Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Synthesis of fixed-structure H_∞ controllers via Constrained Particle Swarm Optimization

Ichiro Maruta * **Tae-Hyoung Kim** ** **Toshiharu Sugie** ***

* *Department of Systems Science, Kyoto University, Uji, Kyoto 611-0011, Japan (e-mail: maruta@robot.kuass.kyoto-u.ac.jp)*
** *CREST, Japan Science and Technology Agency, 4-1-8 Honcho, Kawaguchi-shi, Saitama 332-0012, Japan (e-mail: Kim_Tae_Hyoung@ipc.i.u-tokyo.ac.jp)*
*** *Department of Systems Science, Kyoto University, Uji, Kyoto 611-0011, Japan (e-mail: sugie@i.kyoto-u.ac.jp)*

Abstract: This paper provides a design method of fixed-structure robust controllers satisfying multiple H_∞ norm specifications by using a sort of randomized algorithms. First, a new tool to perform general constrained optimization is developed which does not need any gradient or derivative of the objective function. This tool is based on PSO (particle swarm optimization), which attracts a lot of attention recently in the evolutionary computation area due to its empirical evidence of its superiority in solving various non-convex problems. Second, it is shown how to design a fixed-structure controller satisfying given multiple H_∞ specifications by using the developed optimization tool. Third, its effectiveness is evaluated through various numerical examples, because it is difficult to guarantee the performance of the proposed method theoretically due to a probabilistic nature of the PSO. The simulation results demonstrate its effectiveness clearly.

Keywords: Optimization based controller synthesis; Robust controller synthesis; Controller constraints and structure; Parametric optimization; Evolutionary algorithms; Probabilistic robustness.

1. INTRODUCTION

In practical control engineering, it is crucial to obtain reduced-order/fixed-structure controllers due to limitation of available computer resource and necessity of on-site controller tuning. This paper is concerned with a direct way to attack such a problem.

As for proportional-integral-derivative (PID) or lead-lag compensators, some progress has been made on this problem recently. Various deign methods of such compensators satisfying not only stability but also H_∞ specifications have been proposed [Ho and Lin, 2003, Ho, 2003, Blanchini et al., 2004, Hwang and Hsiao, 2002]. It would be, however, difficult to extend these methods to a broader class of fixed-structured controllers, because they strongly depend on the specific (such as PID) structure. While, as for more general framework of H_∞ controller design subject to the fixed-order/fixed-structure, most approaches utilize linear matrix inequality (LMI) formulae. They try to obtain a local optimal solution through LMI iterations [e.g., Iwasaki and Skelton, 1995, Apkarian et al., 2003], some of which may be suitable for multi-objective controllers. Further, Ebihara et al. [2004] and Saeki [2006a,b] keep the controller variables directly in LMI to cope with the fixed-structure constraints. However, it seems to be difficult for any of these methods to treat both the controller structure and the multiple specifications simultaneously. More importantly, since it requires deep understanding of robust control theory and semi-definite programming (SDP), it may not be easy for most practical engineers to enjoy these sophisticated approaches. This could be a serious barrier from the viewpoint of practical use.

Contrary to the above deterministic approaches, Calafiore et al. [2000] proposed to use a probabilistic one based on randomized algorithms because the problem is inherently an NP hard [Fu and Luo, 1997]. As an extension of this line of research, Fujisaki et al. [2006] provided a mixed probabilistic/deterministic approach to aim at computational efficiency. These approaches give us a discerning remedy when we cannot obtain any solution within reasonable time by the existing deterministic approaches. However, it is not clear how often these methods outperform the existing LMI ones (as shown in the numerical examples of this paper). In addition, they require some skill/understanding of both randomization and robust control theory. So, it might be difficult to claim that the probabilistic approach is easy to use for practical engineers.

On the other hand, Kennedy and Eberhart [1995] recently proposed the particle swarm optimization (PSO) algorithm which is a swarm intelligence technique and is one of the evolutionary computation algorithms. PSO has attracted much attention in recent years, and further a lot of research has been made to improve the performance of the original PSO [see e.g., Sedlaczek and Eberhard, 2006, Parsopoulos and Vrahatis, 2002, Kadirkamanathan et al., 2006, Sedlaczek and Eberhard, 2004, Kim et al., 2007, and the references therein]. In this line of researches, Sedlaczek and Eberhard [2006] developed an augmented Lagrangian PSO (ALPSO) algorithm to handle the optimization problems subject to equality/inequality constraints. Therefore, if we can utilize PSO for the fixed-structured controller design, it would be a great help for practical engineers. However, concerning to the constrained optimization which plays a crucial role in controller design, ALPSO has some drawbacks. Namely, it is based on the assumption that the objective function is differentiable, and the algorithm becomes complex due to the augmented Lagrangian. Therefore, an alternative simple way to handle constraints in PSO is desired.

