Computations of Probabilistic Output Admissible Set for Uncertain Constrained Systems

Takeshi Hatanaka and Kiyotsugu Takaba

Department of Applied Mathematics and Physics
Graduate School of Informatics
Kyoto University

Computations of Probabilistic Output Admissible Set for Uncertain Constrained Systems

Takeshi Hatanaka and Kiyotsugu Takaba

Abstract

This paper considers uncertain constrained systems, and develops two algorighms for computing a probabilistic output admissible (POA) set which is a set of initial states probabilistically assured to satisfy the constraint. The first algorithm is inspired by an existing randomized sequential technique. The second algorithm alleviates the computational effort based on heuristics. The present algorithms terminate in a finite number of iterations and provides a POA set. Additionally, we can obtain information on the size of the resulting set *a posteriori*. A numerical simulation demonstrates the applicability of the POA set to control system designs.

1 Introduction

Most of practical control systems inherently have state and control constraints due to nonlinear characteristics of actuators or for safety of hardware. This can lead to performance deterioration or even instability if not properly accounted for in design stage. Thus, when we design a control system, it is required not only to achieve a good control performance but also to avoid constraint violations.

A so-called *maximal output admissible (MOA) set* is an useful concept in analysis and control of constrained systems. This is the set of all initial states such that the trajectories starting from them never violate the infinite-time constraint. The MOA set provides a necessary and sufficient condition for constraint fulfillment (Gilbert and Tan, 1991), and gives an insight into analysis of constrained systems. Additionally, the MOA set has been extensively used in control system design schemes such as controller switching strategies (Hirata and Fujita, 1998; Hirata and Fujita, 2000), reference governors (Bemporad *et al.*, 1997; Bemporad and Mosca, 1998; Gilbert *et al.*, 1995; Gilbert and Kolmanovsky, 1999), model predictive controls (Goodwin *et al.*, 2004), and it also relates to the minimal l^∞-induced norm (Shamma, 1996; Blanchini *et al.*, 1997).

The MOA set was first defined for linear time-invariant autonomous systems (Gilbert and Tan, 1991). Then Kolmanovsky and Gilbert (1995) has extended the concept of the MOA set to systems subject to unknown but bounded disturbances, where the MOA set assures constraint fulfillment even in the worst-case disturbance scenario. Since the worst-case paradigm often makes the MOA set small, Gilbert and Kolmanovsky (1998) has reformulated the MOA set on the basis of the information on rate limit or stochastic properties of disturbances. Meanwhile, the MOA set of uncertain constrained systems has also been investigated in (Blanchini, 1994; Blanchini and Miani, 1996; Casavola *et al.*, 2000; Hirata and

Ohta, 2004). Blanchini (1994), Blanchini and Miani (1996) and Casavola *et al.* (2000) have considered systems with polytopic uncertainties, and Hirata and Ohta (2004) time-varying memoryless operators bounded with respect to l_∞-induced norm as uncertainties. It should be noted that the main aim of Blanchini (1994) is to construct a set-induced Lyapunov function which contains the theory of the MOA set. Additionally, the extension of the MOA set to nonlinear systems has been attempted in recent years (Hirata and Ohta, 2005). Note that topics on invariant sets with close relationships to the MOA set are well summarized in the survey paper (Blanchini, 1999).

This paper considers uncertain constrained systems and addresses the computation of the set of initial states which 'robustly' guarantee infinite-time constraint fulfillment on the basis of a probabilistic approach. Previous relevant works include Blanchini (1994), Blanchini and Miani (1996), Casavola *et al.* (2000) and Hirata and Ohta (2004). All of them are based on the worst-case paradigm, that is, they aim at computing a set which assures constraint fulfillment even in the worst uncertainty scenario.

In this paper, we introduce an alternative notion of robustness about constraint fulfillment, where the guarantee of constraint fulfillment is intended not in the deterministic sense (satisfaction against *all* possible uncertainty outcomes), but instead in the probabilistic sense (satisfaction *in probability*). This approach can be seen as a relaxation of the worst-case paradigm where one allows a risk level $\epsilon \in (0, 1)$, and the approach enables us to construct a subset of the state space such that if the system is initialized in any element of the set, the constraints are violated by at most a fraction ϵ of the uncertainty family. This paper refers to the set as an *ϵ-level probabilistic output admissible (POA) set*. The computations of our algorithm does not suffer from the complexity of structure of uncertainties, which enables us to deal with a wide class of uncertain systems that the deterministic methods (Blanchini, 1994; Blanchini and Miani, 1996; Casavola *et al.*, 2000; Hirata and Ohta, 2004) cannot deal with without introducing conservatism: for example, the structured uncertainty, the time-invariant uncertainty and so on. Additionally, our approach allows us to incorporate information on probabilistic properties of uncertainties into the computations of the output admissible set.

In recent years, introducing probability in robustness gained increasing interest in robust control theory (Tempo *et al.*, 2004). Three different methodologies are currently available for robust control synthesis: the approach based on the Vapnik-Chervonenkis theory of learning (Tempo *et al.*, 2004), the scenario approach (G. Calafiore and M.C. Campi, 2004), and the sequential method (Oishi, 2003). Our algorithm is based on the sequential method. The main difference from (Oishi, 2003) is that our objective is to compute not a *point* (a design parameter) but a *set* (an ϵ-level POA set).

In this paper, we consider time-invariant uncertain systems with mild assumptions, and present an algorithm for computing an ϵ-level POA set. The paper is organized as follows. Section 2 formulates the system to be considered in this paper. In Section 3, several important sets are defined. In Section 4, we present a sequential algorithm (Algorithm 1) for computing an ϵ-level POA set. Additionally, it is shown that the sample size due to (Oishi, 2003) provides an ϵ-level POA set (Corollary 1) and that the algorithm finitely terminates (Theorem 1). Since, however, Algorithm 1 requires a great amount of computational effort, Section 5 presents an improved algorithm (Algorithm 2) with small computational effort. Section 6 presents a numerical example and shows the applicability of the ϵ-level POA set to a reference governor algorithm. Conclusions are finally drawn in Section 7.

Notations

The following notations and terminology will be used throughout this paper. $\mathbb{Z}_+$ is the set of nonnegative integers, namely, $\mathbb{Z}_+ = \{0, 1, 2, \cdots\}$. $\mathbf{0}$ and $\mathbf{1}$ are the vectors or matrices with appropriate dimensions whose elements are 0 and 1, respectively. Let $\gamma > 0$, $g, h \in \mathbf{R}^n$, $M \in \mathbf{R}^{m \times n}$, $\mathcal{Z} \subset \mathbf{R}^n$, and $\mathcal{Z}_f := \{z^{(1)}, \cdots, z^{(r)}\} \subset \mathbf{R}^n$. Then, $g \geq h, g > h$ describes the element-wise inequalities, $M(i,:)$ the i-th row of M, $\|M\|$ the maximal singular value of M, len(M) the number of rows, m. $\gamma\mathcal{Z} := \{\gamma x \in \mathbf{R}^n |\ x \in \mathcal{Z}\}$ and $|\mathcal{Z}_f| := r$. The boundary and interior of $\mathcal{Z}$ are denoted, respectively, by bd$(\mathcal{Z})$ and int$(\mathcal{Z})$. $H(M) \subset \mathbf{R}^n$ denotes the convex polyhedron $\{x \in \mathbf{R}^n | Mx \leq \mathbf{1}\}$. Let $H_i(M) = \{x \in \mathbf{R}^n | M(j,:)x \leq \mathbf{1}\ \forall j \neq i\},\ i \in \{1, \cdots, m\}$. If $H(M) = H_i(M)$ holds, then the inequality $M(i,:)x \leq 1$ is called a *redundant inequality* for describing $H(M)$. Otherwise, $M(i,:)x \leq 1$ is called an *active inequality*. Additionally, a description of a polyhedron is *irredundant* if and only if none of the inequalities describing the polyhedron are redundant. Let the functions $\Phi(g;M)$ and $\phi(g;M)$ be

$$\begin{aligned}\Phi(g;M) &:= \max_{x \in H(M)} g^\top x,\\ \phi(g;M) &:= \arg\max_{x \in H(M)} g^\top x / \Phi(g;M),\end{aligned}$$

respectively ($\Phi(g;M)$ is called a *support function* of $H(M)$). Especially, we denote

$$\begin{aligned}\Phi_i(M) &:= \max_{x \in H_i(M)} M(i,:)^\top x,\\ \phi_i(M) &:= \arg\max_{x \in H_i(M)} M(i,:)^\top x / \Phi_i(M).\end{aligned}$$

The set $\mathcal{L}_{\mathcal{Z}_f}(M)$ describes

$$\mathcal{L}_{\mathcal{Z}_f}(M) := \{l \in \{1, \cdots, \mathrm{len}(M)\} | \; \exists i \in \{1, \cdots, r\} \\ \text{s.t. } M(l,:)z^{(i)} = 1\}.$$

For a matrix $Q = Q^\top > 0$ and a scalar $\rho > 0$, $\Omega(Q, \rho)$ denotes the ellipsoid $\Omega(Q, \rho) = \{x \in \mathbf{R}^n | x^\top Qx \le \rho^2\}$ and $\mathcal{B}(\rho)$ the open ball $\mathcal{B}(\rho) = \{x \in \mathbf{R}^n | x^\top x < \rho^2\}$.

