

Multiobjective Control Systems Design by Genetic Algorithms

Tung Kuan Liu, Tadashi Ishihara and Hikaru Inooka

Graduate School of Information Sciences,
Tohoku University, Aramaki aza Aoba, Sendai 980-77, Japan

Abstract: For Multiobjective Control Systems Design, we use Genetic Algorithms to find the Pareto Optimal set of various control system performance indices. We also propose a modified multiobjective selection scheme and the use of the improved rank-based fitness assignment. By combining multiobjective genetic algorithm (MGA) with the pole-zero placement algorithm which can avoid specified pole-zero cancellations, we construct a MATLAB based software package for the computer-aided control system design (CACSD) system for two degree-of-freedom discrete-time control systems. This CACSD system provides a large freedom in the choice of controller structure and in the design specifications. Effectiveness of the proposed CACSD system is illustrated by a design example where the multiobjective optimization by GA is compared with a goal attainment method in MATLAB OPTIMIZATION TOOLBOX.

Keywords: Multiobjective Genetic Algorithms, Two Degree-of-Freedom Control System, Pole-Zero Cancellation, Rank-Based Fitness Assignment, Pareto Optimal Set, Modified Multiobjective Selection

1. Introduction

Applications of genetic algorithms (GAs) to optimization-based control systems design have been proposed to optimize a single scalar performance for which standard hill climbing techniques fail to find the global minimum [1][2][3][4]. Recently, numbers of researches *e.g.*, [5][6] aimed to design control systems with various specifications by multiobjective genetic algorithms.

In these researches, the rank-based fitness assignment methods are used to identify the fitness of individuals according to the partially less than relations. Furthermore, in [5] the decision maker (DM) is posed to refine the design specifications during GAs run. They show the effectiveness of MGAs. However, the population diversity decreases quickly, especially while MGA optimizes a multiobjective problem with a vector index consisting of a large number of objectives. When all the individuals converge into a single fitness, GA search acts like a random search. Recently, the method to maintain diversity during GAs run have stated by Eshelman [12] and Tsutsui *et al.* [13]. However, the objective function used in these researches are scalar, they are inadequate to handle more realistic designs which usually require the evaluation of multiple performance indices.

To make multiobjective control systems design feasible, Zakian [7] introduced the method of inequalities, in which design specifications are given in the form of inequality constraints on multiple scalar performance indices. A design is satisfactory if all the inequalities are satisfied. Zakian used a relatively simple controller with a simple parameter search algorithm called the moving boundary process (MBP) [8]. Using the method of inequality, Satoh *et al.* [9] have recently constructed a CACSD system which provides a unity feedback controller with the order specified by a designer. This CACSD system is based on a pole-zero placement algorithm that can avoid pole-zero cancellations

specified by a designer. However, this CACSD system is for continuous time systems only. Many modern industrial control systems are discrete-time systems since they invariably include some elements whose inputs and/or outputs are discrete in time. Of course, one can design a continuous time controller in advance, then employ z transform to obtain the discrete-time controllers. It should be careful that the system specifications satisfied by controllers of continuous type are not necessary satisfied by those of discrete type.

In this paper, we proposed a computer-aided design method for two degree-of-freedom control systems considering the cancellations between plants and controllers. For efficient use of MGA, we introduce a modified multiobjective selection. The CACSD we propose here combines multiobjective GA [6] with the pole-placement algorithm proposed by Usami [14]. To ensure wide utilization, we implement the CACSD system as a toolbox for MATLAB together with MATLAB CONTROL SYSTEM TOOLBOX.

This paper is organized as follows. In section 2, the pole-zero algorithm used in a discrete-time two degree-of-freedom control system is described. In section 3, the improved rank-based fitness assignment method and the modified multiobjective selection used in MGA are stated. In section 4, the effectiveness of the proposed CACSD system is exemplified by its application to a control design problem by comparing with an algorithm in MATLAB OPTIMIZATION TOOLBOX.

