4th WSEAS International Conference on ELECTRONICS, CONTROL and SIGNAL PROCESSING, Miami, Florida, USA, 17-19 November, 2005 (pp.144-149)

Robust 2DOF PID Controller Design of Time-delay Sysytems Based on Evolutionary Computation

TOHRU KAWABE
Department of Computer Science
Graduate School of Systems and Information Engineering
University of Tsukuba, Tsukuba, 305-8573, JAPAN

Abstract: - This paper presents a new evolutionary computing method for the design of robust PID controller with two degrees of freedom (2DOF PID controller). Since the objective plants have a time-delay element and parametric uncertainties, the design problem is formulated as a multi-objective minimax optimization problem. Therefore the high-performed optimization method is required. The proposed evolutionary computing method is to generate a set of pareto-optimal solution that is properly distributed in the trade-off surface generated by multi-objective functions. Numerical examples show the effectiveness of the proposed approach.

Key-Words: - Evorutionary computing, Dividing strategy, PID controller, Time-delay systems, Robustness

1 Introduction

Many engineering practical problems require the simultaneous optimization of multiple, often competing, objectives. Unlike single objective optimization, the solution to this type problem is not a single point, but a family of points known as the Pareto-optimal set, also called non-dominated set. Each point in the tradeoff surface is optimal in the sense that no improvement can be achieved in one cost vector component that does not lead to degradation in at least one of the remaining components.

The parallel optimization techniques by using evolutionary computation such as Genetic algorithms (GAs) [1, 2] have been recognized to be well suited to the multiobjective optimization problem. Multiple individuals can search for multiple solutions in parallel, eventually taking advantage of any similarities available in the family of possible solutions to the problem. Extensions of GAs to multiobjective optimization were proposed in several manners [3, 4, 5]. Schaffer [3] proposed an extension of the simple GA to accommodate vector-valued fitness measures, which he called the Vector Evaluated Genetic Algorithm (VEGA).

While a Pareto-based approach was first proposed by Goldberg[1], as a means of assigning equal probability of reproduction to all non-dominated individuals in the population. Fonseca and Fleming[4] proposed a multiobjective ranking method with the Pareto-based fitness assignment.

On the other hand, the proportional-integral-derivative (PID) controller design problem is one of this kind of optimization problems if PID controller has two degrees of freedom (2DOF). Recently, in the practical control field, the 1DOF PID controllers have a renewed interest due to the recent development of digital controllers. In fact, over 90% of industrial control problems are solved by PID controllers (or by their variants) in spite of the simple structures [6]. Many tuning methods of 1DOF PID controller have been developed consequently [6, 7, 8, 9, 10]. Most of these methods, however, are unable to apply the multiple specification design problem, such as the case required the optimization of reference response and disturbance response at the same time, since these methods are based on 1DOF PID controller.

To overcome these weakpoints, we have proposed a new design method of robust PID controller with two degrees of freedom (2DOF PID controller) based on the H^2 performance measure and exact robust stability check method about time-delay systems [10]. In this method, we need to solve the non-convex and multiobjective optimization problems. Hence, we need the effective optimization tool. This paper, therefore, presents a design method based on a new evolutionary computing method with dividing strategy for this type problem.

The purpose of the proposed evolutionary computing method with dividing strategy is to generate a Pareto-optimal set that is properly distributed in the tradeoff surface of the 2DOF robust PID controller design as multiobjective optimization problem. The search using the proposed method uniformly control the convergence of solutions. Some numerical results to demonstrate the effectiveness of proposed method are also included.

2 New evolutionary computing method

2.1 Dividing strategy

To prevent a partial convergence of non-dominated solutions in the trade-off surface, the dividing method which uniformly controls the distribution of solutions is proposed. The proposed method assigns all non-dominated individuals to pre-specified regions. An example of the dividing strategy in two objective minimizing problem is shown in fig. 1. The proposed method consists of following procedure. First, the objective space is divided pre-specified regions. The edge points of the whole region correspond the best solutions for each objective function. In the fig. 1, the individuals p_1 and p_7 match them. Then, the fitness f_i of the individual p_i is defined as $f_i = 1/n_i$. The value of n_i denotes the number of non-dominated solutions in the identical region with the individual p_i. In the example, the fitness of the individuals illustrated in the figure correspond to the following values $(f_1, f_2, f_3, f_4, f_5, f_6, f_7) = (1/3, 1/3, 1/3, 1, 1, 1/2, 1/2)$. In the proposed evolutionary computing method, let's define a neighborhood to every individual as follow: Two objective functions of m-objective problem are selected by using pre-specified selective probabilities. Individuals are arranged on the two-dimensional coordinates, and the neighborhood of an individual is calculated by using the relative distance between all individuals.