2 Probrem Statement

Consider the uncertain discrete-time system

$$\Sigma \begin{cases} x(t+1) = A(\Delta)x(t), & (2.1) \\ c(t) = C(\Delta)x(t), & (2.2) \end{cases}$$

where $t \in \mathbb{Z}_+$, $x(t) \in \mathbf{R}^n$ is the state of Σ, and $c(t) \in \mathbf{R}^{n_c}$ is the auxiliary output that describes state and control constraints. The vector $c(t)$ must be constrained within a prescribed set $\mathcal{C}$ as

$$c(t) \in \mathcal{C} \; \forall t \in \mathbb{Z}_+, \tag{2.3}$$
$$\mathcal{C} := \{c \in \mathbf{R}^{n_c} | \; M_c c \le \mathbf{1}\}, \; M_c \in \mathbf{R}^{m_c \times n_c}.$$

The matrix $\Delta \in \mathbf{R}^{s_1 \times s_2}$ is the time-invariant uncertainty confined in a bounded set $\mathbb{D}$. We assume that the support $\mathbb{D}$ is endowed with a σ-algebra $\mathcal{D}$, and that the probability measure $\mathrm{Prob}_{\boldsymbol{\Delta}}$ is defined over $\mathcal{D}$. Moreover, we assume the existence of a probability density of Δ, and denote it by $f_{\boldsymbol{\Delta}}(\Delta)$. Generally speaking, $f_{\boldsymbol{\Delta}}(\Delta)$ is estimated from available data or prior information. If such prior information is not available, then the probability distribution of Δ should be chosen very carefully and the choice of uniform distribution, which is the worst-case distribution in a certain class of probability measures, is an option (Tempo *et al.*, 2004). This paper denotes the system for a fixed Δ by $\Sigma(\Delta)$. Without loss of generality, we assume that $\mathbf{0} \in \operatorname{int} \mathbb{D}$, and refer to $\Sigma(\mathbf{0})$ as the nominal system.

We assume that the system Σ, the set $\mathbb{D}$ and the probability density $f_{\boldsymbol{\Delta}}(\Delta)$ satisfy the following assumptions.

Assumption 1

(a) $C(\Delta)$ and $A(\Delta)$ are Lebesgue measurable functions of Δ.

(b) $\sup_{\Delta \in \mathbb{D}} \|M_c C(\Delta)\|$ is bounded.

(c) Samples can be efficiently generated in $\mathbb{D}$ according to $f_{\boldsymbol{\Delta}}(\Delta)$.

(d) $A(\Delta)$ is asymptotically stable for any $\Delta \in \operatorname{cl} \mathbb{D}$.

(e) $(A(\mathbf{0}), C(\mathbf{0}))$ is an observable pair.

(f) The set $\mathcal{C}$ is bounded.

Remark 1

In the real world, almost all uncertain systems satisfy Assumptions 1 (a) and (b). Additionally, the items (d) and (e) are not really limitations for the following reasons. The MOA set is generally defined for closed-loop systems designed *a priori*, when it is used in control system design schemes such as controller switching strategies (Hirata and Fujita, 1998; Hirata and Fujita, 2000), reference governors (Bemporad *et al.*, 1997; Bemporad and Mosca, 1998; Gilbert *et al.*, 1995; Gilbert and Kolmanovsky, 1999) and model predictive controls (Goodwin *et al.*, 2004). Thus, Assumption 1 (d) just requires to design a robust controller. The item (e) is assumed without loss of generality (Gilbert and Tan, 1991).

Remark 2

Assumption 1 (c) is satisfied in the following cases.

Case 1: Δ is a vector (a parametric uncertainty) We can easily generate random samples, if $\mathbb{D}$ is an l_p norm ball and the probability density $f_{\boldsymbol{\Delta}}(\Delta)$ has *radial symmetry* with respect to l_p norms. In addition, $\mathbb{D}$ is allowed to be an ellipsoid or an unit simplex.

Case 2: Δ is a matrix (a memoryless full block operator) We can easily obtain random matrices if Δ is bounded with respect to the Frobenius norm, l_1, or l_∞ induced norm. Though the sample generation in the l_2 induced norm ball is more difficult than the above cases, a sophisticated algorithm for computing them is reported in Calafiore *et al.* (2000).

Interested readers are recommended to refer to Calafiore *et al.* (2000) or Tempo *et al.* (2004) for more detail.

3 Definition of Sets

This section defines several important sets.

Definition 1 MOA Set of $\Sigma(\Delta)$

Let $c(t; x, \Delta)$ denote the response of $\Sigma(\Delta)$ for an initial state x, namely $c(t; x, \Delta) := C(\Delta)A^t(\Delta)x$. Then, the MOA set $\mathcal{S}(\Delta)$ and the i-steps output admissible sets $\mathcal{K}_i(\Delta),\ i \in \mathbb{Z}_+$ are defined by

$$\mathcal{S}(\Delta) := \{x \in \mathbf{R}^n |\ c(t; x, \Delta) \in \mathcal{C}\ \forall t \in \mathbb{Z}_+\},$$
$$\mathcal{K}_i(\Delta) := \{x \in \mathbf{R}^n |\ c(t; x, \Delta) \in \mathcal{C}\ \forall t \in \{0, 1, \cdots, i\}\}.$$

About the sets $\mathcal{S}(\Delta)$ and $\mathcal{K}_i(\Delta)$, we have the following well-known results.

Proposition 1

(i) The sequence of sets $\mathcal{K}_i(\Delta)$ is monotonically decreasing in the sense that $\mathcal{K}_i(\Delta) \supseteq \mathcal{K}_{i+1}(\Delta)\ \forall i \in \mathbb{Z}_+$.

(ii) If $\mathcal{K}_i(\Delta) = \mathcal{K}_{i+1}(\Delta)$ holds for some $i \in \mathbb{Z}_+$, then we obtain $\mathcal{S}(\Delta) = \mathcal{K}_i(\Delta)$. Such an i is finite if $A(\Delta)$ is asymptotically stable.

(iii) The set $\mathcal{S}(\Delta)$ is a convex polyhedron, and is bounded if $(C(\Delta), A(\Delta))$ is an observable pair.

(iv) The set $\mathcal{S}(\Delta)$ is positively invariant for $\Sigma(\Delta)$, that is, $x(0) \in \mathcal{S}(\Delta)$ implies that $A^t(\Delta)x(0) \in \mathcal{S}(\Delta)\ \forall t \in \mathbb{Z}_+$.

Proof

See Gilbert and Tan (1991). □

Definition 2 Output Admissibility Index

The output admissibility index $i_o(\Delta)$ is defined by

$$i_o(\Delta) := \min_{i \in \mathbb{Z}_+} i \text{ subject to } \mathcal{K}_i(\Delta) = \mathcal{K}_{i+1}(\Delta).$$

Moreover, we define $i_m := \sup_{\Delta \in \mathbb{D}} i_o(\Delta)$, which is finite under Assumption 1(d) but is difficult to compute exactly.