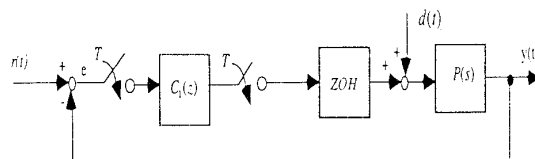


Fig. 1: Discrete-time unity feedback system

2. Discrete-Time Control System Structure

In this section, we consider discrete-time systems where the signal representing the control effort is piecewise constant and changes only at discrete points in time.

2.1 Control System Structure Considering Pole-zero Cancellations

Figure 1 shows the configuration of SISO unity feedback system for which we propose the design method that can avoid specified pole-zero cancellations. Using this method, one can guarantee the steady state characteristics of the system by specifying the number of integrators included in the controller for a general type control system.

For constructing a discrete-time control system, we convert the continuous time plant $P(s)$ into the discrete plant $G(z)$ by z transform with sampling instant T .

Suppose that $G(z)$ is strictly proper and given by

$$G(z) = \frac{b_+(z)b_-(z)}{a_+(z)a_-(z)} \quad (1)$$

where $a_+(z)$, $a_-(z)$, $b_+(z)$, and $b_-(z)$ are polynomials of z , and defined as follows.

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

Now suppose that the controller in the control system satisfies the two conditions below.

- 1) The transfer function of the controller $C_1(z)$ is proper, and has l poles specified by a designer at $z = 1$.
- 2) The transfer function of the controller $C_1(z)$ does not cancel the poles of the plant which are given as zeros of $a_-(z)$, and also does not cancel the zeros of the plant which are given as zeros of $b_-(z)$.

Under the above conditions, we consider a pole-zero placement procedure by which the closed loop transfer function $T(z)$ from $r(t)$ to $y(t)$ is realized as

$$T(z) = \frac{g(z)b_-(z)}{f(z)}, \quad (2)$$

where $f(z)$ and $g(z)$ are polynomials which satisfy certain conditions given later on. Without loss of generality, the

polynomial $f(z)$ is assumed to be monic. It will be shown that the zeros of $f(z)$ and a part of the coefficients of $g(z)$ are freely specified by a designer. The transfer function of the controller realizing the closed loop transfer function (2) is obviously given by

$$C_1(z) = \frac{g(z)a_+(z)a_-(z)}{(f(z) - g(z)b_-(z))b_+(z)}. \quad (3)$$

On the other hand, the above-mentioned conditions 1) and 2) require the transfer function $C_1(z)$ to be in the form

$$C_1(z) = \frac{g(z)a_+(z)}{(z-1)^l d(z)b_+(z)}, \quad (4)$$

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

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

holds. Elaborating the above observation, we have the following result for the pole-zero placement.

Proposition 3.1

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

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

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

Then the Diophantine equation

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

has unique solutions $g_r(z)$ and $d(z)$ that satisfy

$$\begin{aligned} \partial[d(z)] &= n - \alpha_- - l \\ \partial[g_r(z)] &\leq \alpha_- + l - 1. \end{aligned} \quad (11)$$

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

$$f_p(z) = f(z) - g_p(z)b_-(z). \quad (12)$$

Define

$$g(z) = g_p(z) + g_r(z). \quad (13)$$

Then $\partial[g(z)] = m$ and the controller $C_1(z)$ 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

$$n = \partial[f(z)] \geq \alpha_+ + \partial[g(z)] - \beta_+. \quad (14)$$

Define

$$d(z) = f(z) - b_-(z)g(z). \quad (15)$$

Then the controller $C_1(z)$ defined by (4) using $g(z)$ and $d(z)$ is proper and achieves the closed-loop transfer function $T(z)$ given by (2).

Proof: Omitted, see [14].

2.2 Two Degree-of-Freedom Control System

For designing a more efficient control system, we add the following controller $C_2(z)$ to Fig. 1 and restructure the system as Fig. 2.

$$C_2(z) = a_0 + \frac{a_1}{z} + \frac{a_2}{z^2} + \dots + \frac{a_n}{z^n}, \quad (16)$$

where $\{a_0, a_1, a_2, \dots, a_n\}$ are the parameters decided by the designer. Introducing $C_2(z)$ into the control system offers us the following two advantages:

1) Since the open loop transfer function shown in Fig. 2 is given by

$$H(z) = C_1(z)G(z), \quad (17)$$

it does not be effected by $C_2(z)$. Consequently, one can design the controller $C_2(z)$ to obtain desired response of the function from $r(t)$ to $y(t)$ without effecting the open loop characteristics.

