Design of PID Controllers via Genetic Algorithm for Benchmark Systems

C. Thammarat, P. Sukserm, and D. Puangdownreong
Department of Electrical Engineering, Faculty of Engineering,
South-East Asia University, Bangkok, 10160, THAILAND
Email: dp@sau.ac.th

Abstract—This paper presents an intelligent method for designing proportional-integral-derivative (PID) controllers using the genetic algorithm (GA), one of the most efficient AI search techniques. The goal is considered heuristic approach and more flexible design method. With its robustness and effectiveness, GA can well perform finding optimum parameters of a PID controller in order to minimize some particular objectives. To evaluate its performance, the proposed method is applied to several benchmark systems for PID control presented by Åström, et al. [1]. Based on the result obtained from a series of simulations, the proposed GA-based PID design method is found to yield satisfactory performance for various designing PID controllers.

I. INTRODUCTION

The proportional–integral–derivative (PID) control schemes in industrial applications were first introduced in 1939 [2]. Due to ease of use and simple realization, PID controllers have been widely used in most of process control systems over decades. To obtain appropriate controller’s parameters, one can proceed with available analytical design methods or tuning rules. Mostly the analytical design methods assume known plant models [3], [4], [5], while the tuning rules assume known process responses [6], [7], and known plant models [8]. Every analytical design method and every tuning rule, however, have some particular conditions concerning to the plant models, such as dead time or transport lag, fast and slow poles, real and complex conjugated zeros and poles, as well as unstable poles, and to the process responses, such as oscillation and shapes. These conditions make the design methods and tuning rules non-general. In 2000, Åström, et al. [1] proposed a collection of systems which are suitable for testing PID controllers. These collected systems are considered benchmark problems for evaluating post-proposed PID controller design method.

Over two decades, artificial intelligent (AI) techniques have been employed for the controller design in industrial control applications, for example: designing of an adaptive PID controller by genetic algorithm (GA) [9], self-tuning PID controller by GA [10], and finite-precision PID controller by GA [11]. GA is efficient to find the global minimum of the search space. In this paper, GA is briefly reviewed and then applied to design PID controllers for benchmark systems proposed by Åström, et al. [1]. This paper consists of five sections. The problem formulation of PID design is described in Section 2. Section 3 gives GA algorithm in brief. The GA-based design of the PID controllers is illustrated in Section 4, while Section 5 provides the conclusions.

II. PROBLEM FORMULATION

In conventional control loop represented by the block diagram in Figure 1, the PID controller receives the error signal, \( E(s) \), and generates the control signal, \( U(s) \), to regulate the output response, \( C(s) \), referred to the input, \( R(s) \), where \( D(s) \) is disturbance signal, \( G_p(s) \) and \( G_c(s) \) are the plant and the controller transfer functions, respectively.

![Figure 1. Conventional control loop.](image)

The transfer function of the PID controller is stated in (1), where \( K_p \) is proportional gain, \( K_i \) is integral gain, and \( K_d \) is derivative gain. So, the design problem is simplified to determine the parameters \( K_p, K_i, \) and \( K_d \). The closed loop transfer function with PID controller is given in (2).

\[
G_c(s) = K_p + \frac{K_i}{s} + K_ds
\]

\[
C(s) = \frac{\left(K_p + \frac{K_i}{s} + K_ds\right)G_p(s)}{1 + \left(K_p + \frac{K_i}{s} + K_ds\right)G_p(s)}
\]

\[
J = \sum_{t=0}^{T} |r(t) - c(t)|
\]

The use of AI search techniques to design the PID controller can be represented by the block diagram in Figure 2. The cost function, \( J \), sum absolute error (SAE) between \( R(s) \) and \( C(s) \) as stated in (3), is fed back to the AI tuning block. \( J \) is minimized to find the optimum PID controller’s parameters that give a satisfactory response. To minimize \( J \), GA method is applied and explained in the next section.
III. GENETIC ALGORITHM (GA)

