Computing Robust Forward Invariant Sets of Multidimensional Nonlinear Systems via Geometric Deformation of Polytopes

Taha Ameen, Shayok Mukhopadhyay, and Nasser Qaddoumi

Abstract—This article develops an algorithm to compute the sequences of polytopic robust forward invariant sets (RFIS) that vary in size for a nonlinear dynamical system. This is done through a novel approach that geometrically deforms a polytope into an invariant set using a sequence of homeomorphisms, based on an invariance condition that only needs to be satisfied at a finite set of test points. A fast computational test is also developed to check if a given polytope is an RFIS. Our approach is applicable to arbitrary Lipschitz continuous nonlinear systems in the presence of bounded additive disturbances, and its versatility is presented through simulation results on a variety of nonlinear dynamical systems in two and three dimensions.

Index Terms—Computational topology, invariant sets, nonlinear dynamical systems, robust forward invariance.

I. INTRODUCTION

A robust forward invariant set (RFIS) for a dynamical system is a subset of the state space from which trajectories never escape in spite of disturbances, as time runs forward. Invariant sets are useful in performance and safety analysis. For example, finding an invariant set that is disjoint from a set of unsafe states can ensure that system trajectories never reach this undesirable subset. Further, an RFIS of the appropriate size can guarantee safety and robustness without imposing overconservative or overaggressive bounds on system trajectories. Thus motivated, this work focuses on computing RFISs of different size in an \( n \)-dimensional setting.

Methods to compute invariant sets include Laserre’s hierarchy for polynomial optimization [1], [2], [3] and Koopman-operator based approaches [4]. Recently, polytopic invariant sets have gained interest, since they are nonconservative and guarantee stability without the loss of performance [5]. Approaches to calculate invariant polytopes for linear systems include constraint tightening [6], finite-time Aumann integrals [7], and semidefinite programs [8]. Among linear systems, the class of discrete time systems has received considerable attention [9]. Approaches include geometric construction using zonotropic bounds [10], linear matrix inequality (LMI)-based algorithms [5], [11], [12], and linear programming [13]. Recently, viability theoretic approaches to model state disturbances have also gained attraction [14]. For example, the authors in [15] modeled discrete time systems with noise as convex difference inclusions and derive invariance conditions.

In contrast, the literature on continuous-time nonlinear systems is limited. For instance, invariant sets for piecewise affine systems were studied in [16], and polynomial systems were studied in [17] using sum of squares relaxations. Similarly, the authors in [18] used Lyapunov functions to characterize invariant sets of polynomial systems with bounded disturbances. Another approach to compute RFIS for nonlinear systems is using the Hamilton Jacobian Reachability (HJR) condition [19]. Of particular relevance to our work are [20] and [21], where a test for invariance is developed based on the subtangentiality condition. However, the infeasibility of testing for the condition at infinitely many points limits the scope of these works to polynomial systems.

All these works use algebraic rather than geometric formulations. This is motivated by the representation of polytopes as LMIs, which provide formulations for optimization-based approaches. However, most computational approaches deal with linear and polynomial systems [22], [23]. Examples include: (i) [24] and [25] where polytopic invariant sets are constructed between ellipsoidal sets, (ii) [26] where symmetrical polytopic invariant sets are determined with constraints for noiseless systems, (iii) [27] where the authors derive a maximal robust invariant set for linear systems with additive disturbances. Another work with a geometric approach is [28], where the authors derive invariant sets for linear systems using an invariance condition based on the support function of a set. Geometric approaches have the advantage of not relying on algebraic properties of system dynamics. For instance, the authors in [29] and [30] constructed closed polytopes using path planning algorithms on radial graphs for general nonlinear systems in \( \mathbb{R}^n \). Another set of examples is [31] and [32], where simplex-based approaches were used for nonlinear systems in continuous time. Besides these limited works, few computation-oriented results are available for general nonlinear systems [33].

In this work, we make the following contributions.

1) We use ideas from computational topology to quantify the extent of forward invariance for a given polytope.
2) Based on this, we propose an invariance test for a polytopic set that amounts to a simple calculation at a finite set of test points on the polytope boundary.
3) We use this test to implement an algorithm to generate a sequence of RFISs that vary in size, for \( n \)-dimensional Lipschitz continuous nonlinear systems with bounded additive disturbances.

In contrast to the existing methods available in literature, our approach does not impose any restrictions on the system dynamics, besides the usual conditions for existence and uniqueness of solutions [34]. Further, no additional assumptions on the nature of disturbances are made besides their being bounded and additive. Finally, our approach is simple, and it does not involve solving optimization problems or partial differential equations specific to individual systems. Thus, our approach is valid for a large class of nonlinear systems.

Manuscript received 2 May 2022; revised 10 March 2023; accepted 17 July 2023. Date of publication 1 August 2023; date of current version 5 December 2023. Recommended by Associate Editor M. Kanat Camlibel.

Taha Ameen is with the Department of Electrical and Computer Engineering, University of Illinois Urbana-Champaign, Urbana, IL 61801 USA (e-mail: tahaa3@illinois.edu).

Shayok Mukhopadhyay is with the Electrical & Computer Engineering and Computer Science Department, University of New Haven, West Haven, CT 06516 USA (e-mail: smukhopadhyay@newhaven.edu).

Nasser Qaddoumi is with the Department of Electrical Engineering, American University of Sharjah, Sharjah 26666, UAE (e-mail: nqaddoumi@aus.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TAC.2023.3300349.

Digital Object Identifier 10.1109/TAC.2023.3300349

0018-9286 © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.

See https://www.ieee.org/publications/rights/index.html for more information.
II. MATHEMATICAL BACKGROUND

A. Computational Topology

Simplices generalize triangles to higher dimensions [35].

