

Design of Discrete-Time Control Systems by Multiobjective Genetic Algorithms

TungKuan Liu, Tadashi Ishihara, and Hikaru Inooka
 Graduate School of Information Sciences,
 Tohoku University, Aramaki aza Aoba, Sendai 980-77, Japan

Abstract— In this paper we discuss an application of genetic algorithms (GAs) to discrete-time two-degree-of-freedom control systems design where various design specifications are required. The design method is based on a pole-zero placement considering pole-zero cancellations. The tunable parameters of the controllers are optimized by a new multiobjective GA (MGA). Combining the MGA with the design method we construct a computer aided control system design (CACSD) using MATLAB. The effectiveness of the proposed CACSD is demonstrated by a numerical design example. The MGA is compared with an established optimization algorithm in MATLAB OPTIMIZATION TOOLBOX.

1. INTRODUCTION

Control systems design usually requires satisfaction of multiple design specifications that can not be expressed by a single scalar performance index. As a pioneering work dealing with multiple design specifications systematically in computer aided design, Zakian proposed the method of inequalities [7] where design specifications are given in the form of inequality constraints. Controller parameters satisfying these inequalities are searched by a numerical algorithm. Although the method of inequality has successfully been utilized in various practical control systems, it is difficult in general to guarantee the convergence of parameter search. The convergence difficulty can be overcome by the convex optimization approach proposed by Boyd *et al.* [8]. A major difficulty with this approach is that it generally provides an extremely high order controller that requires simplification for practical controller implementation. Recently, multicriteria problems have been formulated as mixed H^2/H^∞ optimal control problems [5]. Linear matrix inequality (LMI) and bilinear matrix inequality (BMI) have been recognized as useful tools for optimization, but general design method guaranteeing the global optimization has not been obtained. In addition, an optimal controller, if it exists, is extremely high order controller requiring the reduction of order for practical implementation.

In this paper, we discuss a systematic method for directly design a discrete-time controller satisfying multiple design specifications implementable complexity. We first propose a flexible design method of discrete-time two-degree-of-freedom (2DOF) controllers which contain parameters to be directly determined by a numerical search. The proposed method is based on a pole-zero placement with considering pole-zero cancellations. The order of the controller can be

arbitrary larger than the minimum order determined by the steady state characteristics and the number of the pole-zero cancellations both of which are specified by a designer. Although the choice of the controller structure is quite flexible, fixing controller structure inevitably sacrifices the convexity of the search space. It is highly possible that a parameter search with a hill climbing type algorithm is trapped in a local minimum. To overcome this dilemma, we use a GA which has potential to elude local minima. So far several researchers [1][2][3][4] have applied GA to control system designs. However, most of them have considered optimization of a simple controller with a single scalar performance index and potential ability of GA has not been fully utilized. For efficient search for multiple performance indices, we propose a MGA with an improved rank-based fitness assignment method and a modified multiobjective selection. Using the proposed controller structure with tunable parameters and the MGA as a parameter search, we construct a CACSD system using MATLAB. A numerical design example is presented to illustrate the effectiveness of the proposed CACSD system.

2. DESIGN METHOD

For a better engineering design, complexity of controllers have to be limited for implementation, therefore we proposed a design procedure based on a pole-zero placement considering pole-zero cancellations in which a controller is given as a fixed but flexible form. Fig. 1 shows the control system structure set-up. Controller $C_{FB}(z)$ is considered for guaranteeing desired feedback properties such as robustness, disturbance rejection and steady state characteristics. Controller $C_{FF}(z)$ is considered for improving the transient response to the reference input $r(z)$.

At first, we use the pole-zero placement with considering pole-zero cancellations to design $C_{FB}(z)$ by letting $C_{FF}(z) = 0$, then construct a general type controller $C_{FF}(z)$. Suppose we have a plant which is strictly proper and given as

$$G(z) = b_+(z)b_-(z)/a_+(z)a_-(z), \quad (1)$$

where $a_+(z)$, $a_-(z)$, $b_+(z)$ and $b_-(z)$ are polynomials of z

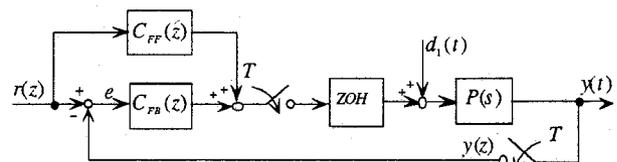


Fig. 1: Discrete-Time Two-Degree-of-Freedom control System

defined as

$a_+(z)$: monic polynomial consists of the poles of the plant that are permitted to be canceled.

