Improved Interval Reachability Bounds for Nonlinear Discrete-Time Systems using an Efficient One-Dimensional Partitioning Method*

Bowen Mu[1], Xuejiao Yang[1], Kai Shen[2], and Joseph K. Scott[1]

***Abstract*— This article presents a new method for accurately enclosing the reachable sets of nonlinear discrete-time systems with unknown but bounded disturbances. This method is motivated by the discrete-time differential inequalities method (DTDI) proposed by Yang and Scott, which exhibits state-of-the-art accuracy at low cost for many problems, but suffers from theoretical limitations that significantly restrict its applicability. The proposed method uses an efficient one-dimensional partitioning scheme to approximate DTDI while avoiding the key technical assumptions that limit it. Numerical result shows that this approach matches the accuracy of DTDI when DTDI is applicable, but, unlike DTDI, is valid for arbitrary systems.**

I. INTRODUCTION

This article introduces a new method for efficiently computing tight enclosures of the reachable sets of nonlinear discrete-time systems with bounded disturbances. Such enclosures are useful for robust control [1], [2] and fault detection [3], [4], and have applications in chemical systems [5], power systems [6], and autonomous vehicles [7], [8].

A variety of methods have been developed for bounding the reachable sets of nonlinear discrete-time systems (methods for continuous-time systems are reviewed elsewhere [9], [10]). Some methods compute reachable set enclosures over the entire horizon simultaneously by solving a large optimization problem [11], [12]. Although accurate bounds can be achieved in this way, they come at a relatively high computational cost. Alternatively, many methods compute enclosures recursively by repeatedly bounding the image of the current enclosure under the dynamics. The simplest of these propagates interval enclosures, which is very inexpensive [13], [14]. However, this typically produces conservative enclosures without partitioning and becomes inefficient with partitioning. More sophisticated methods use polytopes [15], [16], zonotopes [17], [18], [19], and constrained zonotopes [20], which can approximate the true reachable set much more accurately than intervals. However, propagating polytopic enclosures requires solving either many linear programs [15] or one larger nonlinear program [16] in each time step, which is inefficient. In contrast, zonotopes and constrained zonotopes can be propagated efficiently using conservative linearization techniques, but the results can be conservative when the linearization error is large [21]. The method in [22] avoids linearization, but only applies to polynomial systems. Moreover, it propagates parallelotope bundles using a procedure that requires solving multiple linear programs in each time step.

To address these challenges, Yang and Scott [23] developed a fast interval method that improves the accuracy of the standard interval method (without partitioning) using the theory of differential inequalities. We refer to this as the discrete-time differential inequalities (DTDI) method. The key feature of DTDI is that, when propagating the current interval to the next time, the dynamics are only bounded over the boundary of this interval (i.e., its $2n$ faces) rather than the entire interval. This idea originates from the continuous-time setting, where a trajectory cannot leave an enclosure without touching its boundary at some time. Consequently, when propagating enclosures forward in time, it suffices to consider only the values of the vector field on the enclosure boundary. This insight is central to the theory of differential inequalities and is used in many continuous-time reachability methods. Although this reasoning does not hold in discrete-time, Yang and Scott showed that a discrete-time analog of this approach is valid for systems satisfying a certain monotonicity property. In particular, it is valid for systems obtained by forward Euler discretization with a step size less than one over a Lipschitz constant for the system. The basic DTDI method in [23] produces much tighter bounds than the standard interval method without partitioning, but is still very conservative for many examples.

To further improve accuracy, some researchers have recently suggested using redundant model equations to refine the computed enclosures [23], [20]. *Redundant equations* refer to any explicit relationships involving the system states and uncertain parameters that are known to be satisfied by all solutions. Many systems naturally satisfy such relations, which may include conservation laws, forward invariant sets, or physical bounds on certain states. The use of redundant equations is well established in continuous-time reachability and results in very sharp bounds for many problems [24]. Moreover, systems that do not satisfy any known redundant relationships can also benefit from this approach using the concept of *manufactured redundancy* described in [9]. Yang and Scott first applied this approach in discrete-time in [23]. They incorporated redundancy into the standard interval method by using the redundant equations to refine the current interval at each step before propagating it forward. Unfortunately, this did not improve the accuracy much. Two similar methods for incorporating redundancy into constrained zonotope methods were later proposed in [20]. This resulted in much more accurate enclosures for one example, but it is not yet clear how these methods perform more generally.

*This material is based upon work supported by the National Science Foundation under grant no. 1949748

[1] School of Chemical and Biomolecular Engineering, Georgia Institute of Technology, GA 30332, USA (joseph.scott@chbe.gatech.edu)

[2] Aspen Technology, Incorporation, MA 01730, USA

Moreover, these methods are significantly more complex than the interval method and the current implementations involve the costly solution of linear programs in every time step. In [23], Yang and Scott also incorporated the use of redundant equations into their new DTDI method. Critically, the refinement step was applied to each face of the current bounding interval independently, rather than to the entire interval as in the standard interval method. This distinction proved to be decisive. Unlike the standard interval method, the use of redundancy within DTDI resulted in dramatic improvements in accuracy at moderate cost, leading to much tighter bounds than all other methods for several test cases. However, the monotonicity condition required by DTDI is more stringent when using redundancy. For Euler discretized systems, the step size must be lower than the product of two Lipschitz constants, one for the dynamics and one associated with the refinement algorithm. In the authors' experience, this has proven to be restrictive. In the best case, it requires the inconvenient calculation of Lipschitz constants. In more challenging cases, the Lipschitz constants must be reduced by scaling the dynamics or using less effective refinement algorithms (see [23]). In the worst case, the method is not applicable at all.