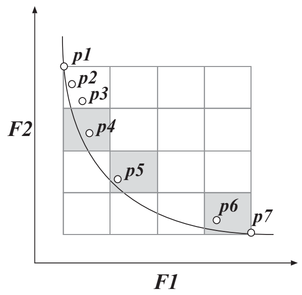


Fig. 1 Dividing strategy in two objective problem

The crossover operator is made locally in each neighborhood in parallel. Even if the fitness of an individual is relatively very high in a population, it can spread over the succeeding populations only through an overlap of the neighborhood. This prohibits a rapid increase of an relatively high performance individual, and then, the population diversity is favorably maintained. The evolutionary operators are defined as follows:

(a) The selection is done by considering the number of individuals in the 2-dimensional objective space. That is, the fitness Γ_i of the individual p_i is defined as $\Gamma_i = 1/n_i$. The value of n_i denotes the number of individuals in the identical region with the individual p_i. The proportional fitness method is employed in the selection process.

(b) BLX-α [11] method is employed for crossover. In the experiments, control parameter α is fixed 0.5. The mate of crossover is chosen randomly in the neighborhood.

(c) The real-code string representation is employed for candidate solution.

(d) Mutation is designed to perform random exchange; that is, it selects some bits randomly in an individual and exchanges their values. Boundary mutation and non-uniform mutation are used to avoid the premature convergence of the solutions.

The proposed procedure consists of the following steps:

Step 1. Set a generation number $t = 0$. Randomly generate an initial population $P(t)$ of M individuals.

Step 2. Calculate the fitness of each individuals in the current population according to the distribution of the objective space.

Step 3. Select M individuals according to above fitness, then the mate of the individuals are chosen randomly in the neighborhood.

Step 4. Generate a new population $P'(t)$ from $P(t)$ by using a crossover operator.

Step 5. Apply a mutation operator to the newly generated population $P'(t)$.

Step 6. Calculate the fitness both of $P(t)$ and $P'(t)$.

Step 7. Select M individuals from all population member on the basis of the fitness.

Step 8. If a terminal condition is satisfied, stop and return the best individuals. Otherwise set $t = t + 1$ and go to [Step 2].

In this procedure, update of the current population size is always constant M. Here, to avoid the rapid loss of genetic diversity, multiple equivalent individuals are eliminated from the current population.

2.2 Performance check

In this section, the performance ability of the proposed evolutionary computing method with dividing strategy is checked by following a two objective problem.

$$\begin{aligned} \min_{x_1, x_2} \quad & F_1(x_1, x_2) \\ \min_{x_1, x_2} \quad & F_2(x_1, x_2) \\ \textbf{subject to} \quad & -4 \leq x_1,\ x_2 \leq 4 \end{aligned}$$

where

$$\begin{aligned} F_1(x_1, x_2) \quad &:= \quad \left[1 - \exp\left(-(x_1 - 1)^2 - (x_2 + 1)^2\right)\right]^{\alpha} \\ F_2(x_1, x_2) \quad &:= \quad \left[1 - \exp\left(-(x_1 + 1)^2 - (x_2 - 1)^2\right)\right]^{\alpha} \end{aligned}$$

and where α is a free parameter. The shape of trade-off surface is changed convex or non-convex depend on the value of α. The following GA parameter specifications are used in this test problem.