Definition 3 MOA Set of Σ

The MOA set $\mathcal{S}$ and the i-steps output admissible sets $\mathcal{K}_i,\ i \in \mathbb{Z}_+$ of Σ are defined by

$$\mathcal{S} := \{x \in \mathbf{R}^n |\ c(t; x, \Delta) \in \mathcal{C}\ \forall t \in \mathbb{Z}_+,\ \forall \Delta \in \mathbb{D}\},$$
$$\mathcal{K}_i := \{x \in \mathbf{R}^n |\ c(t; x, \Delta) \in \mathcal{C}\ \forall t \in \{0, \cdots, i\},\ \forall \Delta \in \mathbb{D}\}.$$

By the definition, $x(0) \in \mathcal{S}$ is necessary and sufficient for constraint fulfillment in the face of all possible time-invariant uncertainties $\Delta \in \mathbb{D}$.

We have the following Lemma about the MOA set $\mathcal{S}$.

Lemma 1

(i) $\mathcal{S} = \bigcap_{\Delta \in \mathbb{D}} \mathcal{S}(\Delta)$ holds. (ii) Under Assumptions 1(b), (d) and (e), the MOA set $\mathcal{S}$ is a bounded convex and nonempty set satisfying $\mathbf{0} \in \text{int } \mathcal{S}$.

Proof

The item (i) is clear from the definitions of $\mathcal{S}$ and $\mathcal{S}(\Delta)$.

We prove $\mathbf{0} \in \text{int } \mathcal{S}$. Assumption 1(b) implies that there exists an open ball $\mathcal{B}(r)$ centered at the origin contained in the set $\mathcal{K}_0 := \{x \in \mathbf{R}^n |\ M_c C(\Delta) x \leq \mathbf{1}\ \forall \Delta \in \mathbb{D}\}$. Note that if $x(t) \in \mathcal{K}_0\ \forall t \in \mathbb{Z}_+$ holds, then the constraint (2.3) is satisfied. Now, from Assumption 1(d), $r_s := \sup_{t \in \mathbb{Z}_+, \Delta \in \mathbb{D}} \|A^t(\Delta)\|^2$ is a finite number. The state trajectory for any initial state in $\mathcal{B}(r/r_s)$ never gets out of $\mathcal{B}(r)$, that is, never violates the constraint. This implies that $\mathcal{B}(r/r_s)$ is a subset of the MOA set $\mathcal{S}$ and that $\mathbf{0} \in \text{int } \mathcal{S}$.

The convexity of $\mathcal{S}$ is clear from the item (i) and Proposition 1(iii). Finally, the MOA set $\mathcal{S}$ is bounded because of $\mathcal{S} \subseteq \mathcal{S}(\mathbf{0})$ and Proposition 1(iii). $\Box$

The MOA set $\mathcal{S}$ does not satisfy each property in Proposition 1. More specifically, we have the following statements: (i) The set $\mathcal{S}$ is not positively invariant for each $\Sigma(\Delta)$. (ii) The geometric characteristics of $\mathcal{S}$ and $\mathcal{K}_i$ are unknown while $\mathcal{S}(\Delta)$ and $\mathcal{K}_i(\Delta)$ are convex polyhedra. This is because $\mathcal{S}$ and $\mathcal{K}_i$ are the intersections of an infinite number of convex polyhedra $\mathcal{S}(\Delta)$ and $\mathcal{K}_i(\Delta)$ respectively. (iii) Because of the definition of i_m, we have $\mathcal{S} = \mathcal{K}_{i_m}$ and hence there is a finite i such that $\mathcal{S} = \mathcal{K}_i$. However, $\mathcal{K}_i = \mathcal{K}_{i+1}$ does not imply $\mathcal{S} = \mathcal{K}_i$ unlike the case of $S(\Delta)$ (Proposition 1 (ii)). This is because $x(1)$ is not always contained in K_i even if $x(0) \in K_{i+1}$. To the best of our knowledge, the previous research works

(Bemporad and Mosca, 1998; Casavola *et al.*, 2000; Gilbert and Tan, 1991; Gilbert and Kolmanovsky, 1998; Kolmanovsky and Gilbert, 1995; Hirata, 2004; Hirata and Ohta, 2004) consider systems with such a property as Proposition 1 (ii). In contrast, the system Σ no longer has the property. It is thus difficult to construct $\mathcal{S}$ based on the ideas of previous works, that is, $\mathcal{K}_i = \mathcal{K}_{i+1} \Rightarrow \mathcal{S} = \mathcal{K}_i$. Moreover, even if we obtain an i satisfying $\mathcal{S} = \mathcal{K}_i$ somehow, it is still difficult to describe $\mathcal{K}_i$ with a finite number of conditions because of the item (ii). We thus relax the objective of the computation of the MOA set, and aim at computing the following set.

Definition 4 ϵ-level POA set

For a real number $\epsilon \in (0,1)$, a set of initial states $\hat{\mathcal{S}}$ is said to be an ϵ-level POA set if it satisfies the following two conditions.

$$c(t;x,\mathbf{0}) \in \mathcal{C} \ \ \forall t \in \mathbb{Z}_+,\ x \in \hat{\mathcal{S}} \tag{3.1}$$

$$\text{Prob}_{\boldsymbol{\Delta}}\{\Delta \in \mathbb{D} |\ \exists t \in \mathbb{Z}_+,\ x \in \hat{\mathcal{S}} \quad \text{s.t. } c(t;x,\Delta) \notin \mathcal{C}\} \leq \epsilon \tag{3.2}$$

The condition (3.1) guarantees $\hat{\mathcal{S}} \subseteq \mathcal{S}(\mathbf{0})$ and hence boundedness of $\hat{\mathcal{S}}$. The inequality (3.2) means that at most a fraction ϵ of the uncertainty family can violate the constraint when the system is initialized in some element of $\hat{\mathcal{S}}$. It is clear that the POA set $\hat{\mathcal{S}}$ is not unique, and especially all the subsets of $\mathcal{S}$ satisfy (3.1) and (3.2) for any $\epsilon \in (0,1)$.

Our objective is to compute as large an ϵ-level POA set as possible for a prescribed $\epsilon \in (0,1)$. For this purpose, we define the following set.

Definition 5

Let $\mathbb{D}^{\mathrm{f}}$ be a subset of $\mathbb{D}$ with a finite number of elements. Then the set $\mathcal{S}(\mathbb{D}^{\mathrm{f}})$

is defined by

$$\mathcal{S}(\mathbb{D}^{\mathrm{f}}) := \{x \in \mathbf{R}^n |\ c(t;\ x, \Delta) \in \mathcal{C}\ \ \forall t \in \mathbb{Z}_+, \Delta \in \mathbb{D}^{\mathrm{f}}\}.$$

The set $\mathcal{S}(\mathbb{D}^{\mathrm{f}})$ is a convex polyhedron, since $\mathcal{S}(\mathbb{D}^{\mathrm{f}})$ is an intersection of a finite number of convex polyhedra $\mathcal{S}(\Delta),\ \Delta \in \mathbb{D}^{\mathrm{f}}$.

Remark 3

The ϵ-level POA set gives an insight into analysis of constrained systems. In addition, it is applicable to several control schemes such as switching control and reference governor. It should be noted that the relevant on-line optimization problems are not always feasible because $\hat{\mathcal{S}}$ (and even $\mathcal{S}$) is not positively invariant. Nevertheless, constraint fulfillment and control objectives are achieved by only adding the operation that the same control signal as the previous time instant is applied if the relevant on-line optimization is infeasible at a certain time instant. See (Gilbert and Kolmanovsky (1999); Gilbert and Kolmanovsky (2002)) for more detail, where output admissible sets without invariance are employed for the design of reference governors.

4 Computation of POA Set

This section presents a sequential algorithm for computing an ϵ-level POA set for a prescribed $\epsilon \in (0, 1)$. The algorithm iteratively updates a polyhedral set $\mathcal{P}_k \subset \mathbf{R}^n$, which is a candidate of the POA set.

Define a *violation function* by

$$v(\Delta, \mathcal{P}) := \begin{cases} g_v(\Delta, \mathcal{P}) - 1 & \text{if } g_v(\Delta, \mathcal{P}) > 1 \\ 0 & \text{otherwise} \end{cases},$$

$$g_v(\Delta, \mathcal{P}) := \max_{\substack{j \in \{1, \cdots, m_c\}, \\ i \in \{0, \cdots, i_m\}}} \max_{x \in \mathcal{P}} M_c(j, :) C(\Delta) A^i(\Delta) x.$$

Let $\mathcal{P} \subset \mathbf{R}^n$ be a fixed convex polyhedron with vertices $\{x_v^{(1)}, \cdots, x_v^{(n_v)}\}$. Then, the violation function $v(\Delta, \mathcal{P})$ is a measurable function (that is, a random variable) of $\Delta \in \mathbb{D}$, because $g_v(\Delta, \mathcal{P})$ is the maximum of a finite number of measurable functions $M_c(j,:)C(\Delta)A^i(\Delta)x_v^{(h)},\ j \in \{1, \cdots, m_c\},\ i \in \{0, \cdots, i_m\},\ h \in \{1, \cdots, n_v\}$.

