Optimal design of PID controller by Multi-objective genetic algorithms

Ouahib Guenounou
Laboratory of Industrial Technology and Information LT2I
University of Bejaia
Algeria, Bejaia 06000
Email: wguenounou@yahoo.fr

Boutaieb Dahhou
Laboratory of Analysis and Architecture of Systems
Toulouse, France
Email: dahhou@laas.fr

Billal Athmani
Laboratory of Industrial Technology and Information LT2I
University of Bejaia
Email: billal1983@hotmail.fr

Abstract—A novel design method for PID controller with optimal parameters is proposed based on the Improved Non-dominated Sorting Genetic Algorithm II (NSGA-II). The design of PID controller is formulated as multi-objective optimization problem where the integral of time multiplied by absolute error and integral of the square of the error (ISE), are optimized simultaneously. By testing two control systems, the proposed method has been able to produce a good performance control.

Index Terms—NSGA-II; PID controller; Laplace crossover; Pareto front

I. INTRODUCTION

Most real control problems are characterized by several objectives, often contradicting, that must be satisfied simultaneously. For example, the engineers are often faced with design problems, where a controller is needed that provide a small overshoot, fast response and economical control action. Since the 1980’s several Multi-objective Genetic Algorithms (MOGAs) have been proposed and applied in Multi-objective optimization problems [1, 2, 3, 4, 5, 6]. These algorithms are mainly classified in two approaches: (1) aggregative approach which consists to transform the Multi-objective optimization problem into a single objective optimization problem. (2) non aggregative approach which solves the Multi-objective optimization problem, based on Pareto’s principle. The aggregative approach is relatively simple but on other hand many design iterations are required to obtain the set of trade-off (Pareto front), leading to a prohibitive number of evaluations. For this reason, non aggregative approach is often used as solving method of Multi-objective optimization problems and generates a Pareto set in a single run.

In this work we use the improved non-dominated sorting genetic algorithm (NSGA-II), which is one of the most powerful non aggregative multi-objective techniques, and able to locate the Pareto front in complex reach space. The design of PID controller is considered here and it formulated as multi-objective optimization problem where the integral of time multiplied by absolute error and integral of the square of the error (ISE), are optimized simultaneously. This controller type is still the one most widely use in industrial control. However the optimal tuning of its three parameters is not an easy task, especially if several performance indices are simultaneously considered. Genetic algorithms (GA) are well suited for searching in large objective space.

The remainder of the paper is organized as follows: Section 2 describes the PID design problem. Section 3 gives a brief overviews of multi-objective optimization problem and NSGA-II algorithm. Section 4 presents a design method based on NSGA-II. Section 5 gives simulation results and Section 6 presents some conclusions.

II. PROBLEM FORMULATION

PID (proportional integral derivative) control is one of the earlier control strategies. The control-loop system is illustrated in Fig. 1, where it can be seen that in a PID controller, the error signal \(e(t)\) is used to generate the proportional, integral, and derivative actions, with the resulting signals weighted and summed to form the control signal \(u(t)\) applied to the plan model. A mathematical description of the PID controller is

\[
u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt}
\]  

(1)

where \(K_p\), \(K_i\) and \(K_d\) are the parameters of PID controller.

Formally, a "performance index" is defined as a quantitative measure to depict the system performance. To a PID control system, there are often four indices to depict the system performance: Integral of the Square of the Error (ISE), Integral of the Absolute magnitude of the Error (IAE), Integral of Time multiplied by Absolute Error (ITAE) and Integral of Time multiplied by the Square Error (ITSE). ITAE provides the best selectivity of the performance indices [7] and it is commonly referred to as a good criterion in designing PID controllers [8, [9]].
where the mathematical descriptions of these objectives are given by:

\[ Ob_1 = ITAE = \int_0^{T_s} t|e(t)|dt \]  \hspace{1cm} (2)

\[ Ob_2 = ISE = \int_0^{T_s} e(t)^2dt \]  \hspace{1cm} (3)

where the \( t \) is the time, the error signal \( e(t) \) is the difference between reference input signal \( r(t) \) and controlled variable \( y(t) \) and \( T_s \) is the simulation time.

III. DESCRIPTION OF MULTI-OBJECTIVE OPTIMIZATION PROBLEM AND NSGA-II

IV. MULTI-OBJECTIVE OPTIMIZATION PROBLEM

Without loss of generality, we consider here a multi-objective minimization problem which can be formulated as:

\[
\begin{align*}
\text{Minimize} & \quad F(X) = (f_1(X), f_2(X), \ldots, f_n(X)) \\
\text{s.t} & \quad g_i(X) \leq 0 \quad (i = 1, 2, \ldots, k) \\
& \quad h_i(X) = 0 \quad (i = 1, 2, \ldots, l)
\end{align*}
\]  \hspace{1cm} (4)

where \((f_1(X), f_2(X), \ldots, f_n(X))\) are the \( n \) different objective functions to be minimized, \( X = (x_1, x_2, \ldots, x_r) \) is the \( r \)-dimensional decision space, \( g_i(X) \) are the \( k \) inequality constraints, \( h_i(X) \) are the \( l \) equality constraints.

