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

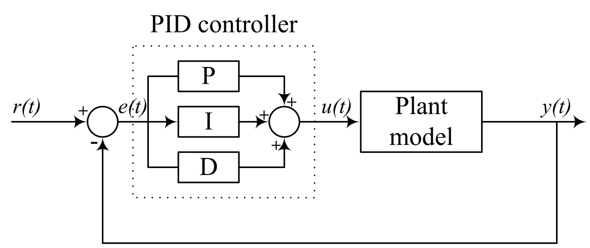

Fig. 1. A typical PID control structure

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt} \tag{1}$$

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].

In this work, two different objectives are considered, the first objective is the ITAE performance index and the second objective is the ISE. Whereas minimizing the first objective will provide good reference tracking and better disturbance rejection, minimizing the second reduces the rise time. The mathematical descriptions of these objectives are given by:

$$Ob_1 = ITAE = \int_0^{T_s} t|e(t)|dt \tag{2}$$

$$Ob_2 = ISE = \int_0^{T_s} e(t)^2 dt \tag{3}$$

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:

$$\begin{cases} Minimize & (F(X) = (f_1(X), f_2(X), ..., f_n(X))) \\ s.t & g_i(X) \leq 0 \ (i = 1, 2, \ldots, k) \\ & h_i(X) = 0 \ (i = 1, 2, \ldots, l) \end{cases} \tag{4}$$

where ($f_1(X)$,$f_2(X)$, ..., $f_n(X)$) are the n different objective functions to be minimized, $X = (x_1, x_2, ..., x_r)$ 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:

$$\begin{cases} \forall i \in \{1, 2, ..., n\}: & f_i(X_1) \leq f_i(X_2) \\ \exists j \in \{1, 2, ..., n\}: & f_j(X_1) < f_j(X_2) \end{cases} \tag{5}$$

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

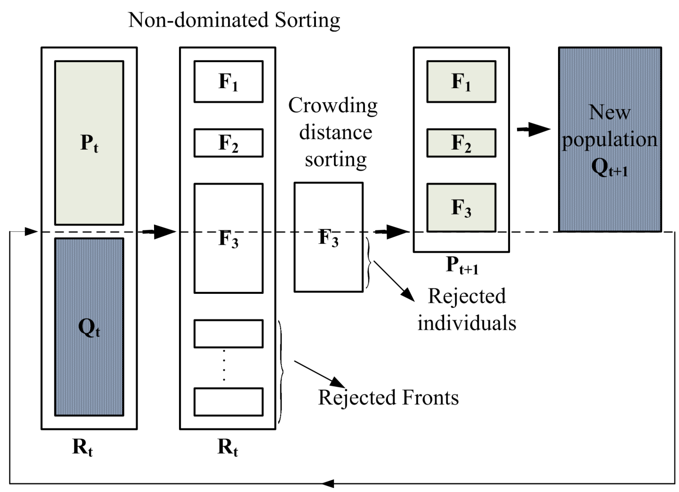

Fig. 2. NSGA-II procedure

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 (Rt - F1) and so on. In general, F_i consists of the non-dominated solutions from the set (R_t -($F_1 \cup F2 \cup ... \cup F_{i-1}$)). 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_i 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 Rt . 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 of magnitude. The first and the last individuals of the front are assigned infinity as their distance with respect to Ob_k. 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 $\{p_1, p_2\}$ 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 $Y^1 = (y_1^{(1)}, y_2^{(1)}, ..., y_6^{(1)})$ $Y^2 = (y_1^{(2)}, y_2^{(2)}, ..., y_6^{(2)})$ and are generated from a pair of parents $X^1 = (x_1^{(1)}, x_2^{(1)}, ..., x_6^{(1)})$ $X^2 = (x_1^{(2)}, x_2^{(2)}, ..., x_6^{(2)})$ in the following way.

