

## Research Article

# Dividing Genetic Computation Method for Robust Receding Horizon Control Design

**Tohru Kawabe**

*Department of Computer Science, University of Tsukuba, Tsukuba, Ibaraki 305-8573, Japan*

Correspondence should be addressed to Tohru Kawabe, kawabe@cs.tsukuba.ac.jp

Received 21 August 2009; Revised 2 December 2009; Accepted 15 January 2010

Academic Editor: Oliver Kramer

Copyright © 2010 Tohru Kawabe. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A new robust Receding Horizon Control (RHC) design approach for the sampled-data systems is proposed. The approach is based on a dividing genetic computation of minimax optimization for a robust finite receding horizon control problem. Numerical example is given to show the effectiveness of the proposed method.

## 1. Introduction

In last few decades, the Receding Horizon Control (RHC) has been widely accepted in the industries [1]. RHC is an online powerful control method which solves a control problem with respect to each sampling frequency [2]. A significant merit of RHC is easy handling of constraints during the design and implementation of the controller [2, 3].

In the standard RHC formulation, the current control action is derived by solving a finite or infinite horizon quadratic cost problem at every sample time using the current state of the plant as the initial state [1]. It is an online optimization with big calculation amount. Then, the RHC has been applied conventionally to systems with relatively slow-moving dynamics such as petrochemical plants and so on. However, recent advance of computer performance has made it possible to use it for systems with relatively fast-moving dynamics, for example, the mechatronics and so on. Therefore, it is important to develop a practical RHC method for such systems.

Incidentally, a drawback of the standard RHC is explicitly lack of robust property with respect to model uncertainties or disturbances since the online minimized cost function is defined in terms of the nominal systems. A possible strategy for realizing the robust RHC is solving the so-called minimax optimization problem, namely, minimization problem over the control input of the performance measure maximized by plant uncertainties or disturbances. An earliest work was

proposed by Campo and Morari [4] and Zheng and Morari [5]. Kothare et al. solve minimax RHC problems with state-space uncertainties through LMIs [6]. Cuzzola et al. improve Kothare's method [6] to reduce conservativeness in [7]. Other methods of minimax RHC for systems with model uncertainty can be found in [8–12]. However, the number of available work of the robust RHC is still limited compared with many methods of robust control synthesis for linear systems being proposed. Main reason of this fact is that the robust (minimax) RHC problem is hard to solve in real-time due to the trade-off with an objective function and constraint conditions in the problem generally. The issue of robust RHC therefore still deserves further attention [2, 3] and the effective approach for the robust RHC is still required, especially in the optimization technique.

Optimization techniques by using evolutionary computation such as Genetic algorithms (GAs) [13, 14] have been recognized to be well suited to many kinds of engineering 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 many kinds of optimization problems were proposed in several manners [15–17]. For example, Schaffer [15] proposed an extension of the simple GA to accommodate vector-valued fitness measures, which he called the Vector Evaluated Genetic Algorithm (VEGA). Moreover, Goldberg [13] firstly proposed the Pareto-based approach as a means of assigning equal

probability of reproduction to all nondominated individuals in the population. Fonseca and Fleming [16] proposed a multiobjective ranking method with the Pareto-based fitness assignment. However, it is uncertain whether to be able to apply these methods effectively to the minimax RHC problem.

Therefore, in this paper, a new minimax robust finite RHC design approach based on a new dividing genetic computation for constrained sampled-data systems with structured uncertainties and disturbance is proposed. This approach is one of the general and practical framework of the minimax robust finite RHC problem of bounded constrained systems. Since the dividing genetic computation uniformly controls the convergence of solutions of optimization problems with some trade-off conditions, it can be effective for the minimax RHC problem. Using this approach, we can expect to reduce the conservativeness of control performance and to improve the control performance. Numerical example is given to show these facts.

## 2. Problem Formulation

The target system in this paper is the sampled-data control system. Hence, the control object with uncertainties is a continuous-time system and the controller is designed in discrete-time. Then, let us consider the following discrete-time LTI (Linear Time-Invariant) state-space model with structured uncertainties and disturbances:

$$\begin{aligned} x(k+1) &= (A + L\Delta R_A)x(k) + (B + L\Delta R_B)u(k), \\ y(k) &= Cx(k) + \eta(k), \end{aligned} \quad (1)$$

where  $L\Delta R_A$  and  $L\Delta R_B$  denote the structured uncertainties expressed by perturbation of elements in  $A$  and  $B$ , respectively.  $A, B, C, L, R_A$ , and  $R_B$  are constant matrices.  $\Delta$  ( $\Delta = \text{diag}\{\delta_1, \delta_2'', \dots\}$ ) is a structured uncertainties parameters matrix satisfied  $\Delta^T \Delta \leq r$ . ( $r$  is a given constant) And  $x(k)$ ,  $u(k)$ ,  $y(k)$ , and  $\eta(k)$  denote the state, input, measured output, and disturbance vector, respectively. All these vectors and matrices have appropriate dimensions.

Then, we can transform this system as

$$x(k+1) = Ax(k) + Bu(k) + Lw(k), \quad (2)$$

$$z(k) = R_A x(k) + R_B u(k), \quad (3)$$

$$y(k) = Cx(k) + \eta(k), \quad (4)$$

where  $w(k) = \Delta z(k)$ . We assumed that the system is constrained with following conditions with  $N-1$  steps:

$$\begin{aligned} w^T(k+j)P_w w(k+j) &\leq 1, \\ \eta^T(k+j)P_\eta \eta(k+j) &\leq 1, \\ u^T(k+j)P_u u(k+j) &\leq 1, \\ (j &= 0, \dots, N-1), \end{aligned} \quad (5)$$

where  $P_w, P_\eta, P_u$  ( $P_w, P_u, P_\eta > 0$ ) are positive symmetric matrices for weights of constraints.