Definition 2.1 (Simplex): A simplex, \( \Delta \subset \mathbb{R}^n \) of dimension \( k \) (or a \( k \)-simplex), with \( 0 \leq k \leq n \) is the convex hull of a set of \( (k+1) \) vertex vectors, \( \text{Vert}(\Delta) = \{ v_0, \ldots, v_k \} \subset \mathbb{R}^n \), such that the matrix \( B_\Delta \) has linearly independent columns

\[
B_\Delta = \begin{bmatrix} v_1 - v_0, \ldots, v_k - v_0 \end{bmatrix} \subset \mathbb{R}^{n \times k}.
\]  

(1)

Since the vertices uniquely define a simplex, we identify \( \Delta \) with a matrix, \( L_\Delta \in \mathbb{R}^{(k+1) \times n} \), where row \( i \) of \( L_\Delta \) is \( v_i - v_{i-1} \).

\[
L_\Delta = \begin{bmatrix} v_0, v_1, \ldots, v_k \end{bmatrix}.
\]  

(2)

Note that an \( n \)-simplex, \( \Delta \), itself contains \( k \)-simplices where \( k < n \). We refer to these "subsimplices" as \( k \)-faces of \( \Delta \). We are interested in \((n-1)\)-simplices embedded in \( \mathbb{R}^n \), and will construct invariant sets with simplicial boundaries. The orientation of a simplex is characterized by its normal vector

\[
(N_\Delta)_i = (-1)^{n+i} \det(B_{\Delta i}),
\]  

(3)

where \((N_\Delta)_i\) is the \( i \)th component of the normal vector, and \( B_{\Delta i} \in \mathbb{R}^{(n-1) \times (n-1)} \) is the matrix obtained by deleting the \( i \)th row of \( B_\Delta \).

Further, an \( n \)-simplex, \( \Delta \subset \mathbb{R}^n \) has a volume

\[
\text{Vol}(\Delta) = \frac{1}{n!} \det(B_\Delta).
\]  

(4)

Definition 2.2 (Homogeneous Simplicial \( k \)-Complex): A homogeneous simplicial complex \( K \) is a set of \( k \)-simplices \( \{ \Delta : \Delta \subset K \} \) such that \( \Delta_1 \cap \Delta_2 \) is either the empty set or a face of both \( \Delta_1 \) and \( \Delta_2 \), whenever \( \Delta_1, \Delta_2 \in K \).

We take a simplicial complex to mean a homogeneous simplicial complex.

We denote the geometric realization of \( K \) by \( |K| \), which is the topological space obtained by gluing all the simplices together. In this work, we will deform polytopes into invariant sets through homeomorphisms induced from vertex maps. A vertex map between two simplicial complexes \( \mathcal{K} \) and \( \mathcal{L} \) is a function \( \phi : \text{Vert}(\mathcal{K}) \to \text{Vert}(\mathcal{L}) \) such that the vertices of every simplex in \( \mathcal{K} \) map to the vertices of a simplex in \( \mathcal{L} \). \( \phi \) can be extended to a unique continuous map \( g : |K| \to |\mathcal{L}| \), called the "induced map" due to \( \phi \) [35]. Further, if \( \phi \) is bijective, and \( \phi^{-1} \) is also a vertex map, then \( g \) is a homeomorphism between \(|K|\) and \(|\mathcal{L}|\). An example is in Fig. 2, where \( \phi \) maps \( v_0 = (1, -1, 1) \to (7/10, 1/2, 7/10) \) and all other vertices to themselves.

B. Robust Forward Invariant Sets

A set \( \mathcal{R} \) is a forward invariant set (FIS) if all system trajectories that begin in \( \mathcal{R} \) stay in \( \mathcal{R} \) for all future time. \( \mathcal{R} \) is a robust FIS (RFIS) if it is an FIS in the presence of disturbances.

Definition 2.3 (Robust Forward Invariant Set): Let \( S \subset \mathbb{R}^n \) represent the state space of a given dynamical system, \( \dot{x}(t) = f(x(t), \omega(t)) \), where \( \omega(t) : [0, \infty) \to \Omega \) is a bounded disturbance function that takes values in \( \Omega \). Further, let \( \mathcal{X} \) represent the set of solutions for this system. A set \( \mathcal{R} \subset S \) is said to be an RFIS if \( \forall x \in \mathcal{X}, x(t_0) \in \mathcal{R} \Rightarrow x(t) \in \mathcal{R} \forall t \geq t_0 \).

A minimal RFIS is a set \( \mathcal{R}_m \) such that no proper subset of \( \mathcal{R}_m \) is an RFIS for the system. Conversely, a maximal RFIS, \( \mathcal{R}_m \) is not a proper subset of any RFIS.

III. PROBLEM FORMULATION

Consider a nonlinear system with bounded additive disturbances modeled by a differential inclusion \( \dot{x}(t) \in F(x(t)) \), where \( F(\cdot) \) is a set-valued map. Specifically

\[
F(x(t)) = \{ f(x(t)) + \omega(t) : \omega(t) \in \Omega \forall t \in [0, \infty) \}.
\]  

(5)

Here, \( f : S \to \mathbb{R}^n \) models the dynamical system with state space \( S \subset \mathbb{R}^n \), and \( \omega : [0, \infty) \to \Omega \) is the disturbance function, where \( \Omega \subset \mathbb{R}^n \) is a bounded set. If \( \Omega = \{0\} \), the dynamics reduce to the usual \( \dot{x}(t) = f(x(t)) \). The standard conditions for the existence and uniqueness of solutions [34] are assumed to be met. Thus, \( F(\cdot) \) is assumed to be Lipschitz continuous, so that \( d_H(F(x), F(y)) \leq \ell \|x - y\| \forall x, y \in \mathcal{X} \), where \( \ell \) is the Lipschitz constant, and \( d_H(\cdot, \cdot) \) is the Hausdorff distance between sets. Starting with a polytope \( \mathcal{P} \subset S \) and a simplicial complex \( \mathcal{K} \) that triangulates \( \partial \mathcal{P} \), the objective is to find families of polytopic RFISs for the system in (5).