$a_-(z)$: monic polynomial consists of the poles of the plant that are not permitted to be canceled.

$b_+(z)$: polynomial consists of the zeros of the plant which are permitted to be canceled and holds no zeros at $z = 1$ and is prime to $a_+(z)a_-(z)$.

$b_-(z)$: polynomial consists of the zeros of the plant which are not permitted to be canceled and hold no zeros at $z = 1$ and is prime to $a_+(z)a_-(z)$.

Now suppose that $C_{FB}(z)$ satisfies the following two conditions.

- 1) The transfer function of $C_{FB}(z)$ is proper, and has l poles specified by a designer at $z = 1$.
- 2) $C_{FB}(z)$ does not cancel the poles of the plant given as zeros of $a_-(z)$ and the zeros of the plant given as zeros of $b_-(z)$.

Under the above conditions, by letting $C_{FF}(z) = 0$, the closed loop transfer function $T(z)$ from $r(z)$ to $y(z)$ is realized as

$$T(z) = g(z)b_-(z)/f(z), \quad (2)$$

where $f(z)$ and $g(z)$ are polynomials satisfying certain conditions given later on. Without loss of generality, the polynomial $f(z)$ is assumed to be monic. A controller realizing (2) is obviously given by

$$C_{FB}(z) = g(z)a_+(z)a_-(z)/(f(z) - g(z)b_-(z))b_+(z). \quad (3)$$

On the other hand, the conditions 1) and 2) require the transfer function $C_{FB}(z)$ to be in the form

$$C_{FB}(z) = g(z)a_+(z)/(z-1)^l d(z)b_+(z), \quad (4)$$

where $d(z)$ is a polynomial. It follows from (3) and (4) that the controller (4) achieves (2) provided the relation

$$f(z) = g(z)b_-(z) + (z-1)^l a_-(z)d(z) \quad (5)$$

holds. Elaborating the above observations, we conclude the following for the pole-zero placement with considering pole-zero cancellations.

Proposition 2.1

Define

$$\begin{aligned} \alpha_+ &\equiv \partial[a_+(z)], & \alpha_- &\equiv \partial[a_-(z)], \\ \beta_+ &\equiv \partial[b_+(z)], & \beta_- &\equiv \partial[b_-(z)], \end{aligned} \quad (6)$$

where $\partial[\cdot]$ denotes a degree of a polynomial.

A) Assume that $\alpha_- + l \geq 1$. Let m be an integer satisfying

$$m \geq \alpha_- + l - 1. \quad (7)$$

Consider a polynomial $g_p(z)$ and a stable polynomial $f(z)$ satisfying the conditions

$$n = \partial[f(z)] \geq m + \alpha_+ + \alpha_- - \beta_+, \quad (8)$$

$$g_p(z) = \begin{cases} g_m z^m + g_{m-1} z^{m-1} + \dots + g_{\alpha_+} z^{\alpha_+}, & (\text{if } m > \alpha_- + l - 1), \\ 0, & (\text{if } m = \alpha_- + l - 1). \end{cases} \quad (9)$$

Then the Diophantine equation

$$f_p(z) = (z-1)^l a_-(z)d(z) + g_p(z)b_-(z) \quad (10)$$

has unique solutions $g_p(z)$ and $d(z)$ that satisfies

$$\partial[d(z)] = n - \alpha_- - l, \quad (11)$$

$$\partial[g_p(z)] \leq \alpha_- + l - 1, \quad (12)$$

where $f_p(z)$ is a polynomial defined by

$$f_p(z) \equiv f(z) - g_p(z)b_-(z). \quad (13)$$

Define

$$g(z) \equiv g_p(z) + g_r(z). \quad (14)$$

Then $\partial[g(z)] = m$ and the controller $C_{FB}(z)$ defined by (4) using $g(z)$ and $d(z)$ is proper and achieves the closed-loop transfer function $T(z)$ given by (2).

B) Assume that $\alpha_- = 0$ and $l = 0$. Consider an arbitrary polynomial $g(z)$ and a stable polynomial $f(z)$ satisfying

$$m \equiv \partial[g(z)], \quad (15)$$

$$n \equiv \partial[f(z)] \geq \alpha_+ + m - \beta_+. \quad (16)$$

From (5), one has

$$d(z) = f(z) - b_-(z)g(z). \quad (17)$$

Then the controller $C_{FB}(z)$ defined by (4) using $g(z)$ and $d(z)$ is proper and achieves the closed-loop transfer function $T(z)$ given by (2).

