



# Reducing the wrapping effect of set computation via Delaunay triangulation for guaranteed state estimation of nonlinear discrete-time systems

Jian Wan<sup>1</sup> · Luc Jaulin<sup>2</sup>

Received: 7 August 2023 / Accepted: 19 February 2024  
© The Author(s) 2024

## Abstract

Set computation methods have been widely used to compute reachable sets, design invariant sets and estimate system state for dynamic systems. The wrapping effect of such set computation methods plays an essential role in the accuracy of their solutions. This paper studies the wrapping effect of existing interval, zonotopic and polytopic set computation methods and proposes novel approaches to reduce the wrapping effect for these set computation methods based on the task of computing the dynamic evolution of a nonlinear uncertain discrete-time system with a set as the initial state. The proposed novel approaches include the partition of a polytopic set via Delaunay triangulation and also the representation of a polytopic set by the union of small zonotopes for the following set propagation. The proposed novel approaches with the reduced wrapping effect has been further applied to state estimation of a nonlinear uncertain discrete-time system with improved accuracy. Similar to bisection for interval and zonotopic sets, Delaunay triangulation has been introduced as a set partition tool for polytopic sets, which has opened new research directions in terms of novel set partition, set representation and set propagation for reducing the wrapping effect of set computation.

**Keywords** Set computation · Wrapping effect · Nonlinear systems · State estimation

**Mathematics Subject Classification** 32B25 · 93-08

---

✉ Jian Wan  
wanj3@aston.ac.uk

Luc Jaulin  
luc.jaulin@ensta-bretagne.fr

<sup>1</sup> Department of Mechanical, Biomedical and Design Engineering, School of Engineering and Technology, College of Engineering and Physical Sciences, Aston University, Aston Street, Birmingham B4 7ET, UK

<sup>2</sup> Lab-STICC, ENSTA Bretagne, 2 rue François Verny, 29806 Brest, France

## 1 Introduction

Set computation for a dynamic system involves the computation of the dynamic evolution of the system with uncertainties on system parameters, noises, inputs, and initial states. All the uncertainties are represented by the corresponding sets and the computation takes a set as the input and returns another set as the output. Set computation is closely related to computational geometry since the sets can be represented by various forms of geometric entities such as zonotopes and polytopes [1]. As one of the oldest fields of computing, computational geometry involves the design, analysis, and implementation of efficient algorithms for solving geometric input and output problems [2]. Intervals or boxes are the simplest geometric entity and interval arithmetic has been developed to conduct interval set computation for dynamic systems with many successful applications in control and robotics [3]. However, the main disadvantage of interval set computation is that the shape of the input set is restricted to be an interval or a box and the corresponding wrapping effect of interval set computation makes the output set to be often excessively over-approximated. A common operation in interval arithmetic is to bisect an interval into smaller intervals for interval set computation, which can reduce the wrapping effect [4]. An interval ellipsoid arithmetic was also proposed in [5] to relax the shape restriction as well as to reduce the wrapping effect.

A zonotope is a centrally symmetric convex polytope and it is an extension of interval arithmetic in terms of both the shape and the wrapping effect [6, 7]. The shape of a zonotope is less rigid with comparison to an interval or a box and also the wrapping effect for zonotopic set computation can be reduced significantly as centred inclusion functions are used in zonotopic set computation while natural inclusion functions are often used in interval arithmetic [8]. It is worth noting that other inclusion functions such as those based on the first-order Taylor expansion in [9] or the combination of the first-order Taylor expansion and the centered inclusion in [10] can also be used for zonotopic set computation. A zonotope can also be bisected into smaller zonotopes for reducing the wrapping effect as in [11].

In many applications such as state estimation, the input set or the state of a dynamic system is a polytope or even a non-convex set [12]. It is therefore needed to compute the dynamic evolution of this polytopic set for a nonlinear discrete-time system [13]. However, a polytopic set is usually represented by linear inequality constraints rather than equality equations, which are used for representing intervals and zonotopes. So the propagation of a polytopic set for a nonlinear system cannot be conducted in a similar way to intervals and zonotopes. A common practice in the literature is to bound the polytopic set by an optimal zonotope and then to compute the propagation of this zonotope instead using zonotopic set computation [13–16]. However, the bounding of a polytopic set by a single zonotope is usually an over-approximation of the polytopic set, which increases the wrapping effect significantly. The concept of constrained zonotope was also proposed in [17] as a new tool for state estimation and fault detection. A constrained

zonotope has an equality constraint on the box that is used to generate an ordinary zonotope and thus a constrained zonotope turns out to be a convex polytope. The specific mathematical format of constrained zonotopes facilitates its basic set operations such as the Minkowski sum and the intersection with other sets, which can be used for state estimation of nonlinear discrete-time systems with linear or nonlinear output functions [18, 19]. However, similar to zonotopic set computation, these approaches based on constrained zonotopes are restricted to use a single constrained zonotope for set propagation and intersection.

An indirectly implemented polytopic set computation method was proposed in [8], where the polytope was represented exactly by the intersection of individual zonotopes and the set computation was conducted by propagating these individual zonotopes and then intersecting the propagated zonotopes to obtain another polytopic set at the next time instant. This indirectly implemented polytopic set computation method can be regarded as a generic polytopic set computation method since the input set is a generic polytope and the output set is also a generic polytope. The proposed polytopic set computation method was applied to state estimation of a nonlinear uncertain discrete-time system with improved accuracy with comparison to the method of over-approximating the polytopic set obtained at each time instant by a single zonotope [8]. A potential issue for the indirectly implemented polytopic set computation method comes from the representation of a polytope exactly by the intersection of zonotopes: the obtained individual zonotopes are much bigger than the original polytope and thus the wrapping effect can be enlarged by these bigger zonotopes although the following intersection operation would reduce the wrapping effect significantly. In order to reduce such wrapping effect for this polytopic set computation method, this paper proposes to partition a polytopic set via Delaunay triangulation and the representation of a polytope exactly by the union of small zonotopes for zonotopic set computation. These proposed approaches are similar to the idea of bisection in interval arithmetic to reduce the wrapping effect of interval set computation. The novelty of the proposed approaches lies in two aspects: first, Delaunay triangulation is introduced as a set-theoretic method to partition a polytopic set for the first time; second, a polytopic set is to be represented exactly by the union of small zonotopes for the first time as well. The proposed novel polytopic set computation method of integrating these two approaches is also applied to state estimation of a nonlinear uncertain discrete-time system with further improved accuracy.

The rest of the paper is organised as follows. Section 2 provides the problem description for guaranteed state estimation of nonlinear uncertain discrete-time systems. Section 3 reviews interval, zonotopic and polytopic set computation methods and their wrapping effect and then proposes the novel approaches of partitioning a polytopic set via Delaunay triangulation and the representation of a polytopic set exactly by the union of small zonotopes for reducing the wrapping effect of polytopic set computation. The detailed algorithm for guaranteed state estimation of nonlinear discrete-time systems via the proposed polytopic set computation method is given in Sect. 4. An illustrative example is provided in Sect. 5 to demonstrate the effectiveness of the proposed method. Finally, some conclusions and future work are provided in Sect. 6.