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

$$\beta = \begin{cases} a - b\log_e(u) & \text{if } u \leq \frac{1}{2} \\ a + b\log_e(u) & \text{if } u > \frac{1}{2} \end{cases} \quad (6)$$

The offsprings are given by the equations

$$y_i^{(1)} = x_i^{(1)} + \beta|x_i^{(1)} - x_i^{(2)}| \quad (7)$$

$$y_i^{(2)} = x_i^{(2)} + \beta|x_i^{(1)} - x_i^{(2)}| \quad (8)$$

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:

$$x_i^{'} = \begin{cases} x_i + \Delta(t, x_i^u - x_i) & \text{if } r \leq \frac{1}{2} \\ x_i - \Delta(t, x_i - x_i^l) & \text{otherwise} \end{cases} \quad (9)$$

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

$$\Delta(t, y) = y(1 - u^{(1-\frac{t}{T})^b}) \quad (10)$$

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: $G_A(s) = (2/s(s + 1.4) + 2)$, $G_B(s) = 2/((s + 1)(s + 2))$.

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

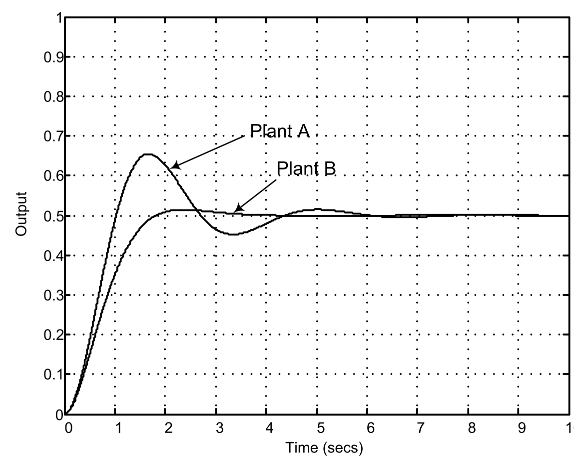

Fig. 3. Step reponses

The performance of the PID is evaluated based on the unit step responses and we used the ITAE (eq. 2) and ISE (eq. 3) criterions as objective functions.

In order to acquire better performance, several parameters of NSGA-II should be set appropriately. In this work, the used parameters are summarized as follows:

- crossover probability $P_c = 0.85$;
- mutation probability $P_m = 0.01$;
- population size $pop\_size = 40$;
- number of generations $T = 100$;

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.

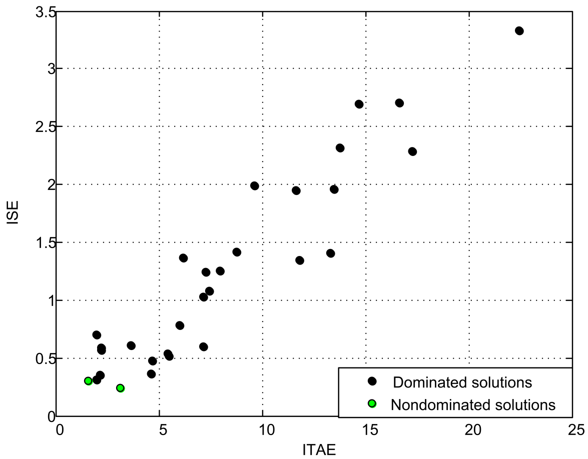

Fig. 4. Initial population (Plant A)

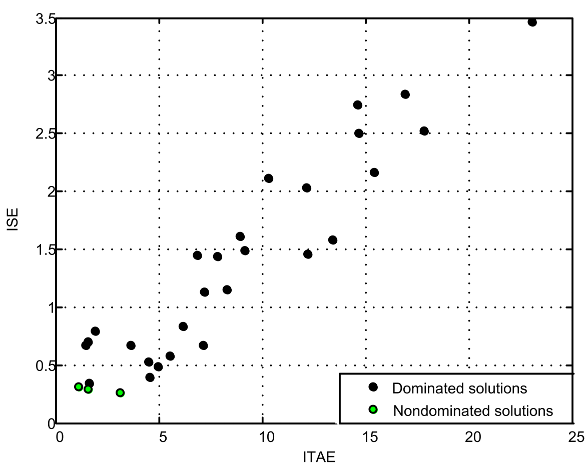

Fig. 5. Initial population (Plant B)

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.