This paper presents a new interval bounding method that uses one-dimensional partitions to mimic DTDI without using differential inequalities theory. Specifically, the current state enclosure is partitioned along dimension i when bounding the range of the i^{th} state variable at the next time. This partitioning scheme is devised specifically so that any calculation in DTDI that would use the i^{th} upper or lower face of the current interval will be done in our method using intervals that have been partitioned along the i^{th} dimension, and are therefore relatively narrow in that dimension. The favorable performance of DTDI suggests that partitioning in this way should lead to significantly more accurate enclosures than the standard interval method, especially when using redundancy. At the same time, the proposed method is valid for any discrete-time system, does not require the computation of any Lipschitz constants, and can be used with any refinement algorithm. Although partitioning is a well known approach for improving the accuracy of interval methods, this typically requires partitioning simultaneously in all dimensions, resulting in exponential complexity. In contrast, the proposed method partitions along one dimension at a time and hence scales only linearly. Numerical results show that this method is nearly as accurate as DTDI with redundancy and only slightly more costly.

II. Problem statement

Consider the nonlinear discrete-time system

$$\mathbf{x}_{k+1} = \mathbf{f}(k, \mathbf{x}_k, \mathbf{w}_k), \tag{1}$$

where $k \in \mathbb{K} \equiv \{0, \ldots, K\}$, $\mathbf{x}_k \in \mathbb{R}^{n_x}$ is the state, $\mathbf{w}_k \in \mathbb{R}^{n_w}$ is a disturbance, and $\mathbf{f} : \mathbb{K} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_w} \to \mathbb{R}^{n_x}$. Assume the initial conditions and disturbances satisfy

$$(\mathbf{x}_0, \mathbf{w}_k) \in C_0 \times W, \quad \forall k \in \mathbb{K}, \tag{2}$$

where C_0 and W are known compact intervals. Define the shorthands $\mathbf{x}_{0:K} \equiv (\mathbf{x}_0, \ldots, \mathbf{x}_K)$, $\mathbf{w}_{0:K} \equiv (\mathbf{w}_0, \ldots, \mathbf{w}_K)$, and $W_{0:K} \equiv W \times \ldots \times W$. When necessary for clarity, the solution of (1) corresponding to a specific $(\mathbf{x}_0, \mathbf{w}_{0:K}) \in C_0 \times W_{0:K}$ will be denoted explicitly by $\mathbf{x}_k(\mathbf{x}_0, \mathbf{w}_{0:K})$.

Definition 1: The *reachable set* of (1) at time $k \in \mathbb{K}$ is

$$\mathcal{R}_k \equiv \{\mathbf{x}_k(\mathbf{x}_0, \mathbf{w}_{0:k}) : (\mathbf{x}_0, \mathbf{w}_{0:K}) \in C_0 \times W_{0:K}\}.$$

Moreover, $\mathbf{x}^L_{0:K}$ and $\mathbf{x}^U_{0:K}$ are *state bounds* for (1) if

$$\mathbf{x}^L_k \le \mathbf{x}_k(\mathbf{x}_0, \mathbf{w}_{0:K}) \le \mathbf{x}^U_k,$$

for all $(k, \mathbf{x}_0, \mathbf{w}_{0:K}) \in \mathbb{K} \times C_0 \times W_{0:K}$.

The objective of this paper is to present a new method for efficiently computing tight state bounds for (1). As mentioned in §I, some methods discussed herein make use of redundant relations satisfied by all solutions of (1) to achieve tighter bounds. We assume this information is available in the form of a general *a priori* enclosure.

Assumption 1: A set $G \subset \mathbb{R}^{n_x}$ is known such that

$$\mathbf{x}_k(\mathbf{c}_0, \mathbf{w}_{0:K}) \in G, \tag{3}$$

for all $(k, \mathbf{c}_0, \mathbf{w}_{0:K}) \in \mathbb{K} \times C_0 \times W_{0:K}$.

This assumption is not restrictive because $G = \mathbb{R}^{n_x}$ is always valid. However, nontrivial choices of G are often available in the form of conservation laws, physical bounds, etc., and can have a profound impact on the results of some bounding methods. When no such constraints are known, a useful set G can often be manufactured by lifting the system into a higher-dimensional space as described in [23], [9].

III. Background

To motivate the proposed one-dimensional partitioning method, this section briefly reviews the standard interval and DTDI methods for computing state bounds for (1).