## 2 Problem description

A set-membership state estimation problem is to be studied, which is to estimate or bound the real state of the system given the mathematical model of the system and also noise-corrupted measurements of the system output. Considering the following nonlinear uncertain discrete-time system [8]:

$$\begin{cases} \mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \omega_{k-1}) \\ \mathbf{y}_k = \mathbf{g}(\mathbf{x}_k, v_k), \end{cases} \quad (1)$$

where  $\mathbf{x}_k \in \mathfrak{R}^n$  and  $\mathbf{y}_k \in \mathfrak{R}^p$  are the system state and the system output at time instant  $k$ , respectively;  $\omega_k \in \mathfrak{R}^{n_\omega}$  denotes to uncertain process parameters and process perturbations;  $v_k \in \mathfrak{R}^{n_v}$  denotes to observation noises. The state function  $\mathbf{f}(\mathbf{x}_{k-1}, \omega_{k-1})$  is assumed to be nonlinear and the output function  $\mathbf{g}(\mathbf{x}_k, v_k)$  is assumed to be linear as in [13] with the format of  $\mathbf{g}(\mathbf{x}_k, v_k) = \mathbf{c}^T \mathbf{x}_k + \mathbf{d}^T v_k$ . The initial state and all the uncertainties are assumed to be bounded by known compact sets:  $\mathbf{x}_0 \in X_0$ ,  $\omega_k \in W_k$  and  $v_k \in V_k$ .

Starting from an initial set  $X_0$ , set-membership state estimation is to estimate recursively the set  $X_k (k = 1, 2, \dots)$  for the system state at future time instants. The estimated set  $X_k$  is usually an over-estimation of any feasible system state under all the uncertainties  $W_k (k = 0, 1, 2, \dots)$  and  $V_k (k = 1, 2, \dots)$ . Furthermore, the estimated set  $X_k$  is also to be consistent with the observed system output  $\mathbf{y}_k (k = 1, 2, \dots)$ .

Assume that  $X_{\mathbf{y}_k} = \{\mathbf{x} \in \mathfrak{R}^n : \mathbf{y}_k \in \mathbf{g}(\mathbf{x}, V_k)\}$  denotes to the system state that is consistent with the observed output  $\mathbf{y}_k$  at time instant  $k$ , then  $X_k$  for the system state at time instant  $k$ , which is the set of all feasible states, can be obtained as follows:

$$X_k = \mathbf{f}(X_{k-1}, W_{k-1}) \cap X_{\mathbf{y}_k}, k = 1, 2, \dots \quad (2)$$

So the set-membership state estimation problem to be considered here needs to compute the dynamic evolution of the nonlinear discrete-time system starting from an initial polytopic set  $X_0$  and also the intersection of two polytopic sets  $\mathbf{f}(X_{k-1}, W_{k-1})$  and  $X_{\mathbf{y}_k}$  at each time instant. The intersection of two polytopes can be obtained directly through the combination of their corresponding linear inequality constraints.

The wrapping effect of the set-membership state estimation problem mainly comes from the task of computing the dynamic evolution of the nonlinear discrete-time system as well as the approximation of the propagated set by a single zonotope. Initially, interval set computation was used for this set computation task where the admissible state space was bisected and selected into subsets to test their consistency with the observations [20–22]. Zonotopic set computation has been increasingly used for this set computation task where an optimised zonotope was obtained at each time instant to bound the intersection in (2) for the corresponding zonotopic set computation with the reduced wrapping effect [13–16]. Constrained zonotopes are also used in a similar way to zonotopic set computation for state estimation where a single constrained zonotope is obtained

at each time instant and the intersection in (2) can be conducted in the format of constrained zonotopes rather than in the format of polytopes [18, 19]. Making full use of zonotopic set computation for set propagation and polytope geometry for set intersection, the indirectly implemented polytopic set computation method was proposed for the set-membership state estimation problem to reduce the wrapping effect further and thus to improve the accuracy of estate estimation [8]. In the following section, various set computation methods and their wrapping effect are to be reviewed for the purpose of deriving novel polytopic set computation methods with the further reduced wrapping effect.

### 3 Set computation methods and their wrapping effect

Set computation methods originate from interval arithmetic, where real numbers are enclosed by intervals and real vectors are enclosed by boxes. The associated interval analysis has become a fundamental set computation tool to represent uncertainties or errors, prove properties of sets, solve equations or inequalities, and optimize in a global way [3]. Combining interval arithmetic with computational geometry, set computation methods have been extended from the initial interval set computation to the more accurate polytopic set computation.

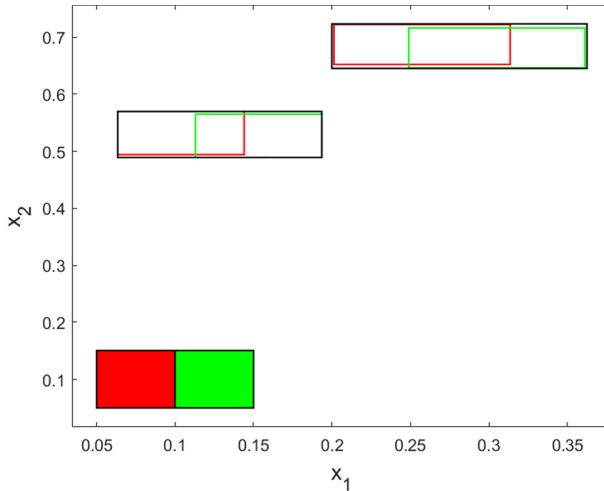
The wrapping effect of set computation methods plays an essential role for the accuracy of their set-based solutions. In the following subsections, the existing set computation methods as well as the proposed novel polytopic set computation methods and their wrapping effect are to be studied and compared on the basis of a benchmark example of computing the dynamic evolution of the following nonlinear uncertain discrete-time system with a set as the initial state:

$$\begin{cases} x_1(k+1) = 0.99x_1(k) + \delta(k)x_2(k), \delta(k) \in [0.28, 0.3] \\ x_2(k+1) = -0.1x_1(k) + \frac{0.5x_2(k)}{1+x_2^2(k)} + \omega(k), \omega(k) \in [0.48, 0.5], \end{cases} \quad (3)$$

where  $x_1(0) \in [0.05, 0.15], x_2(0) \in [0.05, 0.15]$  with an initial volume of 0.01. Two steps are to be computed for the dynamic evolution of this nonlinear uncertain discrete-time system using various set computation methods and the comparison is based on the volumes of the obtained sets at these two steps.

#### 3.1 Interval set computation methods and their wrapping effect

Interval set computation builds on the concept of inclusion functions. Consider a function  $\mathbf{f}(\mathbf{x})$  from  $\mathcal{R}^n$  to  $\mathcal{R}^m$ , the interval function  $\mathbb{F}$  from  $\mathbb{I}(\mathcal{R}^n)$  to  $\mathbb{I}(\mathcal{R}^m)$  is an inclusion function of  $\mathbf{f}$  if  $\forall \mathbf{x} \in \mathbb{I}(\mathcal{R}^n), \mathbf{f}(\mathbf{x}) \subseteq \mathbb{F}(\mathbf{x})$ . The natural inclusion function of  $\mathbf{f}(\mathbf{x})$  can be obtained by replacing each occurrence of every variable with the corresponding interval variable, by executing all operations according to interval arithmetic, and by computing ranges of the standard functions [3]. Taking the nonlinear uncertain discrete-time system in (3) as an example, the dynamic evolution of two steps for the system starting from the initial state of  $x_1(0) \in [0.05, 0.15], x_2(0) \in [0.05, 0.15]$  is shown



**Fig. 1** Interval set computation methods and their wrapping effect

as black boxes in Fig. 1 using interval set computation. This method is denoted as Method 1a: Interval set computation without bisection as no bisection is applied for the approach and the volumes of two obtained sets for Step 1 and Step 2 are 0.0104 and 0.0127, respectively.