For this systems, to use the RHC, a quadratic performance measure with finite horizon with positive weighting constant matrices  $Q$  and  $R$  ( $Q, R > 0$ ) as

$$J(k) = \sum_{j=0}^{N-1} \|y(k+j+1|k)\|_Q^2 + \|u(k+j|k)\|_R^2 \quad (6)$$

is used.  $x(k+j|k)$ ,  $y(k+j|k)$ , and  $u(k+j|k)$  are the predicted state of the plant, the predicted output of the plant and the future control input at time  $k+j$ , respectively.

Then, the robust RHC is need to solve the following minimax optimization problem at each step  $k$ :

$$\begin{aligned} \min_{u(k+j|k)} \quad & \max_{9w(k+j|k), \eta(k+j|k)} J(k) \\ \text{subject to} \quad & w^T(k+j)P_w w(k+j) \leq 1, \\ & u^T(k+j)P_u u(k+j) \leq 1 \\ & \eta^T(k+j)P_\eta \eta(k+j) \leq 1 \\ & (j = 0, \dots, N-1). \end{aligned} \quad (7)$$

Generally, the following state feedback schema is employed:

$$u(k+jk) = -F_{k+j}x(k+jk) \quad (j = 0, 1, \dots, N-1), \quad (8)$$

where  $F_{k+j}$  is a feedback gain matrix.

Finally, the procedure of robust RHC is summarized as follows. At the current step  $k$ , future state values  $x(k+j|k)$  ( $j = 0, \dots, N-1$ ) are predicted by using the state space model (2)–(4), and future feedback gain matrix candidates  $F_{k+j}$  ( $j = 0, \dots, N-1$ ) are calculated by solving (7) under (8). Only first gain matrix  $F(k+1)$  is employed for an actual control input and the others are discarded. Then, go to next step  $k+1$  and repeat same operations.

Namely, the robust RHC requires an online minimax optimization. However, it is difficult to solve this problem as it is at each step, generally. Therefore, the method to eliminate the maximization procedure and transform this problem to simple minimization problem is shown in the next section.

## 3. Transformation of Minimax Finite RHC Problem

Firstly, introducing the following vectors

$$\begin{aligned} X &:= [x(k+1|k) \ x(k+2|k) \ \cdots \ x(k+N|k)]^T, \\ Y &:= [y(k+1|k) \ y(k+2|k) \ \cdots \ y(k+N|k)]^T, \\ U &:= [u(k|k) \ u(k+1|k) \ \cdots \ u(k+N-1|k)]^T, \\ W &:= [w(k|k) \ w(k+1|k) \ \cdots \ w(k+N-1|k)]^T, \\ \Lambda &:= [\eta(k|k) \ \eta(k+1|k) \ \cdots \ \eta(k+N-1|k)]^T \end{aligned} \quad (9)$$

and using state space equations (2)–(4) recursively, we can derive

$$\begin{aligned} X &= \tilde{A}x(k) + \tilde{B}U + \tilde{L}W, \\ Y &= C\tilde{A}x(k) + C\tilde{L}W + \Lambda, \end{aligned} \quad (10)$$

where

$$\begin{aligned} \tilde{A} &:= \begin{bmatrix} A & A^2 & \cdots & A^{N-1} \end{bmatrix}^T, \\ \tilde{B} &:= \begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ A^{N-2}B & A^{N-3}B & \cdots & B \end{bmatrix}, \\ \tilde{L} &:= \begin{bmatrix} L & 0 & \cdots & 0 \\ AL & L & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ A^{N-2}L & A^{N-3}L & \cdots & L \end{bmatrix}. \end{aligned} \quad (11)$$

Hence, we can transform the minimax problem (7) to

$$\begin{aligned} &\min_U \gamma \\ &\max_{W, \Lambda} \Pi \leq \gamma, \\ \text{subject to } &w^T(k+j) \quad P_w \quad w(k+j) \leq 1 \\ &u^T(k+j) \quad P_u \quad u(k+j) \leq 1 \\ &\eta^T(k+j) \quad P_\eta \quad \eta(k+j) \leq 1 \\ &(j = 0, \dots, N-1), \end{aligned} \quad (12)$$

where  $\gamma > 0$  (scalar parameter) and  $P_i$  is defined as follows:

$$\begin{aligned} \Pi &:= \left\{ \left\| \tilde{A}x(k) + \tilde{B}U + \tilde{L}W + \Lambda \right\|_{\hat{Q}}^2 + \|U\|_{\hat{R}}^2 \right\}, \\ \hat{Q} &:= \begin{bmatrix} Q & \mathbf{0} \\ & \ddots \\ \mathbf{0} & Q \end{bmatrix}, \quad \hat{R} := \begin{bmatrix} R & \mathbf{0} \\ & \ddots \\ \mathbf{0} & R \end{bmatrix}. \end{aligned} \quad (13)$$

