Proceedings of the Korean Nuclear Society Autumn Meeting
Seoul, Korea, October 1995

Application of Genetic Algorithm to Control Design

Yoon Joon Lee, Kyung Ho Cho
Cheju National University

Abstract

A classical PID controller is designed by applying the GA (Genetic Algorithm) which searches the optimal parameters through three major operators of reproduction, crossover and mutation under the given constraints. The GA could minimize the designer's interference and the whole design process could easily be automated. In contrast with other traditional PID design methods which allows for the system output responses only, the design with the GA can take account of the magnitude or the rate of change of control input together with the output responses, which reflects the more realistic situations. Compared with other PIDs designed by the traditional methods such as Ziegler and analytic, the PID by the GA shows the superior response characteristics to those of others with the least control input energy.

1. Introduction

A wide variety of engineering works have the problems of trading off the conflicting properties and the works are directed to optimizing these properties under the given constraints. Particularly for the case of control design, almost all the design methods have the optimization problems even either implicit or explicit they are. In general, the trading off procedures are liable to designer's experiences and knowledges and even to his intuition. It has its own advantages in that it might permit more flexibility in design. On the other hand, however, this subjective approach might present many limitations.

All this motivates the new optimization algorithms, one of which is the genetic algorithm (GA). The GA emulates the biological evolutionary theories to solve the optimization problems. By using the three major operators of reproduction, crossover and mutation which are analogous to the biological process in genetics, it searches the optimal design parameters of the problem. Since the GA is the direct searching method independent of the coupling between the parameters, it provides more flexibilities, particularly for the strongly coupled or stiff systems. Also it is a smart algorithm for the multi modal problems because of its capability of concurrent multi points search .

To demonstrate the effectiveness of the GA and its applicability to the controller design, a classical PID controller is designed using the GA. Although the PID design seems to be too simple and elementary, particularly under the present circumstances of modern control, the actual design is somewhat involved, and it includes a wide spectrum of control theories. Further the final implementations of modern control are mainly of PID form. In this study, PIDs for the given feedback system are designed by various methods, including the GA and their response characteristics are compared with each other.

2. Genetic Algorithm

First proposed by J. Holland, the GA has been proved useful in a variety of search and optimization fields over the last years[1-3]. This algorithm is based on the survival-of-the-fitness principle in nature and is an attractive class of computational model that mimics the natural evolution to solve optimization problems.

The GA emulates the biological evolutionary theories to search optimal solution of the problems. It comprises a set of individual elements called a population and a set of biologically inspired operators. According to evolutionary theories, only the most suited elements in the population are likely to survive and generate its own offsprings, thus transmit their biological heredity to next generations. In computing terms, the GA maps a problem onto a set of (typically binary) strings, and each string represents a potential solution. The GA is then manipulates the most promising strings in its search for improved solutions. In the GA, each solution is associated with a *fitness* value that reflects how good it is, compared with other solutions in the population. The higher the fitness value of an individual is, the higher its chances of survival and reproduction become, and the larger its representation in the subsequent generation is. A typical working procedure of a simple GA is as follow :

- Initialize the parameters of the GA;
- Randomly generate the initial population;
- **for** *generation* =: 1 **to** *max_generation*
  - Clear the *new_population*;
  - Compute the *fitness* of each individual in the *old_population*;
  - Copy the *individual* of the highest fitness to the *solution_vector*;
  - **while** the *no_of_individual* < *population_size* **do**

    Select two parents from the *old_population* based on their fitness;
    Perform the crossover of the parents to produce two offsprings;
    Mutate each offspring based on ;
    Place the offsprings to *new_population*;
  - **endwhile**
  - Replace the *old_population* with the *new_population*;
- **endfor**
- Print out the *solution_vector* as the final solution

The above working procedure can be explained further as follows :

(1) An appropriate string representation should be defined to represent the one-to-one mapping of the design parameters.