TABLE I
SUMMARY OF SIMULATION RESULTS RELATED TO NONDOMINATED SOLUTIONS OF INITIAL POPULATION

Plant	K_e	K_i	K_d	$ITAE$	ISE
A	5.8349	1.4745	0.8294	1.5088	0.2978
	15.1942	0.0388	3.8185	3.0540	0.2403
B	5.8349	1.4745	0.8294	1.0775	0.3101
	6.1750	1.4180	0.1845	1.5487	0.2902
	15.1942	0.0388	3.8185	3.0771	0.2603

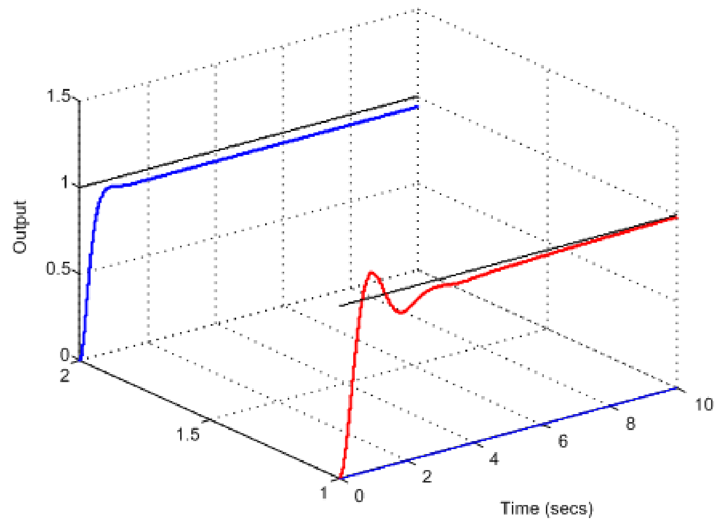

Fig. 6. Unit step response of Plant A

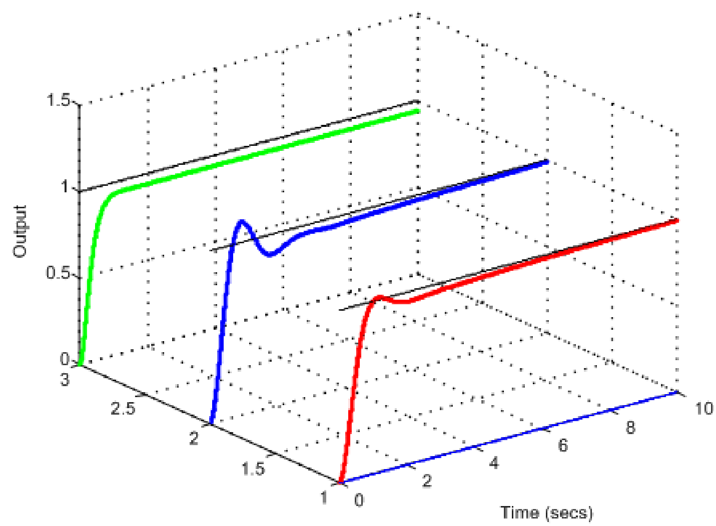

Fig. 7. Unit step response of Plant B

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 $ITEA$ 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