We begin with a simplicial complex \( \mathcal{K} \) such that \( |K| = \partial \mathcal{P} \), where \( \mathcal{P} \) is a convex polytope. We apply a sequence of homeomorphisms on \( |K| \) to geometrically deform \( \mathcal{P} \) into an RFIS. These deformations are vertex maps on \( 0 \)-faces in \( K \) and its subdivisions, subject to a Boundary Condition. We develop this condition and show that if all simplices in \( K \) satisfy it, then \(|K|\) is the boundary of an RFIS. This serves as a computational invariance test to guide the deformations. Our algorithm can be implemented knowing only the dynamics \( F \) and the coordinates of the \( 0 \)-faces in \( K \).
IV. INVARIANCE TEST

We propose an algorithm to check if a simplicial complex $K$ triangulates the boundary of an RFIS, for a system with dynamics given in (5). Let set $A \subseteq \mathbb{R}^n$ and vector $b \in \mathbb{R}^n$. Let $\langle A, b \rangle$ denote $\cap_{a \in A} (a, b)$.

Definition 4.1 (Invariance Condition): Let $K$ be a simplicial complex triangulating $\partial P$, where $P \subseteq \mathbb{R}^n$ is a polytope. A point $x \in \Delta \subseteq K$ is said to satisfy the Invariance Condition if $\langle F(x), N_\Delta \rangle \leq 0$, where $N_\Delta$ is the outward unit normal vector to $\Delta$.

Similar conditions have been used in works such as [21], [29], [30], [31], [32] to determine the invariance of sets, but the novelty of this work is in the development of an implementable invariance test and its invocation to perform geometric deformations.

Theorem 4.2: Suppose $K = T(P)$. If $x_0 \in \Delta \subseteq K$ satisfies the Invariance Condition, then no system trajectory can escape the set $P$ through $x_0$.

Proof: The proof follows from the subtangentiality condition [36]. A full proof is also in [37].

Corollary 4.2.1: If each point $x \in \Delta$ satisfies the Invariance Condition, then no system trajectory can escape $\Delta \subseteq \partial P$. We call such a $\Delta$ as an “invariant simplex.” If each $\Delta \subseteq K$ is an invariant simplex, then $P$ is an RFIS.

If $x$ is in multiple simplices of $K$, our requirement is that it must satisfy the Invariance Condition with respect to all simplices. The caveat is that it is impossible to computationally check the Invariance Condition at every point on $\partial P$. However, when $F(\cdot)$ is Lipschitz continuous, the following is true.

Theorem 4.3: Let $F$ be $\ell$-Lipschitz with respect to the Hausdorff metric. Let $x_0 \in \Delta$, where $\Delta$ is a simplex with normal vector, $N_\Delta$. If $\exists \delta > 0$ such that $F(x_0), N_\Delta \leq -\epsilon$, then $F(x, N_\Delta) \leq 0 \forall x$ such that $\|x - x_0\| \leq \frac{\delta}{\ell}$.

Proof: This follows from [31, Proposition 3].

Based on Theorems 4.2 and 4.3, we develop an algorithm to check if $\Delta$ is an invariant simplex. First, we present a method to generate a finite set of test points on $\Delta$ using only $\Vert(\cdot)\Vert$, and show how a simple calculation at these points can decide whether $\Delta$ is an invariant simplex. Let $m \in \mathbb{N}_0$. The set of test points $T_m$ is a lattice that grows exponentially finer with increasing $m$. To control the coarseness of the lattice, a parameter $\delta_m$ in $[0, 1]$ is introduced. We set $\delta_m = 2^{-m}$.

Definition 4.4 (Lattice Test Points on $\Delta$): Let $\Delta$ be a simplex with $\Vert(\cdot)\Vert = \{v_0, \ldots, v_{n-1}\}$ and let $\delta_0 = \{0, 1, \ldots\}$. For $m \in \mathbb{N}_0$, let $\delta_m = 2^{-m}$, and $g_m : \mathbb{R}^{n-1} \to \mathbb{R}^n$ be the function

$$g_m(a_1, \ldots, a_{n-1}) = \left(1 - \sum_{i=1}^{n-1} \delta_m a_i\right) v_0 + \sum_{j=1}^{n-1} (\delta_m a_j) v_j. \tag{6}$$

The set of test points on $\Delta$, denoted $T_m$ is defined as

$$T_m = \{g_m(a_1, \ldots, a_{n-1}) : a_i \in [0, \delta_m^{-1}], a_i \in \mathbb{N}_0\}. \tag{7}$$

Lemma 4.5: $T_m \subseteq \Delta$.

Proof: Let $x \in T_m$. Thus, $x = g_m(a_1, \ldots, a_{n-1})$ for some $a_i \geq 0$ such that $\sum_{i=1}^{n-1} a_i \in [0, \delta_m^{-1}]$. It follows from (6) that $x$ is a convex combination of the vertices of $\Delta$. Hence $x \in \Delta$.

It follows by construction that the number of test points on the $(n-1)$-simplex $\Delta$ in $\mathbb{R}^n$ with parameter $\delta_m$ is $l_m = (\delta_m^{n-1})$. The $l_m$ test points generated on $\Delta$, $T_m$, are stored as rows of a matrix, $T_m \in \mathbb{R}^{l_m \times n}$. Note that $T_m = T_m L_\Delta$, where $\Lambda$ is the standard simplex and $L_\Delta$ is defined in (2). Since $T_m$ can be computed a priori, the test point generation amounts to simple matrix multiplication. For example, Fig. 3 shows $T_1^3$ and $T_2^3$ for a 3-simplex $\Delta$. We now show that it suffices to check for a variant of the Invariance Condition at only the points in $T_m^\Lambda$ to test whether $\Delta$ is an invariant simplex.

Theorem 4.6: Let $\ell$ be the Lipschitz constant for the system whose dynamics are given by $F$. Let $\Delta$ be a simplex with Vert($\Delta$) = $\{v_0, \ldots, v_{n-1}\}$, $r = \max_{i,j} \|v_i - v_j\|$. For $m \in \mathbb{N}_0$, let $T_m^\Lambda$ be the set of lattice test points on $\Delta$. If

$$\langle F(x), N_\Delta \rangle \leq -r \delta_m \ell \tag{8}$$

for every $x \in T_m^\Lambda$, then $\Delta$ is an invariant simplex.