2) From Fig. 2, the transfer functions from disturbance $d(t)$ to output $y(t)$ is presented by

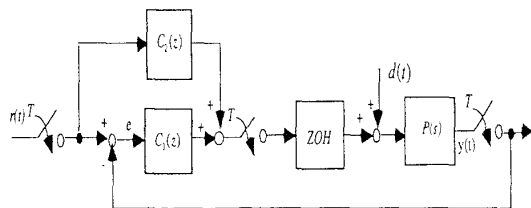


Fig. 2: Two degree-of-freedom discrete-time system

$$F(z) = \frac{G(z)}{1 + C_1(z)G(z)}, \quad (18)$$

and is effected by $C_1(z)$ only. The robustness of the control system can be guaranteed by a good design of $C_1(z)$ independently.

Therefore, with the advantages mentioned above, one can design the controllers $C_1(z)$ and $C_2(z)$ to guarantee the steady state and robust characteristics and to optimize the response of the function from $r(t)$ to $y(t)$ independently by the following two steps.

Step 1: First, design the parameters included in $C_1(z)$ to guarantee the steady state and robust characteristics of systems.

Step 2: Secondly, tuning the parameters included in $C_2(z)$ to optimize the response of the function from $r(t)$ to $y(t)$.

3. Multiobjective Optimization Problems

For a control system with various specifications, the scalar evaluating functions are no more useful, however, fitness defining methods, such as Pareto-based ranking approach of Goldberg [11] and the modified MO ranking scheme [5], are used in GAs. These approaches suffer from too many comparing iterations in deciding the ranks of each individual. To reduce the executing time for the Pareto-based ranking process, the authors have proposed an improved algorithm [6], called improved rank-based fitness assignment. Furthermore, to prevent premature we introduce the modified multiobjective selection to maintain the diversity during GA runs.

3.1 Improved Rank-Based Fitness Assignment

First, we briefly review the improved rank-based fitness of Liu *et al.* [6]. For a set of vectors $\mathbf{F} = \{\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_n\}$, where $\mathbf{f}_i = \{f_{i1}, f_{i2}, f_{i3}, \dots, f_{im}\}$, \mathbf{f}_i is said to be partially less than \mathbf{f}_k , when the following conditions hold:

$$(\mathbf{f}_i < \mathbf{f}_k) \Leftrightarrow (\forall_j)(f_{ij} \leq f_{kj}) (\exists_j)(f_{ij} < f_{kj}). \quad (19)$$

By modifying (19) authors have proposed a less computationally demanding ranking scheme [6] shown as follows

Step 1: Sort \mathbf{F} from the least to the largest according to

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

Step 2: Let the first element \mathbf{f}_1 be the criterion and the inferior or dominated members \mathbf{f}_i are determined by following equation

$$\max_j g_{ij} \leq 0 \wedge (\exists i)(g_{ij} < 0) \quad (20)$$

where $g_i = f_i - f_j$ and with the components g_{ij} , $i=2,3,\dots,n, j=1,2,3,\dots,m$.

- Step 3: Take out f_i and the inferior members f_j from F and save f_i to a temporary non-dominated set Q , f_j 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 of the dominated set of F to be 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 ranks 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: Continue Step 2 to Step 5 until the entire population is ranked.

For the modified multiobjective selection scheme maintained later on, all the newly ranked individuals in Step 5 are classified as the same partial population. From these partial populations we perform the modified multiobjective selection to choose individuals in proportion to their ranks. Consequently, the diversity of current population can be maintained for the next generation.

