MATHEMATICAL ENGINEERING TECHNICAL REPORTS

Analysis of a nonlinear system using interval arithmetic: Computation of an output admissible set

Mizuyo TAKAMATSU, Yasuaki OISHI, and Kenji HIRATA

METR 2005–26 September 2005

DEPARTMENT OF MATHEMATICAL INFORMATICS
GRADUATE SCHOOL OF INFORMATION SCIENCE AND TECHNOLOGY
THE UNIVERSITY OF TOKYO
BUNKYO-KU, TOKYO 113-8656, JAPAN

WWW page: http://www.i.u-tokyo.ac.jp/mi/mi-e.htm

Analysis of a nonlinear system using interval arithmetic: Computation of an output admissible set

Mizuyo TAKAMATSU* Yasuaki OISHI* Kenji HIRATA†

September, 2005

Abstract

It is proposed to use interval arithmetic for analysis of nonlinear systems, in particular, for computation of their output admissible sets, which play an important role in constrained control. A discrete-time system whose dynamics is expressed in polynomials is considered first. When the initial state of such a system is known to belong to some multi-dimensional interval, interval arithmetic enables us to compute an interval that includes all the possible state after some time steps. This fact together with polynomial optimization is used to compute an output admissible set. Numerical examples show effectiveness of the proposed approach. Extensions are considered to non-polynomial systems and continuous-time systems.

1 Introduction

The purpose of this paper is to introduce interval arithmetic into the field of control theory and to show its usefulness, in particular, in computation of an output admissible set, which is important in constrained control.

Interval arithmetic is a technique developed in numerical analysis for evaluation of the effect of a round-off error. The idea is to express an uncertain value using an interval that includes all its possible values and redefine the arithmetic operations so as to use intervals rather than numbers. For example, the sum of two intervals $[\underline{x}, \overline{x}]$ and $[\underline{y}, \overline{y}]$ is defined as $[\underline{x}+\underline{y}, \overline{x}+\overline{y}]$. Similar definition is possible on subtraction, multiplication, division and also elementary functions such as exponential, logarithmic, and sinusoidal functions. It is easily seen how useful this idea is in control. Consider a nonlinear discrete-time system whose initial state is uncertain but is contained in a box-shaped set, that is, each coordinate of the initial state belongs to some known interval. By applying interval arithmetic, it is possible to compute a set that includes all possible states generated after some time steps. In this paper, this idea is extended to computation of an output admissible set.

*Department of Mathematical Informatics, Graduate School of Information Science and Technology, The University of Tokyo. E-mail: {Mizuyo_Takamatsu,oishi}@mist.i.u-tokyo.ac.jp

†Department of Mechanical Engineering, Graduate School of Engineering, Osaka University. E-mail: hirata@mech.eng.osaka-u.ac.jp

An output admissible set plays an important role in controller design for a constrained system, in which the state is required to satisfy some constraints for avoidance of undesirable saturation or damage of the system. An output admissible set is a set of initial states such that the state trajectory starting from that initial state satisfies the constraints for all the time. We are especially interested in the *maximal* output admissible set, which roughly means the set of all initial states that can be handled with the current system configuration. In order to handle an initial state outside of the maximal output admissible set, we need to switch a controller.

Gilbert and Tan (1991) proposed an algorithm to compute the maximal output admissible set for linear systems. Hirata and Ohta (2005) extended this algorithm for nonlinear systems whose dynamics are expressed in polynomials. This latter algorithm enables us to exactly compute the maximal output admissible set. On the other hand, it has a drawback that it requires high computational efforts. Indeed, it repeatedly solves polynomial optimization problems (Lasserre, 2001; Parrilo, 2003) and the degrees of the considered polynomials exponentially increase as the repetition proceeds. In order to overcome this drawback, we propose in this paper a new algorithm based on interval arithmetic. This algorithm only computes an output admissible set, a subset of the maximal output admissible set, while the computed set is guaranteed to converge to the maximal one if we allow large computational efforts. A numerical experiment shows that it gives an almost maximal output admissible set within a much less computational time than the algorithm of Hirata and Ohta. The details can be found in Section 6.

The algorithm proposed in this paper consists of two procedures. First, it computes an invariant set satisfying the given constraints for the given nonlinear system. This invariant set is already one output admissible set though it may be much smaller than the maximal one. The second procedure is to enlarge this set using interval arithmetic. Although we limit our discussion to discrete-time systems whose dynamics are expressed in polynomials, extension to more general nonlinear systems and to continuous-time systems is possible. A basic idea is given in Section 7.

The construction of this paper is as follows. Interval arithmetic is introduced in Section 2. An output admissible set is discussed in Section 3. The first procedure of the algorithm is explained in Section 4 while the second procedure in Section 5. A numerical example is given in Section 6 and extension of the algorithm is considered in Section 7. Section 8 concludes the paper.

The inequality $X \succ O$ means that the real symmetric matrix X is positive definite, that is, all of its eigenvalues are positive. Similarly, $X \succeq O$ means that X is positive semidefinite. The symbol $^{\mathrm{T}}$ stands for a transpose of a matrix or a vector.