(2) The population size and maximum number of generations should be specified. The proper values of the crossover and mutation probability are also assigned to. Then, an initial population is generated randomly.

(3) Evaluate the fitness (or objective function) value of each individual in the current generation.

(4) Apply the genetic operators - reproduction, crossover and mutation - on the old generation to

generate the new population for the next generation.
(5) Repeat steps (3) and (4) until the maximum number of generation is reached.

To perform the GA procedure described above effectively, three GA operators of the reproduction, crossover and mutation are used in general. Based on each individual's fitness, a selection mechanism chooses 'mates' for the genetic manipulation process. The selection policy is ultimately responsible for assuring the survival of the best fitted individuals.

The combined evaluation and selection process is called reproduction. The crossover operator takes two chromosomes and swaps part of their genetic information to produce new chromosomes. This operator together with the selection mechanism is the major search mechanism which locates probabilistically better solutions by exchanging useful information among the visited solutions. Mutation is implemented by occasionally altering a random bit in a string: changing 0 to 1 or vice versa. The mutation operator may introduce new genetic properties to the population helping the search algorithm escape from local traps. More detail information about the GA operators can be found in the related works[1-3].

3. PID Design with GA

The purpose of any control system design boils down to two subjects. They are the configuration of the overall system structure and the design of controllers. In most cases, the specifications of the control design conflicts with each other. If one tries to increase the system speed, then he loses the system stability, or vice versa. This indicates that the design of the controller is to determine the optimal parameters which can yield the best responses under the crashing circumstances. However there is no deterministic metrics to evaluate what the best response is. It depends on the designer's experiences and practical knowledge. And the best controller to those who designs it might be just a plausible controller to others. It should also be noted that almost all the specifications are focused on the output responses only. If the system is stable and the unbounded inputs are permitted, it would be possible to make the ideal system responses. But in reality, there are always limitations on the control input energy . Too large input may cause the mechanical or electrical hazards in addition to the system delay caused by the saturation.

All these problems could be minimized by the GA. The designer does not need to care about the detail procedures. The GA gives out an optimal set of PID parameters. The only thing the designer has to do is to set up a cost function, or a fitness function. For the purpose of demonstration, the PID of the system described in Figure 1 is designed by the traditional Ziegler-Nichols method and by the analytical method as well. Then they are compared with the results of the GA.

1) Ziegler Nichols Method

Ziegler and Nichols developed two methods for the controller tuning. The first one is to choose the controller parameters by setting the decay ratio as 0.25. And the second method, which is used in this paper, is based on a simple stability analysis. In this method, the marginal gain which results in the limit cycle, or marginal stability, and its corresponding frequency are found first either by root locus or by Bode diagram. Then the proportional gain (K_p), integral gain (K_i) and differential gain (K_d) can be obtained from the simple relations[4]

For the model described in Fig. 1, the marginal gain and the limit cycle frequency are found to be 2.0265 and 1.0056, respectively, and the values of the controller described in Fig. 1 is K_p=1.2159, K_i=0.3892, and K_d=0.9497. Figure 2 shows the responses of output and control input of the the uncompensated systems and Fig. 3 describes the responses of the system with this Ziegler PID. The system compensated by the Ziegler PID shows a well damped response, but at the expense of the increased overshooting. The Zigler method uses no design specifications. It is based on that the designed controller provides good responses on the whole.

2) Analytical Method

The analytical method[4] is used when the specific closed loop behavior is to be considered. This method basically has the assumption that the responses of the closed loop system can be approximated to those of the typical second order system. Also it is necessary to specify the gain crossover frequency. For the second order system, the closed loop natural frequency is almost the same as the open loop gain crossover frequency when the feedforward gain is large. Therefore the open loop crossover frequency is frequently used as an initial design value. But the designed controller causes this frequency to be different from the original one, and some trial and errors using a simulation is necessary. The K_p is determined uniquely when the phase margin and the cross over frequency are specified, but K_i and K_d are not unique, which necessitates the designer's discretion. With the design specifications of 10% overshooting, and the settling time of 8 seconds, the crossover frequency is obtained as 1.089 rad/sec from the basic relationships of the second order system. By introducing the phase margin of 30 degrees, the K_i is determined as 0.2 after several trial and error simulations. Then K_p and K_d are found to be 1.9528 and 1.4147, respectively.