Genetic algorithm (GA) is briefly reviewed in this section. GA is a searching method based on two natural processes, i.e. selection and genetic operation. The search process of GA is similar to the nature evolution of biological creatures in which successive generations of organisms are given birth and raised until they themselves are able to breed. In general, the basic goal of GA is to optimize functions, i.e. fitness and objective functions. GA-based approach differs from other conventional optimization techniques in several ways [12],[13]. Firstly GA works with a coding of the parameter set rather than the parameters themselves. Secondly GA searches from a population of points rather than a single point. Thirdly, GA uses the objective function information, not other auxiliary knowledge. Finally GA uses probabilistic transition rules, not deterministic rules. With these properties, GA is made robust, powerful, and problem-dependent. Algorithms of a simple GA are shown step-by-step as follows.

1. Random initial populations or chromosomes and set them as a search space.
2. Evaluate the fitness value of each chromosome via the objective function.
3. Select some chromosomes giving better fitness value to be parents.
4. Reproduce new generation (offspring) by genetic operations, i.e. crossover and mutation.
5. Compute the fitness value of each new chromosome via the objective function.
6. Replace old chromosomes by new ones and go back to (2).

When the termination criteria are met, GA is stopped. The optimum solution found is the best chromosome in a search space.

IV. GA-BASED DESIGN OF PID CONTROLLER

The GA-based design of the PID controllers is illustrated in this section. Referring to Figure 2, the AI tuning block can be well represented by GA. Then, the parameter tuning process is repeatedly performed in order to minimize J (SAE) stated in (3) until the termination criteria are satisfied. In this work, maximum generation (Max-Gen) of the search process is set as the termination criteria. To evaluate the performance of the proposed design method, the GA-based design of the PID controllers is applied to four benchmark systems suggested by Åström, et al. [1]. Such the systems are difficult to design their PID controllers by conventional design methods or tuning rules.

The desired specification is given in (4), where $t_s$ is rise time, P.O. is percent overshoot, $t_{r\text{-max}}$ is maximum rise time allowance, $P.O_{\text{max}}$ is maximum percent overshoot allowance, $t_{s\text{-max}}$ is maximum settling time allowance, and $E_{ss\text{-max}}$ is maximum steady state error allowance. They are set as inequality constraints of the optimization. GA is used to obtain $K_p$, $K_i$, and $K_d$ of the PID controller. The optimization framework can be defined as (4).

\[
\text{Minimize} \quad J \\
\text{Subject to} \quad t_s \leq t_{r\text{-max}}, P.O \leq P.O_{\text{max}}, \quad t_s \leq t_{s\text{-max}}, E_{ss} \leq E_{ss\text{-max}}
\]

In GA applied in this work, Max-Gen = 500 is defined as the termination criteria. 40 chromosomes are randomized of which the intervals are heuristically assumed. The chromosomes are selected by the ranking fashion. Crossover and mutation mechanisms are activated at single point with the probability level of 0.8. Replacement could be done on only one chromosome giving better fitness value in each generation.

A. Fourth Order System

The fourth order system as shown in (5) is the first one used to perform the tests. Such the system has four poles whose spacing is determined by parameter $\alpha$. In this work, $\alpha$ is set as 0.5. The desired specification of this system are given by $t_{r\text{-max}}=1.$sec, $P.O_{\text{max}}=10.0\%$, $t_{s\text{-max}}=2.$sec, and $E_{ss\text{-max}}=0.00$. The parameter ranges to form the search space were given by $K_p=K_i=K_d=0[10]$. After the search process stopped, the PID controller’s parameters are successfully obtained. From the search result, the PID controller listed in Table I yields satisfactorily response as shown in Table I. Additionally, the step responses of the fourth order system without and with PID controller are shown in Figure 3, while Figure 4 represents the convergence of $J$.

\[
G_p(s) = \frac{1}{(1+s)(1+\alpha s)(1+\alpha^2 s)(1+\alpha^3 s)}
\]