The purpose of this paper is to develop an easy-to-use design method for fixed-structure controllers satisfying multiple H_∞ specifications. In order to solve such a design problem without any complicated pre-processing, we first provide a method to handle the optimization problems subject to inequality constraints by PSO in such a way that we can fully enjoy the merits of PSO in contrast with ALPSO. Second, it is shown how to obtain a fixed structure controllers satisfying multiple H_∞ specifications based on the developed optimization technique. Third, its effectiveness is evaluated through extensive simulation studies, because it is difficult to guarantee the performance of the proposed method theoretically due to a probabilistic nature of PSO.

The following notation will be used hereafter: for given vectors $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{y} \in \mathbb{R}^n$, $\boldsymbol{x} \leq \boldsymbol{y}$ means element-wise inequality.

2. PROBLEM FORMULATION

Consider the linear time-invariant closed-loop system $\Sigma[\boldsymbol{x}]$ described by

$$\begin{bmatrix} \boldsymbol{z} \\ y \end{bmatrix} = G(s) \begin{bmatrix} \boldsymbol{w} \\ u \end{bmatrix}, \ u = K(s;\boldsymbol{x})y \tag{1}$$

where $G(s)$ is the generalized plant, $K(s;\boldsymbol{x})$ is the fixed-structure controller which is determined by the design parameter $\boldsymbol{x} := (x_1, x_2, \cdots, x_n)^T \in \mathbb{R}^n$. The vectors $\boldsymbol{z}$ and $\boldsymbol{w}$ are defined by $\boldsymbol{z} := (z_1, z_2, \cdots, z_m)^T$ where $z_i \in \mathbb{R}^{p_i}$ and $\boldsymbol{w} := (w_1, w_2, \cdots, w_m)^T$ where $w_i \in \mathbb{R}^{q_i}$. The signals $z_i \in \mathbb{R}^{p_i}$, $w_i \in \mathbb{R}^{q_i}$, $y \in \mathbb{R}^{p_0}$ and $u \in \mathbb{R}^{q_0}$ are the controlled output vector, the external input vector, the measurement vector and the control input vector, respectively.

Let $\lambda_i(\Sigma[\boldsymbol{x}])$ denote the ith pole of the system $\Sigma[\boldsymbol{x}]$ and $\lambda_{\max}(\Sigma[\boldsymbol{x}])$ be the pole whose real part is greater than any other poles, i.e.,

$$\mathrm{Re}[\lambda_{\max}(\Sigma[\boldsymbol{x}])] = \max_i \{\mathrm{Re}[\lambda_i(\Sigma[\boldsymbol{x}])], \ \forall i\}. \tag{2}$$

Further, let $T_{z_i w_i}(s;\boldsymbol{x})$ denote the transfer matrix from $w_i \in \boldsymbol{w}$ to $z_i \in \boldsymbol{z}$ for $i = 1, 2, \cdots, m$.

Now the optimization-based controller synthesis problem considered in this paper is stated as follows: given the objective function

$$J(\boldsymbol{x}) := \|T_{z_1 w_1}(s;\boldsymbol{x})\|_\infty \tag{3}$$

and the admissible level $\gamma_i > 0$, find the design parameter vector $\boldsymbol{x} \in \mathbb{R}^n$ which minimizes $J(\boldsymbol{x})$ while satisfying the following multiple constraint conditions:

(C1) $\mathrm{Re}[\lambda_{\max}(\Sigma[\boldsymbol{x}])] < 0$,
(C2) $\|T_{z_i w_i}(s;\boldsymbol{x})\|_\infty < \gamma_i$ for $i = 2, 3, \cdots, m$.

The case where the following form of the objective function

$$J(\boldsymbol{x}) := \mathrm{Re}[\lambda_{\max}(\Sigma[\boldsymbol{x}])] \tag{4}$$

is adopted instead of (3) is also considered.

In the following sections, a concrete procedure to determine the design parameter $\boldsymbol{x} \in \mathbb{R}^n$ based on a novel constrained PSO algorithm will be presented.

3. CONSTRAINED PARTICLE SWARM OPTIMIZATION ALGORITHM

In this section, we first briefly describe the conventional PSO algorithm proposed by Kennedy and Eberhart [1995]. Then, it will be presented how to handle the constraint conditions in PSO algorithm, which plays a crucial role in controller design problems.

3.1 Basic PSO algorithm

Consider the following optimization problem:

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}), \ \boldsymbol{x} \in \mathbb{R}^n, \tag{5}$$

where the objective function $f : \mathbb{R}^n \to \mathbb{R}$ and the initial search space $\mathbb{D} \subset \mathbb{R}^n$, which is supposed to contain the desired design parameters x_i $(i = 1, 2, \cdots, n)$, is given by the designer in advance.