To eliminate the maximization procedure, we have to remove  $W$  and  $\Lambda$  terms in the first constraint. For this, in the first place, following basis for all variables and transformation matrices are defined,

$$\zeta = \begin{bmatrix} x(k) & W^T & \Lambda^T & 1 \end{bmatrix}^T, \quad (14)$$

$$U = H_u \zeta, \quad (15)$$

$$Y = H_y \zeta, \quad (16)$$

$$\Lambda = H_\eta \zeta, \quad (17)$$

$$1 = (H_1 \zeta)^T (H_1 \zeta), \quad (18)$$

where

$$\begin{aligned} H_u &:= \begin{bmatrix} F\tilde{A} & F\tilde{L} & F & 0 \end{bmatrix}, \\ H_y &:= \begin{bmatrix} C\tilde{A} & C\tilde{L} & I & 0 \end{bmatrix}, \\ H_\eta &:= \begin{bmatrix} \mathbf{0} & \mathbf{0} & I & 0 \end{bmatrix}, \\ H_1 &:= \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix}, \end{aligned} \quad (19)$$

and where

$I :=$  (identity matrix with appropriate dimension),

$$F := \begin{bmatrix} -F_k & 0 & \cdots & 0 \\ 0 & -F_{k+1} & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & -F_{k+N-1} \end{bmatrix}. \quad (20)$$

By using these, we can express the first constraint condition of problem (12):

$$\max_{W, \Lambda} \left\{ \|H_y \zeta\|_{\hat{Q}}^2 + \|H_u \zeta\|_{\hat{R}}^2 \right\} \leq (H_1 \zeta)^T \lambda (H_1 \zeta). \quad (21)$$

Please take notice that both the left side and the right side of this inequality are expressed by the quadratic forms and they have positive scalar values. Hence, if the inequality is held by maximum values of  $W$  and  $\Lambda$  in left side, this inequality must be held by any other values of them. This fact means that we can eliminate the maximization procedure in the first constraint. We can only check the following condition instead of the first constraint of problem (12):

$$\left\{ \|H_y \zeta\|_{\hat{Q}}^2 + \|H_u \zeta\|_{\hat{R}}^2 \right\} \leq (H_1 \zeta)^T \lambda (H_1 \zeta). \quad (22)$$

In the second place,  $H_w(j)$  is defined. This matrix picks out the suitable block from  $W$  and satisfies the relation of  $w(k+j) = H_w^{(j)} \zeta$ . Then, we can derive

$$\left( H_w^{(j)} \zeta \right)^T P_w \left( H_w^{(j)} \zeta \right) \leq (H_1 \zeta)^T (H_1 \zeta) \quad (j = 0, \dots, N-1). \quad (23)$$

For the constraints of  $\eta$ ,  $u$  and  $z$ , we can derive the following relations in the same way:

$$\left( H_\eta^{(j)} \zeta \right)^T P_\eta \left( H_\eta^{(j)} \zeta \right) \leq (H_1 \zeta)^T (H_1 \zeta),$$

$$\left( H_u^{(j)} \zeta \right)^T P_u \left( H_u^{(j)} \zeta \right) \leq (H_1 \zeta)^T (H_1 \zeta) \quad (j = 0, \dots, N-1). \quad (24)$$

Furthermore, by using (14)–(21), all constraints in minimax problem (12) can be transformed into

$$\begin{aligned} \text{subject to } & \forall \zeta \neq 0; \quad \zeta^T (H_1^T \lambda H_1 - H_x^T \hat{Q} H_x - H_u^T \hat{R} H_u) \zeta \geq 0 \\ & \zeta^T (H_1^T H_1 - (H_w^{(j)})^T P_w H_w^{(j)}) \zeta \geq 0 \\ & \zeta^T (H_1^T H_1 - (H_u^{(j)})^T P_u H_u^{(j)}) \zeta \geq 0 \\ & \zeta^T (H_1^T H_1 - (H_\eta^{(j)})^T P_\eta H_\eta^{(j)}) \zeta \geq 0 \\ & (j = 0, \dots, N-1). \end{aligned} \quad (25)$$

Then, we can transform the original minimax problem (7) to the following one by using *S*-procedure [18]:

$$\begin{aligned} & \min_F \gamma \\ \text{subject to } & H_1^T \gamma H_1 - H_y^T \hat{Q} H_y - H_u^T \hat{R} H_u \\ & - \sum_{j=0}^{N-1} [\tau_j^w S_j^w + \tau_j^u S_j^u + \tau_j^\eta S_j^\eta] \geq 0 \\ & (j = 0, \dots, N-1), \end{aligned} \quad (26)$$

where

$$\begin{aligned} S_j^w &= (H_1^T H_1 - (H_w^{(j)})^T P_w H_w^{(j)}), \\ S_j^u &= (H_1^T H_1 - (H_u^{(j)})^T P_u H_u^{(j)}), \\ S_j^\eta &= (H_1^T H_1 - (H_\eta^{(j)})^T P_\eta H_\eta^{(j)}), \end{aligned} \quad (27)$$

and where  $\tau_j^w$ ,  $\tau_j^u$ ,  $\tau_j^\eta$ , and  $\tau_j^z$  are positive semidefinite scalars. It must be noted that this transformation satisfies only a sufficient condition of *S*-procedure, since *S*-procedure is not the so-called “lossless” in this case. We cannot therefore avoid that the design results are slightly conservative. Nevertheless, we can expect the reduction of conservativeness in design result by this technique in contrast with the results by preexisting methods, because the conservativeness caused by *S*-procedure is too small to put a matter for practical purposes.