2 Interval Arithmetic

Interval arithmetic is the arithmetic using intervals rather than numbers (Moore, 1962; Alefeld and Herzberger, 1983; Neumaier, 1990). For given intervals, the result of interval arithmetic is defined to be the smallest interval that includes all the possible elementwise arithmetic results. More explicitly, for two intervals $X = [\underline{x}, \overline{x}]$ and $Y = [\underline{y}, \overline{y}]$, their addition and multiplication are defined as

$$X + Y = \{x + y \mid x \in X, y \in Y\},$$
$$X \cdot Y = \{xy \mid x \in X, y \in Y\}.$$

It is easy to see that

$$X + Y = [\underline{x} + \underline{y}, \overline{x} + \overline{y}],$$
$$X \cdot Y = [\min\{\underline{x}\underline{y}, \overline{x}\overline{y}, \underline{x}\overline{y}, \overline{x}\underline{y}\}, \max\{\underline{x}\underline{y}, \overline{x}\overline{y}, \underline{x}\overline{y}, \overline{x}\underline{y}\}].$$

Although it is possible to extend this idea to elementary functions, we rather limit our scope in this paper for the sake of simplicity. We also define the width of X as $\overline{x} - \underline{x}$. In some cases, we consider an n-dimensional interval $U = [\underline{u}_1, \overline{u}_1] \times \cdots \times [\underline{u}_n, \overline{u}_n]$. The width of U is defined as the maximum of $\overline{u}_1 - \underline{u}_1, \ldots, \overline{u}_n - \underline{u}_n$.

In order to see how we can use this technique in control theory, let us consider a nonlinear discrete-time time-invariant system

$$\boldsymbol{x}(t+1) = \boldsymbol{f}(\boldsymbol{x}(t)), \tag{1}$$

where the real vector $\boldsymbol{x} = (x_1, \ldots, x_n)^{\mathrm{T}}$ denotes its state. We make two assumptions.

(i) Each element of $\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_n(\boldsymbol{x}))^{\mathrm{T}}$ is a polynomial in $x_1, \ldots, x_n$.

(ii) The origin is an equilibrium of the system (1), i.e., $\boldsymbol{f}(\boldsymbol{0}) = \boldsymbol{0}$.

Suppose that the initial state $\boldsymbol{x}(0)$ is known to be an element of a given n-dimensional interval $U(0)$. Then, by evaluating $\boldsymbol{f}(\boldsymbol{x}(0))$ with interval arithmetic, we can obtain an n-dimensional interval $U(1)$, which includes all $\boldsymbol{x}(1)$ that can result from an $\boldsymbol{x}(0)$ belonging to $U(0)$. By repeating the same procedure to the obtained interval, we can also obtain $U(t)$, an interval that includes all the possible $\boldsymbol{x}(t)$ for $t = 2, 3, \ldots$. Here, we consider that the coefficients of the polynomials $f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_n(\boldsymbol{x})$ are exact, that is, they are intervals of width zero. When we compute $X_1^2 + X_1 \cdot X_2$, for example, we first compute two intervals ${X_1}^2$ and $X_1 \cdot X_2$ and then sum them up just as they are two independent intervals. Note that the resulting interval may be larger than $\{{x_1}^2 + x_1 x_2 \mid x_1 \in X_1, x_2 \in X_2\}$. By the same reason, $U(t)$ may be larger than the smallest interval that includes all $\boldsymbol{x}(t)$.

There are several techniques to make $U(t)$ not too large. The idea is to compute $U(t)$ not only from $U(t-1)$ but also from $U(0), U(1), \ldots, U(t-2)$ in order to avoid accumulation of the error (Nickel, 1985; Corliss, 1995). In this paper, we employ the following technique. Let A be the Jacobian matrix of $\boldsymbol{f}(\boldsymbol{x})$ at $\boldsymbol{x} = \boldsymbol{0}$ and write $\boldsymbol{h}(\boldsymbol{x}) := \boldsymbol{f}(\boldsymbol{x}) - A\boldsymbol{x}$. Then, the system (1) is written as

$$\boldsymbol{x}(t+1) = A\boldsymbol{x}(t) + \boldsymbol{h}(\boldsymbol{x}(t)), \tag{2}$$

where each term of $\boldsymbol{h}(\boldsymbol{x})$ is of order two at least due to Assumption (ii).

Suppose that we have the intervals $U(i)$, $i = 0, 1, \ldots, t-1$, and want to compute $U(t)$. A solution of the difference equation (2) is

$$\boldsymbol{x}(t) = A^t \boldsymbol{x}(0) + \sum_{i=1}^{t} A^{t-i} \boldsymbol{h}(\boldsymbol{x}(i-1)). \tag{3}$$

Noting this formula, we first compute an n-dimensional interval U_i that includes $\{\boldsymbol{h}(\boldsymbol{x}(i-1)) \mid \boldsymbol{x}(i-1) \in U(i-1)\}$ for each $i = 1, 2, \ldots, t$. Then we compute an n-dimensional interval $\hat{U}_0$ containing $A^t U(0)$ and an n-dimensional interval $\hat{U}_i$ containing $A^{t-i} U_i$ for each $i = 1, 2, \ldots, t$. The desired interval $U(t)$ can be computed as

$$U(t) = \hat{U}_0 + \hat{U}_1 \cdots + \hat{U}_t, \tag{4}$$

where the addition is performed elementwise.

Interval arithmetic is considered to be useful for various analysis of nonlinear systems such as computation of a domain of attraction. In the rest of this paper, we show how one can compute an output admissible set, which is useful for control of a constrained system.

3 Output admissible set

We consider the nonlinear discrete-time system (1). In addition to Assumptions (i) and (ii), we assume that

(iii) The Jacobian matrix of $\boldsymbol{f}(\boldsymbol{x})$ at $\boldsymbol{x} = \boldsymbol{0}$ has all the eigenvalues in the open unit disk.