However, interval set computation based on natural inclusion functions tends to be over-approximated, which is also called the wrapping effect. A typical remedy to reduce the wrapping effect is to bisect the interval into a subpaving and the resulting set computation is to be conducted on smaller intervals. The initial state of  $x_1(0) \in [0.05, 0.15], x_2(0) \in [0.05, 0.15]$  for the system in (3) is bisected into two sub-boxes and the dynamic evolution of two steps for these two sub-boxes is also plotted in Fig. 1, where two boxes are obtained at each time instant and their convex hull is used to compute the volume of the obtained set for the comparison. The volumes of two obtained convex hulls for Step 1 and Step 2 are 0.0102 and 0.0119, respectively. This method with bisection is denoted as Method 1b: Interval set computation with bisection. It can be seen that the wrapping effect can be reduced slightly from the dynamic evolution of smaller intervals with comparison to the direct dynamic evolution of the original box.

### 3.2 Zonotopic set computation methods and their wrapping effect

Given a vector  $\mathbf{p} \in \mathfrak{R}^n$  and a matrix  $H \in \mathfrak{R}^{n \times m}$ , a zonotope  $\mathcal{Z}$  of order  $n \times m$  is the set to be represented by:

$$\mathbf{p} \oplus H\mathbf{B}^m = \{\mathbf{p} + H\mathbf{z} | \mathbf{z} \in \mathbf{B}^m\}, \tag{4}$$

where  $\mathbf{B}^m$  is a box composed of  $m$  unitary intervals  $\mathbf{B} = [-1, 1]$  and  $\oplus$  is the Minkowski sum of sets. So a zonotope is derived from  $m$  unitary intervals and it becomes a box if the matrix  $H$  is diagonal.

Zonotopic set computation methods build on the seminal work in [6, 14] and zonotopes are used to compute the dynamic evolution of a nonlinear system with a guaranteed sub-exponential over-estimation. The principle of zonotopic set computation is to be explained as follows.

For a function  $\mathbf{f}(\mathbf{x}) : \mathfrak{R}^n \rightarrow \mathfrak{R}^n, \mathbf{x} \in \mathcal{Z} \subset \mathbb{X} \subset \mathbb{I}^n$  with a zonotopic set  $\mathcal{Z}$  as its initial state,  $\mathcal{Z} = \mathbf{p} \oplus H\mathbf{B}^m$  and  $\mathbb{X}$  is the bounding box for  $\mathcal{Z}$ , the centered inclusion function  $\mathbf{F}_c(\mathcal{Z})$  for  $\mathbf{f}(\mathcal{Z})$ , i.e.,  $\mathbf{f}(\mathcal{Z}) \subseteq \mathbf{F}_c(\mathcal{Z})$ , can be deduced by the mean-value theorem [14]:

$$\mathbf{F}_c(\mathcal{Z}) = \mathbf{f}(\mathbf{p}) + \nabla_{\mathbf{x}}\mathbf{f}(\mathbb{X})(\mathcal{Z} - \mathbf{p}), \tag{5}$$

where  $\mathcal{Z} - \mathbf{p} = H\mathbf{B}^m$ . So the centered inclusion function  $\mathbf{F}_c(\mathcal{Z})$  of  $\mathbf{f}(\mathcal{Z})$  is a family of zonotopes represented by  $\mathcal{Z} = \mathbf{q} \oplus \mathbb{M}\mathbf{B}^m$ , where  $\mathbf{q} = \mathbf{f}(\mathbf{p})$  and  $\mathbb{M} = \nabla_{\mathbf{x}}\mathbf{f}(\mathbb{X})H \subset \mathbb{I}^{n \times m}$  is an interval matrix.  $\mathcal{Z}$  can be further bounded by a single zonotope to be represented by  $\diamond(\mathcal{Z})$  [14]:

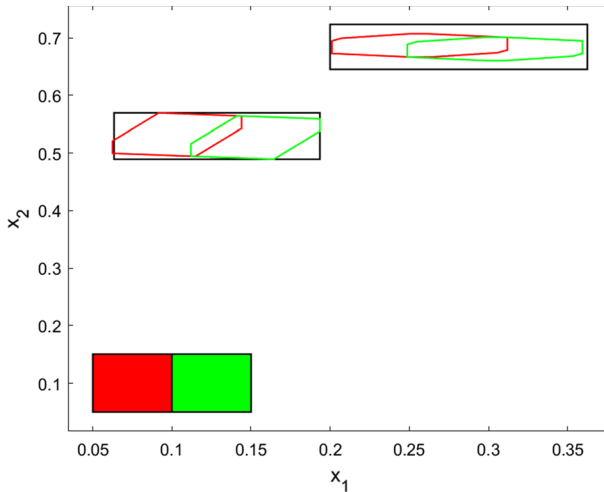
$$\diamond(\mathcal{Z}) = \mathbf{q} \oplus [\text{mid}(\mathbb{M}) \ G] \begin{bmatrix} \mathbf{B}^m \\ \mathbf{B}^n \end{bmatrix}, \tag{6}$$

where  $\text{mid}(\mathbb{M})$  is the centered-point matrix of  $\mathbb{M}$  and  $G \in \mathfrak{R}^{n \times n}$  is a diagonal matrix that satisfies

$$G_{ii} = \sum_{j=1}^m \frac{\text{diam}(\mathbb{M}_{ij})}{2}, i = 1, \dots, n \tag{7}$$

and  $\text{diam}(\mathbb{M}_{ij})$  is the length of the interval  $\mathbb{M}_{ij}$ . So  $\mathbf{f}(\mathcal{Z}) \subseteq \mathbf{F}_c(\mathcal{Z}) \subseteq \diamond(\mathcal{Z})$  where the dynamic evolution of the system can be computed with a zonotopic set  $\mathcal{Z}$  as the system input and another zonotopic set  $\diamond(\mathcal{Z})$  as the system output. This is the primary principle of zonotopic set computation for computing the dynamic evolution of a nonlinear system and it uses centered inclusion functions rather than natural inclusion functions. Other inclusion functions can also be used for zonotopic set computation such as those based on the first-order Taylor expansion [9] or their combination with centered inclusion functions [10].

