2019 IEEE 58th Conference on Decision and Control (CDC)
Palais des Congrès et des Expositions Nice Acropolis
Nice, France, December 11-13, 2019

Tight decomposition functions for mixed monotonicity

Liren Yang Necmiye Ozay

***Abstract*— Mixed monotonicity is a property of a system's vector field that says that the vector field admits a decomposition function, in a lifted space, that has some order-preserving properties. It is recently shown that this property allows one to efficiently over-approximate the system's one-step reachable set with a hyperinterval, which is obtained by evaluating the vector field's decomposition function at two points. Such decomposition functions are usually not unique and some decompositions may not be tight in the sense that the resulting hyperintervals are not the smallest ones that contain the exact one-step reachable set, which leads to conservative over-approximation. In this paper, we show that for a general class of functions, there exists a tight decomposition, which can be implicitly constructed as the solution of certain optimization problems. This implies that any function from $\mathbb{R}^n$ to $\mathbb{R}^m$ (hence any forward complete system) is mixed-monotone. However, the usefulness of the constructed tight decomposition functions is limited by the fact that it might not be possible to evaluate them efficiently. We show that under certain conditions, the tight decompositions can reduce to a function with explicit expression, which can be directly evaluated. This result suggests that it is not mixed monotonicity itself, but other extra properties, which lead to explicitly evaluatable decomposition functions, that enable efficient and tight hyperinterval over-approximation of reachable sets.**

I. Introduction

Reachable set computation and over-approximation is widely used in formal verification and control synthesis of cyber physical systems. Consider an autonomous discrete-time dynamical system in the following form

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

where $\mathbf{x}_k \in \mathbb{R}^n$ is the state, and $f : \mathbb{R}^n \to \mathbb{R}^n$ is the vector field, one-step reachable set computation solves the following problem: given initial state set $X_0 \subseteq \mathbb{R}^n$, what is the (one-step) reachable set $X_1 := \{f(\mathbf{x}) \mid \mathbf{x} \in X_0\}$? Being able to compute X_1 is useful in system verification because one can compute the system's k-step reachable set X_k recursively and verify, for example, if X_k does not intersect with a set X_b of bad states for all k, or if X_k is entirely contained by a set X_g of good states for some k (Fig. 1, left). For these purposes, clearly it is sufficient to over-approximate the reachable sets and check if the over-approximation intersects X_b or is fully contained by the X_g. In abstraction-based verification and control synthesis, one can construct a finite graph structure, called an abstraction, that preserves certain properties of the underlying dynamical system and perform verification and control synthesis on the abstraction leveraging graph search algorithms [16]. To compute the abstraction, one usually discretizes the state space into small regions, maps each region to a node of the abstraction and then constructs the transitions between the nodes via computing the one-step reachable sets starting from each region (Fig. 1, right).

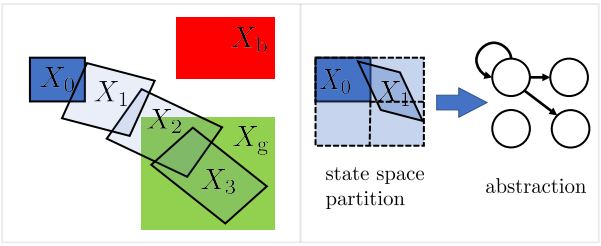

Fig. 1: Left: verification via k-step reachable set computation. Right: abstraction computation via one-step reachable set computation.

In the past several decades, there were considerable amount of research focusing on computing (or over-approximating) reachable sets efficiently. While there are approaches developed for general (both continuous-time and discrete-time) nonlinear systems, e.g., [3], [6], [13], [20], there are many other works that are tailored to systems with special structural properties, for example, linearity [7], [2], piecewise affine property[1], multi-affine property [9], monotonicity [15] and mixed monotonicity [4], [11].