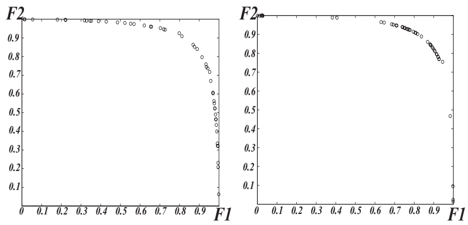
Population size	:	50
Mutation rate	:	0.10
Maximum generation	:	100
The number of dividing region	:	$50^2 = 2500$

For comparison with the proposed evorutionary computing method with dividing strategy, the standard genetic algorithm (GA) with multiobjective ranking[4] is used. In the standard GA, the newly generated population member is randomly selected from current population when the number of non-dominated solutions is over M. The genetic operators are same as the proposed method. The simulation results are shown in figs. 2, 3 and 4 for the cases when the value of α is 1.0, 5.0 and 10.0, respectively. In all figures, small marks indicate the solutions in the final population.

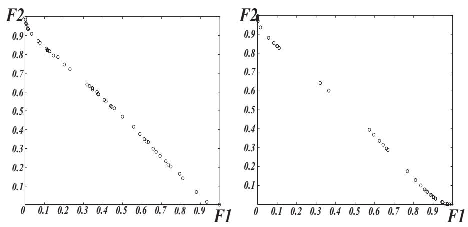
From these simulation results, the following facts are obtained:

- By using the proposed method, the solutions are widely distributed in the trade-off surface, and the search performance does not deteriorate significantly.
- The standard GA approaches cause the partial convergence of the solutions because of stochastic errors in the iterative process.

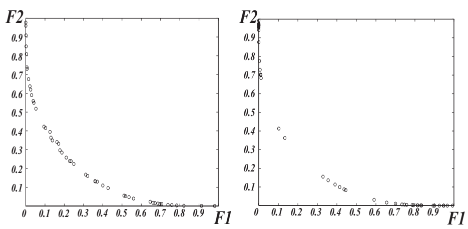
- It is clear that the proposed method can seek for the widely distributed solutions in comparison with the standard GA.



(a) the proposed method (b) the standard GA
Fig. 2 Simulation results ($\alpha = 1.0$)



(a) the proposed method (b) the standard GA
Fig. 3 Simulation results ($\alpha = 5.0$)



(a) the proposed method (b) the standard GA
Fig. 4 Simulation results ($\alpha = 10.0$)

Although the proposed method take more computation time than the standard one naturally, the differences are only a few seconds. Hence, the proposed evolutionary computing method with dividing strategy can be recognized the effective method for multi-objective otimization problems.

3 Design problem

3.1 Problem formulation

Consider the SISO control system shown in Fig. 5, where r is the step input, u the manipulated variable, y the measured variable and d the step disturbance. We assume that the plant model is described by $G(s) := P(s)e^{-Ls}$, where where $P(s)$ is strictly proper rational function with parametric uncertainties and L is a delay.

Let θ be the parameter vector of $P(s)$ and belong to a bounded set $\Theta = \{\theta \,|\, \theta_l \leq \theta \leq \theta_u\}$. The 2DOF PID controller is consist of feedback part, $C_1(s)$, and feedforward part $C_2(s)$ as:

$$C_1(s) = \frac{K_I + K_P s + K_D s^2}{s},$$
$$C_2(s) = -\alpha K_P - \beta K_D s$$

where K_P, K_I, K_D, α, $\beta \in \mathbf{R}$ are parameters of the 2DOF PID controller. Now, if $\alpha = \beta = 0$ then this 2DOF PID controller is the standard 1DOF PID controller. On the other hand, in the case of $\alpha = \beta = 1$, it is the so-called 1DOF I-PD controller.

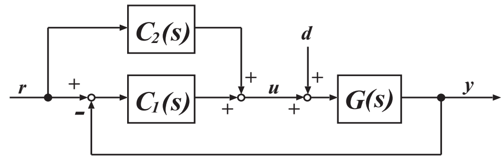


Fig. 5 Control system with 2DOF controller

Let $q_1 := (K_P, K_I, K_D)^T$ and $q_2 := (\alpha, \beta)^T$ be the 2DOF PID controller parameter vectors belonging to bounded sets $Q_1 = \{q_1 \,|\, q_1^l \leq q_1 \leq q_1^u\} \subset \mathbf{R}^3$ and $Q_2 = \{q_2 \,|\, q_2^l \leq q_2 \leq q_2^u\} \subset \mathbf{R}^2$, which are the ranges of adjustable 2DOF PID parameters. It is assumed that Θ and Q_1 are given a priori.