Proof: We give a proof only for the case A) since a proof for the case B) is easier. Because the transfer function of plant $G(z)$ is a strictly proper rational, we have the relation $\alpha_+ + \alpha_- > \beta_+ + \beta_-$. Combining this relation with (8), we can see that

$$n > m + \beta_-. \quad (18)$$

Since the degree of $g_p(z)$ is at most m we obtain

$$\partial[f_p(z)] = \partial[f(z)] = n \quad (19)$$

from (13) and (18). Besides, from (7) and (18), we have

$$\begin{aligned} \partial[f_p(z)] &= n \geq m + \beta_- + 1 \geq \alpha_- + l + \beta_- \\ &= \partial[(z-1)^l a_-(z)] + \partial[b_-(z)]. \end{aligned} \quad (20)$$

It follows from a well-known result on the Diophantine equation [14] that (10) have a unique solution pair $\{g_p(z), d(z)\}$ which satisfies (11) and (12) since we suppose that $(z-1)^l a_-(z)$ is prime to $b_-(z)$.

Using (7), (9), (12) and (14) we obtain

$$\partial[g(z)] = m. \quad (21)$$

In addition, from (10), (13) and (14) we see that the pair $\{g_p(z), d(z)\}$ satisfies (5). Therefore, the controller (4) achieve the desired closed-loop transfer function (2).

The controller transfer function (4) is proper if

$$l + \partial[d(z)] + \beta_+ \geq m + \alpha_+. \quad (22)$$

Using (11), we can rewrite (22) as

$$n - \alpha_- + \beta_+ \geq m + \alpha_+, \quad (23)$$

which is equivalent to (8), so that the transfer function of the controller (4) is proper. ■

The controller $C_{FF}(z)$ is designed to obtain the desired response to the reference input, given as follows

$$C_{FF}(z) = a_0 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_n z^{-n}, \quad (24)$$

where $\{a_0, a_1, a_2, \dots, a_n\}$ are parameters decided by the designer. Note that the transfer function (24) is stable for arbitrary choice of the coefficients $\{a_0, a_1, a_2, \dots, a_n\}$ since it is FIR type.

Introducing $C_{FF}(z)$ into the control system offers us the

following two advantages:

1) From Fig. 1, the transfer function of disturbance $d_1(t)$ and output $y(z)$ is given by

$$G_{yd}(z) = y(z)/d_1(z) = G(z)/(1 + C_{FB}(z)G(z)) \quad (25)$$

which depends only on $C_{FB}(z)$. The steady state characteristics of the response to the reference input and to the disturbance can be guaranteed by specifying the number of integrators included in $C_{FB}(z)$. Moreover, since the characteristic equation of the system includes $C_{FB}(z)$ only, the robust stability of the system can be achieved by a judicious design of $C_{FB}(z)$.

2) The transfer function from $r(z)$ to $y(z)$ is given as

$$G_{yr}(z) = y(z)/r(z) = G(z)(C_{FB}(z) + C_{FF}(z))/(1 + C_{FB}(z)G(z)). \quad (26)$$

Since the characteristic equation of the system is independent of $C_{FF}(z)$, $C_{FF}(z)$ can be designed for improving the transient response to the reference input $r(z)$ without effecting the steady state characteristics obtained by $C_{FB}(z)$.

3. MULTIPLE CRITERIA PROBLEM

In this section we will discuss the formulation of the multiobjective optimization problem. Assume that the performances of a control system are evaluated by various scalar performance indices $\phi_i(\mathbf{p})$ ($i=1,2,\dots,M$), where \mathbf{p} is a tunable parameter vector. Vector performance index can be defined as

$$\Phi(\mathbf{p}) \equiv [\phi_1(\mathbf{p}) \ \phi_2(\mathbf{p}) \ \dots \ \phi_M(\mathbf{p})]'. \quad (27)$$

For performance indices of practical interest, it is hard to characterize the exact Pareto optimal set since $\phi_i(\mathbf{p})$ is usually a highly non-linear function of \mathbf{p} . Even if we apply MGA directly to the vector performance index (27), it would require fine quantization of the parameter space and thus enormous amount of computations are necessary to obtain a "Quasi-Pareto-Optimal Set" (QPOS) sufficiently close to the exact Pareto optimal set.