Taking the same nonlinear discrete-time system in (3) as an example, the initial state of  $x_1(0) \in [0.05, 0.15], x_2(0) \in [0.05, 0.15]$  can be re-represented as a zonotope to compute the dynamic evolution of two steps using zonotopic set computation. Similarly, the initial set of a zonotope can also be bisected into two sub-zonotopes for zonotopic set computation [11]. The corresponding zonotopic set computation methods are denoted as Method 2a: Zonotopic set computation without bisection and Method 2b: Zonotopic set computation with bisection, respectively. The dynamic evolution of two steps using these two methods is plotted in Fig. 2, where the volumes of two obtained zonotopes in Step 1 and Step 2 for Method 2a are 0.0080 and 0.0058, respectively; and the volumes of two obtained convex hulls in Step 1 and Step 2 for Method 2b are 0.0080



**Fig. 2** Zonotopic set computation methods and their wrapping effect

and 0.0057, respectively. The result of Fig. 1 is also plotted in Fig. 2 as outer boxes for the purpose of comparison. It can be seen that the wrapping effect has been reduced significantly by zonotopic set computation instead of interval set computation and Method 2b has also reduced the wrapping effect slightly for Step 2 with comparison to Method 2a, which is consistent to the difference between Method 1a and Method 1b and reflects the benefit of the reduced wrapping effect for smaller set computation.

### 3.3 Polytopic set computation methods and their wrapping effect

In some applications such as the set-membership state estimation problem described in Sect. 2, the initial state of the dynamic system as well as the propagated state at the next time instant is often a polytopic set rather than a box or a zonotope. Therefore, it is often needed to compute the dynamic evolution of a polytopic set for a nonlinear system. As a polytope is often represented by linear inequality constraints or their vertices, polytopic set computation cannot be implemented directly in a similar way to interval and zonotopic set computation. All polytopes used in the paper are represented by linear inequality constraints or their vertices as in [1]. A common practice to implement polytopic set computation is to over-approximate the polytopic set by a single zonotope and then to propagate this single zonotope using zonotopic set computation [13]. Such an over-approximation of the polytopic set by a single zonotope amplifies the wrapping effect and increases the conservatism of the set-based solution.

The indirectly-implemented polytopic set computation method in [8] represents the polytopic set at each time instant exactly by the intersection of individual zonotopes and then to propagate these individual zonotopes using zonotopic set computation. At the next time instant, another polytope is obtained from the intersection of



these propagated zonotopes and then the same procedure is adopted to represent the renewed polytope exactly by the intersection of zonotopes. This original polytopic set computation method is denoted as Method 3a: Polytopic set computation via the intersection of zonotopes. Taking the same nonlinear discrete-time system in (3) as an example, its dynamic evolution of two steps using Method 3a is shown in Fig. 3, where the initial state is assumed to be a polytope with a volume of 0.0074. The polytope for the initial state has four vertices of (0.05, 0.15), (0.15, 0.132), (0.06, 0.05) and (0.12, 0.05). These vertices are all contained in the initial box of  $x_1(0) \in [0.05, 0.15], x_2(0) \in [0.05, 0.15]$ . This box can be assumed to be the bounding zonotope for the initial polytopic set in a similar way to traditional methods [13–16] and thus the result of Method 3a is comparable to the results of the previous methods in Figs. 1 and 2. The initial polytope is represented exactly by the intersection of two zonotopes  $z_1 = \begin{bmatrix} 0.105 \\ 0.091 \end{bmatrix} \oplus \begin{bmatrix} -0.0500 & -0.0050 \\ 0.0090 & 0.0500 \end{bmatrix} \mathbb{B}^2$  and  $z_2 = \begin{bmatrix} 0.0850 \\ 0.1000 \end{bmatrix} \oplus \begin{bmatrix} 0.0183 & 0.0533 \\ 0.0500 & 0 \end{bmatrix} \mathbb{B}^2$ , which are also shown in Fig. 3 as blue parallelograms or zonotopes. The intersection of these two propagated zonotopes at Step 1 generates another polytope with a volume of 0.0064. Accordingly, the polytopic set obtained at Step 2 has a volume of 0.0051. It can be seen that the wrapping effect of Method 3a has been reduced significantly with comparison to the cases of using interval and zonotopic set computation in Figs. 1 and 2, respectively.

Similar to the bisection of an interval and a zonotope in Figs. 1 and 2, Delaunay triangulation is proposed to partition a polytopic set into smaller sets for set computation with reduced wrapping effect. The corresponding polytopic set computation method is denoted as Method 3b: Polytopic set computation via Delaunay triangulation and the intersection of zonotopes,

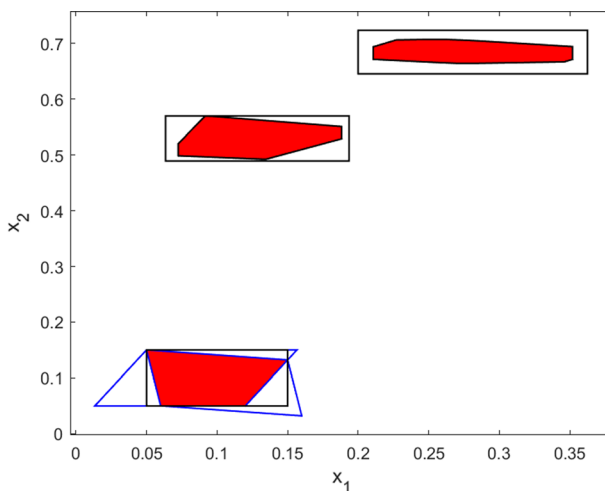
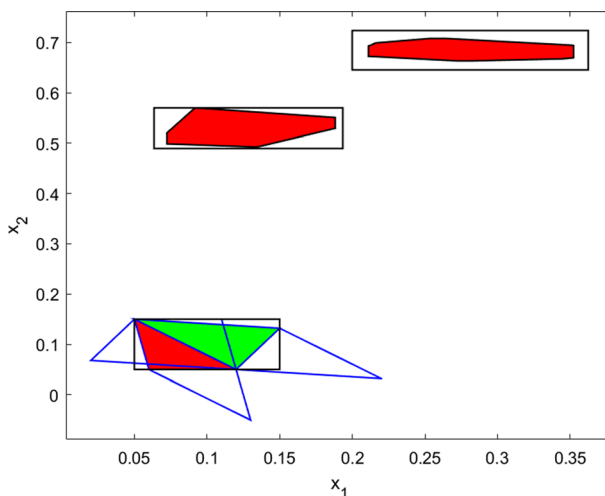


Fig. 3 Method 3a: polytopic set computation via the intersection of zonotopes

where a polytopic set is partitioned into smaller polytopes and each polytope is to be represented as the intersection of zonotopes for zonotopic set computation. Triangulation is a process to organize arbitrarily distributed data points in a triangular mesh and Delaunay triangulation provides an efficient and optimal way to organize distributed data points in a triangular mesh. Here it is used as a mathematical tool to partition a polytopic set on the basis of the following Theorem 1 [23].