From fig. 5, G_{yr} which is the transfer function from r to y and G_{yd}, the transfer function from d to y, are given by

$$G_{yr}(s) = \frac{[C_1(s) + C_2(s)]P(s)e^{-Ls}}{1 + C_1(s)P(s)e^{-Ls}} \quad (1)$$
$$G_{yd}(s) = \frac{P(s)e^{-Ls}}{1 + C_1(s)P(s)e^{-Ls}}. \quad (2)$$

Then the tracking errors of reference response and disturbance response are expressed as

$$E_r(s) = \frac{B_r(s) + D_r(s)e^{-Ls}}{A(s) + C(s)e^{-Ls}} \cdot r(s) \quad (3)$$
$$E_d(s) = \frac{B_d(s) + D_d(s)e^{-Ls}}{A(s) + C(s)e^{-Ls}} \cdot d(s). \quad (4)$$

Where $P(s) = N(s)/D(s)$, and define $A(s)$, $B_r(s)$, $B_d(s)$, $C(s)$, $D_r(s)$ and $D_d(s)$ as

$$\begin{aligned} A(s) &= sD(s), \\ B_r(s) &= sD(s), \\ B_d(s) &= 0, \\ C(s) &= [K_I + K_P s + K_D s^2]N(s), \\ D_r(s) &= [\alpha K_P + \beta K_D s]N(s), \\ D_d(s) &= -sN(s). \end{aligned} \quad (5)$$

We choose the square of H^2 norm of E_r and E_d as the performance measures $J_r(q_1, q_2, \theta)$ and $J_d(q_1, q_2, \theta)$:

$$J(q_1, q_2, \theta) := \frac{1}{2\pi j}\int_{-j\infty}^{+j\infty} E(s)E(-s)ds \quad (6)$$

. where $J(q_1, q_2, \theta)$ is $J_r(q_1, q_2, \theta)$ or $J_d(q_1, q_2, \theta)$, and where $E(s)$ is $E_r(s)$ or $E_d(s)$. In the case of $u(s) = 1/s$, J_r indicates the standard ISE (Integral Squared Error) of step reference input.

Hence, the design problem of robust 2DOF PID controller based on H^2 optimization is formulated as a following minimax optimization problem:

Design problem :

$$\min_{q_1 \in Q_1, q_2 \in Q_2} \max_{\theta \in \Theta} J_d(q_1, q_2, \theta) \quad (7)$$
$$\min_{q_1 \in Q_1, q_2 \in Q_2} \max_{\theta \in \Theta} J_r(q_1, q_2, \theta) \quad (8)$$

s.t. closed-loop system is stable for $\forall\, \theta \in \Theta$

We note that this design problem is a multiobjective optimization problem with a significant constraint, that is the closed-loop system should be robustly stable for all plant parameters in the set Θ. About this point, the characteristic equation of the closed-loop system is given by

$$f(s) := A(s) + C(s)\, e^{-Ls} = 0. \tag{9}$$

Since the plant is strictly proper, the degree of $A(s)$ is always greater than that of $C(s)$, so that this system is a retarded system. Then we can check the exact robust stability of eq.(9) without any approximation for time-delay elements by using the method proposed in [12, 10] which is presented in the nest section.

3.2 Exact robust stability check method

Let's assume that $N(s)$ and $D(s)$ in eq. (3) are the (real) interval polynomials. Let N denote the number of vertices of the hyperrectangle Θ in the parameter space and

$$f^i(s) = A^i(s) + C^i(s)e^{-Ls}, i = 1, \cdots, N, \tag{10}$$

be the quasi-polynomials corresponding to each vertex. Then the characteristic quasi-polynomial is generated by the convex combination of these quasi-polynomials:

$$f(s,k) := \sum_{i=1}^{r} k_i\, f^i(s), \tag{11}$$

$$s.t. \sum_{i=1}^{N} k_i = 1,\ 0 \le k_i \le 1,\ i = 1, \cdots, N.$$

Let $k := (k_1, \cdots, k_N)^T$ and define the convex polyhedron K, the quasi-polynomial family S, and the value set S_ω as

$$\begin{aligned} K &:= \left\{ k \;\middle|\; \sum_{i=1}^{N} k_i = 1,\ 0 \le k_i \le 1 \right\}, \\ S &:= \{f(s,k) \mid k \in K\}, \\ S_\omega &:= \{f(j\omega,k) \mid k \in K\}. \end{aligned} \tag{12}$$

