), from the controller is given as

$$U\,(t) = K_p\,[e(t) + 1/T_i \int e(t)\,dt + T_d\,de/dt] \quad (2)$$

The transfer function of the PID controller is usually represented in more simple terms as

$$G_c(s) = K_p + K_i/s + K_d\,s \quad (3)$$

where

K_p = Proportional gain

K_i = Integral gain

K_d = derivative gain

Controllers for Time-Variant Systems

Figure 1 shows the block diagram to represent the methodology that is generally used for tuning the controllers designed for systems with time varying parameters. It is required that the system parameters are identified before the controller can be tuned. There are many methods available in the literature that can take care of the identification of system parameters. RLS (Recursive Least Squares) is one of the very famous system identification methods that can be used for this purpose. Since the scope of the paper is restricted to developing an online tuner for an adaptive PID controller, the methodologies and techniques of system identification are not elaborated.

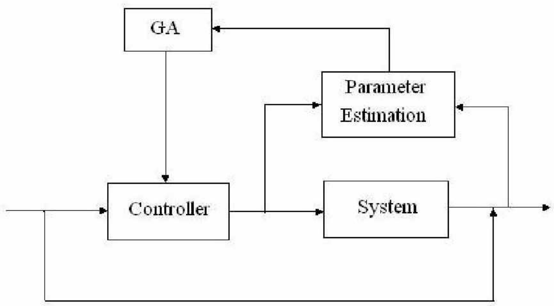

Figure 1 - Block diagram of the control system using Genetic Algorithm tuner

Optimization Problem Statement

A constrained optimization or a mathematical programming problem is generally stated as

Find

$$X = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ : : \\ X_n \end{bmatrix}$$

which minimizes the function F(x) subject to the constraints

$G_i(x) <= L$; $i=1,2,3\ldots m$.

where X is an n x 1 vector called the design factor, F(x) is called the objective function and Gi (x) is the inequality constraint.

The optimization problem for tuning the PID controller can also be considered as a constrained minimization problem. The design vector in this case will be a 3 x 1 vector whose elements are controller parameters – K_p, T_i, T_d. The inequality constraints for this case can be given as

$K_1 <= K_p <= K_2$

$T_1 <= T_i <= T_2$

$M_1 <= T_d <= M_2$

The selection of the objective function F(x) for this problem is discussed in a later part of the paper.

GA Parameters and Schemes

Encoding

In order to use GA to solve the optimization problem, the variables K_p, T_i, T_d are first coded as strings. The binary encoding method, where the control variables are coded as finite-length string of 0s and 1s, is followed in the tuner program. The user can change the length of the string during the initiation of the tuner. Any length string can be formed to represent a point in the search space according to a fixed mapping rule

$X_i = X_i(L) + \{[X_i(U) – X_i(L)]/ 2^{Li-1}$

where X_i (L) and X_i (U) are the lower and upper limits for the control variables and Li is the length of the string.

The decoded value of binary sub strings S_i is calculated as $\sum (2^i S_i)$ summed between I=0 to L-1.

Objective Function

The most important criterion in the GA is the framing of objective function. Generally, methods are patterned to optimize some design criteria that characterize the properties of the closed loop system. Here, the objective function is selected as the Integral Square Error (ISE), which is given as

$ISE= J =[\sum (CV – SP)]^2$.

Hence, the problem of finding the optimum parameters can be deduced as the problem of finding the parameters of the controller in order to minimize the ISE (Integral Square Error).

Fitness Function

Since GA mimics 'the survival of the fittest' principle to make a search process, it is naturally suitable to use GA for solving maximization problems. Hence minimization problems are usually converted to maximization problems before GA is used to solve them. The reciprocal of the ISE is used as the fitness function of the problem.

Reproduction

The initial populations of the controller parameters can be generated by applying well know tuning rules such as Cohen-Coon [2], Haalman [3]. In GA, the creation of new solution-candidates mimics the reproduction process in nature. The individuals in the population having a higher fitness functions have a higher probability of participating in the reproduction process to form new individuals of the next generation. There are various ways of selecting parents as suggested by Jukka Lieslehato [4]. In this case, Roulette-Wheel selection method is used to determine the individuals participating in the reproduction.

Genetic Operators

Generally, the Mutation and Recombination operations are implemented in the algorithm to simulate the biological reproduction process. Hence these two phenomena are rightly called genetic operators. In this case single-point crossover recombination and binary mutation methodologies are used to implement the Genetic operators in the tuner program.