Figure 4 shows the system output and control input responses with the analytical PID so far designed. It should be stressed again that this is not the unique one. There can be numerous combinations of parameters depending on the selections of K_i, phase margin and crossover frequency. The figure shows that the overshooting is more than 40% which deviates greatly from the design specification. This is because the guide parameters are based on the second order system.

3) GA Method

Contrary to those method of Ziegler or analytical, the PID design by the GA is very simple. The only design specification is the definition of the cost function which calculates the fitness value. The cost function of below is used for the PID design by the GA.

$$J = \int_0^t \left(|y(\tau) - y_s| + w \cdot |u(\tau) - u_s| \right) d\tau, \quad \text{fitness} = \frac{1}{J}$$

The weighting variable, w, indicates the relative limitation, or relative penalty on the control input energy. The large value of w imposes more limitation on the input, and the input becomes small. If w is zero, for an extreme case, the control input becomes unbounded. The above cost function implies that the controller to be designed is such one that makes the sum of the output deviations from the steady state value and control energy be minimum. Of course another form of cost functions such as the ITAE, or the LQR quadratic function may be possible.

The weighting factor in the above equation can either be determined by the user as a fixed value, or adjusted by the GA itself. For the case of which the weighting factor is included, the variables to be optimized are K_p, K_i, K_d and w. The GA searches the optimal set of these parameters through generations until there is no more improvements of the fitness. The output and control input responses of the final system are described in Figure 5. The parameters selected by the GA are $K_p=1.8$, $K_i=0$, $K_d=1.5$ and w=0.8. The figure shows a reasonable overshooting, rapid settling and increased speed. The input energy, compared with that of the analytical method, shows a milder variation also. This demonstrates that the GA PID gives the better output response with the less input energy than the analytical PID or Ziegler PID. Figure 6 shows the results when the input weighting value of 7 is given externally. The results show that there is no overshooting, The speed is fast, and the input energy is mild maintaing non negative values.

More than anything else, the design procedure by the GA is very simple. It also provides more flexibility when it comes to actual applications. For instance, if the actual plant is susceptible mechanically or electrically to a large control input, a larger weight value may be used as shown in the above example. Further even the tachometer input, such as reactor control rod speed or acceleration of the actuator, can be easily taken into account by modifying the cost function.

4. Conclusions

The GA emulates the biological evolutionary theories to solve the optimization problems. With the three major operators of reproduction, crossover and mutation it searches the optimal solution of the problem. This GA is applied to the PID controller design, and the PID by GA is compared with those designed by traditional methods. The hidden responses of the input energy as well as the output responses can be controlled in the GA, while the traditional methods focus on the system output responses only. In addition, the output responses of the PID by GA is superior to those of traditional ones. The GA could minimize the designer's interference and the design process might be routinized. By defining the proper fitness function which describes the problem characteristics, the design could be directed to user intended solution.

References

1. D. E. Goldberg., *Genetic Algorithm in Search, Optimization and Machine Learning*, A/H, 1989
2. J. R. Koza, *Genetic Programming on the Programming of Computers by means of Natural Selection*, MIT Press, 1992
3. K. De Jong, "Learning with Genetic Algorithms : An Overview," *Machine Learning 3*, Kluwer Academic Publishers, 1988
4. C. L. Phillips, H. T. Nagle, *Digital Control System - Analysis and Design*, P/H, 1990