The PSO algorithm uses a swarm consisting of n_p particles to search the optimal solution $\boldsymbol{x}^* \in \mathbb{R}^n$ of (5). The position of the ith particle is denoted as

$$\boldsymbol{x}_i := (x_{i,1}, x_{i,2}, \cdots, x_{i,n})^T \in \mathbb{R}^n \tag{6}$$

where $i \in \{1, 2, \cdots, n_p\}$, and its velocity is denoted as

$$\boldsymbol{v}_i := (v_{i,1}, v_{i,2}, \cdots, v_{i,n})^T \in \mathbb{R}^n. \tag{7}$$

Then, the position of the ith particle, $\boldsymbol{x}_i \in \mathbb{R}^n$, is updated by

$$\boldsymbol{x}_i^{k+1} = \boldsymbol{x}_i^k + \boldsymbol{v}_i^{k+1}, \tag{8}$$

$$\boldsymbol{v}_i^{k+1} = c_0 \boldsymbol{v}_i^k + c_1 r_{1,i}^k (\boldsymbol{x}_i^{\mathrm{best},k} - \boldsymbol{x}_i^k) + c_2 r_{2,i}^k (\boldsymbol{x}_{\mathrm{swarm}}^{\mathrm{best},k} - \boldsymbol{x}_i^k), \tag{9}$$

where the inertia factor c_0, the cognitive scaling factor c_1 and the social scaling factor c_2, which are given by the designer, influence on the particle trajectories and thus the convergence and search diversity properties. The random numbers $r_{1,i}^k$ and $r_{2,i}^k$ are uniformly distributed in $[0, 1]$ and represent the stochastic behaviors. In (9), $\boldsymbol{x}_i^{\mathrm{best},k}$ is defined as

$$\boldsymbol{x}_i^{\mathrm{best},k} := \arg\min_{\boldsymbol{x}_i^j} \{f(\boldsymbol{x}_i^j), 0 \leq j \leq k\} \tag{10}$$

and denotes the best previously obtained position of the ith particle. Also, $\boldsymbol{x}_{\mathrm{swarm}}^{\mathrm{best},k}$ is defined as

$$\boldsymbol{x}_{\mathrm{swarm}}^{\mathrm{best},k} := \arg\min_{\boldsymbol{x}_i^k} \{f(\boldsymbol{x}_i^k), \ \forall i\} \tag{11}$$

and denotes the best position in the entire swarm at the current iteration k.

Then, the PSO algorithm consists of the following steps:

[Step 0] Initialize n_p particles with randomly chosen positions in $\mathbb{D}$ and evaluate the corresponding objective function value at each position. Set $k = 0$. Determine $\boldsymbol{x}_i^{\mathrm{best},0}$ and $\boldsymbol{x}_{\mathrm{swarm}}^{\mathrm{best},0}$.

[Step 1] If the termination criterion is satisfied, the algorithm terminates with the solution

$$\boldsymbol{x}^* := \arg\min_{\boldsymbol{x}_i^j}\{f(\boldsymbol{x}_i^j),\ \forall i,j\}. \tag{12}$$

Otherwise, go to Step 2.

[Step 2] Apply (8) and (9) to all particles and evaluate the corresponding objective function value at each position. Set $k = k+1$. Determine $\boldsymbol{x}_{\mathrm{swarm}}^{\mathrm{best},k}$ and $\boldsymbol{x}_i^{\mathrm{best},k}$, and then go to Step 1.

In this paper, the parameters c_0, c_1 and c_2 in (9) are set as $c_0 = 0.9, c_1 = c_2 = 0.8$, and the iteration of (8) and (9) is terminated when the iteration number k exceeds the predefined number $k_{\max}$.

3.2 Constrained PSO algorithm

In optimization-based controller design problems, it is crucial to take the given constraint conditions into account in the optimization process.

Therefore, a reasonable and reliable way to handle such constraints in PSO framework is proposed in this subsection, which exploits the flexibility of PSO and does not destroy any merits of PSO.

First, consider the optimization problem subject to multiple constraint conditions:

$$\min_{\boldsymbol{x}\in\mathbb{F}} f(\boldsymbol{x}), \quad \mathbb{F} = \{\boldsymbol{x} \in \mathbb{R}^n |\ \boldsymbol{h}(\boldsymbol{x}) < 0\}, \tag{13}$$

where the function $\boldsymbol{h} : \mathbb{R}^n \to \mathbb{R}^m$ and $\mathbb{F}$ denote the constraints and the feasible region, respectively. Here, it is assumed that $\mathbb{F}$ is not empty. Next, a novel way to handle the constraints in PSO framework is presented. We first find a virtual objective function $f_v(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}$ which satisfies the following two properties simultaneously:

(P1) $f_v(\boldsymbol{x}) < 0$ holds for any $\boldsymbol{x} \in \mathbb{F}$,
(P2) $f_v(\boldsymbol{x}_a) < f_v(\boldsymbol{x}_b)$ holds whenever $f(\boldsymbol{x}_a) < f(\boldsymbol{x}_b)$ is satisfied.