Proof: See Appendix.

A test point $x$ that satisfies (8) is said to satisfy the Boundary Condition. Algorithm 1 (BCD) uses Theorem 4.6 to test the invariance of a simplex $\Delta$. It takes as input the simplex vertices Vert($\Delta$), the dynamics $F$, Lipschitz constant $\ell$, and the lattice coarseness parameter $m$. The algorithm also has access to the matrix of test points on the standard simplex $T_m^\Lambda$, and the number of test points $l_m$. The algorithm computes the fraction of test points which violate the Boundary Condition, $N(\Delta)$.

Note that $\Delta$ is invariant if $N(\Delta) = 0$.

Lines 1 and 2 of Algorithm 1 compute the matrix $L_\Delta$, the normal vector to the simplex $N_\Delta$, and the length of the longest 1-face of the simplex, $r$. The test points on $\Delta$ are generated as rows of the matrix $T_m^\Lambda$ in line 3 by mapping lattice points on the standard simplex $\Lambda$ to $\Delta$. Since all points in $T_m^\Lambda$ share the same normal vector, the quantity $\langle F(x), N_\Delta \rangle$ can be computed for all points in $T_m^\Lambda$ through matrix-vector multiplication (Line 4). Here, $F(T_m^\Lambda)$ is the matrix obtained by applying $F(\cdot)$ to each row of $T_m^\Lambda$. Finally, the number of test points that violate the Boundary Condition is computed in Line 5 and the fraction of such points is the algorithm output.

V. RFIS COMPUTATION ALGORITHM

In this section, we first present a vertex map in Section V-A and use it successively in Section V-B to propose an RFIS computation algorithm.
Algorithm 1: BCD_Test (Boundary Condition Test).

**Inputs:** Vert\((\Delta)\), \(F\), \(\ell\), \(m\)

**Constants:** \(\delta_m, T^{\Delta}_m, l_m\)

**Outputs:** \(N(\Delta)\)

1: Compute \(L_\Delta\) using (2) and \(N_\Delta\) using (3)
2: Compute \(r = \max \|v_i - v_j\|\), where \(v_i, v_j \in \text{Vert}(\Delta)\)
3: Compute \(T^{\Delta}_m = T^{\Delta}_m L_{\Delta}\)
4: Compute \(\text{IP} = F(T^{\Delta}_m) \backslash N_\Delta\) // Vector of inner products at all test points
5: Set \(N_m = \{n : \text{IP}[n] > -\delta_m \ell\}\) // Number of elements in IP that violate Boundary Condition
6: return \(N(\Delta) = \frac{N_m}{L_m}\) // \(0 \leq N(\Delta) \leq 1\)

A. Proposed Vertex Map

Let \(R_\alpha\) and \(R_m\) denote a minimal and maximal RFIS for a system \(F\). Assume that \(R_\alpha\) has a nonempty interior and that \(R_m\) is bounded. We begin with a convex polytope \(P\), such that \(R_\alpha \subset P \subset R_m\), and triangulate \(P\) by a simplicial complex \(K\), such that \(\text{Vert}(K) = \{v_0, \ldots, v_N\}\).

Let \(c \in \text{int}(R_\alpha)\). Although \(R_\alpha\) is unknown, \(c\) may still be chosen. For instance, \(c\) can be an equilibrium point or a point in the interior of a known limit cycle. We consider deformations where all vertices but one are mapped to themselves. Specifically, \(\phi^c_j: \text{Vert}(K) \rightarrow \text{Vert}(\ell)\) is defined as

\[
\phi^c_j(v_i) = \begin{cases} v_i, & i \neq j \\ (1 - \alpha)c + \alpha v_i, & i = j \end{cases}
\]

where \(\alpha > 0\) is a growth/decay parameter. Hence, the vertex map \(\phi^c_j\) only perturbs the vertex \(v_i\) by moving it along a ray \(B_i(c)\) emanating from \(c\) and containing \(v_i\)

\[B_i(c) = \{c + \lambda(v_i - c) : \lambda > 0\}.\]  

An example is shown in Fig. 4: The vertex \(v_0\) is mapped to \(\phi^c_0(v_0)\), which is constrained on the ray \(B_0(c)\). The \(\ell\) and \(K\) only disagree on the points in the closed star of \(v_0\), i.e., \(g_j(x) = x\) if \(x \notin \text{St}_K(v_0)\), where \(g_j: |K| \rightarrow |\ell|\) is the induced map due to \(\phi^c_j\), as defined in Section II-A. We now show that the two polytopes are homeomorphic.

**Theorem 5.1:** Let \(\phi^c_0\) be the induced map due to \(\phi^c_j\) as defined in (9). If \(\alpha > 0\), then \(\phi^c_j\) is a homeomorphism.

**Proof:** A vertex map induces a homeomorphism if and only if it has an inverse that is also a vertex map [35]. Since \(P\) is a convex polytope, it follows that \(B_i(c) \cap B_k(c)\) is the empty set whenever \(i \neq j\). Since \(\phi^c_j(v_i) \in B_i(c)\) for any \(i\), no two vertices in \(K\) are mapped to the same vertex in \(\ell\). Therefore

\[
(\phi^c_j)^{-1}(v_i) = \begin{cases} v_i, & i \neq j \\ (1 - \alpha^{-1})c + \alpha^{-1}v_i, & i = j \end{cases}
\]

is the inverse map of \(\phi^c_j\). Thus, \(g_j\) is a homeomorphism.

**Corollary 5.1.1:** Let \(k \in \mathbb{N}\) and \(\phi^c: \text{Vert}(K) \rightarrow \text{Vert}(\ell)\) be

\[
\phi^c(v_i) = \phi^c_{j_k} \circ \cdots \circ \phi^c_{j_1}(v_i)
\]

where \(\phi^c_{j_i}\) are vertex maps as defined in (9). Then, the geometric realizations \(|K|\) and \(|\ell|\) are homeomorphic.