In particular, mixed monotonicity has recently attracted some attention in reachable set computation, especially when such computation is involved in abstraction-based control synthesis [4], [5], [10], [12], [17]. Similar ideas can also be found in the literature of interval observers, for example, see [14]. The brief idea is shown in Fig. 2: if a system in the form of Eq. (1) has mixed monotone vector field f, and if the initial set X_0 is a hyperinterval[1] $\{f(\mathbf{x}) \mid \underline{\mathbf{x}} \leq \mathbf{x} \leq \overline{\mathbf{x}}\}$, then one can find another function g, called the decomposition function of f, and approximate the one-step reachable set X_1 with another hyperinterval defined by $\{\mathbf{x} \mid g(\underline{\mathbf{x}}, \overline{\mathbf{x}}) \leq \mathbf{x} \leq g(\overline{\mathbf{x}}, \underline{\mathbf{x}})\}$. The benefits of using mixed monotonicity for reachable set computation are twofold. First, the computation is fairly efficient as it amounts to evaluating the decomposition function g at only two points, whose complexity does not grow dramatically with the dimension of the state space. Secondly, the fact that a hyperinterval is the product of lower dimensional hyperintervals naturally leads to a system decomposition, which enables more efficient computation and storage of abstractions for system verification and synthesis [8].

[1]Here, $\leq$ is the element wise order in $\mathbb{R}^n$.

LY and NO are with the Dept. of Electrical Engineering and Computer Science, Univ. of Michigan, Ann Arbor, MI 48109 {yliren,necmiye}@umich.edu. This work is supported in part by NSF Grant ECCS-1553873.

Authorized licensed use limited to: Huazhong University of Science and Technology. Downloaded on October 30,2022 at 01:07:31 UTC from IEEE Xplore. Restrictions apply.

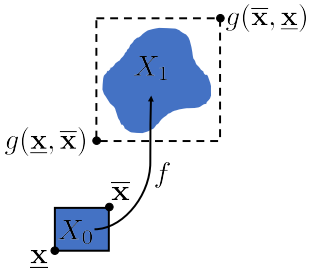

Fig. 2: Illustration: over-approximating the reachable set of a system with mixed monotone vector field f by evaluating the decomposition function at two points, i.e., $(\underline{\mathbf{x}}, \overline{\mathbf{x}})$ and $(\overline{\mathbf{x}}, \underline{\mathbf{x}})$.

Clearly, the usefulness of mixed monotonicity in reachable set computation relies on the following three facts:

- one can show that f has a decomposition function g;
- the decomposition function g can be evaluated efficiently;
- the over-approximation of the reachable set using decomposition function g is not too conservative.

Regarding the above three bullets, the following questions need to be answered:

Q1. How to verify mixed monotonicity of function f (i.e., prove that f has a decomposition function)? Is it a very special property or is it a general property that most functions have?

Q2. When does a mixed monotone function f have a decomposition function in explicit expression so that it can be evaluated efficiently?

Q3. When does a mixed monotone function f have a tight decomposition function?

Works like [4], [18], [19] try to address these questions. In [4], it is shown that every continuously differentiable function with a sign-stable Jacobian is mixed monotone. In particular, the associated decomposition function can be defined directly from the expression of function f and hence can be evaluated efficiently. Moreover, this decomposition function is "tight" (or gives tight one-step reachable set over-approximation) in the sense that hyperinterval $\{\mathbf{x} \mid g(\underline{\mathbf{x}}, \overline{\mathbf{x}}) \leq \mathbf{x} \leq g(\overline{\mathbf{x}}, \underline{\mathbf{x}})\}$ is the smallest (in set inclusion sense) that contains the exact one-step reachable set $\{f(\mathbf{x}) \mid \underline{\mathbf{x}} \leq \mathbf{x} \leq \overline{\mathbf{x}}\}$. Following this, [19] extends similar results to all continuously differentiable functions with bounded partial derivative, except that the associated decomposition functions may not be tight. Despite of this potential drawback, [19] still leads to some successful applications, e.g., [12], [17], in the sense that satisfactory controllers can be still synthesized on the abstraction obtained via such conservative reachable set over-approximation. In a more recent work [18], it is further shown that every function with bounded variation has a decomposition function. However, such decompositions are neither tight nor in explicit form. This hence prevents applying the result to reachable set computation.

In this work, we follow this line of research and try to answer this question: does a general class of functions (other than functions with sign stable Jacobian) have a tight decomposition function, and if yes, when such tight decomposition function can be written in explicit form so that they can be evaluated efficiently. The key result of this work is to show that every function whose extreme values are well defined actually has a (not necessarily unique) tight decomposition, which is "constructed" using optimization. We further show that this result is consistent with [4] when function f has sign-stable Jacobian matrix.