**Theorem 1** *Let  $\mathcal{P}$  be the set of  $n$  points in the plane, not all collinear, and let  $k$  denote the number of points in  $\mathcal{P}$  that lie on the boundary of the convex hull of  $\mathcal{P}$ . Then any triangulation of  $\mathcal{P}$  has  $2n - k - 2$  triangles and  $3n - k - 3$  edges.*

The proof of Theorem 1 as well as its extension to high-dimensional spaces can be found in [23]. As a special format of triangulation, Delaunay triangulation has the property of maximizing the minimum angle of the triangles involved in any triangulation, which is advantageous from the perspective of partitioning a polytope more evenly. Furthermore, Delaunay triangulation returns geometric entities of fixed shape such as triangles in a 2-D space and tetrahedrons in a 3-D space in the format of their vertices, which also facilitates further set operations on them. It is worth noting that Delaunay triangulation is a fundamental concept in computational geometry and there exist efficient algorithms to implement Delaunay triangulation for a generic polytope. The dynamic evolution of two steps using Method 3b for the nonlinear discrete-time system in (3) is shown in Fig. 4, where the initial polytopic set has been partitioned into  $2n - k - 2 = 2$  triangles with  $n = 4, k = 4$  according to Theorem 1 and the propagation of each triangle has been computed by propagating those individual zonotopes whose intersection forms the triangle. The volumes for the convex hulls obtained at Step 1 and Step 2 are 0.0064 and 0.0050,



**Fig. 4** Method 3b: polytopic set computation via Delaunay triangulation and the intersection of zonotopes

respectively. Similar to interval and zonotopic set computation methods, the wrapping effect of Method 3b has been reduced slightly in Step 2 with comparison to Method 3a, which also reflects the benefit of smaller set computation.

The main effort for polytopic set computation via the intersection of zonotopes is to represent the polytope exactly by the intersection of zonotopes, which is a challenging task especially for high-dimensional systems. Furthermore, each individual zonotope should contain the original polytope and thus the obtained zonotopes are much bigger than the original polytope as shown in Figs. 3 and 4, which is disadvantageous for reducing the wrapping effect although the following intersection operation of these propagated zonotopes at the next time instant can reduce the wrapping effect. Instead of representing a polytopic set by the intersection of zonotopes, another novel polytopic set computation method is proposed and it is denoted as Method 3c: Polytopic set computation via Delaunay triangulation and the union of zonotopes. Similar to Method 3b, the initial polytopic set is partitioned into two smaller polytopes via Delaunay triangulation. For the initial polytopic set in Fig. 4, each triangle obtained from Delaunay triangulation can be represented by the union of three parallelotopes or zonotopes. Taking one triangle obtained from Delaunay triangulation as an example, these three zonotopes are shown in Fig. 5 by different colours with  $z_1 = \begin{bmatrix} 0.1631 \\ 0.5279 \end{bmatrix} \oplus \begin{bmatrix} 0.0067 & -0.0050 \\ 0.0047 & 0.0122 \end{bmatrix} \mathbf{B}^2$ ,

$$z_2 = \begin{bmatrix} 0.1698 \\ 0.5326 \end{bmatrix} \oplus \begin{bmatrix} -0.0117 & 0.0067 \\ 0.0076 & 0.0047 \end{bmatrix} \mathbf{B}^2, z_3 = \begin{bmatrix} 0.1581 \\ 0.5401 \end{bmatrix} \oplus \begin{bmatrix} -0.0117 & -0.0050 \\ 0.0076 & 0.0122 \end{bmatrix} \mathbf{B}^2$$

and their union is the

triangle itself. These three parallelotopes or zonotopes are obtained by linking three middle points of the edges of the obtained triangle, which are also plotted in Fig. 5. It can be seen that the combination of three middle points with any vertex of the

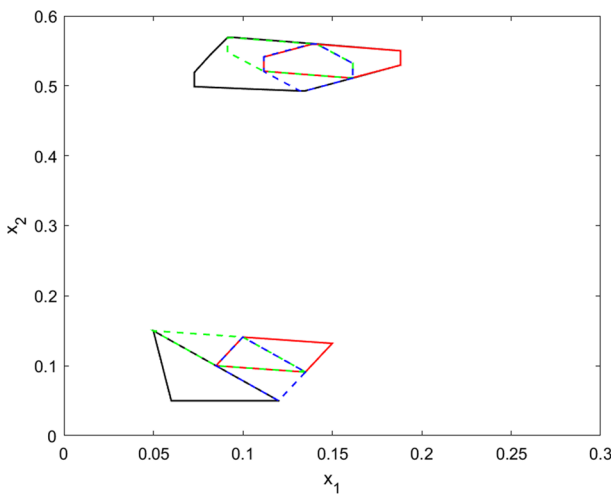
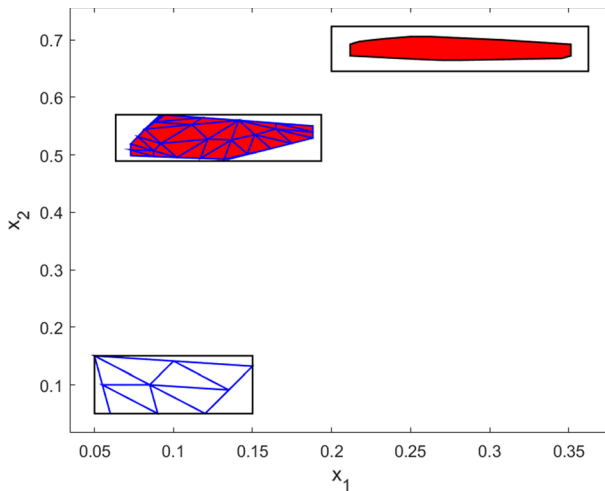


Fig. 5 Represent a polytopic set by the union of zonotopes



**Fig. 6** Method 3c: polytopic set computation via Delaunay triangulation and the union of zonotopes

original triangle would form a parallelopete or a zonotope  $\mathcal{Z} = \mathbf{p} \oplus H\mathbf{B}^2$ , where  $\mathbf{p}$  is the center of the parallelopete and the matrix  $H$  can be derived from the four vertices of the parallelopete. So totally the initial polytopic set can be represented by the union of six unique small zonotopes for zonotopic set computation. The propagation of these six unique small zonotopes as well as their convex hull is also shown in Fig. 5. The convex hull of all these six propagated zonotopes is a new polytopic set at the next time instant and the same procedure is adopted to represent the new polytope exactly by the union of small zonotopes via Delaunay triangulation. It is worth noting that a single parallelopete was also used to bound state in [24]. The approach for the representation of a polytopic set exactly by the union of zonotopes is only demonstrated for a 2-D system here and its extension to higher-dimensional systems needs further investigation.

The result of computing two steps for the dynamic evolution of the nonlinear discrete-time system in (3) using Method 3c is shown in Fig. 6 where the initial polytopic set as well as the polytopic set obtained at Step 1 is represented exactly by the union of small zonotopes. The volumes for the convex hulls obtained at Step 1 and Step 2 are 0.0063 and 0.0047, respectively. It can be seen that the wrapping effect has been further reduced with comparison to Method 3b. However, the principle of Method 3c is only demonstrated for a 2-D case, which is also the focus of this paper.