We prepare a real number $\delta \in (0, 1)$ (confidence parameter) in addition to $\epsilon \in (0, 1)$ before executing the algorithm. Let the initial polyhedron $\mathcal{P}_0$ be the nominal MOA set $\mathcal{S}(\mathbf{0})$, which is computed by the method due to Gilbert and Tan (1991).

Suppose that, after the k-th iteration, we have a polyhedral set $\mathcal{P}_k \subset \mathbf{R}^n$. Then, the set $\mathcal{P}_k$ is updated according to the rule

$$\mathcal{P}_{k+1} := \mathcal{P}_k \cap \mathcal{S}(\mathbb{D}_k^{\mathrm{f}}), \tag{4.1}$$

where $\mathbb{D}_k^{\mathrm{f}} := \{\Delta_k^{(1)}, \cdots, \Delta_k^{(N_k)}\}$ is the set of N_k random samples drawn according to the probability density $f_{\boldsymbol{\Delta}}(\Delta)$. The set $\mathcal{P}_{k+1}$ can be immediately constructed by using Algorithm 3 in Appendix Appendix A, though its description contains redundant conditions. This update rule provides $\mathcal{P}_k$ such that

$$\mathcal{P}_k = \mathcal{S}(\bar{\mathbb{D}}_k^{\mathrm{f}}),\ \bar{\mathbb{D}}_k^{\mathrm{f}} := \left(\bigcup_{i=0}^{k-1} \mathbb{D}_i^{\mathrm{f}}\right) \cup \{\mathbf{0}\}. \tag{4.2}$$

Additionally, after the counter k reaches a prescribed integer $\bar{k}$, we perform the additional operation

$$\mathcal{P}_k := \gamma \mathcal{P}_k \tag{4.3}$$

before executing (4.1) in order to assure the finite termination of the algorithm, where $\gamma \in (0, 1)$ and $\bar{k} \in \mathbb{Z}_+$ are prescribed numbers.

Let the termination condition of the algorithm be

$$v(\Delta_k^{(j)}, \mathcal{P}_k) = 0\ \forall j \in \{1, \cdots, N_k\}, \tag{4.4}$$

which is the same as a standard sequential randomized algorithm (Oishi, 2003). The equation (4.4) is equivalent to

$$\mathcal{P}_k \subseteq \mathcal{S}(\mathbb{D}_k^{\mathrm{f}}). \tag{4.5}$$

In summary, we execute the following algorithm.

Algorithm 1

Parameters : $\gamma \in (0,1)$ and $\bar{k} \in \mathbb{Z}_+$

Step 0 Compute M_0^* such that $H(M_0^*)$ is an irredundant description of $\mathcal{S}(\mathbf{0})$. Set $k := 0$.

Step 1 Draw N_k samples $\mathbb{D}_k^{\mathrm{f}} := \{\Delta_k^{(1)}, \cdots, \Delta_k^{(N_k)}\}$ according to the probability density $f_{\boldsymbol{\Delta}}(\Delta)$.

Step 2 (Update) If $k \geq \bar{k}$, then $M_k := M_k/\gamma$.

Compute M_{k+1} such that $H(M_{k+1}) := H(M_k^*) \cap \mathcal{S}(\mathbb{D}_k^{\mathrm{f}})$ by Algorithm 3.

Step 3 (Elimination) Compute an irredundant description $H(M_{k+1}^*)$ of the polyhedron $H(M_{k+1})$ by eliminating redundant conditions.

Step 4 (Termination condition) If $M_{k+1}^* = M_k^*$, let $\hat{\mathcal{S}} := H(M_k^*)$ and terminate the algorithm. Otherwise, $k := k+1$ and go to Step 1.

In Algorithm 1, $H(M_k)\,(= H(M_k^*))$ describes the set $\mathcal{P}_k$. In Step 4, $M_{k+1}^* = M_k^*$ implies $\mathcal{P}_{k+1} = \mathcal{P}_k \cap \mathcal{S}(\mathbb{D}_k^{\mathrm{f}}) = \mathcal{P}_k$ and hence (4.5).

Lemma 2

Let the sample size N_k be the minimal integer greater than

$$\tilde{N}_k = \frac{\log \frac{\pi^2(k+1)^2}{6\delta}}{\log \frac{1}{1-\epsilon}} \tag{4.6}$$

for prescribed numbers $\delta, \epsilon \in (0,1)$. Suppose that Assumption 1 is satisfied and Algorithm 1 has terminated at the k_T-th iteration. Then, we have

$$\text{Prob}_{\boldsymbol{\Delta}}\{\Delta \in \mathbb{D} |\ v(\Delta, \mathcal{P}_{k_T}) \neq 0\} \leq \epsilon$$

with probability greater than $1-\delta$.

Proof

This Lemma can be proven in a similar manner to (Oishi (2003); Tempo *et al.* (2004)). □

Corollary 1

Suppose that all the assumptions in Lemma 2 hold. Then $\mathcal{P}_{k_T}$ is an ϵ-level POA set with probability greater than $1-\delta$.

Proof

The conditions (3.1) and (3.2) are clearly satisfied from (4.2) and Lemma 2, respectively. □

Theorem 1

Suppose that Assumption 1 is satisfied. Then, for any $\gamma \in (0,1)$, Algorithm 1 terminates in a finite number of iterations.

Proof

After k reaches $\bar{k}$, it follows from $\mathcal{P}_{k+1} \subseteq \mathcal{P}_k$ that

$$\mathcal{P}_k \subseteq \gamma^{(k-\bar{k})}\mathcal{P}_{\bar{k}}.$$

Lemma 1 states that $\mathbf{0} \in \text{int}\mathcal{S}$ holds under Assumption 1. Thus, there is a finite $k \geq \bar{k}$ satisfying $\gamma^{(k-\bar{k})}\mathcal{P}_{\bar{k}} \subseteq \mathcal{S}$. For such a k, $\mathcal{P}_k \subseteq \mathcal{S}$ holds. Then, we have

$\mathcal{P}_k \subseteq \mathcal{S} \subseteq \mathcal{S}(\mathbb{D}_k^{\mathrm{f}})$, which implies the termination of Algorithm 1. This completes the proof. $\square$

Proposition 2

Suppose that Algorithm 1 has terminated at the k_T-th iteration. (i) If k_T is smaller than $\bar{k}$, then the resulting polyhedral set $\mathcal{P}_{k_T}$ satisfies $\mathcal{P}_{k_T} \supseteq \mathcal{S}$. (ii) Otherwise, we have $\mathcal{P}_{k_T} \supseteq \gamma^{k_T-\bar{k}+1}\mathcal{S}$.

Proof

This proposition can be proven in a similar manner to Proposition 3. $\square$

Algorithm 1 gives an ϵ-level POA set for a prescribed ϵ with high probability (greater than or equal to $1-\delta$), and the algorithm terminates in a finite number of iterations. However, the algorithm may not terminate in practical computational time for the following reason, unless Σ is a simple system (n and i_m are quite small). In the elimination of redundant conditions (Step 3), we have to check

$$\Phi_l(M_{k+1}) \leq 1 \tag{4.7}$$

for $l = 1, \cdots, \mathrm{len}(M_{k+1})$ (See Kerrigan (2000) for example). If (4.7) is satisfied, $M_{k+1}(l,:)$ is redundant. This requires to solve LP problems. Since the size of M_{k+1} is in general quite large, enormous amount of computational effort is required in Step 3. In the next section, we present another algorithm with small computational effort based on heuristics.

Remark 4

The size of M_k^* affects on-line computational effort, when the ϵ-level POA set is utilized for control design schemes (A similar argument with respect to the MOA set is made in the reference (Gilbert and Kolmanovsky, 1999).). Thus, it is often required to obtain a POA set with a simple expression. This can be achieved by performing (4.3) not only when the counter k reaches $\bar{k}$ but also when len(M_k^*) exceeds a prescribed upper bound.