II. Preliminaries

A. Notations

Let $\mathbb{R}^n$ be the n-dimensional Euclidean space. In this paper, lower case letters, e.g., x, are often used to denote real scalar, while bold font lower case letters, e.g., $\mathbf{x}$, are used to denote vectors from $\mathbb{R}^n$ and we let x_i be the i^{th} element of vector $\mathbf{x}$. For a function $f : \mathbb{R}^n \to \mathbb{R}^m$, we use $f(\mathbf{x})$ to denote the function value of a vector $\mathbf{x} \in \mathbb{R}^n$ and use $f(x_i)$ as the short notation of $f(\mathbf{x})$ with x_i highlighted as the variables and x_js viewed as function parameters for $j \neq i$. We also use $f_i(\mathbf{x})$ to denote the i^{th} component value of $f(\mathbf{x})$. We use $\geq$ to denote the element wise order on $\mathbb{R}^n$, i.e., for $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, $\mathbf{x} \geq \mathbf{y}$ if and only if (iff) $x_i \geq y_i$ for all $i = 1, 2, \ldots, n$. For $\underline{\mathbf{x}}, \overline{\mathbf{x}} \in \mathbb{R}^n$ such that $\underline{\mathbf{x}} \leq \overline{\mathbf{x}}$, we denote the hyperinterval $\{\mathbf{x} \in \mathbb{R}^n \mid \underline{\mathbf{x}} \leq \mathbf{x} \leq \overline{\mathbf{x}}\}$ by $[\underline{\mathbf{x}}, \overline{\mathbf{x}}]$.

B. Mixed Monotone Functions on $\mathbb{R}^n$

In this part, we give some relevant concepts and preliminary results regarding mixed monotone functions defined on $\mathbb{R}^n$. These concepts are well established in the literature and we present them simply to make this paper more self-contained.

Definition 1: (Mixed Monotone Function) A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is *mixed monotone* if there exists $g : \mathbb{R}^{2n} \to \mathbb{R}^m$ satisfying the following:

1. $g(\mathbf{x}, \mathbf{x}) = f(\mathbf{x})$;
2. $\mathbf{x} \geq \mathbf{x}' \Rightarrow g(\mathbf{x}, \mathbf{y}) \geq g(\mathbf{x}', \mathbf{y})$;
3. $\mathbf{y} \geq \mathbf{y}' \Rightarrow g(\mathbf{x}, \mathbf{y}) \leq g(\mathbf{x}, \mathbf{y}')$.

A function g satisfying the above conditions is called a *decomposition function* of f.

Remark 1: In general, a mixed monotone function can be defined for $f : X \to T$ where X and T are ordered spaces defined via a notion of positive cone [4]. Here, for simplicity, we only consider the case where $X = \mathbb{R}^n$ and $T = \mathbb{R}^m$ and the order is the one induced by the positive orthant (i.e., element wise order).

As described in the introduction, it is possible to use the decomposition function in reachable set computation for a system with mixed monotone vector field. The result is formally stated with the following proposition.

Proposition 1: [4] Let $f : \mathbb{R}^n \to \mathbb{R}^m$ be a mixed monotone function and let g be one of its decomposition functions, then $\{f(\mathbf{x}) \mid \mathbf{x} \in [\underline{\mathbf{x}}, \overline{\mathbf{x}}]\} \subseteq [g(\underline{\mathbf{x}}, \overline{\mathbf{x}}), g(\overline{\mathbf{x}}, \underline{\mathbf{x}})]$.

Motivated by the use of decomposition functions to tightly over approximate $\{f(\mathbf{x}) \mid \mathbf{x} \in [\underline{\mathbf{x}}, \overline{\mathbf{x}}]\}$ with a hyperinterval, we introduce the following definition of tightness of a decomposition function.

Authorized licensed use limited to: Huazhong University of Science and Technology. Downloaded on October 30,2022 at 01:07:31 UTC from IEEE Xplore. Restrictions apply.