We want to make the state $\boldsymbol{x}(t)$ of the system satisfy

$$g_1(\boldsymbol{x}(t)) \geq 0,\ g_2(\boldsymbol{x}(t)) \geq 0,\ \ldots,\ g_m(\boldsymbol{x}(t)) \geq 0 \tag{5}$$

for $t = 0, 1, \ldots$ in order to avoid undesirable saturation or damage of the system. Let us denote the feasible region as

$$G = \left\{ \boldsymbol{x} \;\middle|\; g_1(\boldsymbol{x}) \geq 0, g_2(\boldsymbol{x}) \geq 0, \ldots, g_m(\boldsymbol{x}) \geq 0 \right\}. \tag{6}$$

On these constraints, the following assumptions are made.

(iv) Each of $g_1(\boldsymbol{x}), \ldots, g_m(\boldsymbol{x})$ is a polynomial in $x_1, \ldots, x_n$.

(v) The origin belongs to the interior of G.

Descriptions of constraints considered here are not explicit output constraints and can handle any state and control constraints. Here, we want to find the set of all initial states such that the state trajectory starting from that initial state is included in G. This set is called the *maximal output admissible set*.

The maximal output admissible set and a concept of set invariance play fundamentally important role in analysis and control design of systems with constraints. They have been used extensively throughout analysis and design methodologies including multimode controller switching strategies (Hirata and Fujita, 2000; McClamroch and Kolmanovsky, 2000), reference management problems (Bemporad, 1998; Gilbert and Kolmanovsky, 1999, 2002; Hirata and Fujita, 1999) and persistent disturbance rejection problems (Shamma, 1996; Lu, 1998). More detailed descriptions can be found in references therein and a survey paper (Blanchini, 1999). Gilbert and Tan (1991) gave an algorithm to compute maximal output admissible sets in the special case that $\boldsymbol{f}(\boldsymbol{x})$ is linear in $\boldsymbol{x}$. Recently, Hirata and Ohta (2005) extended the algorithm so as to be applicable to the present setting with polynomial $\boldsymbol{f}(\boldsymbol{x})$, and this makes it possible to determine the maximal output admissible set in a constructive way. An advantage of this algorithm is that exact computation of maximal output admissible sets is possible in case of polynomial $\boldsymbol{f}(\boldsymbol{x})$. On the other hand, the degrees of the polynomials represent specific optimization problems required by the algorithm grow exponentially as iterative computations proceeds. As a result, this algorithm requires high computational efforts.

Motivations of the present work come from controlling nonlinear systems with state and control constrains. It is true that the maximal output admissible set or set invariance properties have been utilized in control design problems in nonlinear settings (Bemporad, 1998; Lu, 1998; McConley et al., 2000; Miller et al., 2000; Gilbert and Kolmanovsky, 2002). However, due to the difficulties of issues for exactly computing maximal output admissible sets for nonlinear systems, certain subset of the maximal output admissible set such as sublevel set of Lyapunov functions or simulation based estimations of constraints admissible regions are used. Computationally efficient exact determinations or precise approximations of maximal output admissible sets may have direct applications with theses control design approaches.

Motivated by this fact, we consider an alternative approach using interval arithmetic. Since our system is nonlinear, it may have an equilibrium at a different point from the origin. Concentrating on the equilibrium at the origin, we define the maximal output admissible set as

$$O_\infty = \left\{ \boldsymbol{x}(0) \mid \boldsymbol{x}(t) \in G \text{ for } t = 0, 1, \ldots \text{ and } \lim_{t\to\infty} \boldsymbol{x}(t) = \boldsymbol{0} \right\}.$$

Our approach is to compute an output admissible set, i.e., a subset of the maximal output admissible set, and to enlarge it using interval arithmetic. The first procedure is explained in Section 4 while the second in Section 5.

4 Finding a Subset

This section explains how we can find an output admissible set. First, we linearize the given system at the origin, and find a quadratic Lyapunov function $V(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}} P \boldsymbol{x}$ for the linearized system. If we choose a level set of $V(\boldsymbol{x})$ small enough, the function $V(\boldsymbol{x})$ works as a Lyapunov function of the original system in that set and the constraints (5) are also satisfied, there. This means that this level set is an output admissible set.

Using the notation in (2), we write the given system as

$$\boldsymbol{x}(t+1) = \boldsymbol{f}(\boldsymbol{x}(t)) = A\boldsymbol{x}(t) + \boldsymbol{h}(\boldsymbol{x}(t)),$$

where A is the Jacobian matrix of $\boldsymbol{f}(\boldsymbol{x})$ at $\boldsymbol{x} = \boldsymbol{0}$. We compute a quadratic Lyapunov function for the linearized system $\boldsymbol{x}(t+1) = A\boldsymbol{x}(t)$. For that purpose, we consider the semidefinite program:

$$\begin{array}{ll} \text{maximize} & t \\ \text{subject to} & P - A^{\mathrm{T}} P A \succeq tI, \\ & P \succeq tI, \end{array} \tag{7}$$

where the optimization variables are the scalar t and the symmetric matrix P. By Assumption (iii), its maximum value is positive and the maximizing P gives a desired Lyapunov function $V(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}} P \boldsymbol{x}$.

Let us define a level set of $V(\boldsymbol{x})$ by

$$\mathrm{Lv}(\gamma) = \{\boldsymbol{x} \mid V(\boldsymbol{x}) < \gamma\}.$$

If we choose γ small enough, $V(\boldsymbol{x})$ can be a Lyapunov function of the original system in $\mathrm{Lv}(\gamma)$. This is stated in the next proposition, whose proof is easy and omitted.