5 Reduction of Computational Effort

This section presents an algorithm getting over the difficulty of Algorithm 1. Notice that Lemma 2 and hence Corollary 1 depend only on the termination condition (4.4), and has no relation to the update rule (4.1) and (4.3). Thus, we modify only the update rule. In this section, we denote the polyhedral set to be updated at each iteration by $\tilde{\mathcal{P}}_k$ in order to distinguish it from $\mathcal{P}_k$ in Section 4.

The algorithm in this section also applies the rule (4.3) to the set $\tilde{\mathcal{P}}_{k+1}$ after the counter k exceeds $\bar{k}$. Then, the finite termination is satisfied if $\tilde{\mathcal{P}}_k$ decreases monotonically, namely, $\tilde{\mathcal{P}}_{k+1} \subseteq \tilde{\mathcal{P}}_k \ \forall k \in \mathbb{Z}_+$.

As well as Algorithm 1, we prepare $\mathcal{S}(\mathbf{0}) = H(M_0^*)$ as the initial polyhedron $\tilde{\mathcal{P}}_0$ before executing the algorithm. Moreover, a set $\mathcal{X}_0 := \{x_0^{(1)}, \cdots, x_0^{(|\mathcal{X}_0|)}\} \subset$ bd$(\tilde{\mathcal{P}}_0)$ is computed by Algorithm 4 in Appendix Appendix B, where $|\mathcal{X}_0| \in \mathbb{Z}_+$ is prescribed by a designer.

Suppose that, after the k-th iteration, an irredundant description $H(M_k^*)$ of a polyhedral set $\tilde{\mathcal{P}}_k$ and a set $\mathcal{X}_k := \{x_k^{(1)}, \cdots, x_k^{(|\mathcal{X}_k|)}\} \subset$ bd $\left(\tilde{\mathcal{P}}_k\right)$ are obtained. Note that these are already obtained at $k = 0$ as shown in the above paragraph.

Similarly to Algorithm 1, we generate samples $\mathbb{D}_k^{\mathrm{f}} := \{\Delta_k^{(1)}, \cdots, \Delta_k^{(N_k)}\}$ according to the probability density $f_{\boldsymbol{\Delta}}(\Delta)$, and compute

$$\hat{\mathcal{P}}_{k+1} := \tilde{\mathcal{P}}_k \cap \mathcal{S}(\mathbb{D}_k^{\mathrm{f}}) \tag{5.1}$$

by Algorithm 3 in Appendix Appendix A. The polyhedral set $\hat{\mathcal{P}}_{k+1}$ is given in the form of

$$\hat{\mathcal{P}}_{k+1} = H(M_{k+1}),\ M_{k+1} := \begin{bmatrix} M_k^* \\ M_{k+1/k} \end{bmatrix}. \tag{5.2}$$

As stated at the end of Section 4, eliminating redundant conditions from $H(M_{k+1})$ takes a great amount of computational time, since len(M_{k+1}) (especially, len$(M_{k+1/k})$) is quite large. Note that if $\tilde{\mathcal{P}}_k = \hat{\mathcal{P}}_k$ is ideally obtained at each iteration k, then $\tilde{\mathcal{P}}_k$ and $\hat{\mathcal{P}}_k$ are equal to $\mathcal{P}_k$ in Section 4.

In this section, we introduce new update procedures called Approximate Construction (AC) and Exact Construction (EC). Before referring to them, we should notice that

$$\Phi(M_{k+1/k}(l,:); M_k^*) > 1 \tag{5.3}$$

is a necessary condition for activity of an inequality $M_{k+1/k}(l,:)x \leq 1$ for describing $\hat{\mathcal{P}}_{k+1}$, and that

$$\exists x \in \mathcal{X}_k \text{ s.t. } M_{k+1/k}(l,:)x > 1 \tag{5.4}$$

is a sufficient condition for (5.3). The AC procedure is based on (5.4), and the EC procedure is both of (5.3) and (5.4).

5.1 Approximate construction of $\hat{\mathcal{P}}_{k+1}$

For $l = 1, 2, \cdots, \text{len}(M_{k+1/k})$, we first make the decision on redundancy of the l-th inequality $M_{k+1/k}(l,:)x \leq 1$ based on (5.4) as follows.

If (5.4) is not satisfied for a given l, then we decide that $M_{k+1/k}(l,:)x \leq 1$ is redundant, though this decision may not be true.

Otherwise, we decide temporarily that the l-th inequality is active, and update all the $x^{(p)} \in \mathcal{X}_k,\ p \in \{1, \cdots, |\mathcal{X}_k|\}$ satisfying $M_{k+1/k}(l,:)x^{(p)} > 1$ by

$$x^{(p)} := \frac{x^{(p)}}{M_{k+1/k}(l,:)x^{(p)}}. \tag{5.5}$$

Then, we obtain $\tilde{\mathcal{P}}_{k+1} = H(\tilde{M}_{k+1})$ by eliminating the possibly redundant inequalities from $H(M_{k+1})$, and the set $\mathcal{X}_k \in \text{bd}(\tilde{\mathcal{P}}_{k+1})$.

We next eliminate exactly redundant conditions from the description of $H(\tilde{M}_{k+1})$ by evaluating

$$\Phi_l(\tilde{M}_{k+1}) > 1. \tag{5.6}$$

However, it is not necessary to evaluate (5.6) for all $l \in \{1, \cdots, \text{len}(\tilde{M}_{k+1})\}$, since we already know that the l-th inequality is active if

$$l \in \mathcal{L}_{\mathcal{X}_k}(\tilde{M}_{k+1}). \tag{5.7}$$

Thus, we carry out the following procedure for $l = 1, 2, \cdots, \text{len}(\tilde{M}_{k+1})$.

If (5.7) holds, then the l-th inequality is active.

Else if (5.6) is satisfied, then the l-th inequality is active, and we add $\phi_l(\tilde{M}_{k+1})$ to the set $\mathcal{X}_k$.

Else the l-th inequality is redundant.

Consequently, we obtain an irredundant description $H(M^*_{k+1})$ of $\tilde{\mathcal{P}}_{k+1}$. The number of computations of Φ and hence the total computational effort are drastically reduced due to (5.4) and (5.7), in comparison with Algorithm 1.

The decisions by (5.4) may be false, that is, active inequalities may be removed, and redundant ones may not be removed. The latter fault does not affect the result, since such inequalities are eventually removed by (5.6). In contrast, due to the former fault, the polyhedral set $\tilde{\mathcal{P}}_{k+1}$ is not always equal to $\hat{\mathcal{P}}_{k+1}$, and

$$\hat{\mathcal{P}}_{k+1} \subseteq \tilde{\mathcal{P}}_{k+1} \subseteq \tilde{\mathcal{P}}_k. \tag{5.8}$$

This causes two problems; Firstly, the resulting POA set may be small. We will discuss this issue at the end of this section. The second problem is that $\tilde{\mathcal{P}}_k = \tilde{\mathcal{P}}_{k+1}$ does not ensure the termination condition (4.4), which is equivalent to

$$\tilde{\mathcal{P}}_k = \hat{\mathcal{P}}_{k+1}. \tag{5.9}$$

Thus, only when $\tilde{\mathcal{P}}_k = \tilde{\mathcal{P}}_{k+1}$, we perform the update procedure called EC which modifies $\tilde{\mathcal{P}}_{k+1}$ so that $\tilde{\mathcal{P}}_{k+1} = \hat{\mathcal{P}}_{k+1}$.

Conversely, $\tilde{\mathcal{P}}_{k+1} \neq \tilde{\mathcal{P}}_k$ ($M_k^* \neq M_{k+1}^*$) implies that the termination condition (5.9) is not satisfied, since $\hat{\mathcal{P}}_{k+1} \subseteq \tilde{\mathcal{P}}_{k+1}$. Thus, if $\tilde{\mathcal{P}}_{k+1} \neq \tilde{\mathcal{P}}_k$, let $\mathcal{X}_{k+1} := \mathcal{X}_k$ and $k := k+1$, and go to the next iteration.

5.2 Exact construction of $\hat{\mathcal{P}}_{k+1}$

The EC procedure decides redundancy of $M_{k+1/k}(l,:)x \leq 1,\ l \in \{1, \mathrm{len}(M_{k+1/k})\}$ for describing $\hat{\mathcal{P}}_{k+1}$ again, based on both (5.3) and (5.4). Namely, we perform the following procedure for $l = 1, 2, \cdots, \mathrm{len}(M_{k+1/k})$.