Let $Z = [\mathbf{z}^L, \mathbf{z}^U]$ denote the n-dimensional compact interval $\{\mathbf{z} \in \mathbb{R}^n : \mathbf{z}^L \le \mathbf{z} \le \mathbf{z}^U\}$. The space of all such intervals is denoted by $\mathbb{IR}^n$. For any $\mathbf{h} : \mathbb{R}^n \to \mathbb{R}^m$, an interval function $H : \mathbb{IR}^n \to \mathbb{IR}^m$ is called an *inclusion function* for $\mathbf{h}$ if

$$\mathbf{h}(Z) \equiv \{\mathbf{h}(\mathbf{z}) : \mathbf{z} \in Z\} \subset H(Z), \quad \forall Z \in \mathbb{IR}^n.$$

Assumption 2:

1) A function $F : \mathbb{K} \times \mathbb{IR}^{n_x} \times \mathbb{IR}^{n_w} \to \mathbb{IR}^{n_x}$ is available such that $F(k, \cdot, \cdot)$ is an inclusion function for $\mathbf{f}(k, \cdot, \cdot)$ for all $k \in \mathbb{K}$.
2) An *interval refinement operator* $\mathcal{I}[G, \cdot] : \mathbb{IR}^{n_x} \to \mathbb{IR}^{n_x}$ is available for the *a priori* enclosure G satisfying

$$(Z \cap G) \subset \mathcal{I}[G, Z], \quad \forall Z \in \mathbb{IR}^{n_x}. \tag{4}$$

An inclusion function satisfying Assumption 2.1 can be readily computed using interval arithmetic [25]. Moreover, several existing refinement algorithms can be used to satisfy Assumption 2.2 provided that G is defined by a system of equality and inequality constraints [9], [23]. In brief, these algorithms attempt to compute a subinterval of Z by eliminating regions in Z that violate one or more of these constraints using iterative methods similar to interval Newton methods. In the trivial case where $G = \mathbb{R}^{n_x}$, the use of this

operator in the methods below should be interpreted with the trivial definition $\mathcal{I}[G,Z] = Z$.

Under these assumptions, the standard interval method (with redundancy) computes state bounds for (1) as the solutions of the following difference equations for all $i \in \{1,\ldots,n_x\}$, where $X_k \equiv [\mathbf{x}_k^L, \mathbf{x}_k^U]$, $F_i = [f_i^L, f_i^U]$ is the i^{th} component of F, and $C_{0,i}$ is the i^{th} component of C_0:

$$\begin{aligned} x_{k+1,i}^L &= f_i^L\left(k, \mathcal{I}[G,X_k], W\right), \\ x_{k+1,i}^U &= f_i^U\left(k, \mathcal{I}[G,X_k], W\right), \\ [x_{0,i}^L, x_{0,i}^U] &= C_{0,i}. \end{aligned} \tag{5}$$

Definition 2: For every $i \in \{1,\ldots,n_x\}$, define the *face selection operators* $\beta_i^L, \beta_i^U : \mathbb{IR}^{n_x} \to \mathbb{IR}^{n_x}$ by

$$\begin{aligned} \beta_i^L\left([\mathbf{z}^L, \mathbf{z}^U]\right) &\equiv \left\{\mathbf{z} \in [\mathbf{z}^L, \mathbf{z}^U] : z_i = z_i^L\right\}, \\ \beta_i^U\left([\mathbf{z}^L, \mathbf{z}^U]\right) &\equiv \left\{\mathbf{z} \in [\mathbf{z}^L, \mathbf{z}^U] : z_i = z_i^U\right\}. \end{aligned} \tag{6}$$

The DTDI method from [21] (with redundancy) computes state bounds for (1) as the solutions of the following difference equations for all $i \in \{1,\ldots,n_x\}$:

$$\begin{aligned} x_{k+1,i}^L &= f_i^L\left(k, \mathcal{I}\left[G, \beta_i^L(X_k)\right], W\right), \\ x_{k+1,i}^U &= f_i^U\left(k, \mathcal{I}\left[G, \beta_i^U(X_k)\right], W\right), \\ [x_{0,i}^L, x_{0,i}^U] &= C_{0,i}. \end{aligned} \tag{7}$$

This differs from (5) in that the refinement operator $\mathcal{I}[G,\cdot]$ is applied to the faces $\beta_i^{L/U}(X_k)$ rather than to X_k, and f_i is only bounded over the refinements of these faces. Even without refinement, this can lead to much tighter bounds than (5) [21]. Moreover, it has been shown that using $\mathcal{I}[G,\cdot]$ is much more effective in DTDI than in the standard interval method [21]. This is explained in Figure 1, where the blue line shows a redundant linear relationship between x_1 and x_2 assumed to hold for all solutions (such affine invariants are common in chemical reaction network models; see Example 1 in §V). Consider the computation of $x_{1,k+1}^L$ from a current interval X_k using both (5) and (7). The top left pane of Figure 1 shows how X_k (black lines) might be refined using the invariant to yield $\mathcal{I}[G,X_k]$ (gray shaded region). Although $\mathcal{I}[G,X_k]$ is tighter than X_k, the improvement is minor. Using (5), $x_{1,k+1}^L$ is computed as a lower bound on f_1 over this entire region. In contrast, using (7), $x_{1,k+1}^L$ is computed as a lower bound on f_1 over $\mathcal{I}[G,\beta_1^L(X_k)]$. This is illustrated in the top right pane of Figure 1, where $\beta_1^L(X_k)$ is the line segment connecting the two blue circles and $\mathcal{I}[G,\beta_1^L(X_k)]$ is a singleton represented by the gray circle. Since this set is much smaller than $\mathcal{I}[G,X_k]$, DTDI is likely to produce a much tighter lower bound $x_{1,k+1}^L$. Although refinement is not always so effective for DTDI, especially in higher dimensions, it is typically much more effective for DTDI than for the standard interval method.