As initiated by Zakian [7], to deal with multiobjective performance indices, a reasonable practical approach is to give design specifications in the form of inequalities $\phi_i(\mathbf{p}) \leq \varepsilon_i$ ($i=1,2,\dots,M$), where ε_i is an admissible performance bound specified by a designer. Assuming that this type of design specifications are given, we apply MGA for the vector performance index which is defined as

$$\Psi(\mathbf{p}) \equiv [\psi_1(\mathbf{p}) \ \psi_2(\mathbf{p}) \ \dots \ \psi_M(\mathbf{p})]', \quad (28)$$

where

$$\psi_i(\mathbf{p}) = \begin{cases} 0 & (\text{if } \phi_i(\mathbf{p}) \leq \varepsilon_i) \\ \phi_i(\mathbf{p}) - \varepsilon_i & (\text{if } \phi_i(\mathbf{p}) > \varepsilon_i) \end{cases} \quad i = 1, 2, \dots, M. \quad (29)$$

Since the Pareto optimal set for the vector performance index (28) is apparently larger than that for (27), the multiobjective GA can effectively be used for (28) with reasonable computational burden.

Remark 1: The optimization based on the vector performance index (28) is closely related to the method of inequalities proposed by Zakian [7]. If the optimal solution for the vector

performance index (28) is given by $\Psi(\mathbf{p}_0) = 0$, a controller corresponding to \mathbf{p}_0 (not necessarily unique) could be found by the method of inequalities using "Moving Boundary Process" (MBP) [9] with an appropriate initial conditions. Since the existence of a controller satisfying all the design specifications $\phi_i(\mathbf{p}) \leq \varepsilon_i$ ($i=1,2,\dots,M$) can not be guaranteed *a priori*, the parameter search in the method of inequalities often fails to converge. The MGA for the performance index (28) overcome this difficulty in the sense that it always generates controllers belonging to QPOS, even if no controller satisfies the given design specifications $\phi_i(\mathbf{p}) \leq \varepsilon_i$ ($i=1,2,\dots,M$). The performance indices for each controller in the Pareto optimal set provide useful information to the designer for modifying the design specifications.

Remark 2: The intention of optimizing the vector performance index (28) is similar to that of using the *goal attainment* method, an optimization procedure of MATLAB, which is commonly used for obtaining a compromising solution of a multiobjective optimization problem. However, notice that the multiobjective GA using the vector performance index (28) seeks a set of Pareto optimal controllers, while the *goal attainment* method provides a single controller possibly dependent on the initial conditions of the parameter search.

Remark 3: In the vector performance index (28), the M scalar performance indices are equally weighted. It is possible to reflect relative importance of the component indices in a vector performance index by appropriate scaling.

4. MULTIOBJECTIVE OPTIMIZATION BY GENETIC ALGORITHM

As an optimization approach to the multiple criteria problem, we use a new MGA. Two main parts, the improved rank-based fitness assignment and the modified multiobjective selection, are included.

4.1 Improved Rank-Based Fitness Assignment

In Pareto optimization, a set $\mathbf{F} = \{\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_n\}$, where $\mathbf{f}_i = \{f_{i1}, f_{i2}, f_{i3}, \dots, f_{im}\}$ is a vector, \mathbf{f}_i is said to be partially less than \mathbf{f}_k , when the following condition holds:

$$(\mathbf{f}_i <_p \mathbf{f}_k) \Leftrightarrow (\forall j)(f_{ij} \leq f_{kj}) (\exists j)(f_{ij} < f_{kj}). \quad (30)$$

This relation is used in MGAs to determine the fitness of individuals. However, in previously suggested method, when the population size is large, the comparison for determining partially less than relations requires heavy computational efforts. For a better way to calculate the fitness, we consider a simple way to decide ranks of individuals. Recalling Section 3, our present vector performance index (28) whose element shown in (29) is nonnegative. Therefore, the relationship (30) is satisfied only if

$$\sum_{j=1}^M f_{ij} \leq \sum_{j=1}^M f_{kj} \quad (31)$$

holds. Using this characteristic we propose an improved rank-based fitness assignment to save the computing time and show it as follows.

Step1: Sort \mathbf{F} from the least to the largest according to

summations of $\sum_{j=1}^m f_{ij}$.