Note that such $f_v(\boldsymbol{x})$ always exists. In fact, one possible candidate of $f_v(\boldsymbol{x})$ is

$$f_v(\boldsymbol{x}) := \arctan\{f(\boldsymbol{x})\} - \frac{\pi}{2}. \tag{14}$$

Then, based on the above virtual objective function $f_v(\boldsymbol{x})$, the constrained optimization problem (13) is modified as the following unconstrained one:

$$\underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}}\ f_m(\boldsymbol{x}) \tag{15}$$

with

$$f_m(\boldsymbol{x}) := \begin{cases} h_{\max}(\boldsymbol{x}) & \text{if } h_{\max}(\boldsymbol{x}) > 0 \\ f_v(\boldsymbol{x}) & \text{otherwise} \end{cases} \tag{16}$$

where

$$h_{\max}(\boldsymbol{x}) := \max\left[h_1(\boldsymbol{x}), h_2(\boldsymbol{x}), \cdots, h_m(\boldsymbol{x})\right] \tag{17}$$

and $h_i(\boldsymbol{x})$ denotes the ith entry of $\boldsymbol{h}(\boldsymbol{x})$ in (13). It is obvious that we can obtain a solution of the constrained optimization problem (13) by optimizing the unconstrained problem (15)-(16) using the ordinary PSO scheme given in Section 3.1.

It may be in order to give some remarks on the merits of the proposed method. Most important advantage of the proposed method is that it is applicable to a broad class of problems, since it does not require for $f(\boldsymbol{x})$ and $h(\boldsymbol{x})$ to be continuous, differentiable nor convex with respect to $\boldsymbol{x}$. This feature is in contrast with ALPSO by Sedlaczek and Eberhard [2006]. In addition, the proposed method does not introduce any additional decision variables such as the coefficients of the augmented Lagrangian in ALPSO.

3.3 Reduction of computational burden

This subsection discusses how to improve the computational efficiency in the proposed method.

First of all, note that since PSO needs only $\boldsymbol{x}_{\mathrm{swarm}}^{\mathrm{best},k}$ and $\boldsymbol{x}_i^{\mathrm{best},k}$ in its update process (8) and (9), it is enough to judge whether each particle could be such best ones or not. Thus, evaluation of the accurate value of the objective function performed in [Step 2] of the basic PSO algorithm (Section 3.1) is can be skipped for most particles.

Suppose $\boldsymbol{x}_{\mathrm{swarm}}^{\mathrm{best},k}$ and $\boldsymbol{x}_i^{\mathrm{best},k}$ are determined by evaluating $f_m(\boldsymbol{x}_1^k), f_m(\boldsymbol{x}_2^k), \cdots, f_m(\boldsymbol{x}_{n_p}^k)$ in this order. In this case, if we can conclude that $f_m(\boldsymbol{x}_i^k)$ is greater than

$$\max\left[f_m(\boldsymbol{x}_i^{\mathrm{best},k-1}), \min_{j=1,2,\cdots,i-1}\{f_m(\boldsymbol{x}_j^k)\}\right], \tag{18}$$

it implies that the particle $\boldsymbol{x}_i^k$ can be neither $\boldsymbol{x}_{\mathrm{swarm}}^{\mathrm{best},k}$ nor $\boldsymbol{x}_i^{\mathrm{best},k}$. Therefore, more precise evaluation of $f_m(\boldsymbol{x}_i^k)$ is not required at all.

Based on this fact, we can reduce the computational burden in the proposed method as shown below:

(A) Exploitation of the structure of $f_m(\cdot)$:

The definition in (16) indicates that the value of f_m equals to f_v or $h_i > 0$ $(i = 1, 2, \cdots, m)$ and if any of them turns out to be greater than the value of (18), no more calculation of other functions are needed. For example, if h_1 (> 0) exceeds (18), then we can skip the calculation of h_j $(2 \le j \le m)$ and f_v. This improves the computational efficiency.

(B) Exploitation of the properties of $f_v(\cdot)$ and $\boldsymbol{h}(\cdot)$:

For some cases, it costs much cheaper to evaluate whether the function value exceeds some given level or not, instead of calculating its precise value. Hence, if an objective function f_m is composed of such functions, the computational burden could be reduced by checking whether it exceeds (18) or not, before the precise calculation of the function value. This is the case for controller design problems dealing with the H_∞ norm and the system poles.

4. SYNTHESIS OF FIXED-STRUCTURE H_∞ CONTROLLERS

In this section, we present a design procedure based on the constrained PSO algorithm developed in Section 3 for a fixed-structure controller satisfying the given performance specifications. Also, several numerical examples are given to evaluate its effectiveness.

4.1 Controller synthesis procedure

First, note that it is straightforward to obtain the desired fixed-structure controller if we introduce the constrained PSO technique developed in Section 3.