Online Tuner Design

Figure 2 shows the flowchart of the genetic algorithm implemented in the tuner program. In order to develop an online tuner, the program is designed to get the varying parameters of the system (assumed to be obtained using RLS or other well established system identification algorithm) during the runtime. The various GA parameters, the limit of convergence for the controller parameters, the sample space for the controller and K_p, Ti and T_d are entered when the tuner is initiated for the first time. The algorithm runs through a number of generations, which is defined by the user. The algorithm will get automatically initiated when a change in the system parameter is

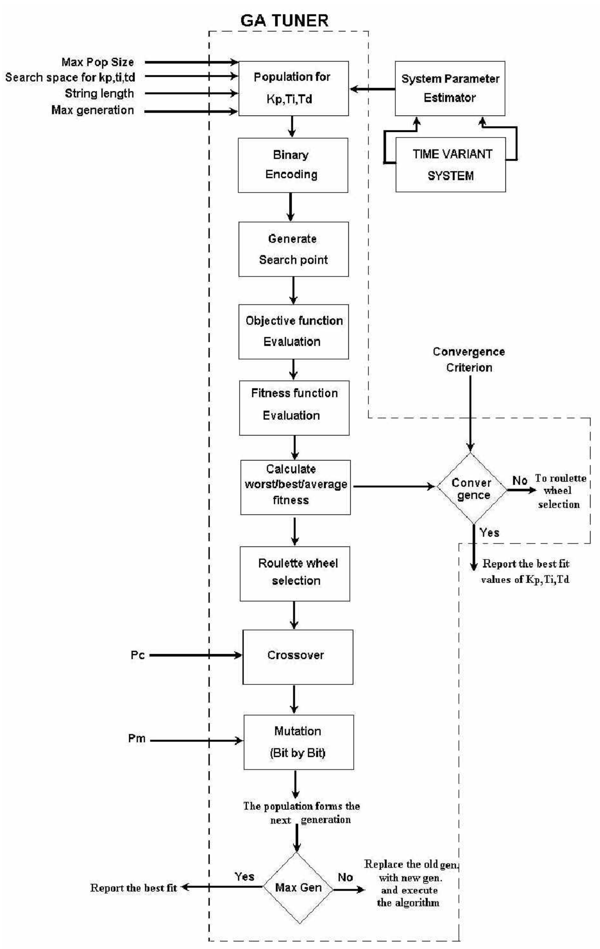

Figure 2 - Flow chart to show the logical sequence followed in the SQA (Simple Genetic Algorithm) for tuning the adaptive PID controller

detected by the online-tuner. Following the suggestion of De Jong [5], the under mentioned parameters are adopted as the default values for the tuner: $P_{mutation}$=0.0333, P_{cross}=0.6 and Population size =30.

Simulation and Results

The simulations were carried out for typical third order systems with the GA parameters fixed according to De Jong's suggestion. Typical test cases were used to evaluate, analyze and compare the performance of the GA based online tuner with the non-evolutionary controller models. The process in the control loop is assumed to be a third order system with three poles at s= - 1(No zeroes and Process Gain=1).

Step change in the set – point of the process loop with time-invariant system

Figure 3 shows the comparison for the step responses of the control loops of the GA based tuner and the conventional Z-N tuner. As indicated by the simulation, the GA based tuner exhibited much better performance in terms of reduced ISE, reduced settling time, reduced peak overshoot and better rise time. Since the algorithm uses ISE as the objective function, it is very important to compare this ISEs of the GA based control loop with that that the Z-N based control loop (Table 1). From the simulations, it is clear that the search algorithm was able to find out a more optimum set of PID parameters.

Table 1 – Comparison of Performance Indices

Performance Index	Z-N tuned PID Controller	GA tuned PID Controller
ISE	3.62	3.02
IAE	5.158	4.86

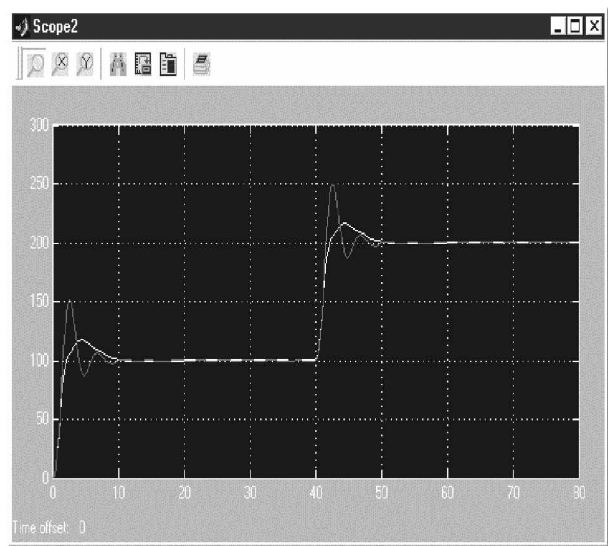

Figure 3 - Multi-Step Response for PID controllers tuned using GA and ZN (The process parameters are assumed to be time-invariant).

Step change in the set – point of the process loop with time-variant system

Since the PID tuning parameters are expected to give an optimum performance for a fixed set of system parameters, a non-adaptive tuner cannot derive an optimum performance from the controller if the system parameters change with time. This fact is well exhibited in the simulation results.