Proposition 1. *There exists a positive number γ such that*

$$(A\boldsymbol{x} + \boldsymbol{h}(\boldsymbol{x}))^T P (A\boldsymbol{x} + \boldsymbol{h}(\boldsymbol{x})) - \boldsymbol{x}^T P\boldsymbol{x} < 0$$

holds for any nonzero $\boldsymbol{x} \in \mathrm{Lv}(\gamma)$.

Suppose that γ is a number as in Proposition 1. Then, if $\boldsymbol{x}(0) \in \mathrm{Lv}(\gamma)$, the state $\boldsymbol{x}(1) = A\boldsymbol{x}(0)+\boldsymbol{h}(\boldsymbol{x}(0))$ satisfies $V(\boldsymbol{x}(1)) < V(\boldsymbol{x}(0))$ and thus $\boldsymbol{x}(1) \in \mathrm{Lv}(\gamma)$. By repeating this reasoning, we can see that $\boldsymbol{x}(t)$ belongs to $\mathrm{Lv}(\gamma)$ for all $t = 1, 2, \ldots$ if it evolves according to the original system $\boldsymbol{x}(t+1) = A\boldsymbol{x}(t) + \boldsymbol{h}(\boldsymbol{x}(t))$. In other words, the level set $\mathrm{Lv}(\gamma)$ is an invariant set. Moreover, $V(\boldsymbol{x}(t))$ approaches zero as t becomes larger (Lemma 5. 3. 40 of Vidyasagar (1993)). Hence, $\lim_{t\to\infty} \boldsymbol{x}(t) = \boldsymbol{0}$. These are desirable properties for us because such $\mathrm{Lv}(\gamma)$ becomes an output admissible set if it has the additional property $\mathrm{Lv}(\gamma) \subseteq G$.

In order to find a γ as in Proposition 1, we solve the following polynomial optimization problem with $\boldsymbol{x}$ being the optimization variable:

$$\begin{array}{ll} \text{minimize} & \boldsymbol{x}^{\mathrm{T}} P\boldsymbol{x} \\ \text{subject to} & (A\boldsymbol{x} + \boldsymbol{h}(\boldsymbol{x}))^{\mathrm{T}} P (A\boldsymbol{x} + \boldsymbol{h}(\boldsymbol{x})) \\ & \qquad\qquad - \boldsymbol{x}^{\mathrm{T}} P\boldsymbol{x} \geq 0, \\ & \boldsymbol{x}^{\mathrm{T}} P\boldsymbol{x} \geq \varepsilon. \end{array} \tag{8}$$

Here, ε is some small positive number, which is introduced for exclusion of the trivial solution $\boldsymbol{x} = \boldsymbol{0}$. In the case that the value obtained as the minimum is equals to ε, replace ε by a smaller number and solve the problem again.

Let us denote by r_0 the value computed as the minimum. Suppose for a while this r_0 is really the minimum of the problem (8). Then the situation is as in Figure 1. Here, the first constraint of the problem (8), i.e., $(A\boldsymbol{x} + \boldsymbol{h}(\boldsymbol{x}))^{\mathrm{T}} P (A\boldsymbol{x} + \boldsymbol{h}(\boldsymbol{x})) - \boldsymbol{x}^{\mathrm{T}} P\boldsymbol{x} \geq 0$, is satisfied in the shaded region while it is not in the white region. The set $\mathrm{Lv}(r_0)$ shown as an ellipse touches but does not deeply intersect the shaded region. This means that r_0 can be a γ that appeared in Proposition 1.

We actually solve a polynomial optimization problem by relaxing the constraints (Lasserre, 2001; Parrilo, 2003), which means that the obtained r_0 may be a lower bound of the minimum value. In this case, $\mathrm{Lv}(r_0)$ is an ellipse smaller than the ellipse in Figure 1. Note that, still in this case, $\mathrm{Lv}(r_0)$ is included in the white region. Therefore, r_0 can be used as a γ in Proposition 1.

As is explained before, if the considered level set is further included in G, this becomes an output admissible set. In order to find such a level set, we solve the following polynomial optimization problem for each $i = 1, \ldots, m$:

$$\begin{array}{ll} \text{minimize} & \boldsymbol{x}^{\mathrm{T}} P\boldsymbol{x} \\ \text{subject to} & g_i(\boldsymbol{x}) \leq 0. \end{array} \tag{9}$$

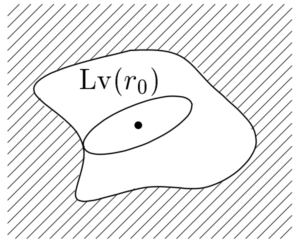

Figure 1: The level set obtained by solving the problem (8).

Let us denote the obtained value by r_i for each $i = 1, \dots, m$. Then, $\mathrm{Lv}(r_i)$ is contained in the region $\{\boldsymbol{x} \mid g_i(\boldsymbol{x}) \geq 0\}$ by a similar reasoning as before. (If r_i is really the minimum of problem (9), $\mathrm{Lv}(r_i)$ is the maximal level set contained in this region.) Define $r := \min\{r_0, r_1, \dots, r_m\}$. We can now see that $\mathrm{Lv}(r)$ is an output admissible set. We conclude this section by summarizing the procedure.

Procedure 1. Step 1 *Solve the semidefinite program* (7) *to obtain a quadratic Lyapunov function of the linearized system* $\boldsymbol{x}(t+1) = A\boldsymbol{x}(t)$. *Write the obtained Lyapunov function as* $V(\boldsymbol{x}) = \boldsymbol{x}^T P \boldsymbol{x}$.