Definition 2: (Tight Decomposition) Let f be a mixed monotone function and g be a decomposition of f. Decomposition function g is called *tight* if for all $\underline{\mathbf{x}}, \overline{\mathbf{x}} \in \mathbb{R}^n$ s.t. $\underline{\mathbf{x}} \leq \overline{\mathbf{x}}$, $[g(\underline{\mathbf{x}}, \overline{\mathbf{x}}), g(\overline{\mathbf{x}}, \underline{\mathbf{x}})]$ is the smallest (in set inclusion sense) hyperinterval that contains $\{f(\mathbf{x}) \mid \mathbf{x} \in [\underline{\mathbf{x}}, \overline{\mathbf{x}}]\}$. That is $[g(\underline{\mathbf{x}}, \overline{\mathbf{x}}), g(\overline{\mathbf{x}}, \underline{\mathbf{x}})] = [\inf_{\boldsymbol{\xi} \in [\underline{\mathbf{x}}, \overline{\mathbf{x}}]} f(\boldsymbol{\xi}), \sup_{\boldsymbol{\xi} \in [\underline{\mathbf{x}}, \overline{\mathbf{x}}]} f(\boldsymbol{\xi})]$.

Remark 2: Tightness of decomposition function g only depends on the definition of g on set $\{(\mathbf{x}, \mathbf{y}) \mid \mathbf{x}, \mathbf{y}$ are comparable$\}$.

The following result is a recently established sufficient condition for a function f in $\mathbb{R}^n$ to be mixed monotone. It comes with the nice properties that the associated decomposition function g is tight and can be constructed directly from the expression of function f. Later, we will prove that g reduces to a special case of the tight decomposition function we construct for more general f using optimization ideas.

Proposition 2: [4] If $f : \mathbb{R}^n \to \mathbb{R}^m$ is continuously differentiable and has sign-stable Jacobian, i.e., either $\frac{\partial f_j}{\partial x_i} \geq 0$ for all $\mathbf{x}$ or $\frac{\partial f_j}{\partial x_i} \leq 0$ for all $\mathbf{x}$, then f is mixed monotone and has a tight decomposition function in the following form:

$$g_j(\mathbf{x}, \mathbf{y}) = f_j(\mathbf{z}), \quad j = 1, 2, \ldots, m, \tag{2}$$

where

$$z_i = \begin{cases} x_i & \text{if } \frac{\partial f_j}{\partial x_i} \geq 0 \ \ \forall \mathbf{x} \\ y_i & \text{if } \frac{\partial f_j}{\partial x_i} \leq 0 \ \ \forall \mathbf{x} \end{cases}. \tag{3}$$

III. Main Results

The key result of this paper is that every function whose extreme values are well defined has a tight decomposition. To prove this, we introduce the following notation: for $x, y \in \mathbb{R}$ and $h : \mathbb{R} \to \mathbb{R}$, define

$$\text{opt}_{\xi}^{(x,y)} h(\xi) = \begin{cases} \inf_{\xi \in [x,y]} h(\xi) & \text{if } x \leq y \\ \sup_{\xi \in [y,x]} h(\xi) & \text{if } x > y \end{cases}. \tag{4}$$

The following simple facts regarding the opt operator are useful in later proofs.

Lemma 1: $\text{opt}_{\xi}^{(x,y)} h(\xi)$ is monotonically increasing in x and monotonically decreasing in y, that is, $x \geq x' \Rightarrow \text{opt}_{\xi}^{(x,y)} h(\xi) \geq \text{opt}_{\xi}^{(x',y)} h(\xi)$ and $y \geq y' \Rightarrow \text{opt}_{\xi}^{(x,y)} h(\xi) \leq \text{opt}_{\xi}^{(x,y')} h(\xi)$.

Proof: We prove $x \geq x' \Rightarrow \text{opt}_{\xi}^{(x,y)} h(\xi) \geq \text{opt}_{\xi}^{(x',y)} h(\xi)$ in the following three cases respectively:

(i) $y \geq x \geq x'$: $\text{opt}_{\xi}^{(x,y)} h(\xi) = \inf_{\xi \in [x,y]} h(\xi) \geq \inf_{\xi \in [x',y]} h(\xi) = \text{opt}_{\xi}^{(x',y)} h(\xi)$;

(ii) $x \geq y \geq x'$: $\text{opt}_{\xi}^{(x,y)} h(\xi) = \sup_{\xi \in [y,x]} h(\xi) \geq h(y) \geq \inf_{\xi \in [x',y]} h(\xi) = \text{opt}_{\xi}^{(x',y)} h(\xi)$;

(iii) $x \geq x' \geq y$: $\text{opt}_{\xi}^{(x,y)} h(\xi) = \sup_{\xi \in [y,x]} h(\xi) \geq \sup_{\xi \in [y,x']} h(\xi) = \text{opt}_{\xi}^{(x',y)} h(\xi)$.