The shape of S_ω is a polygonal region and each quasi-polynomial segment corresponding to the boundary of S_ω is called edge. From the edge theorem [13] and the zero exclusion principle [14], we see that the quasi-polynomial family S is stable if and only if the boundary of S_ω does not contain or pass through the origin for all $\omega \ge 0$. Since the boundary of S_ω is the value set of the segment quasi-polynomial corresponding to two generating points of S, we consider an edge connecting $f^1(s)$ and $f^2(s)$. Define the quasi-polynomial segment

$$f(s,\lambda) := (1-\lambda)f^1(s) + \lambda f^2(s), \tag{13}$$

with $\lambda \in [0,1]$. From (10) we can express (13) as

$$f(s,\lambda) = A(s,\lambda) + C(s,\lambda)\, e^{-Ls}, \tag{14}$$

where $A(s,\lambda)$ and $C(s,\lambda)$ are defined as

$$A(s,\lambda) := (1-\lambda)A^1(s) + \lambda A^2(s),$$

$$B(s,\lambda) := (1-\lambda)C^1(s) + \lambda C^2(s).$$

A stability criterion of the quasi-polynomial segment (13) is given as follows:

Theorem 1
Given $\lambda \in [0,1]$, let ω_λ be

$$\begin{aligned} \omega_\lambda = \sup\{\omega \mid A(j\omega,\lambda)A(-j\omega,\lambda) - \\ C(j\omega,\lambda)C(-j\omega,\lambda) = 0\}. \end{aligned}$$

Also let $\bar\omega$ be

$$\bar\omega = \sup\{\omega_\lambda \mid \lambda \in [0,1]\}.$$

Then, the quasi-polynomial segment (13) contains or passes through the origin for $\omega \ge 0$ if and only if there exist $\omega \in [0, \bar\omega]$ satisfying the following condition

$$\begin{aligned} &\mathrm{Re}[f^1(j\omega)]\ \mathrm{Im}[f^2(j\omega)] - \\ &\qquad \mathrm{Re}[f^2(j\omega)]\ \mathrm{Im}[f^1(j\omega)] = 0, \\ &\mathrm{Re}[f^1(j\omega)]\ \mathrm{Re}[f^2(j\omega)] \le 0, \\ &\mathrm{Im}[f^1(j\omega)]\ \mathrm{Im}[f^2(j\omega)] \le 0. \end{aligned} \tag{15}$$

We can summarize the procedure examining stability of the quasi-polynomial family S as follows:

Step 1: Examine the stability of one quasi-polynomial in S. If the quasi-polynomial is stable, go to Step 2. If not, S is not stable.

Step 2: Check whether $0 \notin S_{\omega_0}$ for one $\omega_0 \ge 0$. If $0 \notin S_{\omega_0}$, go to Step 3. If not, S is not stable.

Step 3: For each edge, check the existence of $\omega \ge 0$ such that (15) holds. If there exist such an ω on at least one edge, S is not stable. If not, S is stable.

A exact computation method of the H^2 performance measures for the dime-delay systems is presented in the next section.