In order to design a controller by minimizing $J(\boldsymbol{x})$ in (3) or (4) subject to (C1) and (C2) in Section 2, it is enough to solve the optimization problem of form (15)-(16). Here, $f_v(\boldsymbol{x})$ and $\boldsymbol{h}(\boldsymbol{x})$ could be set in the following manners:

[**Case A**] If $J(\boldsymbol{x}) := \|T_{z_1w_1}(s;\boldsymbol{x})\|_\infty$ is given, an example of $f_v(\boldsymbol{x})$ satisfying (P1) and (P2) in Section 3.2 is given by

$$f_v(\boldsymbol{x}) = -\|T_{z_1w_1}(s;\boldsymbol{x})\|_\infty^{-1}. \tag{19}$$

Then, the constraint function $\boldsymbol{h}(\boldsymbol{x})$ is set as

$$\boldsymbol{h}(\boldsymbol{x}) = \begin{bmatrix} \mathrm{Re}[\lambda_{\max}(\Sigma[\boldsymbol{x}])] \\ \|T_{z_2w_2}(s;\boldsymbol{x})\|_\infty - \gamma_2 \\ \vdots \\ \|T_{z_mw_m}(s;\boldsymbol{x})\|_\infty - \gamma_m \end{bmatrix}. \tag{20}$$

[**Case B**] If $J(\boldsymbol{x}) = \mathrm{Re}[\lambda_{\max}(\Sigma[\boldsymbol{x}])]$ is given, it is enough to choose

$$f_v(\boldsymbol{x}) = \mathrm{Re}[\lambda_{\max}(\Sigma[\boldsymbol{x}])], \tag{21}$$

and

$$\boldsymbol{h}(\boldsymbol{x}) = \begin{bmatrix} \mathrm{Re}[\lambda_{\max}(\Sigma[\boldsymbol{x}])] \\ \|T_{z_2w_2}(s;\boldsymbol{x})\|_\infty - \gamma_2 \\ \vdots \\ \|T_{z_mw_m}(s;\boldsymbol{x})\|_\infty - \gamma_m \end{bmatrix}. \tag{22}$$

Therefore, all we have to do is to solve (15)-(16) using the above defined $f_v(\boldsymbol{x})$ and $\boldsymbol{h}(\boldsymbol{x})$ via the PSO algorithm given in Section 3.1.

At the current stage, it is difficult to guarantee the performance theoretically because the property of PSO has not been well analyzed. Thus, it would be better to evaluate roughly the probability of finding feasible solutions through the proposed method. Also, it may not be clear if we can find a controller achieving better performance compared to the existing methods. Therefore, in the following, we evaluate these two points in various numerical examples.

4.2 Numerical examples

Example 1 Consider the unity feedback system $\Sigma_1[\boldsymbol{x}]$ consists of the linearized model of the experimental magnetic levitation system

$$P(s) = \frac{7.147}{(s-22.55)(s+20.9)(s+13.99)} \tag{23}$$

and the PID controller

$$K(s;\boldsymbol{x}) = 10^{x_1}\left(1 + \frac{1}{10^{x_2}s} + \frac{10^{x_3}s}{1+10^{(x_3-x_4)}s}\right) \tag{24}$$

where $\boldsymbol{x} := (x_1, x_2, x_3, x_4)^T$ denotes the design parameter vector. Each element of $\boldsymbol{x}$ corresponds to the proportional gain, the integral time, the derivative time, and the parameter to assure the properness of the obtained controller, respectively (Refer to Sugie et al. [1993] and Kim et al. [2007] for details). Suppose that the initial search space of the design parameter is given by

$$\mathbb{D} := \{\boldsymbol{x} \in \mathbb{R}^4 | \ [2,-1,-1,2]^T \le \boldsymbol{x} \le [4,1,1,3]^T\}. \tag{25}$$

Then, our aim is to find $\boldsymbol{x} \in \mathbb{R}^4$ which minimizes $\mathrm{Re}[\lambda_{\max}(\Sigma_1[\boldsymbol{x}])]$ subject to the following multiple H_∞ constraints:

$$\|W_S(s)S(s;\boldsymbol{x})\|_\infty < 1, \ \ \|W_T(s)T(s;\boldsymbol{x})\|_\infty < 1 \tag{26}$$

where $S(s;\boldsymbol{x}) := \frac{1}{1+P(s)K(s;\boldsymbol{x})}$ is the sensitivity function, $T(s;\boldsymbol{x}) := \frac{P(s)K(s;\boldsymbol{x})}{1+P(s)K(s;\boldsymbol{x})}$ is the complementary sensitivity function, and $W_S(s)$ and $W_T(s)$ are set as

$$W_S(s) := 5/(s+0.1), \tag{27}$$

$$W_T(s) := 4.3867\times 10^{-7}(s+0.066) \times (s+31.4)(s+88)(10^4/(s+10^4))^3. \tag{28}$$

Note that the conventional methods are not able to handle problems like this one which has multiple H_∞ constraints.

In order to design $K(s;\boldsymbol{x})$, the developed constrained PSO algorithm is introduced, where the swarm size $n_p = 100$ and the maximum iteration number $k_{\max} = 400$. We run the algorithm 100 times with different initial populations. Then, we succeed in obtaining feasible solutions (i.e., controllers satisfying the constraints are obtained) in 93 trials out of 100 trials.

Fig. 1 shows the convergence property of the proposed constrained PSO algorithm for this example. 93 lines in this figure correspond to 93 succeeded trials and horizontal axis shows the elapsed calculation time. Each line in the figure shows the current best obtained controller performance at elapsed calculation time. This figure shows that a controller satisfying the given multiple robust performance criteria could be found within 400 seconds in average trial. It means that if we run the developed constrained PSO algorithm several times, a feasible solution can be found in an acceptable time and with a high probability of success.