Finally, using “Schur-complement” [19], we can transform the minimax optimization problem (7) into the following problem:

$$\begin{aligned} & \min_F \gamma \\ \text{subject to } & \begin{bmatrix} H_1^T \gamma H_1 - \Sigma & H_y^T & H_u^T \\ H_y & \hat{Q}^{-1} & 0 \\ H_u & 0 & \hat{R}^{-1} \end{bmatrix} \geq 0, \\ & \tau_j^w, \tau_j^u, \tau_j^\eta \geq 0 \quad (j = 0, \dots, N-1), \end{aligned} \quad (28)$$

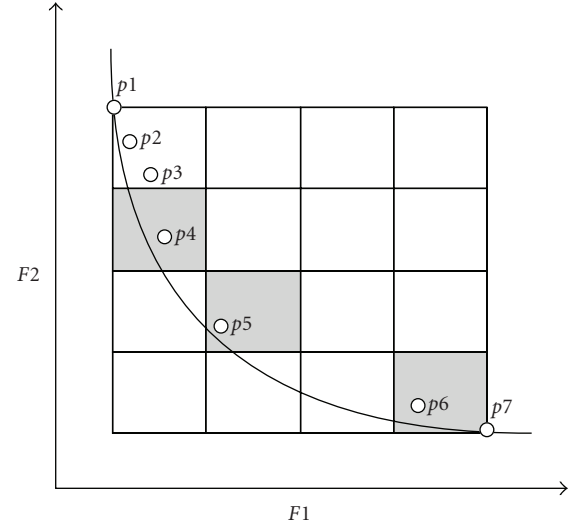


FIGURE 1: Dividing strategy.

where

$$\Sigma := \sum_{j=0}^{N-1} [\tau_j^w S_j^w + \tau_j^u S_j^u + \tau_j^\eta S_j^\eta]. \quad (29)$$

It is known that this problem has trade-off with the objective function and the constraint condition. Therefore, the fast method of finding nondominated solutions with respect to the trade-off as a lot as possible is needed. Then, the method using dividing genetic computation is shown in the next section.

#### 4. RHC Method with Adaptive DA Converter Using Dividing Genetic Computation

**4.1. Dividing Genetic Computation.** To find the best possible nondominated solutions for the RHC problem (28), a partial convergence of nondominated solutions must be avoided. Therefore, a dividing method which uniformly controls the distribution of solutions is proposed. The proposed method assigns all nondominated individuals to prespecified regions. An example of the dividing strategy in two objective minimizing problem is shown in Figure 1. The dividing genetic computing method consists of following procedure. First, the objective space is divided into prespecified regions. The edge points of the whole region corresponds to the best solutions for each objective function. In Figure 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 nondominated solutions in the identical region with the individual  $p_i$ . In the example, the fitness of the individuals illustrated in the figure corresponds 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 us define a neighborhood to every individual as follows: two objective functions of a multiobjective problem are selected by using prespecified 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.

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 a 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  $\Gamma_i$  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- $\alpha$  method is employed for crossover. 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 nonuniform mutation are used to avoid the premature convergence of the solutions.

The procedure of dividing genetic computation approach 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 individual 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 is 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 members 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

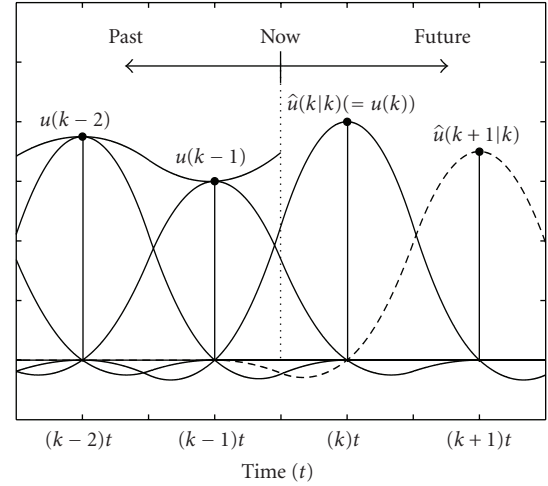


FIGURE 2: Interpolation based on a 2nd-order spline sampling function using predictive future control inputs.

diversity, multiple equivalent individuals are eliminated from the current population.

As a result of exploratory experiments using the multiobjective ranking [16], the following facts have been obtained by comparison with the standard genetic algorithm (GA).

- (i) By using the proposed method, the solutions are widely distributed in the trade-off surface, and the search performance does not deteriorate significantly.
- (ii) The standard GA approaches cause the partial convergence of the solutions because of stochastic errors in the iterative process.
- (iii) It is clear that the proposed method can seek for the widely distributed solutions in comparison with the standard GA.

Therefore, it is judged that the proposed dividing genetic computation method is expected to be effective for the minimax RHC problem.

**4.2. Interpolation Using Predictive Control Inputs.** Generally, the interpolation of samples in the DA conversion is executed by a convolution of samples by using sampling function. In the case of sampled-data control system, to interpolate the current interval, the future sample is needed. But, the information about future sample is unable to be obtained in the present time. Hence, we need to wait to obtain this future sample information. As a result, the time-delay is occurred. However, in the case of controlling systems with relatively fast-moving dynamics, such as robots or vehicles, the method with long time-delay is unable to be applied. That is the point. Therefore, a new idea to use the predictive control inputs obtained by RHC for interpolation is proposed.