To conclude, the summary for all these seven set computation methods for the task of computing the dynamic evolution of the nonlinear discrete-time system is provided in Table 1 where the methods are classified according to the format of input set, the operation of set partition or no, and the types of inclusion functions used for set propagation, respectively. It can be seen that the format of intervals and zonotopes are only used for set propagation and all propagated sets are to be

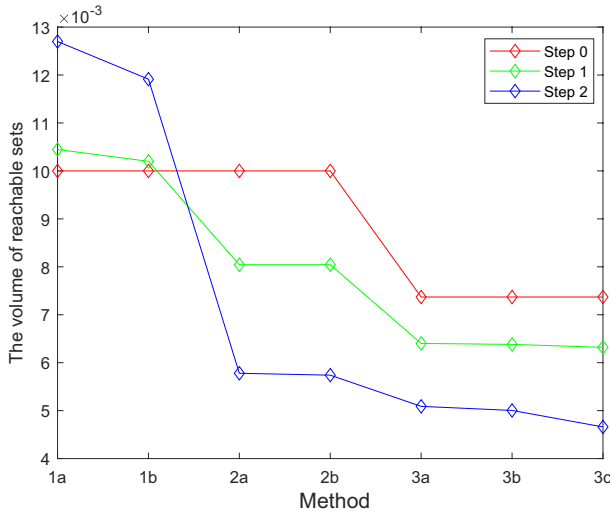


Fig. 7 The comparison of set computation methods in terms of the volumes for the obtained sets

Table 1 Summary of set computation methods

No.	Input set	Set partition	Set propagation	Output set
1a	An interval $\mathbb{X}$	No $\mathbb{X} = \mathbb{X}$	Natural inclusion $\mathbf{f}(\mathbb{X})$	An interval $\mathbb{X}_O = \mathbf{f}(\mathbb{X})$
1b	An interval $\mathbb{X}$	Bisection $\mathbb{X} = \mathbb{X}_1 \cup \mathbb{X}_2$	Natural inclusion $\mathbf{f}(\mathbb{X}_1), \mathbf{f}(\mathbb{X}_2)$	The union of intervals $\mathbb{X}_O = \mathbf{f}(\mathbb{X}_1) \cup \mathbf{f}(\mathbb{X}_2)$
2a	A zonotope $\mathcal{Z}$	No $\mathcal{Z} = \mathcal{Z}$	Centered inclusion $\mathbf{f}(\mathcal{Z})$	A zonotope $\mathcal{Z}_O = \mathbf{f}(\mathcal{Z})$
2b	A zonotope $\mathcal{Z}$	Bisection $\mathcal{Z} = \mathcal{Z}_1 \cup \mathcal{Z}_2$	Centered inclusion $\mathbf{f}(\mathcal{Z}_1), \mathbf{f}(\mathcal{Z}_2)$	The union of zonotopes $\mathcal{Z}_O = \mathbf{f}(\mathcal{Z}_1) \cup \mathbf{f}(\mathcal{Z}_2)$
3a	A polytope $\mathcal{P}$	No $\mathcal{P} = \mathcal{P}$	Centered inclusion $\mathbf{f}(\mathcal{Z}_1), \mathbf{f}(\mathcal{Z}_2) \dots \mathbf{f}(\mathcal{Z}_n)$ $\mathcal{P} = \mathcal{Z}_1 \cap \mathcal{Z}_2 \dots \mathcal{Z}_n$	A polytope $\mathcal{P}_O =$ $\mathbf{f}(\mathcal{Z}_1) \cap \mathbf{f}(\mathcal{Z}_2) \dots \mathbf{f}(\mathcal{Z}_n)$
3b	A polytope $\mathcal{P}$	Delaunay triangulation $\mathcal{P} = \mathcal{T}_1 \cup \mathcal{T}_2 \dots \mathcal{T}_m$	Centered inclusion $\mathbf{f}(\mathcal{Z}_i^1), \mathbf{f}(\mathcal{Z}_i^2) \dots \mathbf{f}(\mathcal{Z}_i^n)$ $\mathcal{T}_i = \mathcal{Z}_i^1 \cap \mathcal{Z}_i^2 \dots \mathcal{Z}_i^n$	A polytope $\mathcal{P}_O =$ $\text{Hull}(\mathbf{f}(\mathcal{T}_1) \cup \mathbf{f}(\mathcal{T}_2) \dots \mathbf{f}(\mathcal{T}_m)),$ $\mathbf{f}(\mathcal{T}_i) = \mathbf{f}(\mathcal{Z}_i^1) \cap \mathbf{f}(\mathcal{Z}_i^2) \dots \mathbf{f}(\mathcal{Z}_i^n)$
3c	A polytope $\mathcal{P}$	Delaunay triangulation $\mathcal{P} = \mathcal{T}_1 \cup \mathcal{T}_2 \dots \mathcal{T}_m$	Centered inclusion $\mathbf{f}(\mathcal{Z}_i^1), \mathbf{f}(\mathcal{Z}_i^2), \mathbf{f}(\mathcal{Z}_i^3)$ $\tau_i = \mathcal{Z}_i^1 \cup \mathcal{Z}_i^2 \cup \mathcal{Z}_i^3$	A polytope $\mathcal{P}_O =$ $\text{Hull}(\mathbf{f}(\mathcal{T}_1) \cup \mathbf{f}(\mathcal{T}_2) \dots \mathbf{f}(\mathcal{T}_m)),$ $\mathbf{f}(\mathcal{T}_i) = \mathbf{f}(\mathcal{Z}_i^1) \cup \mathbf{f}(\mathcal{Z}_i^2) \cup \mathbf{f}(\mathcal{Z}_i^3)$

converted into the format of polytopes for computing the intersection of polytopes or the convex hull of polytopes. The output set or the convex hull of the output set of these seven methods is also a polytope and its volume is used for comparing the corresponding performance, which is shown in Table 2 and Fig. 7, respectively. The volume of the obtained set or the convex hull of small sets obtained at each step reflects the wrapping effect of each set computation method directly. Figure 7

**Table 2** The comparison of set computation methods in terms of the volumes for the obtained sets

Method No.	Step 0	Step 1	Step 2
1a	0.0100	0.0104	0.0127
1b	0.0100	0.0102	0.0119
2a	0.0100	0.0080	0.0058
2b	0.0100	0.0080	0.0057
3a	0.0074	0.0064	0.0051
3b	0.0074	0.0064	0.0050
3c	0.0074	0.0063	0.0047

demonstrates a clear trend of reducing the wrapping effect from interval set computation to zonotopic and polytopic set computation. It also demonstrates the benefit of partitioning the set for smaller set computation to reduce the wrapping effect. Furthermore, the proposed Method 3c have the best performance in terms of having the minimal volume for the reachable set at Step 2 and therefore it is to be used for the set-membership state estimation problem in Sect. 2.

#### 4 Guaranteed state estimation using method 3c

Applying Method 3c in Sect. 3 for the set-membership state estimation problem in Sect. 2, the algorithm for guaranteed state estimation of nonlinear uncertain discrete-time systems is listed as follows:

- *Step 1* partition the past system state of a polytopic set  $X_{k-1}$  into a union of small zonotopes  $X_{k-1} = \mathcal{Z}_1 \cup \dots \cup \mathcal{Z}_{n_z}$  via Delaunay triangulation, where  $n_z$  is the total number of zonotopes whose union forms the polytopic set;
- *Step 2* compute the dynamic evolution of these small zonotopic sets  $\mathbf{f}(\mathcal{Z}_1, W_{k-1})$ ,  $\dots$ ,  $\mathbf{f}(\mathcal{Z}_{n_z}, W_{k-1})$  individually by zonotopic set computation using the centered inclusion function and converted the propagated sets into the format of a polytope;
- *Step 3* compute the convex hull of these propagated sets  $\text{Hull}(\mathbf{f}(\mathcal{Z}_1, W_{k-1}) \cup \dots \cup \mathbf{f}(\mathcal{Z}_{n_z}, W_{k-1}))$ , which is a polytope;
- *Step 4* compute the set of the system state  $X_{y_k}$  that is consistent with the observed system output  $\mathbf{y}_k$  and  $X_{y_k}$  is also a polytope due to the linearity of  $\mathbf{g}(\mathbf{x}_k, v_k)$ ;
- *Step 5* compute the system state at the next time instant  $X_k = \text{Hull}(\mathbf{f}(\mathcal{Z}_1, W_{k-1}) \cup \dots \cup \mathbf{f}(\mathcal{Z}_{n_z}, W_{k-1})) \cap X_{y_k}$ , which is another polytope;
- *Step 6* return to Step 1.