Step 2: Let the first element f_1 be the criterion. The inferior or dominated members f_i are detected if

$$\max_j g_{ij} \leq 0 \wedge (\exists i)(g_{ij} < 0), \quad (32)$$

where g_{ij} are components of $g_i = f_1 - f_i, i=2,3,\dots,n, j=1,2,3,\dots,m$.

Step 3: Take out f_1 and the inferior members f_i from F . Save f_1 to a temporary non-dominated set Q , f_i to a temporary dominated set G . Let the remaining vectors of F be \tilde{F} .

Step 4: Treat \tilde{F} by repeating **Step 2** and **Step 3** until all dominated set of F are removed.

Here, the non-dominated vectors of \tilde{F} and the temporary non-dominated set Q become the non-dominated set of F .

Step 5: The members in non-dominated set are assigned rank of 1. These points are then removed from contention and the next set of non-dominated individuals of G gained from **Step 2** to **Step 4** are identified and assigned as rank 2.

Step 6: All the newly ranked individuals in Step 5 are classified as the same partial population. Continue **Step 2** to **Step 5** until the entire population is ranked.

Step 7: The fitness is given by

$$F(i) = 2(\max - 1)(N - \text{rank}(i))/(N - 1), \quad (33)$$

where $\text{rank}(i)$ is the rank of the individual, \max is a user defined constant satisfying $1 < \max \leq 2$, and N is the population size.

Remark 4: In previously suggested multiobjective GAs [6], to determine the rank-based fitness of a set with n vectors requires the *partially less than relation* to be compared $\sum_{i=1}^{n-1} (n-i)$ times; here i means the i th comparison. However, in our study, we firstly sort all individuals from the least to the largest according to their summations, then choose the first individual as a criterion. Because this criterion is not dominated by any opponents, it detects most dominated individuals and removes them before next contention. Consequently, the i th comparison is less than or equal to $(n-i)$ iterations. This causes the fastest convergence of the comparison iterations.

4.2 Modified Multiobjective Selection

Now we focus on the multiobjective selection deciding which individuals should be remained and which ones should be removed during the calculations by GA. This is a difficult problem for the multicriteria optimization. Many researchers have paid their attentions to this problem, e.g., CHC [11] proposed by Eshelman and forking GA (fGA) [12] proposed by Tsutsui *et al.*. However, the projections from the search space to the evaluation space are not unique. Even if they keep different binary strings in parameter space, the objective population diversity cannot be assured. In addition, the

threshold constants used in fGA are detectors on the *genetic domain*. It needs enormous memory and time to calculate. Therefore, in this research we propose a modified multiobjective selection scheme base on a *objective domain* threshold constant (ODTC). This scheme use two alternative selection methods based on ODTC.

The concept of this method is shown in Fig. 2, "SubP(Rank i)" means the sub-populations consist of the individuals gained from the **Step 6** of the improved rank-based fitness assignment stated in 4.1. The selection methods "select_v" and "select_h" are under the conditions whether the ODTC is reached or not. Pop , $C(t)$ and N are the population size, the child population and the total number of ranks, respectively. The value of ODTC given as V_0/h is calculated by a *diversity maintaining parameter* h and the initial population diversity V_0 . When the threshold constant is reached, we use $select_v$ to choose the number $Pop(N+1 - \text{Rank}_i) / \sum_{i=1}^N \text{Rank}_i$ of children from every sub-populations in proportion to their ranks into next generation. Otherwise, we simply select the best N individuals.

Combining this multiobjective selection method with the improved rank-based fitness assignment mentioned in 4.1, we construct a modified MGA shown as follows.

Step 1: Initialize GA population $P(t)$. Decide the diversity maintaining parameter h . Let $t = 0$.

Step 2: Evaluate individuals in $P(t)$ and determine their fitness by the improved rank-based fitness assignment. If the termination condition is not satisfied go to **Step 3**, else Stop.

Step 3: Detect the diversity V_t of $P(t)$. Let the initial diversity at $t = 0$ be V_0 .

Step 4: Apply GA operators, two-point crossover and mutation, to $P(t)$. Put the new individuals into $P'(t)$.