The proof for $y \geq y' \Rightarrow \text{opt}_{\xi}^{(x,y)} h(\xi) \leq \text{opt}_{\xi}^{(x,y')} h(\xi)$ is similar. ∎

Lemma 2: Let $\overline{h} : \mathbb{R} \to \mathbb{R}$ and $\underline{h} : \mathbb{R} \to \mathbb{R}$ be such that $\overline{h} \geq \underline{h}$ for all $\xi \in \mathbb{R}$, then $\text{opt}_{\xi}^{(x,y)} \overline{h}(\xi) \geq \text{opt}_{\xi}^{(x,y)} \underline{h}(\xi)$ for all $x, y \in \mathbb{R}$.

Proof: This should be clear by the definition of opt. ∎

Now we formally state the main result of this paper.

Theorem 1: Let $f : \mathbb{R}^n \to \mathbb{R}^m$ be such that $\text{opt}_{\xi_i}^{(x_i,y_i)} f(\xi_i)$ is well defined[2], then the following $g : \mathbb{R}^{2n} \to \mathbb{R}^m$ defined element-wise by

$$g_j(\mathbf{x}, \mathbf{y}) = \text{opt}_{\xi_1}^{(x_1,y_1)} \ \text{opt}_{\xi_2}^{(x_2,y_2)} \ldots \text{opt}_{\xi_n}^{(x_n,y_n)} f_j(\boldsymbol{\xi}), \quad j = 1, 2, \ldots, m \tag{5}$$

is a tight decomposition function of f.

Proof: We first prove that g is indeed a decomposition function of f.

1. Clearly, $g(\mathbf{x}, \mathbf{x}) = f(\mathbf{x})$ by definition.
2. To show that $\mathbf{x} \geq \mathbf{x}' \Rightarrow g(\mathbf{x}, \mathbf{y}) \geq g(\mathbf{x}', \mathbf{y})$, it is sufficient to show that this is true for a simple case where $\mathbf{x}$ and $\mathbf{x}'$ differs by only one element, i.e., $x_i \geq x_i'$ and $x_j = x_j'$ for $j \neq i$. For general case, let $\mathbf{x} = \mathbf{x}^0 \geq \mathbf{x}^1 \geq \mathbf{x}^2 \geq \cdots \geq \mathbf{x}^n = \mathbf{x}'$, where $\mathbf{x}^i$ and $\mathbf{x}^{i-1}$ has exactly the same coordinates except for the i^{th} position. Then applying the result for the simple case for n times yields the desired result for the general case.
   Let $\mathbf{x}$ and $\mathbf{x}'$ be such that $x_i \geq x_i'$ and $x_j = x_j'$ for $j \neq i$, and define $h^{(\mathbf{x},\mathbf{y})}(\xi_1, \xi_2, \ldots, \xi_i) := \text{opt}_{\xi_{i+1}}^{(x_{i+1},y_{i+1})} \ldots \text{opt}_{\xi_n}^{(x_n,y_n)} f(\boldsymbol{\xi})$. Since $x_j = x_j'$ for $j \neq i$, $h^{(\mathbf{x},\mathbf{y})}(\xi_1, \xi_2, \ldots, \xi_i) = h^{(\mathbf{x}',\mathbf{y})}(\xi_1, \xi_2, \ldots, \xi_i)$ and we will use h to denote the function in what follows for simplicity. With this notation, $g(\mathbf{x}, \mathbf{y})$ and $g(\mathbf{x}', \mathbf{y})$ can be rewritten as

$$g(\mathbf{x}, \mathbf{y}) = \text{opt}_{\xi_1}^{(x_1,y_1)} \text{opt}_{\xi_2}^{(x_2,y_2)} \ldots \underbrace{\text{opt}_{\xi_i}^{(x_i,y_i)} h(\xi_1, \xi_2, \ldots, \xi_i)}_{=: \overline{h}(\xi_1, \xi_2, \ldots, \xi_{i-1})}, \tag{6}$$

$$g(\mathbf{x}', \mathbf{y}) = \text{opt}_{\xi_1}^{(x_1,y_1)} \text{opt}_{\xi_2}^{(x_2,y_2)} \ldots \underbrace{\text{opt}_{\xi_i}^{(x_i',y_i)} h(\xi_1, \xi_2, \ldots, \xi_i)}_{=: \underline{h}(\xi_1, \xi_2, \ldots, \xi_{i-1})}. \tag{7}$$