Unfortunately, DTDI is not valid for arbitrary systems and arbitrary refinement operators, while the standard interval method is. Specifically, DTDI requires that $\mathcal{I}[G,\cdot]$ is locally Lipschitz continuous on $\mathbb{IR}^{n_x}$ (w.r.t. the Hausdorff metric) and that $\mathbf{f}$ satisfies a certain monotonicity property. If (1) is derived by forward Euler discretization of a continuous-time

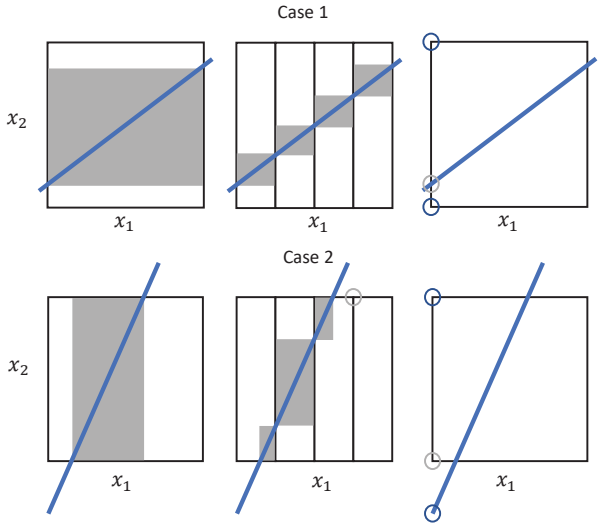

Fig. 1. Schematic of interval refinements of X_k done during the computation of $x_{1,k+1}^L$ using the standard interval method (5) (left), DTDI (7) (right), and the proposed one-dimensional partitioning method (9) with $n_p = 4$ (middle) in two different cases (top and bottom rows). The blue line represents G, the black box is X_k, and the gray regions are the results of refinement.

system, then this monotonicity requirement holds provided that the step size h satisfies $h \leq \frac{1}{MM_{\mathcal{I}}}$, where $M_{\mathcal{I}}$ is a Lipschitz constant for $\mathcal{I}[G,\cdot]$ and M is a Lipschitz constant for the right-hand side function of the continuous-time dynamics.

For Euler systems, this step size limitation is minor when redundancy is not used (i.e., $M_{\mathcal{I}} = 1$). The constant M is easily computed and the requirement $h \leq \frac{1}{M}$ is no more restrictive than well known step size limitations for maintaining numerical stability and preserving physical properties of the system [26]. However, when using redundancy, the step size limitation can be problematic. The constant $M_{\mathcal{I}}$ is difficult to compute and often large [21]. This forces the use of unnecessarily small step sizes or less effective refinements with lower $M_{\mathcal{I}}$. For non-Euler systems, the required monotonicity property is equally difficult to verify and, without a controllable step size, there is no recourse if it fails. Thus, although DTDI produces much tighter bounds than the standard interval method when it applies, it is often inconvenient and sometimes inapplicable.

IV. A One-Dimensional Partitioning Method

This section introduces a new reachable set bounding algorithm designed to capture the key advantages of DTDI while avoiding the limitations discussed in §III. A close look at the theory in support of DTDI [21] shows that the undesirable Lipschitz conditions, monotonicity requirements, and step size limitations are all required to justify the use of the face selection operators $\beta_i^{L/U}$ in (7). To approximate (and hence eliminate) these operators, we define the following one-dimensional partitioning operators.

Definition 3: Choose any partition size $n_p \in \mathbb{N}_+$. For every dimension $i \in \{1,\ldots,n_x\}$ and partition element index $j \in \{1,\ldots,n_p\}$, define the *partition element selection operator*

$\phi_i^{n_p,j} : \mathbb{IR}^{n_x} \to \mathbb{IR}^{n_x}$ by $\phi_i^{n_p,j}(X) \equiv \left[\mathbf{r}^L, \mathbf{r}^U\right]$, where

$$\left[r_q^L, r_q^U\right] \equiv \begin{cases} \left[x_q^L, x_q^U\right], & q \neq i \\ \left[x_q^L + \frac{x_q^U - x_q^L}{n_p}(j-1), x_q^L + \frac{x_q^U - x_q^L}{n_p} j\right], & q = i \end{cases}.$$

In words, $\phi_i^{n_p,j}(X)$ is the subinterval of X corresponding to the j^{th} element of a uniform partition of X along its i^{th} dimension. Evidently, for every $i \in \{1,\ldots,n_x\}$ we have

$$\bigcup_{j=1}^{n_p} \phi_i^{n_p,j}(X) = X. \tag{8}$$

We propose to compute state bounds for (1) as the solutions of the following difference equations for all $i \in \{1,\ldots,n_x\}$:

$$x_{k+1,i}^L = \min_{j\in\{1,\ldots,n_p\}} \left\{ f_i^L\left(k, \mathcal{I}\left[G, \phi_i^{n_p,j}(X_k)\right], W\right)\right\}, \tag{9}$$
$$x_{k+1,i}^U = \max_{j\in\{1,\ldots,n_p\}} \left\{ f_i^U\left(k, \mathcal{I}\left[G, \phi_i^{n_p,j}(X_k)\right], W\right)\right\},$$
$$\left[x_{0,i}^L, x_{0,i}^U\right] = C_{0,i}.$$

In words, this method computes the i^{th} component of X_{k+1} by partitioning X_k into n_p pieces along its i^{th} dimension, refining each element of this partition independently, bounding f_i over each refined element, and finally selecting the minimum and maximum bounds on f_i. In what follows, we first prove that this provides valid state bounds for (1). Notably, this does not require (1) to satisfy any monotonicity properties and does not require $\mathbf{f}$ or $\mathcal{I}[G,\cdot]$ to be Lipschitz. Subsequently, we discuss the conceptual rationale for (9) and its relation to DTDI and existing partitioning approaches.