B. Fast and Slow Modes

The system has a fast time constant at $T=1$sec with a moderate gain and a slow time constant at $T=20$sec with a large gain as shown in (6). The desired specification of this system is given by $t_{r\text{-max}}=0.25$sec, $P.O_{\text{max}}=10.0\%$, $t_{s\text{-max}}=1.$sec, and $E_{ss\text{-max}}=0.00$. The parameter ranges to form the search space were given by $K_p=K_i=K_d=0[10]$. After the search process stopped, the PID controller’s parameters are successfully obtained as listed in Table I. The PID controller yields satisfactorily response as also shown in Table I. The step responses of this system without

ECTI-CON 2007
The 2007 ECTi International Conference

222
and with PID controller are shown in Figure 5, and the convergence of $J$ is represented in Figure 6.

$$G_{p2}(s) = \frac{100}{(s+10)^2} \left( \frac{1}{s+1} + \frac{0.5}{s+0.05} \right)$$  \hspace{1cm} (6)

C. Oscillatory System

The oscillatory system is expressed in (7), where $\omega_0 = 1$ and $\zeta = 0.1$. The desired specification of this system is given by $t_r_{\text{max}} = 2.5\text{sec}$, $P.O._{\text{max}} = 25.0\%$, $t_s_{\text{max}} = 10.0\text{sec}$, and $E_{ss_{\text{max}}} = 0.00$. The parameter ranges to form the search space were given by $K_p = [0, 5]$, $K_i = [0, 5]$ and $K_d = [0, 5]$. After the search process stopped, the PID controller’s parameters are successfully obtained as listed in Table I. The PID controller yields satisfactorily response as also shown in Table I. The step responses of this system without and with PID controller are shown in Figure 7, and the convergence of $J$ is represented in Figure 8.

$$G_{p3}(s) = \frac{\omega_0^2}{(s+1)(s^2 + 2\zeta\omega_0 s + \omega_0^2)}$$  \hspace{1cm} (7)

D. Unstable System

The unstable system as stated in (8) is a simple model of an inverted pendulum. The desired specification of this system is given by $t_r_{\text{max}} = 0.1\text{sec}$, $P.O._{\text{max}} = 15.0\%$, $t_s_{\text{max}} = 0.5\text{sec}$, and $E_{ss_{\text{max}}} = 0.00$. The parameter ranges to form the search space were given by $K_p = [400, 500]$ and $K_i = [0, 50]$. After the search process stopped, the PID controller’s parameters are successfully obtained as listed in Table I. The PID controller yields satisfactorily response as also shown in Table I. The step responses of this system without and with PID controller are shown in Figure 9 and, the convergence of $J$ is represented in Figure 10.

$$G_{p4}(s) = \frac{1}{s^2 - 1}$$  \hspace{1cm} (8)

V. CONCLUSIONS

The intelligent method for designing PID controllers using GA is proposed in this paper. The proposed method is considered the heuristic approach. It is more flexible than the conventional design methods and tuning rules. The presented method has been evaluated via four benchmark systems. As the results, the appropriated parameters of PID controllers have been successfully obtained, and achieved high satisfactorily responses. The usefulness and flexibility of the proposed intelligent design approach have been confirmed.
REFERENCES


<p>| Table I | List of PID Controllers and Step Responses Obtained from GA |</p>
<table>
<thead>
<tr>
<th>Plant PID controllers</th>
<th>System responses</th>
<th>Cost function, J</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G_p(s)$</td>
<td>$K_p$</td>
<td>$K_i$</td>
</tr>
<tr>
<td>$G_p(s)$</td>
<td>3.78</td>
<td>2.16</td>
</tr>
<tr>
<td>$G_p(s)$</td>
<td>8.33</td>
<td>1.50</td>
</tr>
<tr>
<td>$G_p(s)$</td>
<td>0.24</td>
<td>1.21</td>
</tr>
<tr>
<td>$G_p(s)$</td>
<td>449.86</td>
<td>45.75</td>
</tr>
</tbody>
</table>