In RHC, the optimal control inputs  $\{\hat{u}(k|k), \hat{u}(k+1|k), \dots, \hat{u}(k+N-1|k)\}$  are calculated in each step, and only the first control input  $\hat{u}(k|k)$  is used as a real control input. Therefore, we consider to use the other optimal control inputs  $\{\hat{u}(k+1|k), \hat{u}(k+2|k), \dots\}$  as virtual future sampling



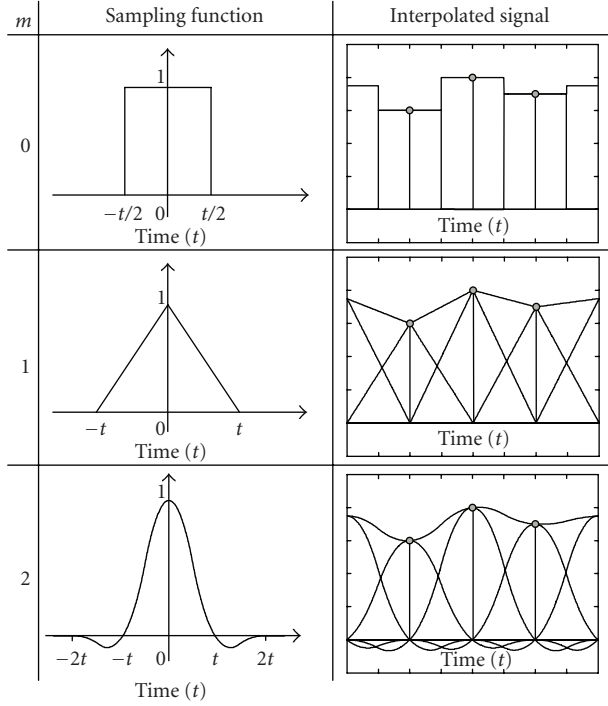


FIGURE 3: Sampling functions  $\Psi$  and their interpolations ( $m = 0, 1, 2, 3$ ) ( $\tau$ : sampling interval).

points. Actually, it is only necessary to use the optimal control inputs which are need for interpolation according to the sampling function. Figure 2 shows an example of this way by using the 2nd-order spline function for interpolations. Only  $\hat{u}(k+1 | k)$  is used as a virtual future sampling point in this case.

Hence, by using the predictive control inputs for interpolation, it becomes possible to reduce the time-delay in the DA conversion, and the total waiting-time is just only computation time of optimization in current step.

Of course, one needs to take account that there is a difference between virtual future sampling points and real sampling points like  $\hat{u}(k+1 | k) \neq u(k+1)$  in future step. However, we consider that this point is not a critical problem because the influence on interpolated waveform due to prediction error is not so big compared to the scale of prediction error. Although the differentiability of each sampling function is lost at sampling points, this also does not become a critical problem compared to the zero-order hold, and it is possible to keep a certain level of smoothness.

**4.3. Adaptive DA Converter.** The spline functions provide various sampling functions with all kinds of orders. Therefore, we consider switching the spline functions optimally according to the system status in the adaptive DA converter. In this paper, we use the spline functions with the order  $m = 0, 1, 2$  as sampling functions. Namely, in the case of  $m = 0$ , the sampling function is equivalent to the staircase function. In the case of  $m = 1$ , it is the 1st-order piecewise polynomial function, and in  $m = 2$ , 2nd-order one as shown in Figure 3.

Appropriate selecting the values of  $m$  according to the object enables to deal with DA conversion flexibly and precisely in the interpolation operation. Although the interpolation is more precisely in the case of using the spline function with  $m = 3$  or more, it is difficult to apply to fast-moving dynamic systems due to the bigger amount of calculation. Therefore we use only the spline functions with the order  $m = 0, 1, 2$ .

The interpolated signals in the closed-open interval  $[k\tau, (k+1)\tau)$  using these sampling functions are obtained as follows:

$$u(t) = \sum_{l=k}^{k+1} \{u(l) \cdot {}^{1,2}\Psi(t-l\tau)\} \quad (m=0,1), \quad (30)$$

$$u(t) = \sum_{l=k-1}^{k+2} \{u(l) \cdot {}^3_{[c]}\Psi(t-l\tau)\} \quad (m=2),$$

where  $\Psi(\cdot)$  is sampling function as shown in Figure 3, and where  $u(t)$  and  $u(l)$  are analog signal and digital signal, respectively.

Figure 4 shows the entire controlled system with the proposed RHC and the adaptive DA converter. As shown in Figure 4, the adaptive DA converter has internal model with sampling interval  $\tau/d$ . Please take notice that this internal model 2 is different from interval model 1 in which sampling interval is just  $\tau$ . The interval to be interpolated is also partitioned to  $d$  sections, and the partitioning points  $u_m(j;k)$ , ( $j = 1, 2, \dots, d-1$ ) on interpolated waveforms are used for the selection of parameter  $m$ , that indicates the degree of spling sampling functions:

$$u_m(j;k) = \sum_{l=k-\phi}^{k+\phi-1} \left\{ u(l) \cdot {}^m\psi \left( (k-1)\tau + \frac{\tau}{d} \cdot j - l\tau \right) \right\} \quad (j = 1, 2, \dots, d-1), \quad (31)$$

where  $\phi$  is the number of samples which the sampling function needs for interpolation, and it is adjusted according to the sampling function.