Theorem 1: For any partition size $n_p \in \mathbb{N}_+$, the solutions of (9) are state bounds for (1).

Proof: Choose any n_p and let $\mathbf{x}_{0:K}^L$ and $\mathbf{x}_{0:K}^U$ be the solutions of (9). Moreover, choose any $\mathbf{x}_0 \in C_0$ and $\mathbf{w}_{0:K} \in W_{0:K}$ and let $\mathbf{x}_{0:K}$ be the corresponding solution of (1). We must prove that $\mathbf{x}_k \in X_k \equiv \left[\mathbf{x}_k^L, \mathbf{x}_k^U\right]$ for $\forall k \in \mathbb{K}$.

Since $\mathbf{x}_0 \in C_0$, we have $\mathbf{x}_k \in X_k$ for $k = 0$. To set up a proof by induction, assume that $\mathbf{x}_k \in X_k$ for some arbitrary $0 \leq k < K$. Choose any $i \in \{1,\ldots,n_x\}$. By (8), there must exist $j \in \{1,\ldots,n_p\}$ such that $\mathbf{x}_k \in \phi_i^{n_p,j}(X_k)$. Moreover, $\mathbf{x}_k \in G$ by Assumption 1. Thus, (4) in Assumption 2.2 ensures that $\mathbf{x}_k \in \mathcal{I}\left[G, \phi_i^{n_p,j}(X_k)\right]$. Since $F(k,\cdot,\cdot)$ is an inclusion function for $\mathbf{f}(k,\cdot,\cdot)$ by Assumption 2.1, this implies that

$$x_{k+1,i} = f_i(k, \mathbf{x}_k, \mathbf{w}_k), \tag{10}$$
$$\in F_i\left(k, \mathcal{I}\left[G, \phi_i^{n_p,j}(X_k)\right], W\right), \tag{11}$$
$$\subset X_{k+1,i}, \tag{12}$$

where the last inclusion follows from (9). Since i is arbitrary, $\mathbf{x}_{k+1} \in X_k$. The result follows by induction over k. ∎

It is well known that partitioning can significantly improve the accuracy of interval methods. However, this typically requires partitioning all dimensions simultaneously, which results in exponential complexity. In contrast, computing a single time step of (9) only requires partitioning X_k in one dimension at a time, and hence scales linearly. Specifically, if complexity is expressed as the number of times a method evaluates $F(k,\mathcal{I}[G,\cdot],W)$ in a single time step, then the complexity of (9) is $n_p n_x$ compared to 1 for the standard interval method, $2n_x$ for DTDI, and $(n_p)^{n_x}$ if X_k were partitioned along all dimensions simultaneously. Thus, (9) is much more efficient than standard partitioning approaches and only a factor of $n_p/2$ more complex than DTDI.

The expectation that accurate results will be obtained by this partitioning scheme is justified by comparison to DTDI. Consider again the computation of $x_{1,k+1}^L$ from X_k. In this case, X_k is partitioned only along the x_1 dimension as illustrated in top middle pane of Figure 1. Notably, the partition element $\phi_1^{n_p,j}(X_k)$ with $j = 1$ approximates the set $\beta_1^L(X_k)$ used in (7). Moreover, if the system satisfies the monotonicity condition required by DTDI, then the minimum over j in (9) will be attained with $j = 1$. Therefore, (9) closely approximates DTDI when DTDI applies. Thus, even in the absence of refinement, this partitioning scheme should lead to significant improvements over the standard interval method. However, like DTDI, the impact of one-dimensional partitioning is expected to be even more pronounced when combined with refinement. In Figure 1, refining each partition element results in the four gray intervals, and $x_{1,k+1}^L$ is computed as a lower bound on f_1 over these regions. Clearly, this is likely to lead to a tighter bound than that computed by the standard interval method using the larger gray region illustrated in the top left pane. While the geometry of G may not always be so favorable, especially in higher dimensions, experience with DTDI suggests that partitioning even in just one dimension will lead to much better utilization of G.

The preceding discussion suggests that the one-dimensional partitioning method should approach the accuracy of DTDI as n_p increases, but can never be more accurate. However, this is not precisely true, as can be seen in the results in §V. To explain, consider the computation of $x_{1,k+1}^L$ from X_k in the case illustrated in the bottom row of Figure 1. The right-most pane shows that $G \cap \beta_1^L(X_k) = \emptyset$. Since DTDI requires $\mathcal{I}[G,\cdot]$ to be Lipschitz continuous on all of $\mathbb{IR}^{n_x}$, it cannot return the empty set even though that would satisfy (4). The algorithm for $\mathcal{I}[G,\cdot]$ in [23] returns the singleton containing the gray circle in Figure 1. In contrast, for the partitioning method in the middle pane, the intersection between G and the first partition element from the left is non-empty and is refined to the indicated gray region. Notably, this region does not contain the gray circle in DTDI schematic, nor do the refined regions corresponding to any other partition element. Due to this technicality, it is possible that the lower bound computed by DTDI may be worse than that from the partitioning method. However, this is expected to happen rarely.