Step 2 *Solve the polynomial optimization problem* (8) *to find* $\mathrm{Lv}(r_0)$ *in which* $V(\boldsymbol{x})$ *is a Lyapunov function of the original system.*

Step 3 *Solve the polynomial optimization problem* (9) *to find* $\mathrm{Lv}(r_i)$ *contained in the region* $\{\boldsymbol{x} \mid g_i(\boldsymbol{x}) \geq 0\}$ *for each* $i = 1, \dots, m$.

Step 4 *Denote by* r *the minimum of* $r_0, r_1, \dots, r_m$. *Then,* $\mathrm{Lv}(r)$ *is an output admissible set.*

5 Enlarging a Set

Since the output admissible set $\mathrm{Lv}(r)$, which was obtained in Section 4, can be much smaller than the maximal output admissible set O_∞, we consider in this section how we can enlarge this set. The idea is to divide the region surrounding $\mathrm{Lv}(r)$ into small n-dimensional intervals and to make them evolve using interval arithmetic as was described in Section 2. See Figure 2. Let $U(0)$ be one such initial interval and $U(t)$ be the interval obtained for the time t. Suppose that there exists t_0 such that

$$U(t_0) \subseteq \mathrm{Lv}(r), \tag{10}$$

$$U(t) \subseteq G \quad \text{for } t = 0, 1, \dots, t_0. \tag{11}$$

In this case, if the initial state $\boldsymbol{x}(0)$ is an element of $U(0)$, the state $\boldsymbol{x}(t)$ resulting from this $\boldsymbol{x}(0)$ belongs to G for $t = 0, 1, \dots, t_0$ because $\boldsymbol{x}(t) \in U(t)$ and $U(t) \subseteq G$. Moreover, $\boldsymbol{x}(t)$ belongs to G also for $t = t_0 + 1, t_0 + 2, \dots$ and $\boldsymbol{x}(t)$ approaches the origin as t becomes larger because $\boldsymbol{x}(t_0) \subseteq \mathrm{Lv}(r)$ and $\mathrm{Lv}(r)$ is an output admissible set. This means that this $U(0)$ is itself an

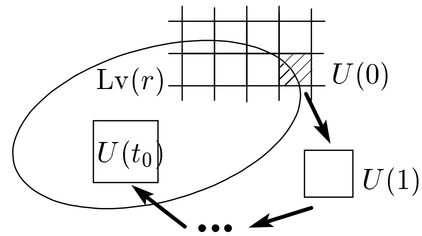

Figure 2: The idea to enlarge $\mathrm{Lv}(r)$.

output admissible set and so is the set $\mathrm{Lv}(r) \cup U(0)$. Thus, we can make an output admissible set larger. It is not difficult to check the inclusions (10) and (11). Note that both $U(t_0)$ and $\mathrm{Lv}(r)$ are convex. Hence, in order to check $U(t_0) \subseteq \mathrm{Lv}(r)$, it suffices to check whether all vertices of $U(t_0)$ belong to $\mathrm{Lv}(r)$. If G happens to be convex, we can use the same technique to see if $U(t) \subseteq G$. If this is not the case, we use interval arithmetic. Compute an interval $C_j = [\underline{c}_j, \overline{c}_j]$ containing $g_j(U(t))$ for $j = 1, 2, \ldots, m$. If $\underline{c}_j \geq 0$ for all j, we can conclude $U(t) \subseteq G$. This latter method may be useful even for a convex G when the dimension n is considerably large.

Based on the preceding discussion, we can construct the following algorithm. Here, $\mathcal{I}_{\mathrm{acc}}$ stands for the set of intervals judged to be an output admissible set; $\mathcal{I}_{\mathrm{rej}}$ the set of intervals not judged to be an output admissible set; $\mathcal{I}_{\mathrm{che}}$ the set of intervals not yet checked.

Procedure 2. Step 1 *Set* $\mathcal{I}_{\mathrm{acc}} \leftarrow \emptyset$, $\mathcal{I}_{\mathrm{rej}} \leftarrow \emptyset$, *and* $\mathcal{I}_{\mathrm{che}} \leftarrow \{U \mid U$ *is an* n*-dimensional interval intersecting* $\mathrm{Lv}(r)$ *but not contained in* $\mathrm{Lv}(r)\}$.

Step 2 *If* $\mathcal{I}_{\mathrm{che}}$ *is empty, then stop.*

Step 3 *Pick one interval* W *from* $\mathcal{I}_{\mathrm{che}}$ *and set it to be* $U(0)$. *Set* $\mathcal{I}_{\mathrm{che}} \leftarrow \mathcal{I}_{\mathrm{che}} \setminus \{W\}$ *and* $t \leftarrow 0$.

Step 4 *If we cannot verify* $U(t) \subseteq G$, *set* $\mathcal{I}_{\mathrm{rej}} \leftarrow \mathcal{I}_{\mathrm{rej}} \cup \{U(0)\}$ *and go back to Step 2.*

Step 5 *If we can see* $U(t) \subseteq \mathrm{Lv}(r)$, *set* $\mathcal{I}_{\mathrm{acc}} \leftarrow \mathcal{I}_{\mathrm{acc}} \cup \{U(0)\}$ *and* $\mathcal{I}_{\mathrm{che}} \leftarrow \mathcal{I}_{\mathrm{che}} \cup \{U \mid U$ *is an* n*-dimensional interval adjacent to* $U(0)$ *and not belonging to* $\mathcal{I}_{\mathrm{acc}} \cup \mathcal{I}_{\mathrm{rej}}\}$. *Go back to Step 2.*

Step 6 *Set* $t \leftarrow t + 1$ *and go back to Step 4.*

After executing this procedure, consider the union of $\mathrm{Lv}(r)$ and all the intervals in $\mathcal{I}_{\mathrm{acc}}$. This union is an output admissible set and is expected to be considerably larger than $\mathrm{Lv}(r)$. Indeed, the obtained output admissible set converges to the maximal one O_∞ as we make the width of the initial intervals smaller, which we show in Proposition 2.