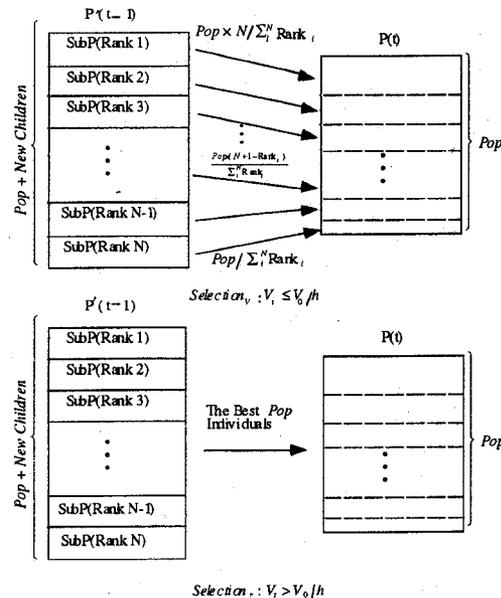


Fig. 2: Modified Multiobjective Selection

Step 5: Evaluate individuals in $P'(t)$ and determine the fitness of $P(t) + P'(t)$ by the improved rank-based fitness assignment.

Step 6: Detect the diversity V_t of the whole population.

Step 7: If $V_t \leq V_o/h$ *select_t* is used to select $C(t)$ from $P(t) + P'(t)$, else *select_t* is used to select $C(t)$ from $P(t) + P'(t)$. Let $t = t + 1$ and $P(t) = C(t-1)$. If termination condition is not satisfied go to **Step 4**, else Stop.

5. DESIGN EXAMPLE

In this section, we present a design example. We apply our design method to the control of a mobile robot. The transfer function of the motor and power amplifier is given as

$$P(s) = 10/s(s+10)(s+20). \quad (34)$$

The control system structure is shown in Fig. 1. For discrete-time controllers, we convert the continuous time plant transfer function $P(s)$ into a discrete-time form using z transform with a zero-order holding device and the sampling time period $T = 0.025$. The discrete-time transfer function is shown as

$$G(z) = \frac{2.1681 \times 10^{-5}(z+3.1095)(z+0.2211)}{(z-1)(z-0.7788)(z-0.6065)}. \quad (35)$$

The system specifications are set according to the number of performance requirements. Six objective are used

- Settling time: $t_s \leq 0.75s$.
- Overshoot: $O_s \leq 10\%$.
- Maximum value of controller output: $U_c \leq 200$.
- Maximum value of $y(t)$ response against a step disturbance input: $D_y \leq 0.5$.
- Maximum value of controller output against a step disturbance input: $U_d \leq 200$.

These five system specifications are set into a vector performance index in the form given in (28). Using the MGA stated in 4.2, GA parameters are chosen as follows: 1) modified multiobjective selection (Let $h=3$), 2) two-point crossover and 3) mutation. Population 100, reproduction rate 0.8, crossover probability 1.0 and mutation probability 0.001 are used. Furthermore, to avoid the pole-zero cancellations of $G(z)$, we define the polynomials included in the plant transfer function (1) as follows

$$\begin{aligned} a_+(z) &= 1 \\ a_-(z) &= (z-1)(z-0.7788)(z-0.6065) \\ b_+(z) &= 1 \\ b_-(z) &= 2.1681 \times 10^{-5}(z+3.1095)(z+0.2211). \end{aligned} \quad (36)$$

In addition, for considering disturbance rejection, we specify one integrator in controller $C_{FB}(z)$. From (7), we have inequality $m \geq 3$. Let $m = 3$ and substitute it into (8), then the inequality $n \geq 6$ should be satisfied. Let $n = 6$ and specify the order of the controller $C_{FF}(z)$ to be 6. In this case, the tuning parameters of $C_{FB}(z)$ include six roots of $f(z)$ represented by $r_1 e^{\pm j\theta_1}$, $r_2 e^{\pm j\theta_2}$ and $r_3 e^{\pm j\theta_3}$ in polar coordinate only, having no coefficient of $g_p(z)$. The tuning

parameters of $C_{FF}(z)$ are $\{a_0, a_1, a_2, a_3, a_4, a_5, a_6\}$. Here, we set $a_0 = 0$. Thus twelve parameters are needed to be tuned for the optimal controllers. For GA calculating we discretize each radius(r_i), angle(θ_i) and the coefficients a_i by mapping from a smallest possible value to a largest possible value. This mapping used a 10-bit binary unsigned integer. In this coding, a string code 0000000000 maps to the smallest possible value and a code 1111111111 maps to the largest possible value with a linear mapping in between. Next, the six 10-bit parameter sets of r_i and θ_i are chained together to form a 60-bit string represents one of the $2^{60} = 1.1529 \times 10^{18}$ alternative solutions for controller $C_{FB}(z)$. The coefficients a_i of $C_{FF}(z)$ is set by the same way.