Finally, note that the partitioning method does not require $\mathcal{I}[G,\cdot]$ to be Lipschitz continuous, and therefore it could use a refinement algorithm that returns the empty set when permitted by (4). For example, the empty set could be returned for the fourth partition element in Figure 1.

V. Numerical Results

This section compares the one-dimensional partitioning method (9) with the standard interval method (5) and the DTDI method (7), with and without the use of an *a priori* enclosure G. For the methods without G, the refinement operator in (5), (7), and (9) is the trivial refinement $\mathcal{I}[G,Z] = Z$. For the methods with G, $\mathcal{I}[G,Z]$ is defined by Algorithm 1 in [23]. Unless stated otherwise, $n_p = 10$ is used for partitioning. DTDI has been compared with other state-of-the-art methods in [23] using the same case study. Therefore, we do not reproduce these additional comparisons here. We report CPU times from MATLAB R2019a on a ThinkPad X390 with an i7-8565U CPU @1.80GHz and 16.0GB RAM.

Example 1: Consider the enzymatic reaction network from [27], where the concentrations of six species x_1–x_6 (M) evolve according to the following difference equations:

$$\begin{aligned}
x_{1,k+1} &= x_{1,k} + h\left(-k_{1,k}x_{1,k}x_{2,k} + k_{2,k}x_{3,k} + k_{6,k}x_{6,k}\right) \\
x_{2,k+1} &= x_{2,k} + h\left(-k_{1,k}x_{1,k}x_{2,k} + k_{2,k}x_{3,k} + k_{3,k}x_{3,k}\right) \\
x_{3,k+1} &= x_{3,k} + h\left(k_{1,k}x_{1,k}x_{2,k} - k_{2,k}x_{3,k} - k_{3,k}x_{3,k}\right) \\
x_{4,k+1} &= x_{4,k} + h\left(k_{3,k}x_{3,k} - k_{4,k}x_{4,k}x_{5,k} + k_{5,k}x_{6,k}\right) \\
x_{5,k+1} &= x_{5,k} + h\left(-k_{4,k}x_{4,k}x_{5,k} + k_{5,k}x_{6,k} + k_{6,k}x_{6,k}\right) \\
x_{6,k+1} &= x_{6,k} + h\left(k_{4,k}x_{4,k}x_{5,k} - k_{5,k}x_{6,k} - k_{6,k}x_{6,k}\right)
\end{aligned} \tag{13}$$

The initial concentrations are $\mathbf{c}_0 = (34,20,0,0,16,0)$ and the parameters $\mathbf{k} = (k_1,\ldots,k_6)$ are disturbances bounded in $[\hat{\mathbf{k}}, 10\hat{\mathbf{k}}]$, where $\hat{\mathbf{k}} = (0.1, 0.033, 16, 5, 0.5, 0.3)$.

All solutions of this system are known to be nonnegative and satisfy three affine reaction invariants [23]. Based on these observations, a valid *a priori* enclosure can be derived as $G \equiv \{\mathbf{z} \in X_{\text{nat}} : \mathbf{M}\mathbf{z} = \mathbf{b}\}$, where

$$\mathbf{M} = \begin{bmatrix} 0 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 \\ 1 & -1 & 0 & 1 & -1 & 0 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} -20 \\ -16 \\ -2 \end{bmatrix}, \tag{14}$$

and $X_{\text{nat}} = [0,34] \times [0,20] \times [0,20] \times [0,34] \times [0,16] \times [0,16]$. Using this G, it was shown in [23] that DTDI is valid with $h = 9 \times 10^{-5}$, which we also use here for comparison.

Figure 2 shows the state bounds for x_3 and x_5 computed by all methods, as well as the total enclosure volume metric $(\text{vol}(X_k))^{1/n_x}$. When the *a priori* enclosure G is not used, the standard interval method rapidly diverges. DTDI gives significantly better results, but the bounds are still too weak to be useful. Nonetheless, the results show that the partitioning method is able to closely match the improvements made by DTDI. When G is used, all methods improve significantly. However, whereas the standard interval method still provides weak bounds, both DTDI and the partitioning method become fairly accurate. Notably, the partitioning method achieves nearly the same accuracy as DTDI, which has previously been shown to be significantly more accurate than other state-of-the-art methods for this example [23]. Although we expect partitioning to be less accurate than DTDI in general, it is more accurate for x_3 in Figure 2 for the reason explained at the end of §IV.

A key conclusion from [21] is that combining DTDI with redundancy leads to improvements far beyond those achieved by applying DTDI or redundancy independently. Our results show that the one-dimensional partitioning method shares this behavior. Specifically, the partitioning method enables redundancy to be used much more effectively than in the standard interval method. In fact, the improvements caused by using G in the standard interval method arise entirely from intersection with X_{nat}, while refinements based on the affine invariants $\mathbf{M}\mathbf{z} = \mathbf{b}$ are wholly ineffective. In contrast, these invariants are highly effective when combined with one-dimensional partitioning, as illustrated in Figure 1.