If (5.4) is satisfied, then we decide temporarily that the l-th inequality is active, and update all the elements $x^{(p)} \in \mathcal{X}_k$ satisfying $M_{k+1/k}(l,:)x^{(p)} > 1$ by (5.5).

Else if (5.3) is satisfied, then we decide temporarily that the l-th inequality is active, and we add $\phi(M_{k+1/k}(l,:); M_k^*)$ to $\mathcal{X}_k$.

Else the l-th inequality is (certainly) redundant.

Then, we obtain $\tilde{\mathcal{P}}_{k+1}$ by eliminating all the inequalities decided to be redundant, and $\mathcal{X}_k \in \mathrm{bd}(\tilde{\mathcal{P}}_{k+1})$. Finally, we compute an irredundant description $H(M_{k+1}^*)$ of $\tilde{\mathcal{P}}_{k+1}$ by the same strategy as the AC procedure.

As a result, we obtain $\tilde{\mathcal{P}}_{k+1}$ satisfying

$$\tilde{\mathcal{P}}_{k+1} = \hat{\mathcal{P}}_{k+1}. \tag{5.10}$$

From (5.9) and (5.10), if $\tilde{\mathcal{P}}_{k+1} = \tilde{\mathcal{P}}_k$ (equivalently $\tilde{M}^*_{k+1} = M^*_k$) holds, the termination condition (4.4) is satisfied. Otherwise, let $\mathcal{X}_{k+1} := \mathcal{X}_k$ and $k := k+1$, and go to the next iteration.

Remark 5

In the practical execution, we do not need to check (5.4) for the elements of $\mathcal{X}_k$ existing from the beginning of this update procedure, since the EC procedure is performed only when (5.4) is not satisfied for these elements. Hence, we have only to check (5.4) for the elements newly included at the "Else if" operation, and it is not necessary to check it until we find an $l \in \{1, \cdots, \mathrm{len}(M_{k+1/k})\}$ satisfying (5.3).

Remark 6

After performing one iteration (whichever procedure is performed), we get $\tilde{\mathcal{P}}_{k+1} = H(M^*_{k+1})$ and $\mathcal{X}_{k+1} \subset \mathrm{bd}\left(\tilde{\mathcal{P}}_{k+1}\right)$. Then, all the $l \in \{1, \cdots, \mathrm{len}(M^*_{k+1})\}$ are contained in $\mathcal{L}_{\mathcal{X}_{k+1}}(M^*_{k+1})$, that is, all the inequalities of $H(M^*_{k+1})$ have at least one element of $\mathcal{X}_{k+1}$ on its boundary. This alleviates the next iteration, because a lot of inequalities can be known to be active by (5.7).

5.3 Total Algorithm

Consequently, the algorithm is described as below.

Algorithm 2

Parameters : $\gamma \in (0,1)$, $\bar{k} \in \mathbb{Z}_+$, and $|\mathcal{X}_0| \in \mathbb{Z}_+$.

Step 0 Compute M_0^* such that $H(M_0^*)$ is an irredundant description of $S(\mathbf{0})$. Compute $\mathcal{X}_0$ by Algorithm 4. Set $k := 0$.

Step 1 Draw N_k samples $\mathbb{D}_k^{\mathrm{f}} := \{\Delta_k^{(1)}, \cdots, \Delta_k^{(N_k)}\}$ according to the probability density $f_{\boldsymbol{\Delta}}(\Delta)$.

Step 2 If $k \geq \bar{k}$, then $M_k := M_k/\gamma$ and $x_k^{(p)} := \gamma x_k^{(p)},\ p \in \{1, \cdots, |\mathcal{X}_k|\}$.

Compute M_{k+1} such that $H(M_{k+1}) = H(M_k^*) \cap \mathcal{S}(\mathbb{D}_k^{\mathrm{f}})$ by Algorithm 3.

Step 3 (AC) Compute M_{k+1}^* and $\mathcal{X}_{k+1}$ according to the AC procedure in Subsection 5.1.

Step 4 If $M_{k+1}^* \neq M_k^*$, then $k := k+1$ and go to Step 1.

Step 5 (EC) Compute M_{k+1}^* and $\mathcal{X}_{k+1}$ according to the EC procedure in Subsection 5.2.

Step 6 (Termination condition) If $M_{k+1}^* = M_k^*$, then terminate the algorithm. Otherwise, let $k := k+1$ and go to Step 1.

From (5.8) and (5.10), the inclusion $\tilde{\mathcal{P}}_{k+1} \subseteq \tilde{\mathcal{P}}_k$ holds. Therefore, Algorithm 2 terminates in a finite number of iterations.

As pointed out previously, the resulting POA set can be small due to the faults of the decisions by (5.4). The size of the set depends on the parameters $|\mathcal{X}_0|$, $\bar{k}$ and γ. Though it is not easy to know their appropriate values *a priori*, the following Proposition 3 (similar to 2) enables us to evaluate the size *a posteriori*. If $\gamma^{k_T - \bar{k}+1}$ is small, we execute Algorithm 2 again with making $|\mathcal{X}_0|$ large, γ close to 1, and/or $\bar{k}$ large. By this procedure, we obtain a large ϵ-level POA set.

Proposition 3

Suppose that Algorithm 2 has terminated at the k_T-th iteration. (i) If k_T

is smaller than $\bar{k}$, then the resulting polyhedral set $\tilde{\mathcal{P}}_{k_T}$ satisfies $\tilde{\mathcal{P}}_{k_T} \supseteq \mathcal{S}$. (ii) Otherwise, we have $\tilde{\mathcal{P}}_{k_T} \supseteq \gamma^{k_T-\bar{k}+1}\mathcal{S}$.

Proof

The item (i) is clear because of $\tilde{\mathcal{P}}_{k_T} \supseteq \mathcal{S}(\bar{\mathbb{D}}^{\mathrm{f}}_{k_T}) \supseteq \mathcal{S}$.

After $\bar{k}$-th iteration, $\tilde{\mathcal{P}}_{k+1} \supseteq \left(\gamma\tilde{\mathcal{P}}_k \cap \mathcal{S}(\mathbb{D}^{\mathrm{f}}_k)\right)$ holds. Thus, we have

$$\begin{aligned}\tilde{\mathcal{P}}_{k_T} &\supseteq \gamma^{k_T-\bar{k}+1}\tilde{\mathcal{P}}_{\bar{k}} \cap \left(\bigcap_{i=0}^{k_T-\bar{k}} \gamma^{k_T-\bar{k}-i}\mathcal{S}(\bar{\mathbb{D}}^{\mathrm{f}}_{\bar{k}+i})\right)\\ &\supseteq \bigcap_{i=0}^{k_T-\bar{k}+1} \gamma^{i}\mathcal{S} = \gamma^{k_T-\bar{k}+1}\mathcal{S}.\end{aligned}$$

This completes the proof of (ii). □

Remark 7

Though the computational effort of the EC procedure is still large, it is not carried out very often for the following reason. The EC procedure adds some elements to $\mathcal{X}_k$ according to the complexity of the description of $\tilde{P}_k$ (Remark 6), and hence it becomes rare that active inequalities are eliminated by (5.4) after performing the EC procedure once or several times. As a result, the total computational effort is drastically reduced compared with Algorithm 1. It should be noted that the effort of the EC procedure is smaller than that of Algorithm 1.