The objectives in the multi-objective optimization problems are often conflict each other, so it does not exist an absolutely optimal solution to optimize all objectives in same time, but can just get a set of satisfactory solutions i.e. Pareto solutions, which describe the trade-off among contradicted objectives.

For a multi-objective minimization problem mentioned above, any two solutions \( X_1 \) and \( X_2 \) can have one of two possibilities: one covers or dominates the other or none dominates the other. We say that the solution \( X_1 \) dominates \( X_2 \), if the following two conditions are satisfied:

\[
\begin{align*}
\forall i \in \{1, 2, \ldots, n\} : & \quad f_i(X_1) \leq f_i(X_2) \\
\exists j \in \{1, 2, \ldots, n\} : & \quad f_j(X_1) < f_j(X_2)
\end{align*}
\]  \hspace{1cm} (5)

If \( X_1 \) dominates solution \( X_2 \), \( X_1 \) is called the nondominated solution. The solutions that are nondominated within the entire search space are denoted as Pareto-optimal and constitute the Pareto-optimal front.

The goal of a multi-objective optimization algorithm is not only to guide the search towards the Pareto-optimal front, but, also to maintain population diversity in the set of the nondominated solutions.

A. GENERAL PRINCIPLE OF NSGA-II

The genetic algorithm, used in this paper, is an adaptation of a general structure of multi-objective genetic algorithms, called NSGA-II (non-dominated sorting genetic algorithm II) proposed by Deb [4]. This algorithm provides excellent results compared to others proposed multi-objective genetic algorithms such as its first version [10]. In this section, we present the general principle of NSGA-II, using the same notations adopted in [4]. Then we specify its components, for our problem, in Section 4. NSGA-II algorithm uses an elitist approach which can significantly speed up the performance of GA [11], a sorting procedure, based on a fast algorithm and a comparison operator based on calculus of a crowding distance instead a sharing function [12] which has been proved to be problematic (depends largely on the chosen sigma value). The general principle of NSGA-II is presented in Fig. 1: at each generation \( t \), a parent population \( P_t \) of size \( N \) and an offspring population \( Q_t \) of the same size are merged for forming a population \( R_t (P_t \cup Q_t) \), of size \( 2N \). Then, the population \( R_t \) is partitioned into a number of sets called fronts \( F \), which are constructed iteratively. Front \( F_1 \) consists of the non-dominated solutions from \( R_t \). \( F_2 \) consists of non-dominated solution from the set \((R_t - F_1)\) so on. In general, \( F_i \) consists of the non-dominated solutions from the set \((R_t - (F_1 \cup F_2 \cup \ldots \cup F_{i-1})\) . Deb et al. have proposed a fast partitioned algorithm called fast non-dominated sorting algorithm. Once all fronts are identified, a new parent population \( P_{t+1} \) of size \( N \) is formed by adding the fronts to \( P_{t+1} \) in order (front one \( F_1 \) followed by front two and so on) as long as the size of \( P_{t+1} \) do not exceed \( N \) individuals. If the number of individuals present in \( P_{t+1} \) is lower than \( N \), a crowding procedure is applied to the first front \( F_1 \) not included in \( P_{t+1} \). The aim of this procedure is to insert the \( N - P_{t+1} \) best individuals which miss to fill all population \( R_t \) . For each solution \( i \) in \( F_i \) a crowding distance \( d_i \) is calculated based on each objective \( Ob_k \). The front \( F_i \) is sorted according to objective function value \( Ob_k \) in ascending order.
order of magnitude. The first and the last individuals of the front are assigned infinity as their distance with respect to $O_R$. For all other intermediate individuals are assigned a distance value equal to the absolute normalized difference in the objective function values of two adjacent solutions. This calculation is repeated for other objective functions. The overall crowding distance for each solution $i$ is calculated as the sum of individual distance values corresponding to each objective function. The individuals with the highest crowding distance values are added to $P_{t+1}$ until $P_{t+1} = N$. Once the individuals appertaining to the population $P_{t+1}$ are identified, a new population $Q_{t+1}$ of size $N$ is created by selection, crossover and mutation using individuals of the population $P_{t+1}$. It is important to note that with NSGA-II, the selection procedure is based on a crowded-comparison operator defined as follows: to select an individual, two solutions are chosen randomly and uniformly from $P_{t+1}$ and that of smaller number of front $i_{rank}$ is preserved. If both solutions belong to the same front, one separates them by calculating the crowding distance for each solution and the solution with the higher distance is preferred. The NSGA-II operation will repeat the procedure until the stopping criteria is satisfied.

V. DESIGN OF PID CONTROLLER BY NSGA-II

A detailed method of the NSGA-II learning algorithm for the PID is introduced in this section. The proposed NSGA-II has been implemented using real-coded genetic algorithm (RCGA). Each coefficient of the PID is encoded by two real variables $\{p_{1,2}\}$ using the exponential form $K = p_1 10^{p_2}$. This allows the NSGA-II to search a wide range of values, with a small change of the exponential variable. So the chromosome uses strings of length 6.