   Since $x_i \geq x_i'$, by Lemma 1, we know that for all $\xi_1, \xi_2, \ldots \xi_{i-1}$:

$$\overline{h}(\xi_1, \xi_2, \ldots, \xi_{i-1}) \geq \underline{h}(\xi_1, \xi_2, \ldots, \xi_{i-1}). \tag{8}$$

   Applying Lemma 2 for $i-1$ times leads to $g(\mathbf{x}, \mathbf{y}) \geq g(\mathbf{x}', \mathbf{y})$.
3. Proving that $\mathbf{y} \geq \mathbf{y}' \Rightarrow g(\mathbf{x}, \mathbf{y}) \leq g(\mathbf{x}, \mathbf{y}')$ is similar as bullet 2.

This hence proves that g is a decomposition function of f. Next, we show that g is a tight decomposition. To see this, let $\underline{\mathbf{x}}, \overline{\mathbf{x}} \in \mathbb{R}^n$ to be such that $\underline{\mathbf{x}} \leq \overline{\mathbf{x}}$. Since $\underline{x}_i \leq \overline{x}_i$ for all i,

[2]For reachable set computation, it makes sense to assume that f is bounded on any bounded set so that the system $\mathbf{x}_{k+1} = f(\mathbf{x}_k)$ is forward complete. With such assumption, $\text{opt}_{\xi_i}^{(x_i,y_i)} f(\xi_i)$ is always well defined. However, if we only want to talk about mixed monotonicity of the function f, it is enough to assume that the domain of f is $\mathbb{R}^n$.

Authorized licensed use limited to: Huazhong University of Science and Technology. Downloaded on October 30,2022 at 01:07:31 UTC from IEEE Xplore. Restrictions apply.

by definition of g in Eq. (5) and Eq. (4), we have

$$g(\underline{\mathbf{x}},\overline{\mathbf{x}}) = \inf_{\xi_1\in[\underline{x}_1,\overline{x}_1]}\inf_{\xi_2\in[\underline{x}_2,\overline{x}_2]}\dots\inf_{\xi_n\in[\underline{x}_n,\overline{x}_n]} f(\xi_1,\xi_2,\dots,\xi_n) = \inf_{\boldsymbol{\xi}\in[\underline{\mathbf{x}},\overline{\mathbf{x}}]} f(\boldsymbol{\xi}), \quad (9)$$

$$g(\overline{\mathbf{x}},\underline{\mathbf{x}}) = \sup_{\xi_1\in[\underline{x}_1,\overline{x}_1]}\sup_{\xi_2\in[\underline{x}_2,\overline{x}_2]}\dots\sup_{\xi_n\in[\underline{x}_n,\overline{x}_n]} f(\xi_1,\xi_2,\dots,\xi_n) = \sup_{\boldsymbol{\xi}\in[\underline{\mathbf{x}},\overline{\mathbf{x}}]} f(\boldsymbol{\xi}), \quad (10)$$

and this shows that g is a tight decomposition function. ■

Corollary 1: A mixed monotone function f may not have a unique tight decomposition function.

Proof: Note that the proof of Theorem 1 does not depend on the fact that g is constructed with $\text{opt}_{\xi_i}^{(x_i,y_i)}$ being arranged in an ascending order of i. Therefore one can rearrange the $\text{opt}_{\xi_i}^{(x_i,y_i)}$ operators and this does not change the fact the resulting g is still a tight decomposition, yet it is well know that g may be different in general after such a rearrangement. For example, let $f:[0,2]\times[0,2]\to\mathbb{R}^2$ be such that

$$f_1(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{x}\in[0,1)\times[0,1]\cup[1,2]\times(1,2] \\ 1 & \text{otherwise} \end{cases}$$
$$f_2(\mathbf{x}) = 0 \ \forall \mathbf{x}\in[0,2]\times[0,2]. \quad (11)$$