6 Numerical Simulation

Consider the positioning servo system shown in Casavola *et al.* (2000). As well as Casavola *et al.* (2000), we assume that the input voltage u is subject to saturation $|u| \leq U_{max} = 220[\mathrm{V}]$, and an upper bound on the absolute value of the torsional torque of the load τ has to be enforced as $|\tau| \leq \tau_{max} = 78.54[\mathrm{Nm}]$ because of the finite shear strength of the shaft. The motor inertia J_M, the load inertia

J_L, the motor viscous friction coefficient β_M and the load viscous friction coefficient β_M, is assumed constant but only known within the intervals $J_M \in [0.5, 1]$, $J_L \in [9, 11]$, $\beta_M \in [0.07, 0.13]$ and $\beta_L \in [24, 26]$. Let the other parameters be the same values as Casavola ***et al.*** (2000). Assume that the nominal system is given by $J_M = 0.75, J_L = 10, \beta_M = 0.1$ and $\beta_L = 25$. The continuous-time model has been discretized with sampling time 0.1[s] by using the zero-th order hold. Since the open-loop plant is unstable, we design a closed-loop system by the same controller as Casavola ***et al.*** (2000) $u(t) = K_x x(t) + K_g r(t),\ K_x = \begin{bmatrix} -994 & 104 & 29.6 & -4.2 \end{bmatrix},\ K_g = 401$. Then the closed-loop system is robustly stable. We attempt to fulfill the constraints by using an auxiliary mechanism called a reference governor, which modifies a reference signal and inputs the resulting signal to a closed-loop system. For this purpose, we first configured an augmented system with state $\xi = [x^\top\ g]^\top$, where g denotes any step reference. Though, the matrix $A(\Delta)$ has an eigenvalue on the unit disk, this problem can be dealt with by putting an additional constraint on g such that the corresponding equilibria are contained in $\mathcal{C}$ for all $\Delta \in \mathbb{D}$. In this example, we set $|g| \leq 50$[deg].

We computed an ϵ-level POA set $\hat{\mathcal{S}}$ by Algorithm 2 with $\delta = \epsilon = 0.01$ and $\gamma = 0.995$ under the assumption that Δ is uniformly distributed. The rule (4.3) is performed when k reaches $\bar{k} = 100$ or when len(M_k^*) exceeds 1000. In this example, len(M_k^*) exceeded 1000 at $k = 19$, and Algorithm 2 terminated at this iteration. Namely, from Proposition 3, the resulting POA set is greater than $\gamma\mathcal{S}$.

We next performed computer simulations by applying a reference governor to the closed-loop system, where we used the ϵ-level POA set instead of the deterministic MOA set. The reference governor algorithm is similar to Gilbert and Kolmanovsky (1999). We ran all the simulations setting $r(t) \equiv 40[\mathrm{deg}]\ \forall t \in \mathbb{Z}_+$ and $x(0) = 0$.

In this example, we compare the responses by using $\hat{\mathcal{S}}$ with those by using

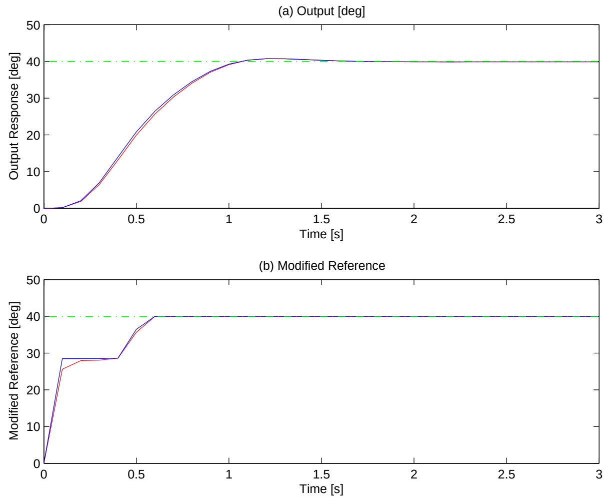

Fig. 1 Responses for nominal system

$\mathcal{S}(\mathbf{0})$, namely in the case where uncertainties are not accounted for. We first show the responses of the nominal system in Fig. 1. Since the constraints are never violated for $\Delta = \mathbf{0}$ because of (3.1), we show only the outputs and the modified references. The blue curves illustrate the responses for $\mathcal{S}(\mathbf{0})$ and the red curves those for $\hat{\mathcal{S}}$. The responses for $\hat{\mathcal{S}}$ are slower than those for $\mathcal{S}(\mathbf{0})$, because the former takes uncertainties into account. However, the deterioration is not so large, and $\hat{\mathcal{S}}$ still achieves a good tracking performance. We next checked the responses for random 10000 uncertainties. When we used $\mathcal{S}(\mathbf{0})$, the constraints were violated 8938 times out of 10000. In contrast, $\hat{\mathcal{S}}$ did not violate them once.

The result suggests that the POA set achieves low conservatism and safe system behavior.

7 Conclusion

We have presented algorithms for computing an ϵ-level POA set for time-invariant uncertain constrained systems. Algorithm 1 was inspired by an existing randomized sequential technique (Oishi, 2003). Since the algorithm requires enormous amount of computational effort, we have presented Algorithm 2 in order to reduce the computational effort. Numerical simulations have shown the applicability of the POA set, instead of the MOA set, to a control system design.

References

Bemporad, A., A. Casavola and E. Mosca (1997). Nonlinear control of constrained linear systems via predictive reference management. *IEEE Transactions on Automatic Control* **42**(3), 340–349.

Bemporad, A. and E. Mosca (1998). Fulfilling hard constraints in uncertain linear systems by reference managing. *Automatica* **34**(4), 451–461.

Blanchini, F. (1994). Ultimate boundedness control for uncertain discrete-time systems via set-induced lyapunov functions. *IEEE Transactions on Automatic Control* **39**(2), 428–433.

Blanchini, F. (1999). Set invariance in control. *Automatica* **35**(11), 1747–1767.

Blanchini, F. and S. Miani (1996). On the transient estimate for linear systems with time-varying uncertain parameters. *IEEE Transactions on Circuits and Systems Part 1* **43**(7), 592–596.

Blanchini, F., S. Miani and M. Sznaier (1997). Robust performance with fixed and worst-case signals for uncertain time-varying systems. *Automatica* **33**(12), 2183–2189.

Calafiore, G. C., F. Dabbene and R. Tempo (2000). Randomized algorithms for probabilistic robustness with real and complex structured uncertainty. *IEEE Transactions on Automatic Control* **45**(12), 2218 – 2235.

Casavola, A., E. Mosca and D. Angeli (2000). Robust command governors for constrained linear systems. *IEEE Transactions on Automatic Control* **45**(11), 2071–2077.

G. Calafiore and M.C. Campi (2004). A new bound on the generalization rate of sampled convex programs. *Proceedings of 43rd IEEE Conference on Decision and Control* pp. 5328–5333.

Gilbert, E. G. and I. Kolmanovsky (2002). Nonlinear tracking control in the presence of state and control constraints : a generalized reference governor. *Automatica* **38**(12), 2063–2073.

Gilbert, E.G. and I. Kolmanovsky (1998). Toward less conservative designs of multimode controllers for systems with state and control constraints. *Proceedings of the 37th IEEE Conference on Decision and Control* pp. 1647–1652.

Gilbert, E.G. and I. Kolmanovsky (1999). Fast reference governors for systems with state and control constraints and disturbance inputs. *International Journal of Robust and Nonlinear Control* pp. 1117–1141.

Gilbert, E.G. and K.T. Tan (1991). Linear systems with state and control constraints: The theory and application of maximal output admissible sets. *IEEE Transactions on Automatic Control* **36**(9), 1008–1019.

Gilbert, E.G., I. Kolmanovsky and K.T. Tan (1995). Discrete-time reference governors and the nonlinear control of systems with state and control constraints. *International Journal of Robust and Nonlinear Control* **5**(5), 487–504.

Goodwin, G. C., M. M. Seron and J. D. Dona (2004). Constrained control and estimation. *Springer*.

Hirata, K. (2004). On exact computations of the maximal output admissible set for a class of nonlinear systems. *Proceedings of the 33rd SICE Symposium on Control Theory* pp. 5–8.

Hirata, K. and M. Fujita (1998). Multimode switching state feedback control of constrained linear discrete-time systems. *Proceedings of the IEEE International Conference on Control Applications* pp. 148–152.

Hirata, K. and M. Fujita (2000). Measurement feedback switching control for systems with state and control constraints. *Proceedings of the American Control Conference* pp. 3321–3325.

Hirata, K. and Y. Ohta (2004). The maximal output admissible set for a class of uncertain systems. *Proceedings of the 43rd IEEE Conference on Decision and Control* pp. 2686–2691.

Hirata, K. and Y. Ohta (2005). Exact determinations of the maximal output admissible set for a class of nonlinear systems. *Proceedings of 44th IEEE Conference on Decision and Control and European Control Conference 2005* pp. 8276–8281.

Kerrigan, E. C. (2000). Robust constraint satisfaction: Invariant sets and predictive control. *Department of Engineering, University of Cambridge, UK*. `http://www-control.eng.cam.ac.uk/Homepage/papers/cued_control _155.pdf`.