As  $\mathbf{f}(X_{k-1}, W_{k-1}) \cap X_{y_k} \subseteq \text{Hull}(\mathbf{f}(\mathcal{Z}_1, W_{k-1}) \cup \dots \cup \mathbf{f}(\mathcal{Z}_{n_z}, W_{k-1})) \cap X_{y_k} = X_k$ , the system states are guaranteed to be contained in the obtained polytopic sets  $X_k (k = 1, 2, \dots)$  at each time instant. The intersection of two polytopes in Step 5 can be obtained by combining their linear inequality constraints together and there is

no need to approximate the intersection of two polytopes by a single zonotope for set propagation. The obtained polytopetic set  $X_k$  is still an over-approximation of the real state. However, as demonstrated in Sect. 3, such over-approximation is mainly from the limited wrapping effect of zonotopic set computation for small zonotopes and their convex hull. It is anticipated that the accuracy of such set-membership estimation can be improved further with comparison to the original approach using Method 3a. Furthermore, Delaunay triangulation can also be efficiently implemented using the latest algorithms and hardware as reviewed in [25].

### 5 An illustrative example

Applying the set-membership state estimation algorithm in Sect. 4 to the following nonlinear uncertain discrete-time system [8]:

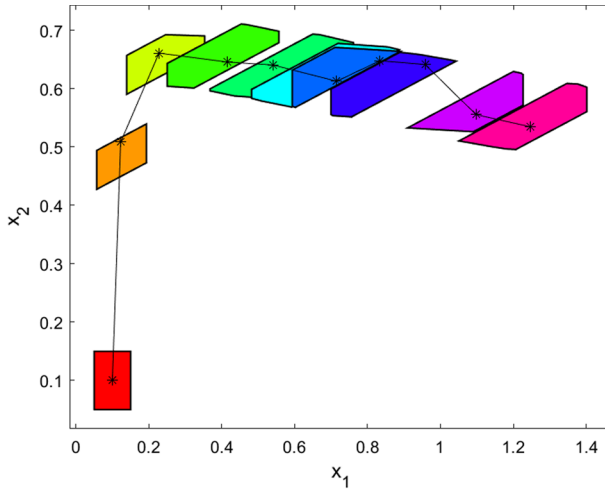
$$\begin{cases} x_1(k+1) = 0.99x_1(k) + \delta(k)x_2(k) \\ x_2(k+1) = -0.1x_1(k) + \frac{0.5x_2(k)}{1+x_2^2(k)} + \omega(k), \end{cases} \tag{8}$$

$$y(k) = x_1(k) - 3x_2(k) + v(k), \tag{9}$$

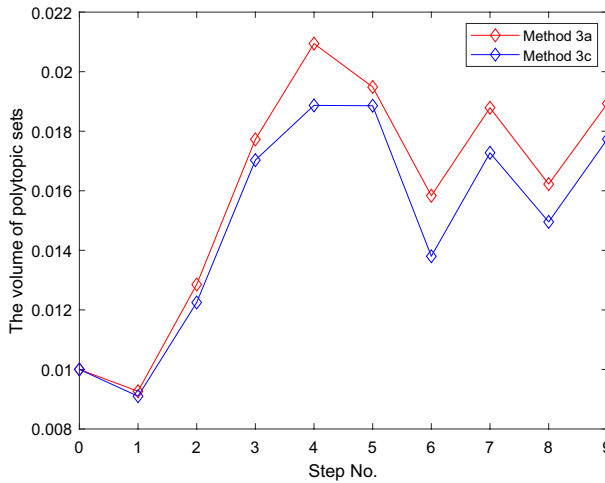
where  $\delta(k) \in [0.2, 0.3]$  is the system uncertain parameter;  $\omega(k) \in [0.4, 0.5]$  is the process perturbation;  $|v(k)| \leq 0.1$  is the bounded measurement noise. The initial state is assumed to be within a box  $x_1(0) \in [0.5, 0.15]$  and  $x_2(0) \in [0.5, 0.15]$  with the real initial state is set to be  $x_1(0) = 0.1$  and  $x_2(0) = 0.1$ . The output of the system  $y(k)$  is assumed to be measured at each time instant.

A polytopetic set is obtained at each time instant for the set-membership state estimation and this polytopetic set is the intersection of  $\text{Hull}(\mathbf{f}(\mathcal{Z}_1, W_{k-1}) \cup \dots \cup \mathbf{f}(\mathcal{Z}_{n_z}, W_{k-1}))$  and  $X_{y_k}$ , where  $\mathcal{Z}_1, \dots, \mathcal{Z}_{n_z}$  are small zonotopes whose union is the initial polytopetic set at the previous time instant and  $X_{y_k}$  is derived from the measurement. These small zonotopes are to be obtained by partitioning the polytopetic set via Delaunay triangulation and representing the resulting triangles exactly by the union of small zonotopes as shown in Fig. 5. The system state has been estimated for nine steps and the obtained polytopetic sets for these nine steps are shown in Fig. 8. It can be seen that the real state of the system is always contained in the computed polytopetic set at each time instant, which demonstrates the effectiveness of the proposed set-membership state estimation method.

Furthermore, the obtained polytopetic sets after observation for Method 3c have an average volume of 0.0155, which is smaller than the average volume of 0.0167 obtained from Method 3a for the same set-membership state estimation task, which stands for an average improvement of 7.19% for these nine steps. It is also worth noting that the polytopetic set obtained from Method 3c at each time instant is consistently smaller than those obtained from Method 3a as shown in Fig. 9, which confirms the benefit of using the union of small zonotopes rather than the intersection of large zonotopes for reducing the wrapping



**Fig. 8** Guaranteed state estimation using method 3c



**Fig. 9** The comparison of method 3a and method 3c for the set-membership state estimation task

effect of set computation. This is also consistent to the findings in Table 2 and Fig. 7, respectively. Furthermore, both Method 3a and Method 3c avoid the approximation of a polytopic set by a single zonotope at each step and the corresponding wrapping effect for such an approximation has also been avoided.



## 6 Conclusions

This paper has studied the wrapping effect of various set computation methods and it has also proposed novel approaches to reduce the wrapping effect for polytopic set computation methods. These approaches include the partition of a polytopic set via Delaunay triangulation and the representation of a polytopic set exactly by the union of zonotopes for zonotopic set computation. The integration of these novel approaches leads to a novel polytopic set computation method with the reduced wrapping effect. The proposed polytopic set computation method via Delaunay triangulation and the union of zonotopes has been successfully applied to solve the set-membership state estimation problem for a nonlinear uncertain discrete-time system with improved accuracy over an existing method for the same problem. The studies of the paper have highlighted the benefit of smaller set computation through set partition of bisection and Delaunay triangulation and also the representation of a generic polytope exactly by the union of zonotopes or the intersection of zonotopes to avoid the approximation of a polytopic set by a single zonotope or a box. Furthermore, set partitioning via Delaunay triangulation results in basic set units such as triangles in a 2-D space and tetrahedrons in a 3-D space, which could simplify the following set representation and propagation via zonotopes for reducing the wrapping effect of set computation.