Consider candidate decomposition function g and g' where

$$g_1(\mathbf{x},\mathbf{y}) = \text{opt}_{\xi_1}^{(x_1,y_1)}\text{opt}_{\xi_2}^{(x_2,y_2)} f_1(\xi_1,\xi_2), \quad (12)$$

$$g_1'(\mathbf{x},\mathbf{y}) = \text{opt}_{\xi_2}^{(x_2,y_2)}\text{opt}_{\xi_1}^{(x_1,y_1)} f_1(\xi_1,\xi_2), \quad (13)$$

and $g_2(\mathbf{x},\mathbf{y}) = g_2'(\mathbf{x},\mathbf{y}) = 0$. By Theorem 1, both g and g' are tight decomposition functions of f. However, at point $\mathbf{x} = [2,0]^T$ and $\mathbf{y} = [0,2]^T$, it can be verified that $g_1(\mathbf{x},\mathbf{y}) = 1$ while $g_1'(\mathbf{x},\mathbf{y}) = 0$. Hence $g \neq g'$ and we have two different tight decomposition functions of f. ■

Remark 3: Note that Corollary 1 is not surprising because decomposition functions are defined on the space of $(\mathbf{x},\mathbf{y})$, while the tightness of a decomposition function g only depends on its value $g(\mathbf{x},\mathbf{y})$ on the set $S := \{(\mathbf{x},\mathbf{y}) \mid \mathbf{x},\mathbf{y} \text{ are comparable}\}$. Therefore, tight decomposition functions are not unique on the entire space of $(\mathbf{x},\mathbf{y})$, although they do coincide with each other on set S.

The main purpose of Theorem 1 is to show that for any function on $\mathbb{R}^n$ whose supremum and infimum are well-defined, there exists a tight decomposition function, so all these functions are mixed monotone. However, this is an existential result rather than a computational one. Indeed, the tight decomposition function defined above is not directly useful in the reachable set computation as its construction involves computing the infimum and supremum of the function f, which is already equivalent to solving for the extreme coordinates of the reachable set, and this defeats the purpose of constructing the decomposition function. However, we show in what follows that the tight decomposition function defined by Eq. (5) reduces to a function with explicit form that can be directly derived from the expression of f, whenever f has sign-stable Jacobian, and this coincides with the result presented in [4].

Theorem 2: Assume that f is continuously differentiable that $\frac{\partial f}{\partial x_i}$ either ≥ 0 everywhere or ≤ 0 everywhere, then the tight decomposition function given by Eq. (5) reduces to exactly the form of Eq. (2)-(3).

Proof: Clearly, if $\frac{\partial f_j}{\partial x_i} \geq 0$ everywhere, then by definition of opt

$$\text{opt}_{\xi_n}^{(x_n,y_n)} f(\boldsymbol{\xi}) = \begin{cases} \inf_{\xi_n\in[x_n,y_n]} f(\xi_1,\xi_2,\dots,\xi_n) \\ \quad \text{if } x_n \leq y_n \\ \sup_{\xi_n\in[x_n,y_n]} f(\xi_1,\xi_2,\dots,\xi_n) \\ \quad \text{if } x_n \geq y_n \end{cases} = f(\xi_1,\xi_2,\dots,x_n). \quad (14)$$

Similarly, if $\frac{\partial f_j}{\partial x_i} \leq 0$ everywhere, $\text{opt}_{\xi_n}^{(x_n,y_n)} f(\boldsymbol{\xi}) = f(\xi_1,\xi_2,\dots,y_n)$. Applying this argument n times leads to exactly the same definition of g in Eq. (2)-(3). ■

IV. Conclusion

In this paper, we proved that any function from $\mathbb{R}^n$ to $\mathbb{R}^m$ is mixed monotone and has a tight decomposition function. From the definition of mixed-monotonicity (see Definition 1), it is not clear that it could be such a generic property. However, by using its interpretation in terms of reachable sets, we showed that a tight decomposition function for any function can be implicitly constructed as a solution to an optimization problem. These tight decompositions, however, are in general not useful in reachable set computation as they cannot be evaluated directly. This indicates that it is neither mixed monotonicity nor the tightness of the decomposition function, but other extra properties (e.g., Jacobian matrix being sign-stable) that make the decomposition directly evaluatable, which, in turn, enable efficient reachable set computation.

Acknowledgments: The authors would like to thank one of the anonymous reviewers of [18] who asked whether there are any functions that are not mixed monotone and how to characterize the tightest mixed monotone decomposition, which motivated this work.