In this study the Laplace crossover operator, recently proposed in [13] for RCGA, has been employed. This operator provides a good results and outperforms other crossover operators such as heuristic crossover [14].

Using LX, two offsprings $Y^1 = \{y^{1,1}, y^{1,2}, ..., y^{1,6}\}$ $Y^2 = \{y^{2,1}, y^{2,2}, ..., y^{2,6}\}$ and are generated from a pair of parents $X^1 = \{x^{1,1}, x^{1,2}, ..., x^{1,6}\}$ $X^2 = \{x^{2,1}, x^{2,2}, ..., x^{2,6}\}$ in the following way.

First, a uniformly distributed random number $u \in [0,1]$ is generated. Then, a random number $\beta$ is generated which follows the Laplace distribution by simply inverting the distribution function of Laplace distribution as follows:

$$\beta = \begin{cases} 
    a - b \log_2(u) & \text{if } u \leq \frac{1}{2} \\
    a + b \log_2(u) & \text{if } u > \frac{1}{2} 
\end{cases}$$

(6)

The offsprings are given by the equations:

$$y_i^{1,1} = x_i^{1,1} + \beta |x_i^{1,1} - x_i^{1,2}|$$

(7)

$$y_i^{2,1} = x_i^{2,1} + \beta |x_i^{2,1} - x_i^{2,2}|$$

(8)

A real value mutation is designed [15,16,17] for the real-string. Here we use the non-uniform mutation [15] which is one of the most widely used mutation operators in RCGA. So, at the $t^{th}$ generation, a parameter $x_i$ of the chromosome $X$ will be transformed to other parameter $x_i'$ with a probability $P_m$ as follows:

$$x_i' = \begin{cases} 
    x_i + \Delta(t, x_i^0 - x_i^1) & \text{if } r \leq \frac{1}{2} \\
    x_i - \Delta(t, x_i - x_i^0) & \text{otherwise} 
\end{cases}$$

(9)

where $r$ is a uniformly distributed random number between 0 and 1. $x_i^0$ and $x_i^1$ are lower and upper bounds of $x_i$, respectively. The function $\Delta(t,y)$ given below takes value in the interval $[0,y]$.

$$\Delta(t,y) = y(1-u(1-\frac{t}{T})^{b})$$

(10)

where $u$ is a uniformly distributed random number in the interval $[0,1]$. $T$ is the maximum number of generations and $b$ is a parameter, determining the strength of the mutation operator. In the initial generations nonuniform mutation tends to search the space uniformly and in the later generations it tends to search the space locally, i.e. closer to its descendants [15].

VI. SIMULATION EXPERIMENTS AND RESULTS

In order to examine the effectiveness of our approach, two typical control systems as used in [18,19] were tested. The transfer functions of the plants (Plant A and Plant B) in the two control systems are given as follows: $G_A(s) = (2/s(s + 1.4) + 2)$, $G_B(s) = 2/(s + 1)$.

Fig. 3 shows the closed-loop step responses of the two plants without a PID controller.

![Fig. 3. Step reponses](image-url)
Fig. 4 and Fig. 5 show the distribution of the two objectives corresponding to chromosomes of initial population for both plants A and B respectively.

The values of two objectives are relatively important and we have only two nondominated solutions in the case of Plant A and three nondominated solutions in the case of plant B. This is due to initialization step that creates chromosomes in randomly way. The PID’s parameters and two objective functions (ITAE and ISE) corresponding to the nondominated solutions of each study case (Plant A and Plant B) were summarized in table 1. Using these PID controller parameters, the unit step responses of each study case were obtained as shown in Fig.6 and Fig. 7.

From generation to generation, the values of Ob$_1$ and Ob$_2$ are significantly improved. Fig. 8 and Fig. 9 represent the distribution of obtained chromosomes at the end of optimization process related to the Plant A and Plant B respectively. It is important to note the simultaneous improvement of objectives ITAE and ISE. The Table 2 gives PID’s parameters and two objective functions corresponding to some solutions (four solutions) in the Pareto Front of each study case. Using these PID controller parameters, the unit step responses of each study case were obtained as shown in Fig.10 and Fig. 11.

Observing the values of the two objectives in Table 2 and the unit responses in figures 10 and 11, we can find that the PID controller has good unit step responses in each study case. Therefore, we can conclude that the NSGA-II algorithm is able
to find the optimal parameters of PID controller that provide a good performance control.

VII. CONCLUSION

A method to design a PID controller has been proposed. All the parameters of PID ($K_p$, $K_i$, and $K_d$) are optimally adjusted using a NSGA-II.

Two systems are used as test problems to analyze the performance of the proposed method. The effectiveness of the approach is well justified through simulation results. We have a good performance control for each system.

REFERENCES


[6] O. Guenounou, A. Belmehdi and B. Dahhou, Multi-