3.3 Exact computing method

By using the technique of [15], we develop a method of computing the standard H^2 performance measure of eq. (6). It is assumed that the system is stable, so that all the poles of (9) lie in the left half-plane. For the notational convenience, we suppress the Laplace variable s of each polynomial and denote its paraconjugate by $\bar{(\cdot)}$, for example, we write $A := A(s)$ and $\bar A := A(-s)$. Then the integrand of (6) is expressed as

$$E(s)E(-s) = \frac{B + De^{-Ls}}{A + Ce^{-Ls}} \cdot \frac{\bar B + \bar D e^{Ls}}{\bar A + \bar C e^{Ls}} \tag{16}$$

where B indicates $B_r(s)$ or $B_d(s)$, and where D is $D_r(s)$ or $D_d(s)$. It can be shown that (16) is additively decomposed as

$$\begin{aligned} E(s)E(-s) &= \frac{\Lambda + \Pi e^{-Ls}}{A + Ce^{-Ls}} + \frac{\bar\Lambda + \bar\Pi e^{Ls}}{\bar A + \bar C e^{Ls}} \\ &=: H(s) + H(-s), \end{aligned} \tag{17}$$

where

$$\begin{aligned} \Lambda &:= \frac{A(B\bar B + D\bar D) - 2BC\bar D}{2(A\bar A - C\bar C)}, \\ \Pi &:= \frac{2AD\bar B - C(B\bar B + D\bar D)}{2(A\bar A - C\bar C)}. \end{aligned} \tag{18}$$

Since the system is stable, $A + Ce^{-Ls} = 0$ has no root. Then the singularities of the integrand are limited to the roots of $A\bar{A} - C\bar{C} = 0$, which are denoted by s_i. It is crucial that $\{s_i\}$ is a finite set. It follows from the residue theorem that

$$I := -\sum_i \mathrm{Res}\left[\frac{\Lambda + \Pi e^{-Ls}}{A + Ce^{-Ls}}, s_i\right], \tag{19}$$

where $\mathrm{Res}\,[f(s), s_i]$ denotes the residues of $f(s)$ at s_i. If s_i is a singularity of $H(s)$, $-s_i$ is also a singularity of $H(-s)$. With respect to residues, we have

$$\mathrm{Res}\,[H(s), s_i] = -\mathrm{Res}\,[H(-s), -s_i]. \tag{20}$$

It follows from (20) that the integral of $H(-s)$ is equal to I. Since $E(s)$ has no singularity on the imaginary axis, we can obtain $J = 2I$.

4 Design algorithm

For computational purpose, let $Q_{1d} := \{q_1^1, \cdots, q_1^N\}$ and $Q_{2d} := \{q_2^1, \cdots, q_2^N\}$ be discrete approximations of the sets Q_1 and Q_2. Then the design algorithm of robust 2DOF PID parameters is summarized as follows.

Step 1: Check robust stability of q_1^i for all $\theta \in \Theta$, where q_1^i and q_2^j are generated by an optimization tool.

Step 2: If the closed-loop system with q_1^i is robustly stable, compute

$$Jd_{max}^{ij} := \max_{\theta\in\Theta} J_d(q_1^i, q_2^j, \theta), \tag{21}$$

$$Jr_{max}^{ij} := \max_{\theta\in\Theta} J_r(q_1^i, q_2^j, \theta). \tag{22}$$

If the system is not robustly stable, set $Jd_{min}^{ij} = \infty$ and $Jr_{min}^{ij} = \infty$.

Step 3: If an algorithm of optimaization tool stops, go to Step 4. Otherwise, go to Step 1.

Step 4: Let the minimum of Jd_{max}^{ij} and J_{max}^{ij} be J_d^{oi} and J_r^{oi} respectively. Then the corresponding q_1^{io} and q_2^{io} yields the minimax robust 2DOF PID controller.

It should be noted that the H^2 performance measure, $J(q_1, q_2, \theta)$, does not have saddle point in this case. Since there are many local optimal solutions for the controller parameters, we use the hybrid strategy with local search and the genetic computing method with dividing strategy which is explained in the next chapter. This means grid search is used for maximization, and the proposed method for minimization.