By the procedural flow using GA in Section 4.2, we obtained the controllers $C_{FB}^{GA}(z)$ and $C_{FF}^{GA}(z)$ which achieve all the system specifications after 53 and 30 generations, respectively.

$$C_{FB}^{GA}(z) = 1.7869 \times 10^2 (z - 0.9844)(z - 0.9091) / (z - 0.5927)(z - 1)(z - 0.7626 \pm j0.2396) \quad (37)$$

$$C_{FF}^{GA}(z) = -1.9795z^{-1} + 4.0225z^{-2} - 10.0929z^{-3} - 8.5288z^{-4} + 10.5327z^{-5} - 18.4995z^{-6} \quad (38)$$

Furthermore, to compare MGA with an established optimization algorithm, the goal attainment method, attgoal, in MATLAB OPTIMIZATION TOOLBOX (OPT) is used to design controllers for the same plant with the same specifications and design condition. The initial values of $f(z)$ are given as a 6th-order Butterworth pattern whose radius is 1 and the initial coefficients of $C_{FF}(z)$ are given as 0's. This search terminates after 4949 iterations for $C_{FB}^{OPT}(z)$ and 2120 iterations for $C_{FF}^{OPT}(z)$, respectively, without satisfying all the system specifications.

$$C_{FB}^{OPT}(z) = 1.0352(z - 0.9488 \pm j0.1439)(z - 0.8687 \pm j0.0388) / (z - 1)(z - 0.9229)(z - 0.5437)((z - 0.8259 \pm j0.2940)) \quad (39)$$

$$C_{FF}^{OPT}(z) = 1.0897z^{-1} + 0.7256z^{-2} + 0.4541z^{-3} + 0.1342z^{-4} - 0.5886z^{-5} - 1.4566z^{-6} \quad (40)$$

The system performances obtained by MGA and OPT are shown in Tab. 1. Figs. A1 to A4 show the step responses, the controller outputs, responses of $y(t)$ and $u(t)$ when the

Table 1: Performances of Examples

	$C_{FB}^{GA}(z)$	$C_{FB}^{OPT}(z)$	$C_{FF}^{GA}(z)$	$C_{FF}^{OPT}(z)$
Rise Time	0.2250	0.2000	0.8250	2.6000
Overshoot	9.9545	4.9821	13.9422	0.4727
Setting Time	3.1000	0.7250	4.8250	3.3750
Max. Controller Output	184.9863	194.3785	199.6426	199.6426
Gain Margin (dB)	9.6861	9.6861	13.8397	13.8397
Phase Margin (deg)	83.7208	83.7208	82.9090	82.9090
Crossover Frequency (rad/s)	3.9984	3.9984	1.8199	1.8199
Peak Value (to Disturbance)	0.0117	0.0117	0.0202	0.0202
Max. Controller Output (to Disturbance)	1.0995	1.0995	1.1394	1.1394
Execute Iterations	4340	2500	4949	2120