Figure 5 shows the difference of the interpolation and partition points according to the sampling function with  $m = 0, 1, 2$  in the case of  $d = 5$ . How to select the values of  $m$  in each interval is summarized as follows. Each value of sample on the partition point is calculated from the state-space equation (2) in the current interval. The evaluation values using  $J(k)$  in (6) are calculated in each  $m$ . Then, the parameter  $m$  whose evaluation value is the smallest is selected for this interval.

From several test simulation results, it has been obtained that the partition number  $d = 5$  is most appropriate by the trade-off between computation time and precision.

## 5. Numerical Example

In this section, an example that illustrates the effectiveness of the proposed method is given. The example is adopted from the benchmark problem in [20] as shown in Figure 6.

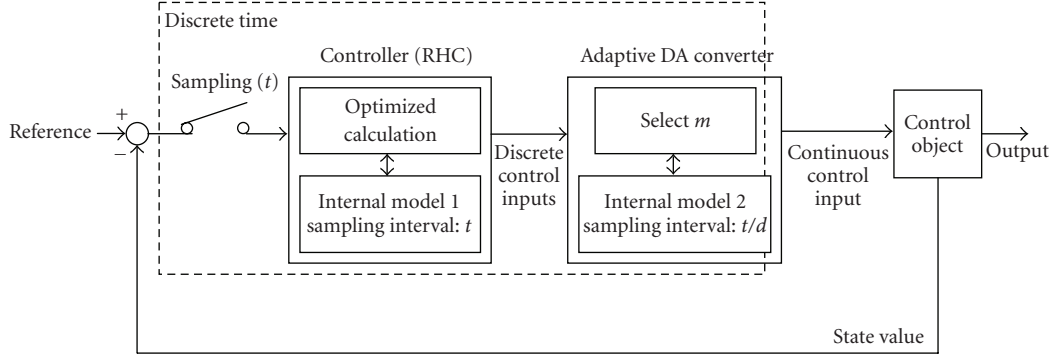


FIGURE 4: Proposed RHC systems with adaptive DA converter.

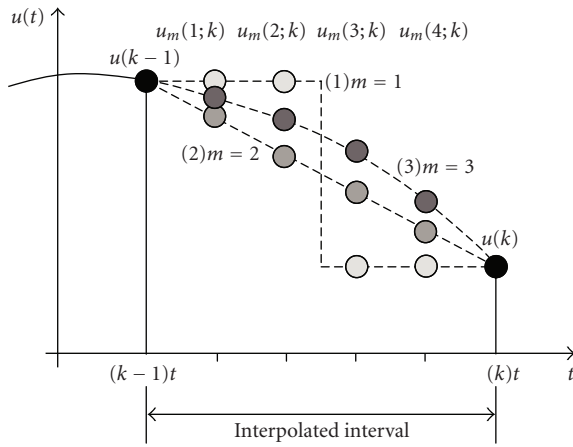
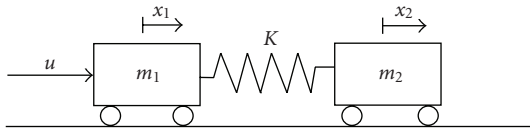
FIGURE 5: Interpolation ways ( $d = 5$ ).

FIGURE 6: Coupled spring-mass system.

Although the original system consists of a two-mass-spring system without noise for output, the bounded noise is added to demonstrate the proposed method. The state-space equations are obtained as follows:

$$\frac{d}{dt} \begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \\ x_4(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0.1 & 0 \\ 0 & 1 & 0 & 0.1 \\ -0.1K/m_1 & 0.1K/m_1 & 1 & 0 \\ 0.1K/m_2 & 0.1K/m_2 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0.1/m_1 \\ 0 \end{bmatrix} u(k), \quad (32)$$

TABLE 1

$\alpha$ in the BLX- $\alpha$ method	0.5
Population size	50
Mutation rate	0.10
Maximum generation	100
The number of dividing region	$50 \times 50 = 2500$

TABLE 2: Upper bound values of  $K_{\max}$ .

Method	$K_{\max}$
Proposed method	79.47
Cuzzola's method [7]	68.28
Kothare's method [6]	45.36

where  $m_1$  and  $m_2$  are the two masses and  $K$  is the spring constant. State variables  $x_1$  and  $x_2$  are the positions of the two masses, respectively, and  $x_3$  and  $x_4$  are their velocities. Now we assume the following perturbations of  $m_2$  and  $K$ :

$$\begin{aligned} m_2 &\in \{m_2 \mid 1 \leq m_2 \leq 10\}, \\ K &\in \{K \mid 0.5 \leq K \leq K_{\max}\}, \end{aligned} \quad (33)$$

and  $m_1$  is constant equaled to 1. The weighting matrices are fixed as  $Q = I$  and  $R = 0.5$ . The constraint condition of input  $|u(k)| \leq 1$  must be satisfied. This means that  $P_u = 1$ . And the constraints of external disturbance,  $\eta$ , are set  $P_\eta = 36.0$ . A prediction horizon of the RHC is set  $N = 10$ .

Furthermore, the following GA parameter specifications are used in the simulation. These values have been selected from several tests (see Table 1).

The results as follows are the best ones in having repeated 20 times.

Figure 7 shows the closed-loop response of the output. From this figure, we can say that the proposed method has good robust performance.