5 Numerical Examples

We consider these plants with transfer functions

$$G_1(s) = \frac{K_p}{1 + T_p s} e^{-Ls} \tag{23}$$

$$G_2(s) = \frac{K_p}{(1 + T_{p1}s)(1 + T_{p2}s)} e^{-Ls} \tag{24}$$

and assume that the set Q_1 and Q_2 for controller are given by

$$Q_1 = \{q_1 \mid \underbrace{\begin{bmatrix}0.10\\0.01\\0.01\end{bmatrix}}_{q_1^l} \le \underbrace{\begin{bmatrix}K_P\\K_I\\K_D\end{bmatrix}}_{q_1} \le \underbrace{\begin{bmatrix}15.00\\120.00\\120.00\end{bmatrix}}_{q_1^u}\} \tag{25}$$

$$Q_2 = \{q_2 \mid \underbrace{\begin{bmatrix}0.00\\0.00\end{bmatrix}}_{q_2^l} \le \underbrace{\begin{bmatrix}\alpha\\\beta\end{bmatrix}}_{q_2} \le \underbrace{\begin{bmatrix}1.00\\1.00\end{bmatrix}}_{q_2^u}\} \tag{26}$$

5.1 Evolutionary computing formulation

The following parameter specifications are employed for all experiments.

Population size	:	200
Mutation rate	:	0.20
Maximum generation	:	15000

The control parameters are held constant during all experiments. The algorithm stops when the best function value is smaller than 1.0×10^{-7}. In bit-based encodings, each parameter has 20 bit of precision, giving a total search space of 2^{400} points.

5.2 Numerical result of $G_1(s)$

Let the uncertainty set of the plant parameters be given by

$$\Theta = \{\theta \mid \underbrace{\begin{bmatrix}0.8\\5.6\end{bmatrix}}_{\theta_l} \le \underbrace{\begin{bmatrix}K_p\\T_p\end{bmatrix}}_{\theta} \le \underbrace{\begin{bmatrix}1.2\\8.4\end{bmatrix}}_{\theta_u}\} \tag{27}$$

The delay-time of e^{-Ls} is $L = 2.5$. The design result is given in table 1 and values of J_d and J_r are 0.7955 and 0.2557 respectively.

Table 1 Design result ($G_1(s)$)

K_P	K_I	K_D	α	β
6.653	1.974	1.189	0.209	0.0186

Figs. 6 and 7 show the step responses of closed-loop system by using the values in table 1. Solid line shows the worst case response and dashed lines shows the best response in the Θ. In the worst case, step and disturbance responses are not oscillated and the deteriorating of response is suppressed to minimum compred with best case. We therefore see the good robust performance.

5.3 Numerical result of $G_2(s)$

Let

$$\Theta = \{\theta \mid \underbrace{\begin{bmatrix}0.8\\4.57\\1.16\end{bmatrix}}_{\theta_l} \le \underbrace{\begin{bmatrix}K_p\\T_{p1}\\T_{p2}\end{bmatrix}}_{\theta} \le \underbrace{\begin{bmatrix}1.2\\6.85\\1.74\end{bmatrix}}_{\theta_u}\} \tag{28}$$

The delay-time is $L = 2.5$. The numerical result is shown in table 2 and values of J_d and J_r are 12.782 and 1.236 respectively. Closed-loop responses are depicted in figs. 7 and 8.

Table 2 Design result ($G_2(s)$)

K_P	K_I	K_D	α	β
8.331	1.642	8.987	0.187	0.0124

Figs. 8 and 9 show the step responses of closed-loop system with the values in table 2. We also see the good robust preformance.

By comparison between the proposed evolutionary computing method and the standard GA in the same ploblems mentioned above, the following facts are obtained. By using the proposed method, the solutions are widely distributed in the trade-off surface. The performance with the design result by the proposed method is bit better than the standard GA method. Although the proposed method take more computation time than the standard one naturally, the differences are only a few seconds in the all cases.

6 Conclusion

In this paper, new evolutionary computing method with ddividing strategy for the design of robust 2DOF PID controller design problem of time-delay systems has been proposed. And the effectiveness of proposed method has been recognized from simulation results. The automatic tuning method of dividing numbers, the appropriate selection method of best solution among the pareto-optimal set, and so on, are remained as future works.

References

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

[2] Z. Michalewics and C. Janikow, Genetic Algorithms + Data Structures = Evolution Programs, *Springer Verlag*, 1992.

[3] J.D. Schaffer, Multiple Objective Optimization with Vector Evaluated Genetic Algorithms, *Proc. of the 1st ICGA*, 1985, pp.93-100.

[4] C.M. Fonseca and P.J. Fleming, An Overview of Evolutionary Algorithms in Multiobjective Optimization, *Evolutionary Computation*, 3(1), 1995, pp.1-16.