In our experiment, the best design parameter is

$$\boldsymbol{x}^* = [3.2583, -0.8157, -0.7564, 2.3286]^T, \tag{29}$$

and thus the corresponding PID controller is obtained as

$$K(s) = 1821.6\left(1 + \frac{1}{0.1529s} + \frac{0.1752s}{1+(8.2224\times 10^{-4})\,s}\right). \tag{30}$$

Though the figure is omitted due to the page limitation, it is confirmed that constraints on the sensitivity function $S(s;\boldsymbol{x}^*)$ and the complementary sensitivity function $T(s;\boldsymbol{x}^*)$ in (26) are satisfied.

Example 2 Consider the linear time-invariant generalized plant $G(s)$ described by

$$\begin{bmatrix} \dot{x}_p \\ z \\ y \end{bmatrix} = \begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{bmatrix} \begin{bmatrix} x_p \\ w \\ u \end{bmatrix}. \tag{31}$$

In this example, the plant is the linearized model of F-8 aircraft treated in Apkarian et al. [2003], Calafiore et al. [2000] and Saeki [2006b]. (see them for the detail of system parameters) The controller $K(s)$ is the first-order output feedback one described by

$$\begin{aligned} \dot{x}_k &= A_k x_k + B_k y, \\ u &= C_k x_k + D_k y. \end{aligned} \tag{32}$$

For the above system, the optimization-based controller design problem is as follows: minimize $\|T_{zw}(s)\|_\infty$ of the closed-loop system $\Sigma_2[\boldsymbol{x}]$ consists of $G(s)$ and $K(s)$ subject to the constraint on internal stability.

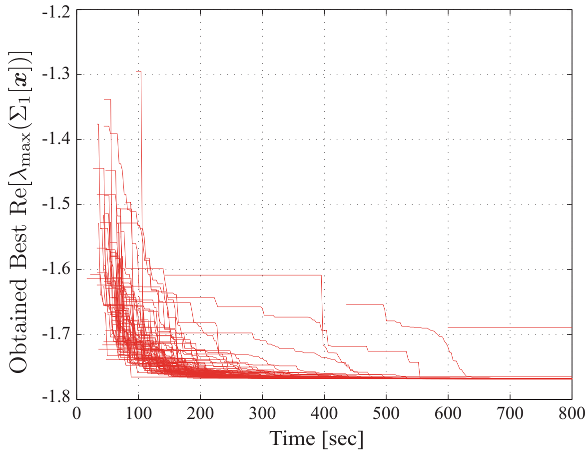

Fig. 1. Convergence property for Example 1

The above-mentioned controller design problem can be solved via the proposed constrained PSO method by setting the design parameter $\boldsymbol{x} := (x_1, x_2, \cdots, x_9)^T \in \mathbb{R}^9$ as

$$\left[\begin{array}{c|c} A_K & B_K \\ \hline C_K & D_K \end{array}\right] = \left[\begin{array}{c|cc} x_1 & x_2 & x_3 \\ \hline x_4 & x_5 & x_6 \\ x_7 & x_8 & x_9 \end{array}\right]. \tag{33}$$

The initial search space is supposed to be given by

$$\mathbb{D} := \{\boldsymbol{x} \in \mathbb{R}^9 | -5 \le x_i \le 5,\ i = 1, 2, \cdots, 9\}. \tag{34}$$

Now, in order to obtain $\boldsymbol{x}^*$, we solve (15)-(16) with

$$f_v(\boldsymbol{x}) = -\|T_{zw}(s;\boldsymbol{x})\|_\infty^{-1}, \tag{35}$$

$$h(\boldsymbol{x}) = \mathrm{Re}\,[\lambda_{\max}(\Sigma_2[s;\boldsymbol{x}])], \tag{36}$$

which corresponds to [Case A] in Section 4.1. We run the algorithm 81 times with the number of particles $n_P = 300$ and the maximum iteration number $k_{\max} = 400$. In all 81 trials, feasible solutions are found. Average time consumed to achieve 400 iterations was about 650[sec]. The experimental results are summarized in Table 1. It is important to note that, if required, we can readily add an additional performance specification on the pole placement, and a desirable controller can be developed through the proposed method without any difficulty.

The obtained best design parameter $\boldsymbol{x}^*$ is

$$\left[\begin{array}{c|cc} x_1^* & x_2^* & x_3^* \\ \hline x_4^* & x_5^* & x_6^* \\ x_7^* & x_8^* & x_9^* \end{array}\right] = \left[\begin{array}{c|cc} -21.1183 & -1.5886 & 11.0822 \\ \hline -2.9907 & 0.4011 & -0.5268 \\ -20.0049 & 0.4298 & -0.9064 \end{array}\right], \tag{37}$$

and the corresponding H_∞ performance index is

$$\|T_{zw}(s;\boldsymbol{x}^*)\|_\infty = 1.7092. \tag{38}$$

In Calafiore et al. [2000], they applied a randomized algorithm to solve the above problem, and then obtained $\|T_{zw}(s)\|_\infty = 4.8937$. Also, in Apkarian et al. [2003], they used a partially augmented Lagrangian method to obtain $\|T_{zw}(s)\|_\infty = 1.821$. From these observations, we can see that although our method is by far the simplest one, it can produce better results in comparison with the conventional methods.