After the ranking procedure and the modified multiobjective section, the rank-based fitness function for GA implementation can be obtained from a linear function

$$V_n(i) = \frac{2(max - 1)}{N - 1}(N - rank(i)), \quad (21)$$

where $rank(i)$ is ranked individual, max is a user defined value, $1 < max \leq 2$, and N is the population size.

3.2 Modified Multiobjective Selection

The method to prevent premature during GAs run have stated by Eshelman [12] and Tsutsui *et al.* [13]. However, the objective function used in these researches are scalar, they are inadequate to handle more realistic designs which usually require the evaluation of multiple performance indices. In the design of a multiobjective control system with multiple specifications we introduce a modified multiobjective selection. To maintain population diversity during MGA runs, we used the following outline of GA

- Step 1: Initialize GA population $P(t)$. Here, let $t=0$.
- Step 2: Evaluate individuals in $P(t)$ and determine their fitness by the improved rank-based fitness assignment. While termination condition is not satisfied go to Step 3. 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 $P'(t)$.
- Step 5: Evaluate individuals in $P'(t)$ and determine the fitness of $P(t)+P'(t)$ by the improved rank-based

fitness assignment.

- Step 5: Detect the diversity V_t of the whole population.
- Step 6: If $V_t \leq \frac{1}{h} V_0$ $select_v$ $C(t)$ from $P(t)+P'(t)$ else $select_v$ $C(t)$ from $P(t)+P'(t)$. Let $t=t+1$ and $P(t)=C(t-1)$. While termination condition is not satisfied go to Step 4. Stop.

Here, in Step 6, h is a integer defined by the designer, $select_v$ is the reproduction selection which is biased toward selecting the better performing individuals, and $select_v$ is the modified multiobjective selection which select the individuals from each partial population in proportion to the ranks.

4. Design Example

This section presents an application of GAs to the following plant

$$P(s) = \frac{1}{(s+1)(s+0.5 \pm j1.2)}. \quad (22)$$

To construct a two degree-of-freedom discrete-time control system, shown in Fig. 2, we convert the continuous time plant transfer function $P(s)$ into a discrete-time transfer function by z transform with a zero-order holding device and the time sampling period $T = 0.1$ shown in (23).

$$G(z) = \frac{1.584 \times 10^{-3}(z + 3.5485)(z + 0.2551)}{(z - 0.9004)(z - 0.9466 \pm j0.1149)}. \quad (23)$$

The system specifications are set according to a number of performance requirements. Six objectives are used:

- Rise time: $t_r \leq 1.6s$
- Settling time: $t_s \leq 8s$.
- Overshoot: $O_s \leq 12\%$.
- Maximum value of controller output: $U_c \leq 20$
- Maximum value of $y(t)$ response against a step disturbance input: $D_y \leq 0.04$.
- Maximum value of controller output against a step disturbance input: $U_d \leq 20$

To solve this problem, a multiobjective optimization GA was chosen with the following operators: 1) modified multiobjective selection ($h = 3$, see 3.2), 2) two-point crossover and 3) mutation. Population 80 for designing $C_1(z)$ and 200 for designing $C_2(z)$, reproduction rate 0.3