**Proof:** \(\phi^c(v_i) \in B_i(c)\) for all \(i\), so no two vertices are mapped to the same point. Since composition of homeomorphisms is also a homeomorphism, \(|K|\) and \(|\ell|\) are homeomorphic.

The vertex map (12) deforms the polytope into a new nonintersecting polytope. In Algorithm 2, we use the fraction of test points on \(\Delta\) that violate the Boundary Condition as a measure of how “close” \(\Delta\) is to being invariant, and propose that a vertex perturbation \(\phi^\prime: \text{Vert}(K) \rightarrow \text{Vert}(\ell)\) be performed if and only if

\[
\left(\sum_{\Delta \in \ell} N(\Delta) < \sum_{\Delta \in K} N(\Delta)\right) \text{ or } \sum_{\Delta \in \ell} N(\Delta) = 0.\]  

B. Polytopic RFIS Computation Algorithm

Let \(K\) triangulate a convex polytope \(P\). To compute RFISs of different sizes for the system (5), we perturb vertices in \(K\) through vertex maps that satisfy (13) until the Boundary Condition is satisfied for all points in \(T^{\Delta}_m\). Once a stage is reached where no vertex can be perturbed further (all perturbations would violate (13)), the simplices in the simplicial complex \(\ell\) are subdivided. Since \(\text{St}_K(v) \subseteq \text{St}_\ell(v)\), the deformations are finer, allowing the new polytope to better approximate the shape of the RFIS. Our algorithm is presented as Algorithm 2, where the polytope is deformed by perturbing a single vertex and checking whether or not to discard the perturbation immediately after.

Algorithm 2 takes as input the simplicial complex \(K\), the homeomorphism parameters \(\alpha, \epsilon\) and the maximum number of permissible simplicial subdivisions \(t_{\text{max}}\). Since each iteration involves simplic subdivision, \(t_{\text{max}}\) bounds the maximum number of simplices in the simplicial complex that triangulates the boundary of the final polytopic set. The algorithm returns as output an ordered list of sets \(H\), where the \(j\)th entry \(H[j]\) is the simplicial complex that triangulates the boundary of the deformed polytope after \(j\) iterations of the algorithm. We associate with each vertex \(v\) a variable \(\text{ind}[j] \in \{0, 1\}\) which is set to 0 if the perturbation \(\phi_j(v)\) in the previous iteration violated (13), and 1 otherwise. \(\text{ind}[j]\) is initialized to 1 for all \(j\) in line 3. Thus, each vertex is perturbed at least once (line 6), but the perturbation is kept only if (13) is satisfied. Since \(\phi^c_j(v)\) only affects points in \(\text{St}_K(v)\), it suffices to check for (13) over only the simplices in \(\text{St}_K(v)\), as in lines 7–8. According to Remark 5.2.1, the parameters can be initialized as follows. \(t_{\text{max}}\) is chosen based on the complexity of the desired polytope to be found, since larger values of \(t_{\text{max}}\) allow finer perturbations and consequently polytopes with a large number of faces. We find that in practice, \(t_{\text{max}} \approx 7\) is sufficient for obtaining good results. Similarly, \(\alpha \approx 0.99\) provides a good balance between fine perturbations and algorithm run time. Finally, the initial simplicial complex \(K\) can be chosen to be a simplic polytopic RFIS of a known RFIS. For example, if a Lyapunov function \(V\) for the system can be constructed [38], then \(K\) can be simply chosen to be a polytopic approximation of any level set of \(V\). Alternatively, one can linearize the system about a point and use a polytopic RFIS for the linear system as a starting polytope.
Algorithm 2: RFIS Computation Algorithm.

Inputs : $K$, $\alpha$, $F$, $L$, $m$, $t_{max}$

Outputs: $H$

1. Set $H[0] = K$

2. for $t = 1, \ldots, t_{max}$ do

3. Initialize $IND[j] = 1 \forall j \in \{1, \ldots, N\}$ // $N$ is the cardinality of Vert($K$)

4. while $\exists i$ such that $IND[i] = 1$ do

5. foreach $j \in \{1, \ldots, N\}$ do

6. Apply $\phi_j^\alpha : \text{Vert}(K) \to \text{Vert}(L)$ on $v_j$

7. foreach $\Delta \in \overline{\text{St}}_K(v_j) \cup \overline{\text{St}}_L(v_j)$ do

8. Compute $N(\Delta) = \text{BCD}_\text{Test}(\text{Vert}(\Delta), F, \ell, m)$

9. end

10. if (13) is true then

11. Set $K = L$ and set $IND[j] = 1$

12. else

13. Set $IND[j] = 0$

14. end

15. end

16. Set $H[t] = K$

17. Set $K = \frac{K}{\text{Barycentric subdivision}}$

18. end

19. if $\sum_{\Delta \in \text{H}[t_{max}]} N(\Delta) \neq 0$ then

20. display “RFIS Not Found”

21. return $H = \{H[0], \ldots, H[t_{max}]\}$

Finally, if other RFIS finding algorithms are applicable, for instance if the system is described by polynomial dynamics, then one can start with a polytopic RFIS itself. In this case, our algorithm serves as an RFIS refining mechanism, finding many RFISs of different sizes and shapes.

Let $c \in \text{int}(K)$. We invoke (4) to compute the volume enclosed by the polytope defined by $K$, using only $\text{Vert}(K)$. Notice that $H[t]$ triangulates only the boundary of a polytope, but the polytope itself is triangulated by the set

$$H'[k] = \{\text{Vert}(\Delta') : \{c\} \cap \text{Vert}(\Delta), \Delta \in H[k]\}.$$

Thus, $\text{Vol}(H'[t]) = \sum_{\Delta \in H'[t]} \text{Vol}(\Delta')$, and $\text{Vol}(H'[\cdot])$ is as in (4).