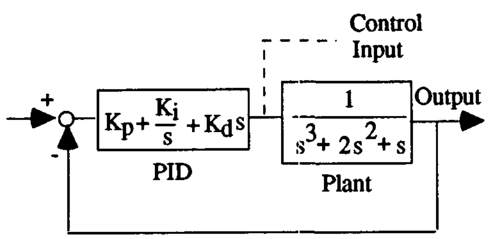

Fig. 1 Closed Loop system with PID Controller

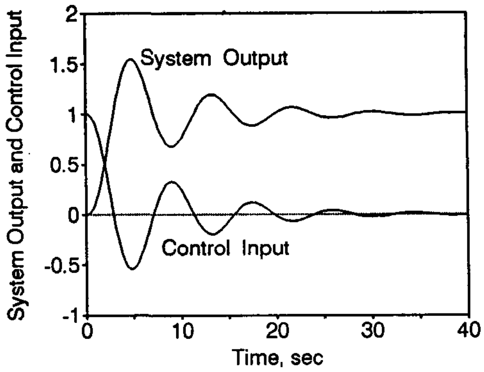

Fig. 2 Responses of Uncompensated System

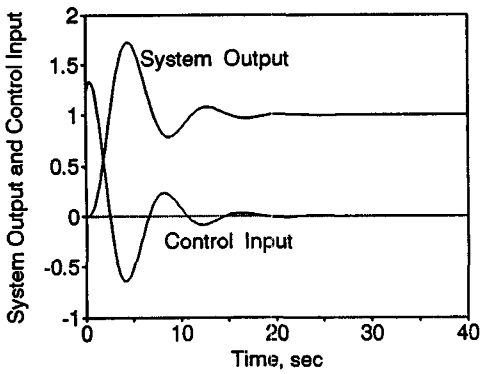

Fig. 3 Responses of Ziegler PID system

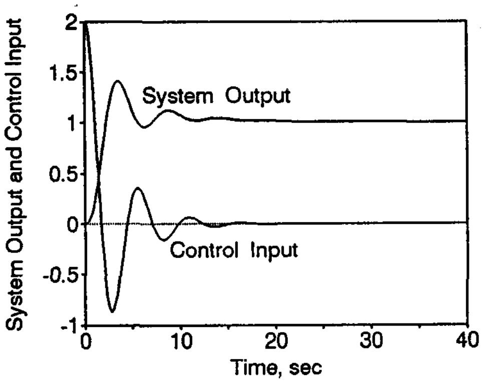

Fig. 4 Responses of Analytical PID System

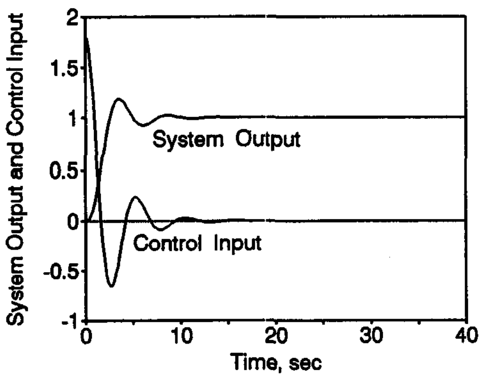

Fig. 5 Responses of GA PID System,
Weighting is determined internally

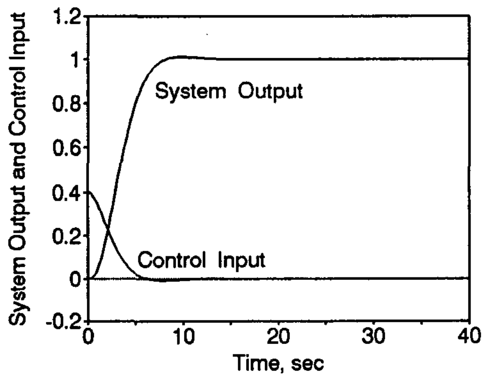

Fig. 6 Responses of GA PID System,
Weighting is provided externally