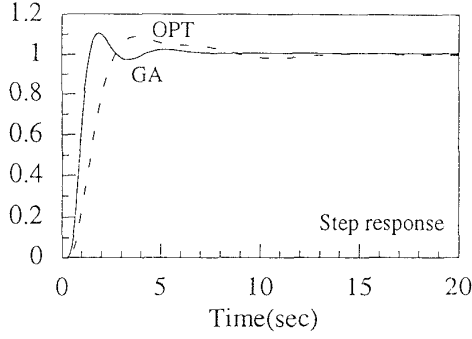


Fig. A1: Step response

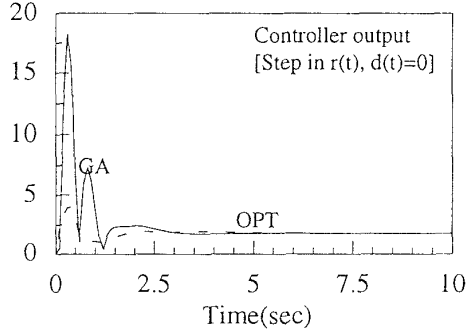


Fig. A2: Controller output

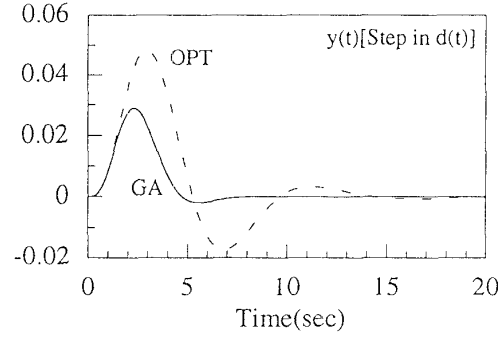


Fig. A3: Step response to disturbance

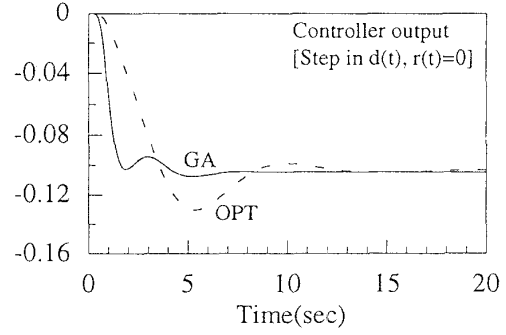


Fig. A4: Controller output to disturbance

for $C_1(z)$ and 0.8 for $C_2(z)$, crossover probability 1.0 and mutation probability 0.001 are used. Furthermore, in this example the poles $(0.9900, 0.9949 \pm j0.0119)$ of the plant $G(z)$ are not permitted to be canceled by the controllers. The polynomials included in the plant transfer function (1) are chosen as follows

$$\begin{aligned} a_+(z) &= 1 \\ a_-(z) &= (z - 0.9004)(z - 0.9466 \pm j0.1149) \\ b_+(z) &= 1 \\ b_-(z) &= 1.584 \times 10^{-3}(z + 3.5485)(z + 0.2551). \end{aligned} \quad (24)$$

In addition, one integrator is specified to be included in controller $C_1(z)$. From (7), we have inequality $m \geq 3$. Let $m = 4$ and substitute it into (8), then the inequality $n \geq 8$ should be satisfied. Let $n = 8$ and specify the orders of the controller $C_2(z)$ to be 6. In this case, the tuning parameters of $C_1(z)$ are one coefficient of $g_p(z)$ and eight roots of $f(z)$ represented by $r_1 e^{\pm j\theta_1}$, $r_2 e^{\pm j\theta_2}$, $r_3 e^{\pm j\theta_3}$, and $r_4 e^{\pm j\theta_4}$ on polar coordinate. On the other hand, the tuning parameters of $C_2(z)$ are $(a_0, a_1, a_2, a_3, a_4, a_5, a_6)$. Here, we set $a_0 = 0$ to make $C_2(z)$ to be proper. Thus fifteen parameters need to be optimized for the optimal controllers. Each radius (r_i), angle (θ_i), and the coefficients in the parameter set is represented by a 10-bit binary code, and the binary codes are concatenated to produce a 90-bit string for $C_1(z)$ and a 60-bit string for $C_2(z)$.

By the procedural flow using GA, we obtained the controllers $C_1^{GA}(z)$ and $C_2^{GA}(z)$ which achieve all the system

specifications after 28 and 4 generations, respectively.

$$C_1^{GA}(z) = \frac{1.1095(z + 9.5835)(z - 0.9459 \pm j0.1022)(z - 0.8900)}{(z - 1.0000)(z - 0.6631 \pm j0.4384)(z - 0.4837 \pm j0.2609)} \quad (25)$$

$$C_2^{GA}(z) = -\frac{0.5440}{z} + \frac{0.5147}{z^2} + \frac{0.7815}{z^3} + \frac{0.5968}{z^4} - \frac{1.0601}{z^5} + \frac{0.0044}{z^6} \quad (26)$$