Proposition 2. *Let* $\boldsymbol{x}_0$ *be any point in the interior of* O_∞. *Let* t_0 *be a number such that* $\boldsymbol{x}(t_0)$ *belongs to the interior of* $\mathrm{Lv}(r)$ *when the system* (1) *evolves from* $\boldsymbol{x}(0) = \boldsymbol{x}_0$. *Then, there exists a positive number* w *such that, if an interval* $U(0)$ *contains* $\boldsymbol{x}_0$ *and has the width less than* w, *the corresponding* $U(t)$ *satisfies* (10) *and* (11).

Proof. As the width of $U(0)$ approaches zero, the width of $U(t)$ converges to zero by the definition of interval arithmetic. Thus, the proposition follows. □

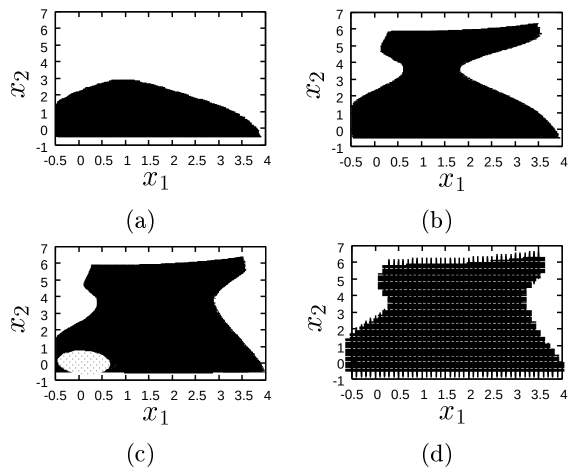

Figure 3: The output admissible sets computed when the width of the initial intervals is (a) 0.05, (b) 0.03, and (c) 0.01, respectively. The rough shape of the maximal output admissible set is shown in (d).

6 Numerical Examples

In this section, we show two examples. From them, one can see that the obtained output admissible set converges to the maximal one as the width of the intervals gets smaller. It is also seen that the present algorithm stops with much shorter running time than the algorithm of Hirata and Ohta (2005).

The algorithm is implemented with MATLAB. The polynomial optimization problems are solved by SOSTOOLS (Prajna et al., 2004) with SeDuMi (Sturm, 1999) as a solver for semidefinite programming. All the experiments were made on the 2.2 GHz Athlon processor with 2.0 GByte memory. Consider the system

$$\begin{aligned} x_1(t+1) &= -0.07071{x_1}^3 + 0.07071{x_1}^2 + 0.7086x_1 - 0.12x_2, \\ x_2(t+1) &= 0.03536{x_1}^3 - 0.03536{x_1}^2 + 0.07071x_1 \\ &\qquad - 0.03536{x_2}^3 + 0.03536{x_2}^2 + 0.92929x_2, \end{aligned}$$

with the constraints $-0.5 \le x_1 \le 15$ and $-0.5 \le x_2 \le 20$. Here, we omit the dependence in t to make the description simple. Since this system satisfies Assumptions (i)–(v), we can apply the proposed algorithm to the system. Figure 3 (a), (b), and (c) show the computed output admissible sets (the black regions) when the width of the initial intervals is set to 0.05, 0.03, and 0.01, respectively in Procedure 2. In Figure 3 (c), the level set $\mathrm{Lv}(r)$ obtained in Procedure 1 is also shown (the ellipse). The rough shape of the maximal output admissible set O_∞ is shown in Figure 3 (d). Here, we check membership to O_∞ for each grid point of distance 0.1 by letting the system evolve from each grid point. One can see from the numerical results that the output of the present algorithm converges to the set O_∞ as is expected in Proposition 2. The time to compute (c) was 389.4 seconds. Although we applied the algorithm of Hirata and Ohta to the same example, we were not able to get a result due to a shortage of memory. With the present algorithm, it is possible to make the running time even shorter by implementing the program for interval arithmetic with Java. The result was 104.5 seconds.

Next, we computed output admissible sets for the same system but with replacing the constraints by $-0.5 \leq x_1 \leq 1.0$ and $-0.5 \leq x_2 \leq 1.0$. The running time of the present algorithm was 99.9 seconds whereas that of Hirata and Ohta's is 2865.7 seconds. The obtained results are almost the same between the two algorithms.

7 Extensions

It is possible to extend the present algorithm so as to be applicable to two classes of systems: non-polynomial systems and continuous-time systems. We give in this section an idea for the extensions.

Suppose that the function $\boldsymbol{f}(\boldsymbol{x})$ of the given system $\boldsymbol{x}(t+1) = \boldsymbol{f}(\boldsymbol{x}(t))$ is expressed not only with polynomials in $\boldsymbol{x}$ but also with elementary functions like sinusoidal, exponential, and logarithmic functions. Suppose the same for $g_1(\boldsymbol{x}), \ldots, g_m(\boldsymbol{x})$ in the given constraints. Even in this case, we can execute the latter half of our algorithm, i.e., Procedure 2, to enlarge an output admissible set, since we can perform interval arithmetic with elementary functions. Only the problem is that we cannot utilize polynomial optimization in the former half of the algorithm, i.e., Procedure 1, to obtain a small output admissible set $\mathrm{Lv}(r)$. Hence, we take the following approach instead.