The test case was taken for a system whose poles tend to move towards the origin (hence making the system less stable). The amplitude of oscillations clearly increases in the case of non-adaptive ZN tuner as the pole moves away from the stable region (Figure 4). Clearly, the GA based tuner (being an adaptive tuner) performed much better as compared to the ZN based non-adaptive tuner. This clearly shows that the GA tuner is relatively more robust to changes in the system parameters. In order to evaluate the impact of the change in the process design on the ZN controller, the simulation was extended to compare the performance of a non- adaptive ZN tuner with an adaptive Z-N tuner. The GA tuner proved to be very robust as compared to the ZN tuner, which showed increased settling time, and peak overshoots as the poles move further towards the origin (Figure 5).

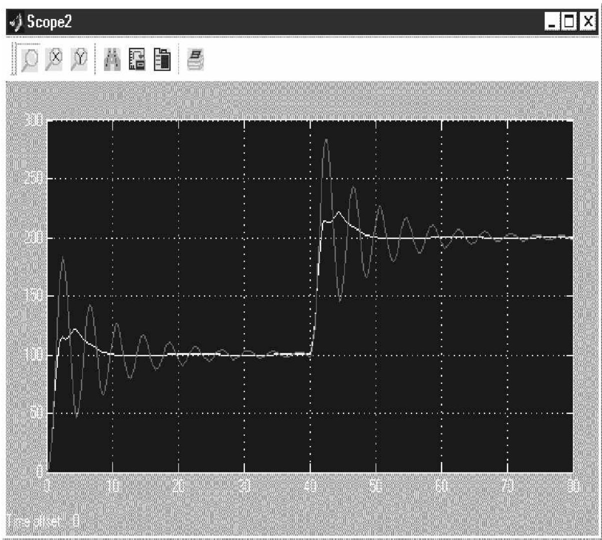

Figure 4 - Multi-Step Response for PID controllers tuned using adaptive GA and non-adaptive ZN tuners. (It is assumed that one of the poles is shifted from s=- 1 to s= - 0.5 after the initial tuning of the controller)

Conclusion

The GA based adaptive tuner presented in this paper offers several advantages. Many tuning methods require that the process model be of a certain type, for example a first order plus dead time model (FOPDT). Hence, in order to use these methods to tune the PID controller, the higher order systems have to be reduced using approximations. GA obviates this requirement of reduction of the process model and hence more accurate (and devoid of any approximations). Evolutionary algorithms are usually criticized for being computationally heavy. Since the PID tuning is a small-scale problem, the computational complexity is not really a problem here [4].

A better convergence to the optimal set of parameters can be obtained by fixing the search range of the parameters based upon an approximate tuning model. With the ever-increasing processor speeds, the computational time is not

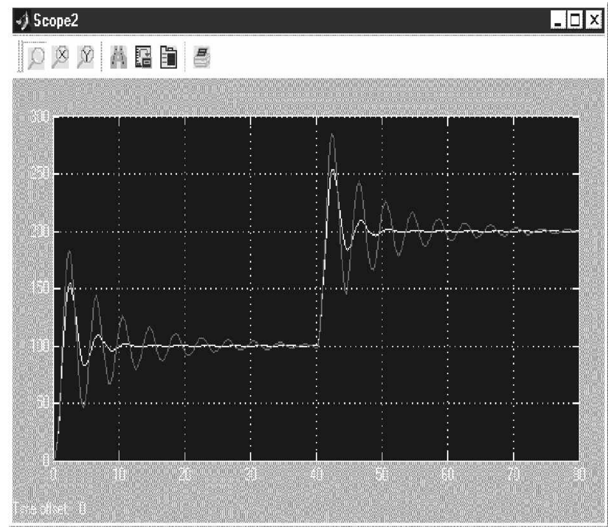

Figure 5 - Multi-Step Response for PID controllers tuned using a ZN non-adaptive tuner. (The initial response is taken after tuning the PID controller for this system .The second response is taken after a system pole shifted from s = -1 to s = - 0.5).

going to be a problem. It will take just a few seconds to calculate the optimum parameters using the normally available microprocessors.

Hence the GA based online adaptive tuner proposed in this paper provides a reliable, quicker and robust way of tuning the PID controllers involved in the controller loops of time-varying systems.

References

[1] Goldberg, D. E. eds. 1989. *Genetic Algorithm in Search, Optimization and Machine Learning*. Addison Wesley.

[2] Cohen, G.H & Coon, G.A. 1953. Theoretical Consideration of Retarded Control, In Proceedings of ASME.

[3] Haalman, A. 1965. Adjusting Controllers for Dead time Processes. *Control Engineering*.

[4] Jukka Liesleheto. 2001. PID Controller Tuning Using Evolutionary Programming, In Proceedings of the American Control Conference.

[5] Davis L., & Ritter F. 1987. Schedule Optimization with Probabilistic Search. In Proceedings of the 3rd IEEE Conference on Artificial Intelligence Applications.

[6] Katsuhiko Ogata. eds. 1995. *Modern Control Engineering.* Prentice Hill Publications.

[7] Mitsuo Gen, Runwei Cheng. eds. 1997. *Genetic Algorithms and Engineering Design.* John Willey and Sons.