Table 1. Statistical result of Example 2

	Best	Mean	Worst	St. Dev.
$\|T_{zw}(s)\|_\infty$	1.7092	1.7775	2.3732	0.0996

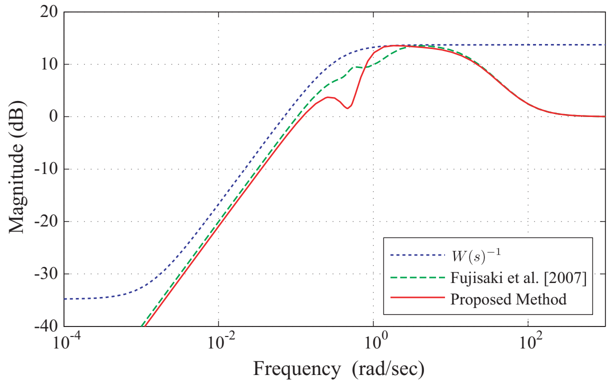

Fig. 2. Bode plot of the sensitivity function with obtained controller

Example 3 In order to show the effectiveness of our controller design methodology over against the mixed probabilistic/deterministic approach by Fujisaki et al. [2006, 2007], the example presented in Fujisaki et al. [2007] is handled by the proposed method.

Consider the unity feedback system $\Sigma_3[\boldsymbol{x}]$ consists of

$$P(s) = \frac{17\,(1+s)\,(1+16s)\,(1-s+s^2)}{s\,(1-s)\,(90-s)\,(1+s+4s^2)} \tag{39}$$

and

$$K(s) = \frac{\theta_0 + \alpha_0 s + \theta_2 s^2}{1 + \mu_0 s + \beta_2 s^2}. \tag{40}$$

Let $\boldsymbol{x} := (\theta_0, \alpha_0, \theta_2, \mu_0, \beta_2)^T$ denote the design parameter vector. Its initial search space is supposed to be given by

$$\mathbb{D} := \{\boldsymbol{x} \in \mathbb{R}^5 |\ -5 \le x_i \le 5,\ i = 1, 2, \cdots, 5\} \tag{41}$$

based on the problem setting in Fujisaki et al. [2007]. Note that the search spaces of θ_0 and θ_2 are not specified in their method, since these are not determined in a probabilistic way. In Fujisaki et al. [2007], two types of fixed-structure controllers are designed following two different performance specifications: (i) the first one is designed to satisfy the following pole placement condition

$$\mathrm{Re}\,[\lambda_{\max}(\Sigma_3[s;\boldsymbol{x}])] < -0.2, \tag{42}$$

(ii) the second one is designed to satisfy the following H_∞ performance condition

$$\|W(s)S(s;\boldsymbol{x})\|_\infty < 1 \tag{43}$$

where $W(s)$ is given as

$$W(s) := \frac{55\,(1+3s)}{1+800s}. \tag{44}$$

Here, in order to find $\boldsymbol{x} \in \mathbb{R}^5$ which satisfies both of these conditions (42)-(43) simultaneously, we solve (15)-(16) with

$$f_v(\boldsymbol{x}) = \mathrm{Re}\,[\lambda_{\max}(\Sigma_3[s;\boldsymbol{x}])], \tag{45}$$

$$h(\boldsymbol{x}) = \begin{bmatrix} \mathrm{Re}\,[\lambda_{\max}(\Sigma_3[s;\boldsymbol{x}])] \\ \|W(s)S(s;\boldsymbol{x})\|_\infty - 1 \end{bmatrix}, \tag{46}$$

which corresponds to [Case B] in Section 4.1. The number of particles is set as $n_p = 300$ and the maximum iteration number is $k_{\max} = 200$. Then, we run the algorithm 173 times, and feasible solutions are found in 68 trials (i.e., 39%).

The best controller obtained from the above procedure is

$$K(s;\boldsymbol{x}^*) = \frac{-0.5891 - 0.7339s + -2.5918s^2}{1 - 0.5578s - 1.555s^2}, \qquad (47)$$

and the poles of the corresponding closed-loop system are -19.4482, $-0.5780 \pm 0.4914j$, $-0.5834 \pm 0.2620j$ and $-0.5885 \pm 0.0737j$. Thus, $\mathrm{Re}\,[\lambda_{\max}(\Sigma_3[s;\boldsymbol{x}])] = -0.5780$, which verifies that the pole placement specification is guaranteed with a considerable margin. Fig. 2 shows the gain plot of the corresponding sensitivity function $S(s;\boldsymbol{x}^*)$, which verifies the given constraint condition is guaranteed.