Let us write as $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}(\boldsymbol{x})$ the Jacobian matrix of $\boldsymbol{f}(\boldsymbol{x})$ evaluated at $\boldsymbol{x}$. Note that $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}(\boldsymbol{x})$ is again expressed with elementary functions. Choose a small n-dimensional interval X containing the origin. By evaluating $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}(\boldsymbol{x})$ with interval arithmetic, we can obtain matrices $A^{(1)}, \ldots, A^{(p)}$ such that their convex hull includes $\{\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}(\boldsymbol{x}) \mid \boldsymbol{x} \in X\}$. Next, find a positive definite matrix P such that $P - {A^{(i)}}^{\mathrm{T}} P A^{(i)} \succ O$ for $i = 1, \ldots, p$ by semidefinite programming. If there exists such a P, choose γ so that the level set $\{\boldsymbol{x} \mid \boldsymbol{x}^{\mathrm{T}} P \boldsymbol{x} < \gamma\}$ is contained in X and also in $G = \{\boldsymbol{x} \mid g_1(\boldsymbol{x}) \geq 0, \ldots, g_m(\boldsymbol{x}) \geq 0\}$. This task is accomplished with interval arithmetic again. The obtained level set is an output admissible set. If there is no P satisfying the above inequalities, choose the interval X smaller and repeat the same procedure.

Next, suppose that we are given a continuous-time system expressed in polynomials. We can adapt Procedure 1 to this case with replacing the Lyapunov inequality by the continuous-time counterpart. In order to adapt Procedure 2, however, we need a device to solve $\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t))$ with interval arithmetic. There are several techniques to solve an ordinary differential equation with interval arithmetic (Corliss, 1995). We use here the technique of Iri and Amemiya (1995) (see also Kubota and Iri (1998)). Its basic idea is as follows.

We solve $\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t))$ with the initial value $\boldsymbol{x}(0)$ belonging to the given interval $U(0)$. Suppose that we have an interval $U(t_k)$ that includes all the possible values of $\boldsymbol{x}(t_k)$. For some $t_{k+1} > t_k$, we will obtain an interval $U(t_{k+1})$ that includes all the possible values of $\boldsymbol{x}(t_{k+1})$. For this purpose, we note that the Taylor expansion of $\boldsymbol{x}(t)$ gives

$$\boldsymbol{x}(t_{k+1}) = \boldsymbol{x}(t_k) + (t_{k+1} - t_k)\dot{\boldsymbol{x}}(t_k) + \cdots + \frac{(t_{k+1} - t_k)^p}{p!}\boldsymbol{x}^{(p)}(t_k) + \frac{(t_{k+1} - t_k)^{p+1}}{(p+1)!}\boldsymbol{x}^{(p+1)}(\tau), \quad (12)$$

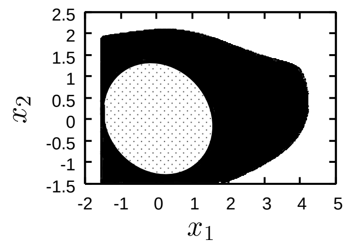

Figure 4: The output admissible set computed for the given continuous-time system (the black region) and the associated level set $\mathrm{Lv}(r)$ (the ellipse).

where τ is some number between t_k and t_{k+1}. Using the relation $\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t))$, it is possible to express $\dot{\boldsymbol{x}}(t), \ldots, \boldsymbol{x}^{(p)}(t), \boldsymbol{x}^{(p+1)}(t)$ as functions of $\boldsymbol{x}(t)$. Namely, they are polynomials in $\boldsymbol{x}(t)$ since so is $f(\cdot)$. Thus, it is possible to evaluate all the terms of (12) but the last using interval arithmetic under the condition $\boldsymbol{x}(t_k) \in U(t_k)$. If we further have an interval V_k that includes all the possible values of $\boldsymbol{x}(t)$, $t_k \leq t \leq t_{k+1}$, we can also evaluate the last term of (12) and obtain an interval $U(t_{k+1})$ that includes $\boldsymbol{x}(t_{k+1})$. In order to compute V_k, we consider the inequality

$$V \supseteq U(t_k) + \boldsymbol{f}(V)(t_{k+1} - t_k), \tag{13}$$

where V is an interval and $\boldsymbol{f}(V)$ is the result of interval arithmetic to evaluate $\boldsymbol{f}(\boldsymbol{x})$ for $\boldsymbol{x} \in V$. If we can find V satisfying this inequality, we can use it for V_k. Hence, we begin with some small interval V, check (13), and enlarge V if we fail. Once we can find an appropriate interval V_k, we can compute $U(t_{k+1})$ that includes $\boldsymbol{x}(t_{k+1})$ and we can step further.

As an example, consider the system

$$\begin{aligned}
\frac{\mathrm{d}x_1}{\mathrm{d}t} &= -0.00707{x_1}^3 + 0.00707{x_1}^2 - 0.02914x_1 - 0.024x_2, \\
\frac{\mathrm{d}x_2}{\mathrm{d}t} &= 0.00354{x_1}^3 - 0.00354{x_1}^2 + 0.00707x_1 \\
&\quad - 0.00354{x_2}^3 + 0.00354{x_2}^2 - 0.00707x_2,
\end{aligned}$$

with the constraints $-1.5 \leq x_1 \leq 15$ and $-1.5 \leq x_2 \leq 20$. We executed the algorithm sketched so far. Figure 4 shows the computed output admissible set (the black region) and the level set $\mathrm{Lv}(r)$ used in the former half of the algorithm (the ellipse).