**Theorem 5.3:** Let $R_4$ and $R_m$ be a minimal and maximal RFIS, respectively, for the system $F$ in (5). Let $K$ triangulate $\partial \mathcal{P}$, where $\mathcal{P}$ is a convex polytope satisfying $R_4 \subset \mathcal{P} \subset R_m$. Starting with $H[0] = K$ and $\alpha \in (0, \infty)$, Algorithm 2 terminates. Further, if $H[t]$ is an RFIS for any $t$, then the sequence $(H[t])_{t \geq 1}$ is a sequence of RFISs that successively reduce in volume if $\alpha < 1$, and increase in volume if $\alpha > 1$.

**Proof:** Assume instead that for some $t$, the while loop does not terminate, so that infinitely many perturbations satisfy (13). Since $|\text{Vert}(K)| < \infty$, it follows that there exists a vertex $v_j$ for which (13) is satisfied infinitely many times. Further, since $M_K := \sum_{\Delta \in K} N(\Delta)$ takes values in a finite set, and since each perturbation strictly reduces $M_K$ when it is positive, it follows that $M_K = 0$ after finitely many iterations of the while loop. Equivalently, there exists $K \in N$ such that $K$ triangulates the boundary of an RFIS after $K$ iterations. Let $\Phi_k(v_j)$ denote the image of $v_j$ after $k$ iterations of the while loop.

1. If $\alpha < 1$, it follows from the definition of $\phi^\alpha$ in (9) that $\lim_{k \to \infty} \Phi_k(v_j) = c$. Thus, for any $\varepsilon > 0$, there exists $k > K$ such that an $\varepsilon$-ball centered at $c$ is not entirely contained in the polytope defined by $K$. This contradicts the fact that $c \in \text{int}(R_m)$.

2. If $\alpha > 1$, then $v_j$ is mapped to infinity along the ray $B_1(c)$ under $\Phi_k$ as $k \to \infty$. This contradicts the fact that $R_m$ is bounded.

We conclude that the while loop must terminate in finitely many iterations.

Suppose $K = H[t]$ is an RFIS for some $t$, so $M_k = 0$. Any further perturbation $\phi^\alpha_j : \text{Vert}(K) \to \text{Vert}(L)$ is only accepted if it satisfies (13). Thus, $M_k = 0$ and $L$ is also an RFIS. Let simplicial complexes $\mathcal{K}$ and $\mathcal{L}$ be triangulations of the polytopes $\mathcal{P}_K$ and $\mathcal{P}_L$, obtained as in (14). Then

$$\text{Vol}(\mathcal{P}_K) = \sum_{\Delta \in \text{St}_K(v_j)} \frac{1}{n!} \det(B_\Delta) + \sum_{\Delta \in \text{St}_L(v_j)} \frac{1}{n!} \det(B_\Delta).$$

The second term in (15) is not affected by $\phi^\alpha_j$. Let $\Delta_0 \in \overline{\text{St}}_K(v_j)$, and let $\text{Vert}(\Delta_0) = \{v_0, \ldots, v_{n-1}, c\}$, so that

$$\text{Vol}(\Delta_0) = \frac{1}{n!} \det \begin{bmatrix} v_0 - v_0, & \ldots, & v_{n-1} - v_0, & c - v_0 \end{bmatrix} = \frac{1}{n!} \det \begin{bmatrix} 0 & \cdots & v_j & \cdots & v_{n-1} & c \end{bmatrix}.$$

Denote by $\Delta$ the simplex corresponding to $\Delta_0$ after the perturbation $\phi^\alpha_j$. By definition of $\phi^\alpha_j$, it follows that

$$\text{Vol}(\Delta) = \frac{1}{n!} \det \begin{bmatrix} v_0 & \cdots & v_j & (1-\alpha)c & \cdots & v_{n-1} & c \end{bmatrix}. (18)$$

But the matrix in (18) can be obtained using elementary column transformations on the matrix in (17), so $\text{Vol}(\Delta) = \alpha \text{Vol}(\Delta_0)$. It follows from (15) that $\phi^\alpha_j$ increases (resp. decreases) the polytope volume if $\alpha > 1$ (resp. $\alpha < 1$).

**Corollary 5.3.1:** If $H[0]$ is an RFIS, then so is $H[k]$ for all $k$.

Note: The algorithm may terminate without finding an RFIS if the deformations do not significantly change $K$. However, our simulations show that a good choice of initial polytope results in fast convergence to a polytopic approximation of an $R_m$ when $\alpha < 1$. Algorithm 2 requires only $\text{Vert}(\Delta)$ for all $\Delta \in K$, since the vertices fully describe perturbations and subdividing [35], as well as $N_\Delta$, $\text{Vol}(\Delta)$ and $\mathcal{T}_m$.

**VI. SIMULATIONS**

We apply Algorithm 2 to dynamical systems in two and three dimensions. In all figures, red curves and blue arrows represent the system trajectories and vector field, respectively. In these simulations, $H[0]$ is an RFIS, so the polytope remains an RFIS after each accepted perturbation. The computation times and polytope volumes for each iteration of the algorithm is presented in Table I. All computations were done on a Dell Workstation with an Intel Xeon processor @ 3.30 GHz and 16 GB of RAM.

**A. Fitzhugh–Nagumo Neuron Model**

The Fitzhugh–Nagumo system is modeled in [20] as

$$\dot{x}_1 = x_1 - x_1^3/3 - x_2 + 7/8, \quad \dot{x}_2 = 0.08(x_1 + 0.7 - 0.8x_2).$$

Setting $H[0]$ as a convex quadrilateral RFIS, and $\alpha = 0.95$, $m = 8$, $t_{max} = 7$ and $c = [0 \ 1]^T$, Algorithm 2 converges and there is no change.
Algorithm result for 2-D systems. (a) Fitzhugh–Nagumo neuron model, (b) Curve tracking problem, (c) Van der Pol oscillator, (d) Reversed Van der Pol oscillator.