Nevertheless, the proposed polytopic set computation method has only been implemented and verified for a 2-D system through an illustrative example. The extension of the proposed method to higher-dimensional systems with nonlinear output functions and the application of similar approaches to reduce the wrapping effect for other engineering problems are to be studied in the future work.

**Acknowledgements** This work was supported in part by the start-up grant of Aston University. The authors are very grateful for anonymous reviewers' comments and suggestions to improve the paper significantly.

**Funding** No other funding was involved in this work.

**Data availability** All of the material is owned by the authors and no permissions are required.

## Declarations

**Conflict of interest** The authors have no competing interests as defined by Springer, or other interests that might be perceived to influence the results and discussion reported in this paper.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Herceg M, Kvasnica M, Jones CN, Morari M (2013) Multi-parametric toolbox 3.0. In: Proceedings of the European control conference, Zürich, Switzerland, pp 502–510. <http://control.ee.ethz.ch/~mpt>
2. Mount DM (2012) CMSC754 computational geometry. In: Lecture note. University of Maryland, College Park, p 161. <https://www.cs.cmu.edu/afs/cs/academic/class/15456-s14/Handouts/cmsc754-lects.pdf>
3. Jaulin L, Kieffer M, Didrit O, Walter E (2001) Applied interval analysis with examples in parameter and state estimation, robust control and robotics. Springer, London, p 398
4. Barbroie C (1995) Reducing the wrapping effect. *Computing* 54(4):347–357. <https://doi.org/10.1007/BF02238232>
5. Neumaier A (1993) The wrapping effect, ellipsoid arithmetic, stability and confidence regions. In: Albrecht R, Alefeld G, Stetter HJ (eds) *Validation numerics*. Springer, Vienna, pp 175–190. [https://doi.org/10.1007/978-3-7091-6918-6\\_14](https://doi.org/10.1007/978-3-7091-6918-6_14)
6. Kühn W (1998) Rigorously computed orbits of dynamical systems without the wrapping effect. *Computing* 61(1):47–67. <https://doi.org/10.1007/BF02684450>
7. Le VTH, Stoica C, Alamo T, Camacho EF, Dumur D (2013) Zonotopic guaranteed state estimation for uncertain systems. *Automatica* 49(11):3418–3424. <https://doi.org/10.1016/j.automatica.2013.08.014>
8. Wan J, Sharma S, Sutton R (2018) Guaranteed state estimation for nonlinear discrete-time systems via indirectly implemented polytopic set computation. *IEEE Trans Autom Control* 63(12):4317–4322. <https://doi.org/10.1109/TAC.2018.2816262>
9. Calafiore G (2005) Reliable localization using set-valued nonlinear filters. *IEEE Trans Syst Man Cybern Part A Syst Hum* 35(2):189–197. <https://doi.org/10.1109/TSMCA.2005.843383>
10. Zhao Z, Wang Z, Zou L, Chen Y, Sheng W (2023) Zonotopic distributed fusion for nonlinear networked systems with bit rate constraint. *Inf Fusion* 90:174–184. <https://doi.org/10.1016/j.inffus.2022.09.014>
11. Wan J, Vehi J, Luo N (2009) A numerical approach to design control invariant sets for constrained nonlinear discrete-time systems with guaranteed optimality. *J Glob Optim* 44(3):395–407. <https://doi.org/10.1007/s10898-008-9334-6>
12. Althoff M, Frehse G, Girard A (2021) Set propagation techniques for reachability analysis. *Annu Rev Control Robot Auton Syst* 4(1):369–395. <https://doi.org/10.1146/annurev-control-071420-081941>
13. Chai W, Sun X, Qiao J (2013) Set membership state estimation with improved zonotopic description of feasible solution set. *Int J Robust Nonlinear Control* 23(14):1642–1654. <https://doi.org/10.1002/rnc.2842>
14. Alamo T, Bravo JM, Camacho EF (2005) Guaranteed state estimation by zonotopes. *Automatica* 41(6):1035–1043. <https://doi.org/10.1016/j.automatica.2004.12.008>
15. Alamo T, Bravo JM, Redondo MJ, Camacho EF (2008) A set-membership state estimation algorithm based on dc programming. *Automatica* 44(1):216–224. <https://doi.org/10.1016/j.automatica.2007.05.008>
16. Pan Z, Liu F (2023) Nonlinear set-membership state estimation based on the Koopman operator. *Int J Robust Nonlinear Control* 33(4):2703–2721. <https://doi.org/10.1002/rnc.6536>
17. Scott JK, Raimondo DM, Marseglia GR, Braatz RD (2016) Constrained zonotopes: a new tool for set-based estimation and fault detection. *Automatica* 69:126–136. <https://doi.org/10.1016/j.automatica.2016.02.036>
18. Rego BS, Raffo GV, Scott JK, Raimondo DM (2020) Guaranteed methods based on constrained zonotopes for set-valued state estimation of nonlinear discrete-time systems. *Automatica* 111:108614. <https://doi.org/10.1016/j.automatica.2019.108614>
19. Rego BS, Scott JK, Raimondo DM, Raffo GV (2021) Set-valued state estimation of nonlinear discrete-time systems with nonlinear invariants based on constrained zonotopes. *Automatica* 129:109638. <https://doi.org/10.1016/j.automatica.2021.109638>
20. Jaulin L, Kieffer M, Braems I, Walter E (2001) Guaranteed non-linear estimation using constraint propagation on sets. *Int J Control* 74(18):1772–1782. <https://doi.org/10.1080/00207170110090642>

21. Kieffer M, Jaulin L, Walter E (2002) Guaranteed recursive non-linear state bounding using interval analysis. *Int J Adapt Control Signal Process* 16(3):193–218. <https://doi.org/10.1002/acs.680>
22. Meslem N, Hably A, Raïssi T (2023) State and unknown input set-valued estimation for uncertain linear discrete-time systems. *Proc Inst Mech Eng Part I J Syst Control Eng* 237(4):571–584. <https://doi.org/10.1177/09596518231153316>
23. Berg M, Cheong O, Kreveld M, Overmars M (2008) *Computational geometry: algorithms and applications*, 3rd edn. Springer Berlin, Heidelberg. <https://doi.org/10.1007/978-3-540-77974-2>
24. Chisci L, Garulli A, Zappa G (1996) Recursive state bounding by parallelotopes. *Automatica* 32(7):1049–1055. [https://doi.org/10.1016/0005-1098\(96\)00048-9](https://doi.org/10.1016/0005-1098(96)00048-9)
25. Elshakhs YS, Deliparaschos KM, Charalambous T, Oliva G, Zolotas A (2024) A comprehensive survey on Delaunay triangulation: applications, algorithms, and implementations over CPUs, GPUs, and FPGAs. *IEEE Access* 12:12562–12585. <https://doi.org/10.1109/ACCESS.2024.3354709>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.