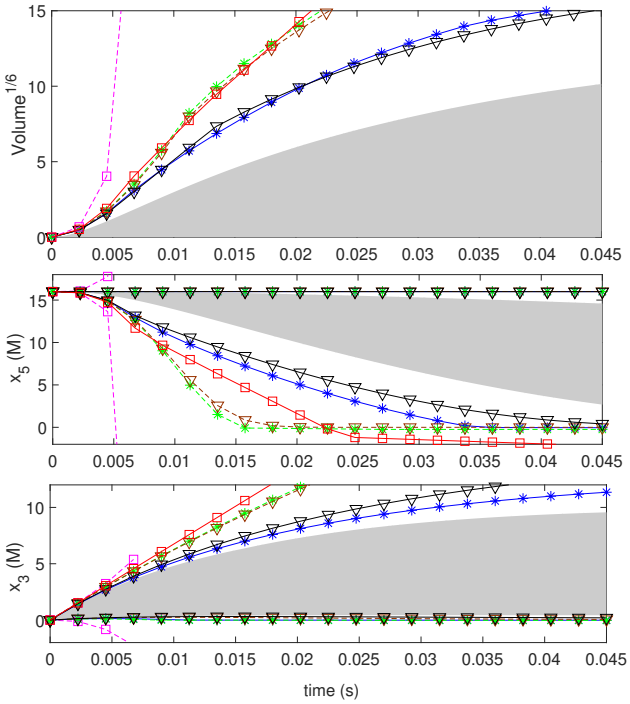

Fig. 2. Enclosure volume and selected state bounds for Example 1 computed using the standard interval method (□), DTDI (▽), and the one-dimensional partitioning method (⋆), both with (solid lines, □, ▽, ⋆) and without (dashed lines, □, ▽, ⋆) using G. The gray shaded regions show the true volume and reachable sets estimated by sampling.

Compared to DTDI, one-dimensional partitioning has several advantages. First, it does not require the system to satisfy any monotonicity properties. Consequently, it is valid for any step size, whereas DTDI is not known to be valid for this example with h greater than 9×10^{-5}. Thus, one-dimensional partitioning may be implementable with fewer time steps, leading to higher efficiency despite having a higher per step complexity. Second, there is no need to compute Lipschitz constants for the dynamics or the refinement operator, and in fact no requirement that they are Lipschitz at all. This is more convenient, but also enables the use of more effective refinement operators. Since large Lipschitz constants were a significant limiting factor for several refinement methods in [21], this is expected to enable more accurate bounds than DTDI in many cases. However, for this example, equipping the one-dimensional partitioning method with the more accurate refinement method in Algorithm 1 of [9] did not lead to a significant improvement. Third, it is possible to achieve

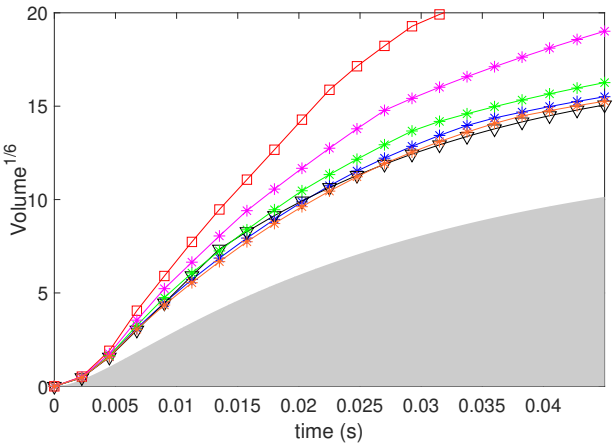

Fig. 3. Enclosure volume for Example 1 computed using the standard interval method (□), the DI method (▽), and the one-dimensional partitioning method with n_p equal to 2, 5, 10 and 15 (⋆, ⋆, ⋆, ⋆). All methods use G. The gray shaded region show the true volume estimated by sampling.

tighter bounds than DTDI even when the same refinement operator is used, as can be seen in Figure 2. However, this is due to technical details of how DTDI handles the case where $G \cap \beta_i^{L/U}(Z) = \emptyset$ and is not expected to happen often.

Figure 3 shows how the number of partition elements n_p influences the volume of the resulting bounds. The volumes obtained by the standard interval and DTDI methods are also plotted for reference. The *a priori* enclosure G was used for all methods. The standard interval method is equivalent to the partitioning method with $n_p = 1$. By increasing n_p to 2, the bounds are already tightened significantly. With $n_p = 10$, the bounds are nearly the same as those from DTDI. Increasing n_p further does not result in significant improvements.

For the methods without G, the standard interval method has the lowest cost, requiring roughly 1.5×10^{-4} s per step. The partitioning method with $n_p = 10$ has the highest cost, requiring 3×10^{-3} s per step. This is roughly 6 ($\approx \frac{n_p}{2}$) times higher than the cost of DTDI. When using G, the computational burden of all three methods doubles, but their relative costs remain the same.

References

[1] G. A. A. Gonçalves and M. Guay. Robust discrete-time set-based adaptive predictive control for nonlinear systems. *J. Process Control*, 39:111–122, 2016.

[2] S. Yu, C. Maier, H. Chen, and F. Allgöwer. Tube MPC scheme based on robust control invariant set with application to Lipschitz nonlinear systems. *Syst. Control. Lett.*, 62(2):194–200, 2013.