Kolmanovsky, I. and E.G. Gilbert (1995). Maximal output admissible sets for discrete-time systems with disturbance inputs. *American Control Conference* pp. 1995–2000.

Oishi, Y. (2003). Probabilistic design of a robust state-feedback controller based on parameter-dependent lyapunov functions. *Proceedings of 42nd IEEE Conference on Decision and Control* pp. 1920 – 1925.

Shamma, J. S. (1996). Optimization of the l^{∞}-induced norm under full state feedback. *IEEE Transactions on Automatic Control* **41**(4), 533 – 544.

Tempo, R., G. Calafiore and F. Dabbene (2004). *Randomized Algorithms for Analysis and Control of Uncertain Systems*. Springer. London.

Appendix A Computation of $\mathcal{P} \cap \mathcal{S}(\mathbb{D}^{\mathrm{f}})$

This section presents a method for computing $\mathcal{P} \cap \mathcal{S}(\mathbb{D}^{\mathrm{f}})$ under the situation where a polyhedral set $\mathcal{P} \subseteq \mathcal{S}(\mathbf{0})$ and $\mathbb{D}^{\mathrm{f}} := \{\Delta^{(1)}, \cdots, \Delta^{(N)}\}$ are given ($\mathcal{P} = \mathcal{P}_k$ in Algorithm 1, and $\mathcal{P} = \tilde{\mathcal{P}}_k$ in Algorithm 2).

Gilbert and Tan (1991) presented methods for computing the output admissibility index $i_o(\Delta)$ or an upper bound of $i_o(\Delta)$. However, both of them require to solve several convex programming problems. Thus, to utilize the methods in Algorithm 1 or 2, we have to solve these problems for each fixed uncertainty, and total computational effort becomes quite large.

We improve the method for computing an upper bound of $i_o(\Delta)$ so that it adapts to the present algorithms. The algorithm below is performed only by algebraic calculations. Before executing the algorithm, we prepare an ellipsoid $\Omega(Q^*, \rho^*)$ containing $\mathcal{S}(\mathbf{0})$. It is obtained by the following two step optimization procedure (Hirata and Fujita, 1998).

$$Q^* := \arg\max \ \log\det Q^{-1} \ \text{ subject to } \ \Omega(Q^*, 1) \subset \mathcal{S}(\mathbf{0})$$
$$\rho^* := \min \rho \text{ subject to } \mathcal{S}(\mathbf{0}) \subset \Omega(Q^*, \rho)$$

We focus on the fact that, if the system $\Sigma(\Delta)$ is initialized from $x(0) \in \Omega(Q^*, \rho^*)$, then the state $x(t)$ of $\Sigma(\Delta)$ is confined in $\Omega(Q^*, \rho^*\|\tilde{A}^t(\Delta)\|)$ at each time instant $t \in \mathbb{Z}_+$, where $\tilde{A}(\Delta) := \{Q^*\}^{1/2} A(\Delta) \{Q^*\}^{-1/2}$. Now, define

$$\begin{aligned} r^* := \max_{r>0} \ & r \text{ subject to} \\ & \Omega(Q^*, \, r) \subset \mathcal{K}_0(\Delta^{(j)}) \ \forall j \in \{1, \cdots, N\}. \end{aligned} \tag{A.1}$$

Then, $\rho^*\|\tilde{A}^t(\Delta^{(j)})\| \le r^*$ implies that the constraint (2.3) is satisfied at time t for $\Delta^{(j)} \in \mathbb{D}^{\mathrm{f}}$, if $x(0) \in \Omega(Q^*, \rho^*)$.

Let $t^{(j)}$ and $i^{(j)} \in \mathbb{Z}_+, \ j \in \{1, \cdots, N\}$ be the minimal integers satisfying

$$\|\tilde{A}^{t^{(j)}}(\Delta^{(j)})\| \le 1, \tag{A.2}$$

$$\rho^* \|\tilde{A}^t(\Delta^{(j)})\| \le r^* \ \ \forall t \in \{i^{(j)}, \cdots, i^{(j)} + t^{(j)} - 1\}, \tag{A.3}$$

respectively. Both $t^{(j)}$ and $i^{(j)}$ are finite numbers for any $j \in \{1, \cdots, N\}$, since $\tilde{A}(\Delta^{(j)}),\ j \in \{1, \cdots, N\}$ are stable.

Lemma 3

If the system $\Sigma(\Delta^{(j)})$ is initialized in any element of $\Omega(Q^*, \rho^*)$, then the constraint is not violated after the time $i^{(j)}$.

Proof

Define the interval $\mathcal{T}_\kappa^{(j)},\ \kappa \in \mathbb{Z}_+$ by $\mathcal{T}_\kappa^{(j)} := \{i^{(j)} + \kappa t^{(j)}, \cdots, i^{(j)} + (\kappa + 1)t^{(j)} - 1\}$. To prove the lemma, it is sufficient to show

$$\rho^* \|\tilde{A}^t(\Delta^{(j)})\| \le r^* \ \forall t \in \mathcal{T}_\kappa^{(j)} \tag{A.4}$$

for all $\kappa \in \mathbb{Z}_+$. When $\kappa = 0$, (A.4) holds from (A.3). Assume that (A.4) is satisfied for a $\kappa' \in \mathbb{Z}_+$. Then, we have

$$\begin{aligned} \rho^* \ & \|\tilde{A}^{i^{(j)} + (\kappa'+1)t^{(j)} + k}(\Delta^{(j)})\| \\ & \le \rho^* \|\tilde{A}^{i^{(j)} + \kappa' t^{(j)} + k}(\Delta^{(j)})\| \|\tilde{A}^{t^{(j)}}(\Delta^{(j)})\| \\ & \le \rho^* \|\tilde{A}^{i^{(j)} + \kappa' t^{(j)} + k}(\Delta^{(j)})\| \le r^* \end{aligned}$$

for all $k \in \{0, \cdots, t^{(j)} - 1\}$. Namely, (A.4) also holds for $\kappa' + 1$. This completes the proof. $\square$

Lemma 3 and $\mathcal{P} \subseteq \mathcal{S}(\mathbf{0}) \subset \Omega(Q^*, \rho^*)$ imply $\mathcal{P} \cap \mathcal{S}(\Delta^{(j)}) = \mathcal{P} \cap \mathcal{K}_{i^{(j)}}(\Delta^{(j)})$. Namely, if $i^{(j)},\ j \in \{1, \cdots, N\}$ are computed by (A.2) and (A.3), then we have

$$\mathcal{P} \cap \mathcal{S}(\mathbb{D}^{\mathrm{f}}) = \mathcal{P} \cap \left(\bigcap_{j=1}^{N} \mathcal{K}_{i^{(j)}}(\Delta^{(j)}) \right).$$

Thus, $\mathcal{P} \cap \mathcal{S}(\mathbb{D}^{\mathrm{f}})$ is computed by the following algorithm.

Algorithm 3

Step 0 Compute r^* by (A.1). Let M be the matrix such that $\mathcal{P} = H(M)$ and $j := 1$.

Step 1 Compute $i^{(j)}$ by (A.2) and (A.3).

Step 2 Let M be

$$M := \begin{bmatrix} M \\ \bar{M} \end{bmatrix}, \ \bar{M} = \begin{bmatrix} C(\Delta^{(j)}) \\ C(\Delta^{(j)})A(\Delta^{(j)}) \\ \vdots \\ C(\Delta^{(j)})A^{i^{(j)}}(\Delta^{(j)}) \end{bmatrix}$$

Step 3 If $j = N$, then terminate the algorithm. Otherwise, $j := j + 1$ and go to Step 1.

Appendix B Generation of Samples over $\mathrm{bd}(H(M_0^*))$

The set $\mathcal{X}_0$ is computed by the following algorithm.

Algorithm 4

Step 0 Set $p := 1$.

Step 1 Draw an uniform sample $y^{(p)}$ on the boundary surface of the unit ball $\mathcal{B}(1)$. (This can be implemented in polynomial time (Tempo *et al.*, 2004))

Step 2 Compute

$$\alpha_p := \min_l \frac{1}{M_0^*(l,:)y^{(p)}} \text{ subject to } M_0^*(l,:)y^{(p)} > 0,$$

and let $x_0^{(p)} := \alpha_p y^{(p)}$.

Step 3 If p is equal to the prescribed sample number, the algorithm terminates. Otherwise, let $p := p + 1$ and go to Step 1.