In addition, to compare GA with an established optimization algorithm, the goal attainment method (attgoal) in MATLAB OPTIMIZATION TOOLBOX is used to design controllers for the same plant with the same specifications and design condition. Differing from GA, the initial values of the roots of $f(z)$ by means of the 8th-order Butterworth pattern whose radius is 2 are given. The coefficient of $g_p(z)$ and is given as 1 and those of $C_2(z)$ are given as 0's. This search terminates after 267 iterations without satisfying all the system specifications, and gains the following controllers

$$C_1^{opt}(z) = \frac{1.0352(z - 0.9488 \pm j0.1439)(z - 0.8687 \pm j0.0388)}{(z - 1)(z - 0.92291)(z - 0.5437)(z - 0.8259 \pm j0.2940)} \quad (27)$$

$$C_2^{opt}(z) = \frac{1.0897}{z} + \frac{0.7256}{z^2} + \frac{0.4541}{z^3} + \frac{0.1342}{z^4} - \frac{0.5886}{z^5} - \frac{1.4566}{z^6} \quad (28)$$

The results obtained are shown in Table 1. Figs. A1 to A4 show the step responses, the controller outputs, responses of $y(t)$ and $u(t)$ when the system suffers a step disturbance $d(t)$, respectively. From Fig. A2, one can see that the

Table 1: Design performances

Controllers		$C_1^{GA}(z)$	$C_1^{GA}(z)$ + $C_2^{GA}(z)$	$C_1^{OPT}(z)$	$C_1^{OPT}(z)$ + $C_2^{OPT}(z)$
Response to $r(t)$	Rise time (s)	0.8000	0.7000	2.2030	1.5470
	Overshoot (%)	2.7812	10.7119	28.6800	12.0000
	Settling time (s)	3.7000	2.5000	11.1000	6.6400
	Max. Controller output	17.4895	18.2418	2.1570	4.0480
Response to $d(t)$	Peak Value (to Disturbance)	0.0289		0.0485	
	Max. Controller output (to Disturbance)	0.1076		0.1287	

system designed by GA uses higher energy than MATLAB OPTIMIZATION TOOLBOX (attgoal), but obtain much better system performances shown in Figs. A1, A3, and A4. Furthermore, from Table 1 we find that by adding $C_2(z)$ to systems the performances of the input to output response can be improved without effecting the robust characteristics against the disturbance. This provides designers with a large freedom in the design of controllers for a multiobjective control system.

5. Conclusions

An application of a multiobjective GA to realistic control systems design has been discussed. By combining the multiobjective GA with the pole-zero placement algorithm which can avoid specified pole-zero cancellations, we have developed a MATLAB based CACSD system software package for two degree-of-freedom control systems. This CACSD system provides a large freedom in the choice of controller structures and in the design specifications. We design the controllers to guarantee the characteristics of the steady state and robust of the system independently. For efficient use of a MGA, we introduce an modified multiobjective selection to prevent premature convergence. To illustrate the effectiveness of the proposed CACSD system, we have given a design example where the multiobjective optimization by GA is compared with an existing parameter optimization algorithm, the goal attainment method in MATLAB OPTIMIZATION TOOLBOX. The results show that genetic algorithm works better in multiobjective control systems design.

The values of the vector performance index given by the multiobjective GA provide information on the achievable performances. The effective use of the information to automate decision making is under current investigation.

6. 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] 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
- [6] T.K. Liu, T. Satoh, T. Ishihara, and H. Inooka. An application of genetic algorithms to control system design. In Proc. 1st Asian Contr. Conf., volume III, pp. 701-704, Tokyo, 1994
- [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] H. H. Rosenbrock, Computer-Aided Control System Design, Academic Press, 1973
- [9] S. Boyd, *et al.*, A New CAD Method and Associated Architectures for Linear Controllers, IEEE Trans., AC-33-3, pp. 268-283, 1988
- [10] 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
- [11] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learn. Reading, MA, Addison-Wisely, 1989
- [12] 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
- [13] 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
- [14] K. Usami, Discrete-Time Controllers Design by Multiobjective Optimization (in Japanese), Master Thesis, Tohoku University, 1995