Figure 8 shows the change of the parameter  $m$  of the adaptive DA converter. From this figure, we can see that the spline function with  $m = 0$  (staircase function) is likely to be selected when the control input stays flat, and the function with  $m = 1$  (piecewise linear function) is selected when the control input changes rapidly. The function with

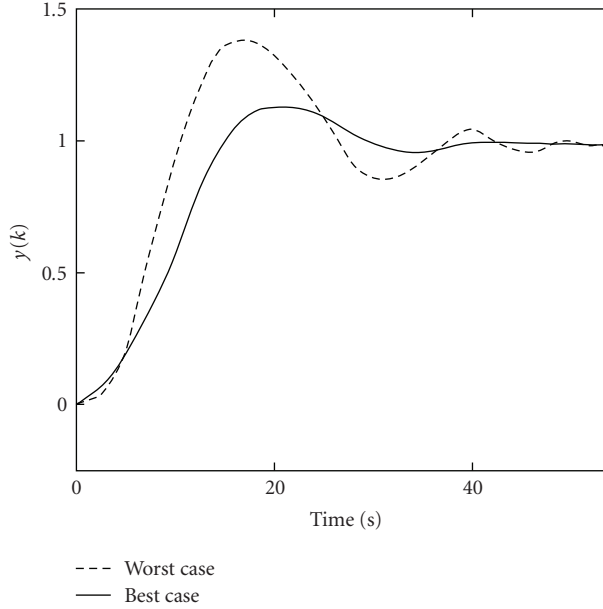


FIGURE 7: Closed-loop response.

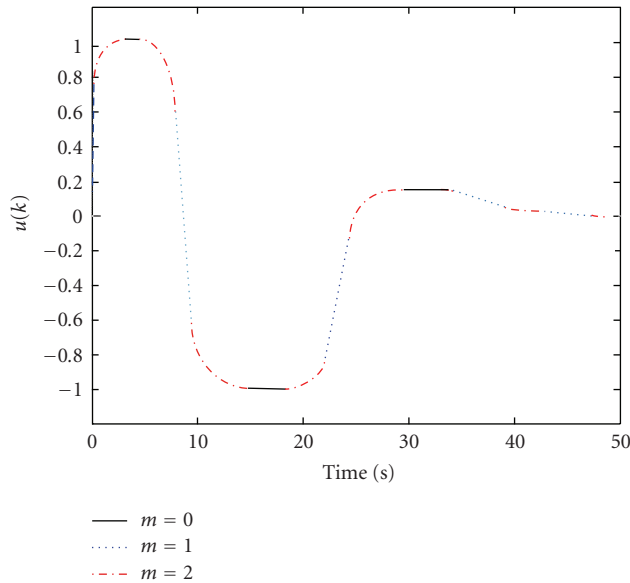


FIGURE 8: Illustration of Switching ways of sampling functions in the interpolation.

$m = 2$  (piecewise quadratic function) is also likely to be selected when the control input changes smoothly. These are natural results, but please take notice that the tiny difference of control input causes a big influence on the result, in the case of the systems with fast-moving dynamics. Therefore, it is important to select the appropriate value of  $m$  in each interval flexibly for better control performance. Moreover, longer the sampling interval, the improvement of control performance is expected to be more conspicuous using the proposed method.

To show the fact that the proposed method can reduce the conservativeness, the maximum values of  $K_{\max}$  by the

TABLE 3

CPU	Intel Core2 Duo U7700 1.33 GHz
Memory	2 GB RAM
OS	Windows Vista Business

proposed method, the technique in [6], and the one in [7] are searched, respectively, within the limits of holding the feasibility of the robust RHC problem. Then results obtained are indicated in Table 2.

From Table 2, we can see that the result by proposed method is much better than the one in [6] and slightly better than the one in [7]. We can see therefore that the proposed method with the dividing genetic computation can realize the less conservative control performance than the preexisting methods in [6, 7].

To examine the performance of the dividing genetic computation method, comparisons of computation time with NSGA-II [21] and SPEA-II [22] are done.

Computation environment using a software, "MATLAB 7.8.0", is as shown in Table 3.

Then, maximum computation time of the feedback gain matrix  $F$  per each step in the robust RHC by using the dividing genetic computation method is 0.04 second. On the other hand, the times by NSGA-II and by SPEA-II are indicated by the same value 0.02 second. Although the proposed method takes twice time compared with NSGA-II and SPEA-II, it can be said that the proposed method can be practicable in such a time.

Moreover, the upper bound values of  $K_{\max}$  by the proposed method, by NSGA-II [21], and by SPEA-II [22] are calculated, respectively. As a result, values are obtained: 79.47 by proposed method as above mentioned, 78.98 by NSGA-II, and 79.28 by SPEA-II, respectively. Judging from this, we can say that the proposed method might be somewhat excellent for the reducing the conservativeness of control performance compared with them.

## 6. Conclusion

The new approach of minimax robust finite RHC method based on dividing genetic computation for constrained sampled-data control systems with structured uncertainties and disturbance has been proposed. At the same time, a numerical example is given to show that the proposed method improves the control performance.

Although the dividing genetic computation method is able to uniformly control the convergence of solutions of the minimax RHC design problems, more performance investigation of this method is compared with other GA methods, for example, NSGA-II, SPEA-II, and so on, or other heuristic optimization methods, for example, Particle Swarm Optimization [23], Ant Colony Optimization [24], and so on, as future works.

Moreover, I also need to develop the selection method of the best sampling function according to the control object, and need to make sure the effectiveness of the proposed method in various control objects as future works.