[3] F. Xu, V. Puig, C. Ocampo-Martinez, S. Olaru, and S.-I. Niculescu. Robust MPC for actuator–fault tolerance using set–based passive fault detection and active fault isolation. *Int. J. Appl. Math. Comput. Sci.*, 27(1):43–61, 2017.

[4] D. Silvestre, P. Rosa, J. P. Hespanha, and C. Silvestre. Set-based fault detection and isolation for detectable linear parameter-varying systems. *Int. J. Robust Nonlinear Control*, 27(18):4381–4397, 2017.

[5] F. Parise, M. E. Valcher, and J. Lygeros. Reachability analysis for switched affine systems and its application to controlled stochastic biochemical reaction networks. In *Proc. 55th IEEE Conference on Decision and Control*, pages 746–751, Dec 2016.

[6] H. N. V. Pico and D. C. Aliprantis. Voltage ride-through capability verification of DFIG-based wind turbines using reachability analysis. *IEEE Trans. Energy Convers.*, 31(4):1387–1398, 2016.

[7] M. Althoff, D. Althoff, D. Wollherr, and M. Buss. Safety verification of autonomous vehicles for coordinated evasive maneuvers. In *IEEE Intelligent Vehicles Symposium*, pages 1078–1083, June 2010.

[8] P. Falcone, M. Ali, and J. Sjoberg. Predictive threat assessment via reachability analysis and set invariance theory. *IEEE Trans. Intell. Transp. Syst.*, 12(4):1352–1361, 2011.

[9] K. Shen and J. K. Scott. Rapid and accurate reachability analysis for nonlinear dynamic systems by exploiting model redundancy. *Comput. Chem. Eng.*, 106:596–608, 2017.

[10] M. E. Villanueva, B. Houska, and B. Chachuat. Unified framework for the propagation of continuous-time enclosures for parametric nonlinear ODEs. *J. Glob. Optim.*, 62(3):575–613, 2015.

[11] T. Alamo, J. M. Bravo, M. J. Redondo, and E. F. Camacho. A set-membership state estimation algorithm based on DC programming. *Automatica*, 44(1):216–224, 2008.

[12] V. Magron, P.-L.Garoche, D. Henrion, and X. Thirioux. Semidefinite approximations of reachable sets for discrete-time polynomial systems. *SIAM J. Control Optim.*, 57(4):2799–2820, 2019.

[13] M. Kieffer, L. Jaulin, and É. Walter. Guaranteed recursive non-linear state bounding using interval analysis. *Int. J. Adapt. Contr. Signal Process*, 16(3):193–218, 2002.

[14] M. Kieffer and E. Walter. Interval analysis for guaranteed non-linear parameter and state estimation. *Math. Comp. Model Dyn.*, 11(2):171–181, 2005.

[15] J. S. Shamma and K.-Y. Tu. Approximate set-valued observers for nonlinear systems. *IEEE Trans. Automat. Contr.*, 42(5):648–658, 1997.

[16] C. E. Valero, M. E. Villanueva, B. Houska, and R. Paulen. Set-based state estimation: a polytopic approach. In *Proc. 21st IFAC World Congress*, page (TBA), 2020.

[17] T. Alamo, J.M. Bravo, and E.F. Camacho. Guaranteed state estimation by zonotopes. *Automatica*, 41(6):1035–1043, 2005.

[18] C. Combastel. A state bounding observer for uncertain non-linear continuous-time systems based on zonotopes. In *Proc. 44th IEEE Conference on Decision and Control*, pages 7228–7234, Dec 2005.

[19] J. M. Bravo, T. Alamo, and E. F. Camacho. Bounded error identification of systems with time-varying parameters. *IEEE Trans. Automat. Contr.*, 51(7):1144–1150, 2006.

[20] B. S. Rego, G. V. Raffo, J. K. Scott, and D. M. Raimondo. Guaranteed methods based on constrained zonotopes for set-valued state estimation of nonlinear discrete-time systems. *Automatica*, 111:1–14, 2020.

[21] X. Yang and J. K. Scott. Accurate uncertainty propagation for discrete-time nonlinear systems using differential inequalities with model redundancy. *IEEE Trans. Automat. Contr.*, 65(12):5043–5057, 2020.

[22] T. Dreossi, T. Dang, and C. Piazza. Parallelotope bundles for polynomial reachability. In *Proc. 19th International Conference on Hybrid Systems: Computation and Control*, page 297–306, 2016.

[23] X. Yang and J. K. Scott. Efficient reachability bounds for discrete-time nonlinear systems by extending the continuous-time theory of differential inequalities. In *Proc. American Control Conference*, pages 6242–6247, June 2018.

[24] K. Shen and J. K. Scott. Exploiting nonlinear invariants and path constraints to achieve tighter reachable set enclosures using differential inequalities. *Math. Control. Signals, Syst.*, 32(1):101–127, 2020.

[25] R. E. Moore, R. B. Kearfott, and M. J. Cloud. *Introduction to interval analysis*. SIAM, 2009.

[26] Z. Horváth. On the positivity step size threshold of Runge–Kutta methods. *Appl. Numer. Math*, 53(2):341–356, 2005.

[27] J. K. Scott and P. I. Barton. Bounds on the reachable sets of nonlinear control systems. *Automatica*, 49(1):93–100, 2013.