Fig. 5. Algorithm result for 2-D systems. (a) Fitzhugh–Nagumo neuron model. (b) Curve tracking problem. (c) Van der Pol oscillator. (d) Reversed Van der Pol oscillator.

in volume over the last iteration. This evolution is shown pictorially in Fig. 5(a). These polytopic approximations are much tighter than previous works such as [21, Fig. 1] and [20, Fig. 4].

B. Curve Tracking Problem

The curve tracking problem [39] is considered in the presence of disturbances, with \( \omega = [\omega_1 \omega_2]^\top : [0, \infty) \to \Omega \), with the noise space \( \Omega = \{0\} \times [-1.5, 1.5] \subset \mathbb{R}^2 \). The dynamics are modeled as

\[
\dot{x}_1 = -\sin(x_2) + \omega_1,
\]

\[
\dot{x}_2 = (x_1 - \rho) \cos(x_2) - \mu \sin(x_2) + \omega_2.
\]

The results in Fig. 5(b) show the set valued map \( F \) with the parameters \( c = [1 0]^\top, \alpha = 0.95, m = 8, t_{\text{max}} = 6 \), and extreme rays obtained by using \( \omega(t) = \pm 0.15 \). The blue cones represent all directions for \( \dot{x} \) due to the disturbance. Some trajectories starting at the obtained RFIS boundary are also shown, for a disturbance \( \omega_1(t) = 0, \omega_2(t) = 0.15 \sin(t) \), with \( \rho = 1 \) and \( \mu = 0.42 \). These are the same parameters as [29] and [30], where the RFIS is computed using a different method. The results agree, and Algorithm 2 converges in six iterations (Table I).

C. Van der Pol (VdP) Oscillator

The VdP system is

\[
\dot{x}_1 = x_2,
\]

\[
\dot{x}_2 = \mu(1-x_1^2)x_2-x_1
\]

where \( x_1, x_2 : [0, \infty) \to \mathbb{R} \) are the states and \( \mu \) indicates the strength of damping. The VdP oscillator has a stable limit cycle, which is also its minimal RFIS. \( \Omega(0) \) is set as a polytopic RFIS with six vertices, and \( c \) is chosen as the origin. The results of Algorithm 2 with \( c = [0 0]^\top, \mu = 1, m = 8 \), and \( t_{\text{max}} = 5 \) are shown pictorially in Fig. 5(c) for \( \alpha = 0.98 \) and \( \alpha = 1.02 \), respectively.

D. Reversed VdP (RvdP) Oscillator

The RvdP system is

\[
\dot{x}_1 = -x_2 + \omega_1,
\]

\[
\dot{x}_2 = x_1 - x_2 + x_1^3 + \omega_2.
\]

We consider a polytopic noise space \( \Omega = [-0.03, 0.03] \times [-0.03, 0.03] \). Fig. 5(d) shows the evolution of the polytope due to Algorithm 2, with \( c = [0 0]^\top, \alpha = 0.99, m = 8 \), and \( t_{\text{max}} = 5 \). It also shows three sets of system trajectories: The first two use constant disturbances \( [\omega_1(t) \omega_2(t)]^\top = [0.03 - 0.03]^\top \) and \( [-0.03 0.03]^\top \), respectively, while the third set uses \( \omega_1(t) = 0.01 \sin(2t) + 0.005 \sin(\pi t) + 0.015 \sin(6.53t) \) and \( \omega_2(t) = -0.01 \cos(0.2t) + 0.02 \sin(5\pi t) \). The algorithm converges in four iterations as seen in Table I.

E. Phytoplankton Growth Model

Phytoplankton growth is described by the dynamical system

\[
\dot{x}_1 = 1 - x_1 - \frac{x_1 x_2}{4},
\]

\[
\dot{x}_2 = (2x_3 - 1)x_2,
\]

\[
\dot{x}_3 = \frac{x_1}{4} - 2x_3^3.
\]

The authors in [20] used optimization methods to obtain a polytopic FIS (see Fig. 5 in [20]. Setting \( H(0) \) as an approximate recreation of this, and \( c = [0.9969 0.01 0.3571]^\top, \alpha = 0.9, m = 8, t_{\text{max}} = 3 \), we obtain a much tighter approximation of \( R \), for the system. In just three iterations of Algorithm 2, the polytope is deformed to a much smaller RFIS than previous work, enclosing just 0.15% of the volume that \( H(0) \) did. Fig. 6 shows the invariant sets \( H(0) \) and \( H(3) \), and Fig. 6(c) shows some system trajectories.

F. Thomas’ Cyclically Symmetric Attractor

Thomas’ Cyclically Symmetric Attractor is described by the dynamical system

\[
\dot{x}_1 = \sin(x_2) - bx_1,
\]

\[
\dot{x}_2 = \sin(x_3) - bx_2,
\]

\[
\dot{x}_3 = \sin(x_1) - bx_3.
\]

| Iteration | Van der Pol (VdP) | Fitzhugh–Nagumo | Curve Tracking | Reversed VdP | Phytoplankton Growth | Thomas Attractor |
|-----------|-------------------|-----------------|---------------|--------------|---------------------|-----------------|
|           | Time [s] | Volume | Time [s] | Volume | Time [s] | Volume | Time [s] | Volume | Time [s] | Volume |
| 0         | 59.051  | 99.121 | 0.063  | 0.3717 | 0.5827 | 8000.0 |
| 1         | 0.152  | 54.851 | 0.121  | 0.1575 | 1.81   | 371.22 |
| 2         | 0.281  | 31.104 | 0.122  | 0.0661 | 8.65   | 6.5610 | 303.4 |
| 3         | 0.787  | 23.102 | 0.341  | 0.0278 | 12.6   | 0.0014 | 6.5610 | 303.4 |
| 4         | 2.752  | 17.738 | 1.051  | 0.0333 | 27.8   | 0.0009 | 6.5610 | 303.4 |
| 5         | 10.37  | 16.961 | 3.739  | 0.0347 | 174.0  | 0.0347 |
| 6         | 41.62  | 16.771 | 14.18  | 0.0198 |        |        |        |        |
| 7         | 56.02  | 16.151 | 16.99  | 0.0198 |        |        |        |        |
We consider $b = 0.3$, for which the system has two stable attractive limit cycles. Setting $c = [0 0 0]^\top$, $\alpha = 0.95$, $t_{\text{max}} = 3$, $m = 8$, $t_{\text{max}} = 3$, and $H[0]$ as a triangulation of the boundary of the cube with vertices $(\pm 10, \pm 10, \pm 10)$, the algorithm converges to an FIS in three iterations. Fig. 7 shows the invariant sets $H[0]$ and $H[3]$, and Fig. 7(c) shows some system trajectories.