8 Conclusion

In this paper, we proposed a new algorithm to find an output admissible set for a nonlinear discrete-time time-invariant system expressed in polynomials. It is based on interval arithmetic. The obtained output admissible set converges to the maximal one as we take higher computational efforts. Numerical examples show that the proposed algorithm stops in a much shorter running time than the existing algorithm of Hirata and Ohta (2005).

In order to shorten the running time of our algorithm, it may be effective to use n-dimensional intervals of different widths in the phase of enlarging an output admissible set. First, we enlarge the set using intervals of width θ_1 and we get a larger output admissible set Θ_1. Secondly, we divide the region surrounding Θ_1 into intervals of width θ_2 ($< \theta_1$) and we get an even larger output admissible set Θ_2. In this way, we can gain a larger output admissible set efficiently.

Acknowledgements

The authors are obliged to Sunao Murashige for his helpful comments.

References

A. Bemporad: Reference governor for constrained nonlinear systems. *IEEE Transactions on Automatic Control*, Vol. 43, No. 3, pp. 415–419, 1998.

F. Blanchini: Set invariance in control. *Automatica*, Vol. 35, No. 11, pp. 1747–1767, 1999.

G. Alefeld and J. Herzberger: *Introduction to Interval Computations.* Academic Press, New York, USA, 1983.

G. F. Corliss: Guaranteed error bounds for ordinary differential equations. *Theory and Numerics of Ordinary and Partial Differential Equations*, Oxford University Press, New York, USA, pp. 1–75, 1995.

E. G. Gilbert and I. Kolmanovsky: Fast reference governors for systems with state and control constraints and disturbance inputs. *International Journal of Robust and Nonlinear Control*, Vol. 9, No. 15, pp. 1117–1141, 1999.

E. G. Gilbert and I. Kolmanovsky: Nonlinear tracking control in the presence of state and control constraints: A generalized reference governor. *Automatica*, Vol. 38, No. 12, pp. 2063–2073, 2002.

E. G. Gilbert and K. T. Tan: Linear systems with state and control constraints: The theory and application of maximal output admissible sets. *IEEE Transactions on Automatic Control*, Vol. 36, No. 9, pp. 1008–1020, 1991.

K. Hitara and M. Fujita: Set of admissible reference signals and control of systems with state and control constraints. *Proceedings of the 38th IEEE Conference on Decision and Control*, pp. 1427–1432, Phoenix, USA, 1999.

K. Hitara and M. Fujita: Control of systems with state and control constraints via controller switching strategy. *Proceedings of the Workshop on Systems with Time-domain Constraints*, Eindhoven, The Netherlands, 2000.

K. Hirata and Y. Ohta: Exact determinations of the maximal output admissible set for a class of nonlinear systems. *The 44th IEEE Conference on Decision and Control*, Seville, Spain, 2005, to be presented.

M. Iri and J. Amemiya: Experimental studies on guaranteed-accuracy solutions of the initial-value problem of nonlinear ordinary differential equations. *Numerical Analysis of Ordinary Differential Equations and Its Applications*, World Scientific, River Edge, USA, pp. 195–212, 1995.

K. Kubota and M. Iri: *Automatic Differentiation of Algorithms and Applications*. Corona Publishing, Tokyo, Japan, 1998 (in Japanese).

J. B. Lasserre: Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, Vol. 11, No. 3, pp. 796–817, 2001.

W.-M. Lu: Rejection of persistent $\mathcal{L}_\infty$-bounded disturbance for nonlinear systems. *IEEE Transactions on Automatic Control*, Vol. 43, No. 12, pp. 1692–1702, 1998.

N. H. McClamroch and I. Kolmanovsky: Performance benefits of hybrid control design for linear and nonlinear systems. *Proceedings of the IEEE*, Vol. 88, No. 7, pp. 1083–1096, 2000.

M. W. McConley, B. D. Appleby, M. A. Dahleh, and E. Feron: A computationally efficient Lyapunov-based scheduling procedure for control of nonlinear systems with stability guarantees. *IEEE Transactions on Automatic Control*, Vol. 45, No. 1, pp. 33–49, 2000.

R. H. Miller, I. Kolmanovsky, E. G. Gilbert, and P. D. Washabaugh: Control of constrained nonlinear systems: A case study. *IEEE Control Systems Magazine*, Vol. 20, No. 2, pp. 23–32, 2000.

R. E. Moore: Interval arithmetic and automatic error analysis in digital computing. Ph.D. Dissertation, Department of Mathematics, Stanford University, 1962.

A. Neumaier: *Interval Methods for Systems of Equations*. Cambridge University Press, Cambridge, England, 1990.

K. Nickel: How to fight the wrapping effect. *Interval Analysis 1985*, Springer, Berlin, Germany, pp. 121–132, 1985.

P. A. Parrilo: Semidefinite programming relaxations for semialgebraic problems. *Mathematical Programming*, Ser. B, Vol. 96, pp. 293–320, 2003.

S. Prajna, A. Papachristodoulou, P. Seiler, and P. A. Parrilo: SOSTOOLS: Sum of squares optimization toolbox for MATLAB, 2004. Available from `http://www.mit.edu/~parrilo/sostools/index.html`.

J. S. Shamma: Optimization of the ℓ^∞-induced norm under full state feedback. *IEEE Transactions on Automatic Control*, Vol. 41, No. 4, pp. 533–544, 1996.

J. F. Sturm: Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, Vols. 11–12, pp. 625–653, 1999.

M. Vidyasagar: *Nonlinear Systems Analysis, Second Edition*. Prentice Hall, Englewood Cliffs, USA, 1993.