[5] D.H. Loughlin and S. Ranjithan, The Nighborhood Constraint Method; A Genetic Algorithm-Based Multiobjective Optimization Technique, *Proc. of the 7th ICGA*, 1995, pp.666-673.

[6] K.J. Åström and T. Hägglund, *PID Control – Theory, Design and Tuning, 2nd ed.* Instrument Society of America, 1995.

[7] M. Morari and E. Zafiriou, *Robust Process Control.*, 1989, Prentice-Hall.

[8] B.S. Chen, Y.M. Cheng and C.H. Lee, A Genetic Approach to Mixed H_2/H_∞ Optimal PID Control. *IEEE Control Systems Magazine*, **15**, 1995, pp.51-60.

[9] T. Kawabe and T. Katayama, A Minimax Approach to Design of Robust I-PD Controller. *Proc. 2nd Asian Contr. Conf.* , 1997, pp.661-664.

[10] T. Kawabe, Robust PID Control Based on Partial Knowledge about Time-Delay Systems, *WSEAS Trans. Systems*, 5(3), 2004, pp.1999-2004.

[11] T. Tagami and T. Kawabe, Genetic Algorithms with General Encoding for Combinatorial Optimization Problems, *Proc. 4th Asian Fuzzy Systems Sympo.*, 2000, pp.284-287.

[12] K. Hirata, Y. Yanase, T. Katayama and T. Kawabe, A Minimax Design of Robust I-PD Controller for Time-Delay Systems with Parametric Uncertainty, *Proc. IFAC 14th World Congress (C)*, 1999, pp.259-264.

[13] M. Fu, A.W. Olbrot and M.P. Polis, Robust Stability for Time-Delay Systems, The Edge Theorem and Graphical Tests. *IEEE Trans. Automat. Contr.,* **34**, 1989, 813-820.

[14] J. Kogan, *Robust Stability and Convexity,* Springer-Verlag, 1995.

[15] J.E. Marshall, H. Górecki, A. Korytowski and K. Walton, *Time-Delay Systems: Stability and Performance Criteria with Applications.*, Ellis Horwood. 1992.

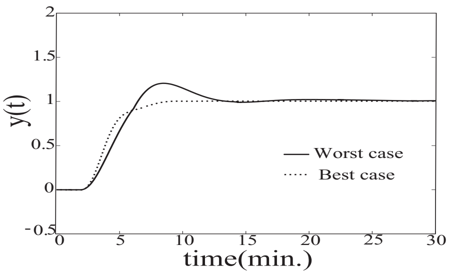


Fig. 6 Step reference responses of closed-loop system

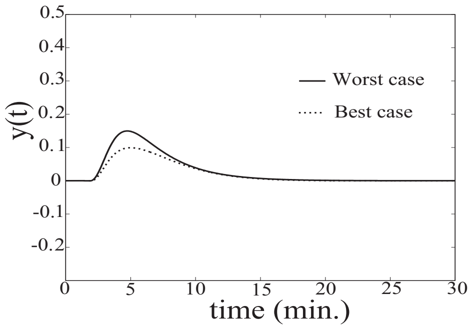


Fig. 7 Step disturbance responses of closed-loop system

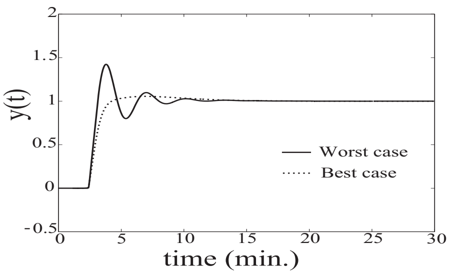


Fig. 8 Step reference responses of closed-loop system

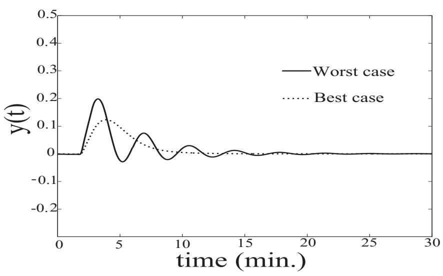


Fig. 9 Step disturbance responses of closed-loop system