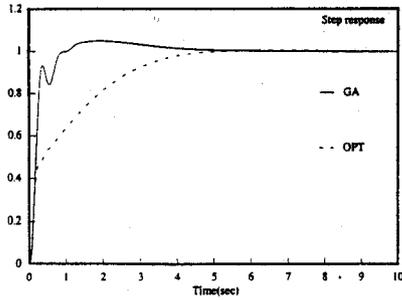


Fig. A1: Step response of $y(t)$ to $r(t)$

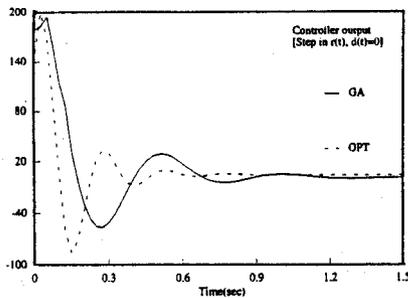


Fig. A2: Controller output

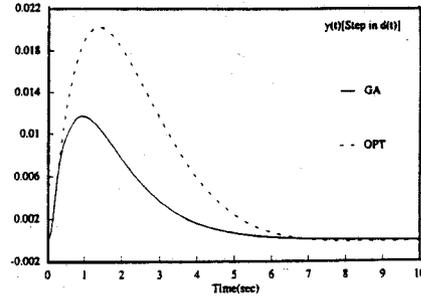


Fig. A3: Step response of $y(t)$ to $d(t)$

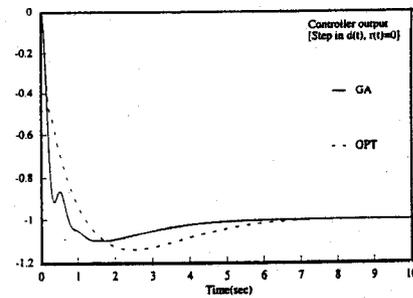


Fig. A4: Controller output to disturbance

system suffers a step disturbance $d_1(t)$, respectively. From Tab. 1 we see that MGA calculates in less times as compared with OPT (6840 and 7069, respectively). However, MGA satisfies all specifications. On the other hand, OPT gains an excellent overshoot characteristic but violates the rise-time and settling-time constraints. Furthermore, from Fig. A3 we find that the maximum value of the step responses against a step disturbance is 0.0117 of MGA and is 0.0202 of OPT. It can be said that in this example MGA proposes both better transient response and disturbance rejection characteristics than OPT.

6. CONCLUSIONS

In this paper, we have discussed a systematic approach for directly designing discrete-time 2DOF controllers using multiobjective genetic algorithm. The proposed method based on the pole-zero placement with considering pole-zero cancellations designs controllers with simple and flexible structure. For efficient parameter search, we have proposed a new multiobjective GA (MGA) with an improved rank-based assignment method and a modified multiobjective selection scheme to elude local minima. Combining the design method with the MGA we construct a computer aided control systems design system using MATLAB. We compare the MGA with an established optimization algorithm, the goal attainment method, attgoal, in MATLAB OPTIMIZATION TOOLBOX by a numerical design example. The result shows that comparing with OPT, MGA is highly possible to find the optimization solution.

References

- [1] K. Krishnakumar and D. E. Goldberg, Control System Optimization Using Genetic Algorithms, *Journal of Guidance, Control and Dynamics*, Vol. 15 No. 3, pp.735-740, 1992
- [2] K. Kristinsson and G. A. Dumont, System Identification and Control using Genetic Algorithms, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 22, No. 5, pp. 1033-1046, 1992
- [3] B. Porter and A. H. Jones, Genetic Tuning of Digital PID Controllers, *Electronics Letters*, Vol. 22 No. 9, pp. 834-844, 1992
- [4] A. Varsek, T. Urbinc and B. Filipic, Genetic algorithms in Controller Design and Tuning, *IEEE Transactions On Systems, Man and Cybernetics*, Vol. 23, No 5, pp. 1330-1339, 1993
- [5] P. P. Khargoneker and M. A. Rotea, Multiple Objective Optimal Control of Linear system: The Quadratic Norm Case, *IEEE Transactions on Automatic Control*, Vol. 36, No. 1, pp. 14-24, 1991
- [6] C. M. Fonseca and P. J. Fleming, Genetic algorithms for multiobjective optimization, Formulation, discussion and generalization, *Proc. of the Fifth International Conference*, pp. 416-423, 1993
- [7] V. Zakian and U. Al-Naib, Design of Dynamical and Control System by the Method of Inequalities, *Proc. IEE*, 120-11, pp. 1421-1427, 1973
- [8] S. Boyd, *et al.*, A New CAD Method and Associated Architectures for Linear Controllers, *IEEE Trans.*, AC-33-3, pp. 268-283, 1988
- [9] T. Satoh, T. Ishihara and H. Inooka, Computer-Aided Control System Design Accounting Pole-Zero Cancellations by the Method of Inequalities, *Proc. IEEE/IFAC Joint symposium on Control System Design*, pp. 481-488, 1994
- [10] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learn.* Reading, MA, Addison-Wisely, 1989
- [11] L. J. Eshelman, The CHC Adaptive Search algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination, *Foundations of Genetic Algorithms* edited by Gregory J.E. Rawlins, Morgan Koufmann, pp. 265-283,1991
- [12] S. Tsutsui and Y. Fujimoto, Forking Genetic Algorithm with Blocking and Shrinking Modes (fGA), *Proc. of the Fifth ICGA* pp. 206-213, XII +665, 1993
- [13] *Optimization Toolbox User's Guide*, The Math Works Inc. (1990)
- [14] T. Kailath, *Linear systems*, Englewood Cliffs, NJ: Prentice-Hall, 1984