5. CONCLUSION

In this paper, we have proposed a design method of fixed-structure robust controllers satisfying multiple H_∞ norm specifications by using a sort of randomized algorithms, which is easy to use for most practical engineers. First, a new PSO based optimization tool which can handle inequality constraints is developed. Since the method requires few assumptions on the objective function and the algorithm is so simple, it is applicable for a broad class of non-convex problems directly. In addition, it is shown how to reduce the computation burden including the exploitation of the specific property of H_∞ control performance. Second, it is shown how to obtain a fixed-structure controllers satisfying multiple H_∞ specifications via the developed method. It directly deals the multiple specifications without introducing any relaxations or conservative assumptions, and it is straightforward to take any fixed-structure into account in the controller design. It should be stressed that the method is so simple that most practical engineers can use it without any difficulty. Third, its effectiveness has been evaluated through extensive numerical examples, where it is observed that the proposed method always succeeds to find a fixed-structure controller which outperforms the existing methods in various cases such as constant output feedback and PID controllers.

Note, however, that it is not guaranteed that the proposed method always can find a solution even if feasible solutions exist, because the PSO inherently relies on the randomized variables. Therefore, it is important to study the global convergence of the PSO theoretically. Nevertheless, all numerical examples exhibit that it is very likely to obtain a controller which outperforms the existing ones if we run the algorithm a few dozen times with different initial conditions. Therefore, combined with its easy-to-use property, the method is NOT trivial at all from the practical point of view.

REFERENCES

P. Apkarian, D. Noll, and H. Duong Tuan. Fixed-order H_∞ control design via a partially augmented Lagrangian method. *Int. J. Robust Nonlin. Contr.*, 13(12):1137–1148, 2003.

F. Blanchini, A. Lepschy, S. Miani, and U. Viaro. Characterization of PID and lead/lag compensators satisfying given H_∞ specifications. *IEEE Trans. Automat. Contr.*, 48(5):736–740, 2004.

G. Calafiore, F. Dabbene, and R. Tempo. Randomized algorithms for reduced order H_∞ controller design. In *Proc. American Contr. Conf.*, pages 3837–3839, 2000.

Y. Ebihara, K. Tokuyama, and T. Hagiwara. Structured controller synthesis using LMI and alternating projection method. *Int. J. of Contr.*, 77(12):1137–1147, 2004.

M Fu and Z.-Q. Luo. Computational complexity of a problem arising in fixed order output feedback design. *Systems & Control Letters*, 30(5):209–215, 1997.

Y. Fujisaki, Y. Oishi, and R. Tempo. A mixed probabilistic/deterministic approach to fixed order H_∞ controller design. In *The 45th IEEE Conf. Decision and Control*, pages 3554–3559, 2006.

Y. Fujisaki, Y. Oishi, and R. Tempo. Mixed deterministic/randomized methods for fixed order controller design. Technical report, March 2007.

M. Ho. Synthesis of H_∞ PID controllers: A parametric approach. *Automatica*, 39(6):1069–1075, 2003.

M. Ho and C. Lin. PID controller design for robust performance. *IEEE Trans. Automat. Contr.*, 48(8): 1404–1409, 2003.

C. Hwang and C.-Y. Hsiao. Solution of a non-convex optimization arising in PI/PID control design. *Automatica*, 38(11):1895–1904, 2002.

T. Iwasaki and R. E. Skelton. All fixed order H_∞ controllers: Observer-based structure and covariance bounds. *IEEE Trans. Automat. Contr.*, 40:512–516, 1995.

V. Kadirkamanathan, K. Selvarajah, and P.J. Fleming. Stability analysis of the particle dynamics in particle swarm optimizer. *IEEE Trans. on Evolutionary Computation*, 10(3):245–255, 2006.

J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proc. IEEE Int. Conf. Neural Networks*, volume 4, pages 1942–1948, 1995.

T.-H. Kim, I. Maruta, and T. Sugie. Particle swarm optimization based on robust PID controller tuning scheme. In *Proceedings of the 46th IEEE Conference on Decision and Control*, pages 200–205, December 2007.

K. E. Parsopoulos and M. N. Vrahatis. Recent approaches to global optimization problems through particle swarm optimization. *Natural Computing*, 1(2–3):235–306, 2002.

M. Saeki. Fixed structure PID controller design for standard H_∞ control problem. *Automatica*, 42(1):93–100, 2006a.

M. Saeki. Static output feedback design for H_∞ control by descent method. In *The 45th IEEE Conf. Decision and Control,*, pages 5156–5161, 2006b.

K. Sedlaczek and P. Eberhard. Optimization of Nonlinear Mechanical Systems under Constraints with the Particle Swarm Method. *PAMM*, 4(1):169–170, 2004.

K. Sedlaczek and P. Eberhard. Using augmented Lagrangian particle swarm optimization for constrained problems in engineering. *Structural and Multidisciplinary Optimization*, 32(4):277–286, 2006.

T. Sugie, K. Simizu, and J. Imura. H_∞ control with exact linearization and its application to magnetic levitation systems. In *IFAC 12th World Congress*, volume 4, pages 363–366, July 1993.