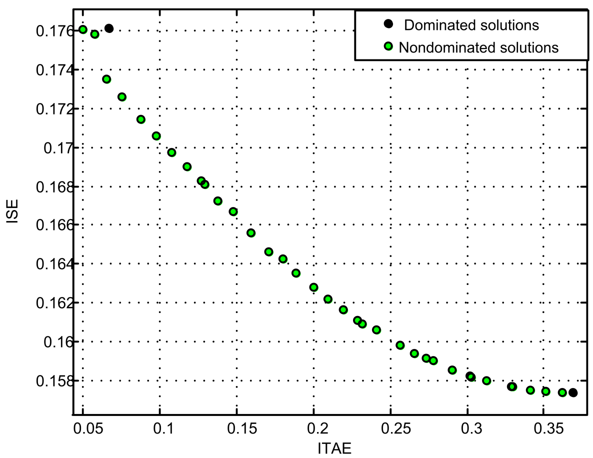

Fig. 8. Final population (Plant A)

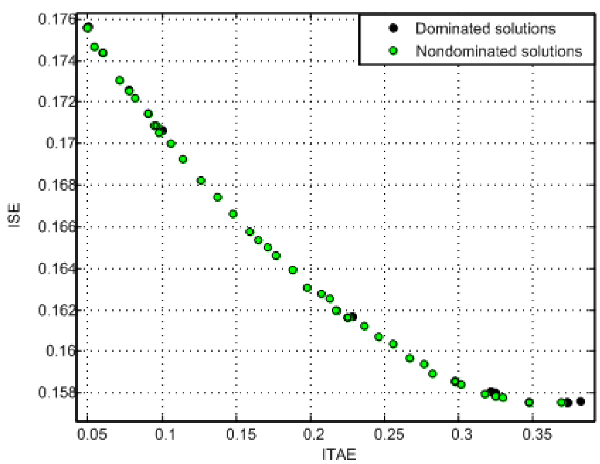

Fig. 9. Final population (Plant B)

TABLE II
SUMMARY OF SIMULATION RESULTS RELATED TO SOME SOLUTIONS IN PARETO FRONT

Plant	K_e	K_i	K_d	$ITAE$	ISE
A	19.9759	3.9966	4.4507	0.0578	0.1758
	19.9269	3.9950	4.0031	0.1074	0.1698
	19.9934	3.9737	2.5517	0.3514	0.1575
	19.9931	3.9845	2.4766	0.3623	0.1574
B	19.9066	3.9833	3.5269	0.0598	0.1744
	19.9453	3.9871	3.1740	0.1141	0.1693
	19.9759	3.9908	1.7540	0.3474	0.1576
	19.9759	3.9896	1.9285	0.3178	0.1580

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.

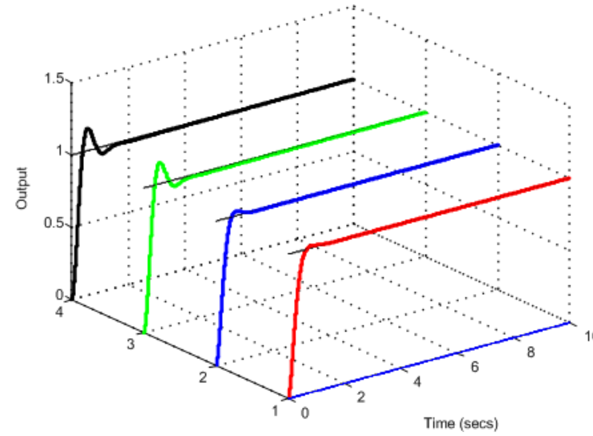

Fig. 10. Unit step response of Plant A

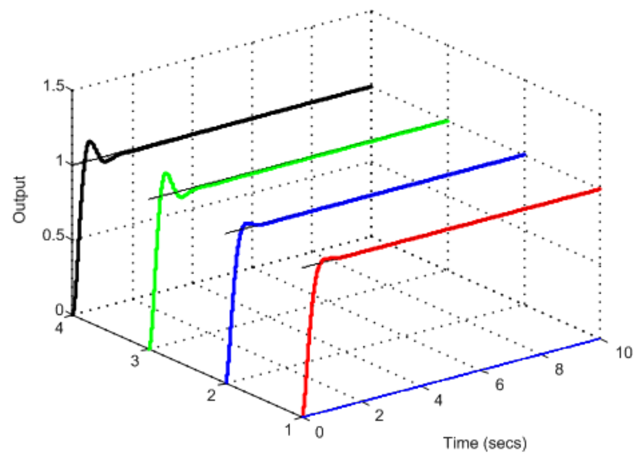

Fig. 11. Unit step response of Plant B

References

[1] J. D. Schaffer,*Multiple objective optimization with vector evaluated genetic algorithms*, In: Proceedings of the 1st International Conference on Genetic Algorithms, pp. 93-100, 1985.