References

[1] E. Asarin, O. Bournez, T. Dang, and O. Maler. Approximate reachability analysis of piecewise-linear dynamical systems. In *International Workshop on Hybrid Systems: Computation and Control*, pages 20–31. Springer, 2000.

[2] O. Botchkarev and S. Tripakis. Verification of hybrid systems with linear differential inclusions using ellipsoidal approximations. In *International Workshop on Hybrid Systems: Computation and Control*, pages 73–88. Springer, 2000.

[3] X. Chen, E. Ábrahám, and S. Sankaranarayanan. Flow*: An analyzer for non-linear hybrid systems. In *International Conference on Computer Aided Verification*, pages 258–263. Springer, 2013.

[4] S. Coogan and M. Arcak. Efficient finite abstraction of mixed monotone systems. In *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*, pages 58–67. ACM, 2015.

[5] M. Dutreix and S. Coogan. Specification-guided verification and abstraction refinement of mixed-monotone stochastic systems. *arXiv preprint arXiv:1903.02191*, 2019.

[6] M. Franzle, C. Herde, T. Teige, S. Ratschan, and T. Schubert. Efficient solving of large non-linear arithmetic constraint systems with complex boolean structure. *Journal on Satisfiability, Boolean Modeling and Computation*, 1:209–236, 2007.

Authorized licensed use limited to: Huazhong University of Science and Technology. Downloaded on October 30,2022 at 01:07:31 UTC from IEEE Xplore. Restrictions apply.

[7] A. Girard. Reachability of uncertain linear systems using zonotopes. In *International Workshop on Hybrid Systems: Computation and Control*, pages 291–305. Springer, 2005.

[8] F. Gruber, E. S. Kim, and M. Arcak. Sparsity-aware finite abstraction. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 2366–2371. IEEE, 2017.

[9] M. Kloetzer and C. Belta. Reachability analysis of multi-affine systems. In *International Workshop on Hybrid Systems: Computation and Control*, pages 348–362. Springer, 2006.

[10] P.-J. Meyer, S. Coogan, and M. Arcak. Sampled-data reachability analysis using sensitivity and mixed-monotonicity. *IEEE Control Systems Letters*, 2(4):761–766, Oct 2018.

[11] P.-J. Meyer, A. Devonport, and M. Arcak. Tira: Toolbox for interval reachability analysis. *arXiv preprint arXiv:1902.05204*, 2019.

[12] P.-J. Meyer and D. V. Dimarogonas. Hierarchical decomposition of ltl synthesis problem for mixed-monotone control systems. *arXiv preprint arXiv:1712.06014*, 2017.

[13] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin. A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on automatic control*, 50(7):947–957, 2005.

[14] M. Moisan and O. Bernard. Interval observers for non monotone systems. application to bioprocess models. *IFAC Proceedings Volumes*, 38(1):43–48, 2005.

[15] N. Ramdani, N. Meslem, and Y. Candau. Computing reachable sets for uncertain nonlinear monotone systems. *Nonlinear Analysis: Hybrid Systems*, 4(2):263–278, 2010.

[16] P. Tabuada. *Verification and control of hybrid systems: a symbolic approach*. Springer, 2009.

[17] L. Yang, A. Karnik, B. Pence, M. T. B. Waez, and N. Ozay. Fuel cell thermal management: Modeling, specifications and correct-by-construction control synthesis. In *Proceedings of American Control Conference*, 2017.

[18] L. Yang, O. Mickelin, and N. Ozay. On sufficient conditions for mixed monotonicity. *arXiv preprint arXiv:1803.04528*, 2018. accepted to IEEE Transactions on automatic control.

[19] L. Yang and N. Ozay. A note on some sufficient conditions for mixed monotone systems. Technical report, University of Michigan, Department of EECS, 2017. Available at `http://hdl.handle.net/2027.42/136122`.

[20] M. Zamani, G. Pola, M. Mazo, and P. Tabuada. Symbolic models for nonlinear control systems without stability assumptions. *IEEE Transactions on Automatic Control*, 57(7):1804–1809, 2012.

Authorized licensed use limited to: Huazhong University of Science and Technology. Downloaded on October 30,2022 at 01:07:31 UTC from IEEE Xplore. Restrictions apply.