VII. CONCLUSION

In this work, we developed an algorithm to generate a sequence of RFIS of different sizes. First, we showed that if a nonlinear system’s dynamics are Lipschitz continuous, then testing for invariance of a given (polytopic) set can be easily done by checking for a Boundary Condition on finitely many points, amounting to matrix-vector multiplication. Based on this, we developed an algorithm to deform a polytope into an RFIS, through a sequence of homeomorphisms induced by vertex maps on a simplicial complex that triangulates the boundary of the polytope. The geometric nature of the approach allows the algorithm to be used on any Lipschitz continuous nonlinear dynamical system in the presence of additive bounded disturbances. Application of the algorithm to a variety of nonlinear systems showed fast convergence and resulted in a sequence of RFIS of different sizes.

Our algorithm provides interesting directions for future research. A study of vertex maps and their induced homeomorphisms on convergence of the algorithm would be interesting. One may also modify the algorithm to adaptively adjust the parameters such as $\alpha$ and $c$, based on the behavior of the vector field at the boundary of the polytope. This may help to speed up the algorithm. One may also optimize the sequence in which to apply the maps, since homeomorphisms need not commute. Other measures of how close a simplex is to being invariant, besides $\mathcal{N}(\Delta)$, may also be investigated.

APPENDIX A

We prove Theorem 4.6, which shows that testing for the Boundary Condition at a subset of test points is sufficient to check the invariance of the simplex.

Definition A.1 (Adjacent Lattice Point): Let $w_1 \in T_m^\Delta \subset \Delta$ be a lattice test point on the simplex $\Delta$. Another lattice point $w_2 \in T_m^\Delta$ is said to be adjacent to $w_1$ if the two test points are related as

$$w_2 = w_1 \pm \delta_m (v_i - v_j)$$

for some $v_i, v_j \in \text{Vert}(\Delta)$, where $\delta_m = 2^{-m}$. We denote the set consisting of $w$ and all its adjacent points by $A(w)$.

Adjacent points are shown in blue in Fig. 8(a). The convex hull of $A(w_1)$ is shaded and denoted by $\text{Conv}(A(w_1))$. We now show that these test points are appropriately located.

Lemma B.2: Let $r = \max_{i,j} \|v_i - v_j\|$, where $v_i, v_j \in \text{Vert}(\Delta)$. Let $w_1 \in T_m^\Delta$ be any lattice test point. Let $w_j$ be another lattice test point adjacent to $w_1$. Then $\|w_2 - w_1\| < r$ for some $w_2 \in T_m^\Delta$.
point such that \( w_j \in A(w_i) \). Then, the Euclidean distance between \( w_i \) and \( w_j \) satisfies \( \| w_i - w_j \| \leq r \delta_m \).

Proof: Since \( w_j \in A(w_i) \), there exist vertices \( v_i, v_j \in \text{Vert}(\Delta) \) such that \( w_j = w_i + \delta_m(v_i - v_j) \). We then have

\[
\| w_i - w_j \| = \delta_m \| v_i - v_j \| \leq \delta_m \max_{i,j} \| v_i - v_j \| = \delta_m r
\]
as desired.

We now prove Theorem 4.6. Fig. 8(b) provides an illustration to help visualize the proof.

Proof: Let \( B_{\delta}(z) \) be the ball of radius \( k \) centered at \( z \). For any lattice test point \( x \in \mathbb{T}_k^\Delta \), it follows from Lemma B.2 that \( B_{\delta_k}(x) \) contains all adjacent lattice test points of \( x \). Thus, \( A(x) \subset B_{\delta_k}(x) \). Further, \( B_{\delta_m}(x) \) is a convex set, implying that the convex hull of the set \( A(x) \) satisfies \( \text{Conv}(A(x)) \subset B_{\delta_m}(x) \). Taking the union over all lattice test points, we have

\[
\bigcup_{x \in \mathbb{T}_k^\Delta} \text{Conv}(A(x)) \subset \bigcup_{x \in \mathbb{T}_k^\Delta} B_{\delta_m}(x).
\]

Since the vertices of the simplex \( \text{Vert}(\Delta) \subset \mathbb{T}_k^\Delta \) by construction, the left-hand side of (21) is \( \Delta \) itself. Further, from Theorem 4.3, we know that the Invariance Condition is satisfied at all points in each of the balls, and therefore, also on all subsets of their union. Since \( \Delta \) is such a subset, it follows that \( \Delta \) is an invariant simplex. This concludes the proof.

REFERENCES

[1] B. Legat, P. Tabuada, and R. M. Jungers, “Sum-of-squares methods for controlled invariant sets with applications to model-predictive control,” Nonlinear Anal.: Hybrid Syst., vol. 36, 2020, Art. no. 108858.

[2] J. A. Foster, M. Rossetti, D. R. Ramirez, D. Limon, and E. Garone, “Reference dependent invariant sets: Sum of squares based computation and applications in constrained control,” Automatica, vol. 129, 2021, Art. no. 109614.

[3] F. Tahir and I. M. Jaimoukha, “Iterative computation of constrained forward invariant sets via finite time Aumann integrals,” in Proc. 53rd IEEE Conf. Decis. Control, 2014, pp. 6348–6350.

[4] J. G. Da Silva and S. Tarbouriech, “Antiwindup design with guaranteed regions of stability: An LMI-based approach,” IEEE Trans. Autom. Control, vol. 50, no. 1, pp. 106–111, Jan. 2005.