[2] J. Horn and N. Napfliotis, *Multiobjective optimization using the niched pareto genetic algorithm*, In: Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence. Vol. 1, pp. 82-87, 1994

[3] N. Zitzler, K. Deb and L. Thiele, *Comparison of multiobjective evolutionary algorithms: Empirical results*, Evolutionary Computation, vol. 8, pp. 173-195, 2000.

[4] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, *A fast and elitist multiobjective genetic algorithm: NSGA-II*, IEEE Trans. Evolutionary Computation, vol. 6, pp. 182-197, 2002.

[5] K. P. Ferentinos and L. D. Albright, *Optimal design of plant lighting system by genetic algorithms*, Enginnering Applications of artificial intelligence, vol. 18, pp. 473-484, 2005.

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

objective optimization of TSK fuzzy models, Expert Systems with Applications, vol. 36, pp. 7416-7423, 2009.

[7] R. C. Dorf and R. H. Bishop, *Modern control systems*, Prentice Hall, 2008.

[8] F. G. Martins, *Tuning PID Controllers using the ITAE Criterion*, International Journal of Engineering Education, vol. 21, pp. 867-873, 2005.

[9] D. E. Seborg, T. F. Edgar and D. A. Mellichamp, *Process Dynamics and Control*, Wiley, 2004.

[10] N. Srinivas and K. Deb, *Multiobjective function optimisation using nondominated sorting function genetic algorithms*, Evolutionary Computation, vol. 62, 221248, 1995

[11] H. Meunier, E.-G. Talbi and P. Reininger, *A multiobjective genetic algorithm for radio network optimization*, In Proceedings of congress on evolutionary computation, San Diego, USA ,pp. 317324, 2000.

[12] J. Horn, N. Nafpliotis and D. E. Goldberg, *A niched pareto genetic algorithm for multiobjective optimization*, In Proceedings of the first IEEE conference on evolutionary computation, pp. 8287, 1994.

[13] K. Deep and M. Thakur, *A new crossover operator for real coded genetic algorithms*, Applied Mathematics and Computation, vol. 188, pp. 895-911, 2007.

[14] A. H. Wright, *Genetic algorithms for real parameter optimization*, Ain: G.J.E. Rawlins (Ed.), Foundations of Genetic Algorithms I, Morgan Kaufmann, San Mateo, pp. 205-218, 1991.

[15] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*,Springer-Verlag, New York, 1992.

[16] R. A. E. Makinen, J. Periaux and J. Toivanen, *Multidisciplinary shape optimization in aerodynamics and electromagnetic using genetic algorithms*, International Journal for Numerical Methods in Fluids, vol. 30, pp. 149-159, 1999.

[17] K. Deep and M. Thakur, *A new mutation operator for real coded genetic algorithms*, Applied Mathematics and Computation, vol. 193, pp. 211-230, 2007.

[18] F. Cheong and R. Lai, *Constraining the Optimization of a Fuzzy Logic Controller Using an Enhanced Genetic Algorithm*, IEEE Trans. on systems, Man, and Cybernetics, vol. 30(1), pp. 31-46, 2000.

[19] T. C. Chin, H. C. Hua, K. W. Yeo and W. L. Yeo, *Genetic algorithm based optimal fuzzy controller*, In Proceedings 2nd In ternational Conference on Intelligent Systems, pp. 135-140, 1994.