## Appendix

The proposed minimax approach is easily extended to systems with other constraints which are specified by ellipsoidal bounds, for example, state estimation errors, and so on as follows.

In the case that  $x(k)$  is not full measured, we need to estimate  $x(k)$ , where the bound of estimation error  $e(k) = x(k) - \hat{x}(k)$  is guaranteed as an ellipsoidal set as

$$e^T(k)P_e e(k) \leq 1, \quad (\text{A.1})$$

where  $P_e$  is a positive symmetric matrix for weight. This specification of estimation error is standard one. Now we introduce  $H_e$  as

$$H_e := \begin{bmatrix} 1 & 0 & \cdots & 0 & -\hat{x}(k) \end{bmatrix}, \quad (\text{A.2})$$

and then the relation of  $e(k) = H_e \zeta$  is hold. And the condition below is also held:

$$\zeta^T (H_1^T H_1 - H_e^T P_e H_e) \zeta \geq 0. \quad (\text{A.3})$$

Since this condition has same form as other constraints in (7), we can include this condition into the condition of problem (26) by using a new variable  $\tau_e$ . Furthermore, in this case, a new output equation with measurement noise  $\psi(k)$  is needed as follows instead of (4):

$$y(k) = Cx(k) + \psi(k) \quad (\psi^T(k)P_\psi \psi(k) \leq 1). \quad (\text{A.4})$$

We can also include this constraint into the condition of problem (26) by using a new variable  $\tau_\psi$ .

## Acknowledgment

The author gratefully thanks the anonymous reviewers for their valuable comments and editorial assistance.

## References

- [1] C. E. García, D. M. Prett, and M. Morari, "Model predictive control: theory and practice—a survey," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.
- [2] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.
- [3] A. Bemporad and M. Morari, "Robust model predictive control: a survey," in *Robustness in Identification and Control*, A. Garulli, A. Tesi, and A. Vicino, Eds., vol. 245 of *Lecture Notes in Control and Information Sciences*, pp. 207–226, Springer, London, UK, 1999.
- [4] P. J. Campo and M. Morari, "Robust model predictive control," in *Proceedings of the American Control Conference*, pp. 1021–1026, 1987.
- [5] Z. Q. Zheng and M. Morari, "Robust stability of constrained model predictive control," in *Proceedings of the American Control Conference*, pp. 379–383, 1993.
- [6] M. V. Kothare, V. Balakrishnan, and M. Morari, "Robust constrained model predictive control using linear matrix inequalities," *Automatica*, vol. 32, no. 10, pp. 1361–1379, 1996.
- [7] F. A. Cuzzola, J. C. Geromel, and M. Morari, "An improved approach for constrained robust model predictive control," in *Proceedings of the European Control Conference*, 2001.
- [8] J. C. Allwright and G. C. Papavasiliou, "On linear programming and robust model-predictive control using impulse-responses," *Systems and Control Letters*, vol. 18, no. 2, pp. 159–164, 1992.
- [9] J. H. Lee and Z. Yu, "Worst-case formulations of model predictive control for systems with bounded parameters," *Automatica*, vol. 33, no. 5, pp. 763–781, 1997.
- [10] A. Bemporad, "Reducing conservativeness in predictive control of constrained systems with disturbances," in *Proceedings of the 37th IEEE Conference on Decision and Control (CDC '98)*, vol. 2, pp. 1384–1389, December 1998.
- [11] A. Bemporad and A. Garulli, "Output-feedback predictive control of constrained linear systems via set-membership state estimation," *International Journal of Control*, vol. 73, no. 8, pp. 655–665, 2000.
- [12] P. O. M. Scokaert and D. Q. Mayne, "Min-max feedback model predictive control for constrained linear systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 8, pp. 1136–1142, 1998.
- [13] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley, New York, NY, USA, 1989.
- [14] Z. Michalewicz and C. Janikow, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer, London, UK, 1992.
- [15] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," in *Proceedings of the 1st International Conference on Genetic Algorithms (ICGA '85)*, pp. 93–100, 1985.
- [16] C. M. Fonseca and P. J. Fleming, "An overview of evolutionary algorithms in multiobjective optimization," *Evolutionary Computation*, vol. 3, no. 1, pp. 1–16, 1995.
- [17] D. H. Loughlin and S. Ranjithan, "The neighborhood constraint method: a genetic algorithm-based multiobjective optimization technique," in *Proceedings of the 7th International Conference on Genetic Algorithms (ICGA '95)*, pp. 666–673, 1995.
- [18] S. Boyd and C. H. Barratt, *Linear Controller Design: Limits of Performance*, Prentice-Hall, Upper Saddle River, NJ, USA, 1991.
- [19] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*, Prentice-Hall, Upper Saddle River, NJ, USA, 1996.
- [20] B. Wie and D. S. Bernstein, "Benchmark problems for robust control design," *Journal of Guidance, Control, and Dynamics*, vol. 15, pp. 1057–1059, 1992.
- [21] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [22] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: improving the strength Pareto evolutionary algorithm," in *Proceedings of the International Congress on Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems (EUROGEN '01)*, K. Giannakoglou, D. Tsahalis, J. Periaux, P. Papailou, and T. Fogarty, Eds., pp. 95–100, 2001.

- [23] C. A. Coello Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 256–279, 2004.
- [24] T. Stützle and H. H. Hoos, "MAX-MIN ant system," *Future Generation Computer Systems*, vol. 16, no. 8, pp. 889–914, 2000.

