ARTICLE IN PRESS
Nonlinear Analysis: Hybrid Systems ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Nonlinear Analysis: Hybrid Systems

journal homepage: www.elsevier.com/locate/nahs

# Finite abstraction of mixed monotone systems with discrete and continuous inputs

Samuel Coogan *, Murat Arcak

*Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, United States*

ARTICLE INFO

*Article history:*
Available online xxxx

*Keywords:*
Mixed monotone systems
Monotone systems
Finite state abstractions

ABSTRACT

We present an efficient computational procedure for finite abstraction of discrete-time *mixed monotone* systems by considering a rectangular partition of the state space. Mixed monotone systems are decomposable into increasing and decreasing components, and significantly generalize the well known class of monotone systems. We tightly overapproximate the one-step reachable set from a box of initial conditions by computing a *decomposition function* at only two points, regardless of the dimension of the state space. We first consider systems with a finite set of operating modes and then extend the formulation to systems with continuous control inputs. We apply our results to verify the dynamical behavior of a model for insect population dynamics and to synthesize a signaling strategy for a traffic network.

© 2016 Elsevier Ltd. All rights reserved.

## 1. Introduction

Complex systems often possess intrinsic structure that significantly simplifies analysis and control. An important class of systems exhibiting such structure is *monotone systems* for which trajectories maintain a partial ordering on states [1,2]. The notion of monotonicity is applicable to both continuous-time systems [2] and discrete-time systems [3], and has been extended to control systems with inputs in [4].

Refs. [5–8] have observed that dynamics which are not monotone may nonetheless be decomposable into increasing and decreasing components. Such systems are called *mixed monotone* and significantly generalize the class of monotone systems. Unlike the references above which exploit mixed monotonicity for stability analysis, here we demonstrate that mixed monotonicity enables efficient finite state abstraction.

Increased interest in verification and synthesis of *cyber–physical* systems has motivated symbolic models that abstract the underlying system into a finite set of symbols and transitions between symbols which reflect the dynamics [9–11]. The main reason for obtaining finite state abstractions is to allow formal verification and synthesis for specifications given in, *e.g.*, temporal logic [12–15].

In rare cases, exact symbolic models exactly capture the underlying dynamics [16,17]. In other cases, exact symbolic models are either impossible to obtain or computationally prohibitive; however it is still useful to obtain an abstraction which approximately captures the underlying dynamics [13,18–20]. For example, in [21,22] the authors consider piecewise affine (PWA) systems and construct a finite state abstraction using polyhedral computations.

In this work, we compute finite state abstractions of mixed monotone, discrete-time systems by considering a rectangular partition of the state space. In particular, we show that the reachable set from a box of initial conditions is efficiently

* Corresponding author.
*E-mail addresses:* scoogan@eecs.berkeley.edu (S. Coogan), arcak@eecs.berkeley.edu (M. Arcak).

http://dx.doi.org/10.1016/j.nahs.2016.04.005
1751-570X/© 2016 Elsevier Ltd. All rights reserved.

Please cite this article in press as: S. Coogan, M. Arcak, Finite abstraction of mixed monotone systems with discrete and continuous inputs, Nonlinear Analysis: Hybrid Systems (2016), http://dx.doi.org/10.1016/j.nahs.2016.04.005

overapproximated by evaluating a decomposition function, obtained from the mixed monotone system, at only two points. We accommodate disturbance inputs in the dynamics by suitably generalizing the definition of a mixed monotone system in [6]. Furthermore, we characterize a special class of mixed monotone systems in which the dynamics are componentwise monotone and show that our overapproximation is tight in a particular sense to be made precise. Additionally, we suggest an efficient algorithm for identifying a class of spurious trajectories from the abstraction.

For the case of synthesis, we consider two cases whereby a controlled input to the system is available. We first consider systems with a finite set of operating *modes*, each mode corresponding to a particular update map for the dynamical system. At each time step an operating mode is selected, serving as the control input at that time step. We next consider the case when a continuous set of control inputs is available at each time step. In this scenario, we assume the system is affine in control and the available set of inputs is a (possibly state-dependent) polytope. By considering continuous inputs, we significantly extend our prior paper [23].

The importance of monotonicity for reachability computation and abstraction has been noted in [24–26]. In particular, the authors of [24] study discrete-time systems that are monotone with respect to the positive orthant in Euclidean space and show that the reachable set from a box of initial conditions is overapproximated by propagating only the least and greatest points within this box. The present paper studies the broader class of mixed monotone systems and recovers [24] as a special case.

In Section 2, we introduce the notation. In Section 3, we pose the general problem statement and introduce mixed monotone systems. In Section 4, we present an algorithm for efficiently constructing finite state abstractions of mixed monotone systems. We extend the results to the case when the input takes values from a continuous set in Section 5. In the case studies of Section 6, we analyze a model for insect population dynamics and synthesize a signal controller for a traffic network.

## 2. Preliminaries

For $x \in \mathbb{R}^n$, we use superscripts to index the elements of $x$, *i.e.*, $x^i$ is the $i$th component of $x$ and $x = (x^1, \dots, x^n)$, except in the case studies of Section 6 where we use subscripts for clarity. Let $\mathbb{R}_{\geq 0} = \{x \mid x \geq 0\}$ and $\mathbb{R}^n_{\geq 0} = (\mathbb{R}_{\geq 0})^n$. For a set $\mathcal{Z} \subset \mathbb{R}^n$, $\mathrm{int}(\mathcal{Z})$ denotes the interior of $\mathcal{Z}$.

Consider a set $\mathcal{X} \subset \mathbb{R}^n$ along with a *positive cone* $\mathcal{Y}_+ \subset \mathbb{R}^n$ satisfying $\alpha\mathcal{Y}_+ \subset \mathcal{Y}_+$ for all $\alpha \in \mathbb{R}_{\geq 0}$, $\mathcal{Y}_+ + \mathcal{Y}_+ \subset \mathcal{Y}_+$, and $\mathcal{Y}_+ \cap (-\mathcal{Y}_+) = 0$. The positive cone $\mathcal{Y}_+$ induces an order relation $\leq$ on $\mathcal{X}$ defined by: $x \leq y$ if and only if $y - x \in \mathcal{Y}_+$ for $x, y \in \mathcal{X}$. Given $x, y \in \mathcal{X}$ with $x \leq y$, we define the *interval*

$$[x, y] \triangleq \{z \in \mathcal{X} \mid x \leq z \leq y\}. \tag{1}$$

For $\mathcal{Y}_+ = \mathbb{R}^n_{\geq 0}$, $\leq$ denotes coordinate-wise inequality; we distinguish this partial order by $\leq_+$ and generalize it to arbitrary orthants in the following way: Let $\nu = (\nu_1, \dots, \nu_n)$ with $\nu_i \in \{0, 1\}$ for all $i$, and define $K_\nu = \{x \in \mathbb{R}^n \mid (-1)^{\nu_i} x^i \geq 0 \ \forall i\}$. $K_\nu$ is a cone corresponding to an orthant of $\mathbb{R}^n$, and we denote the induced *orthant order* by $\leq_{K_\nu}$.

For a matrix $M \in \mathbb{R}^{n \times p}$, we interpret $0 \leq_+ M$ to mean $M$ is elementwise nonnegative.

A set $\mathcal{Z} \subset \mathbb{R}^n$ is a *box* if it is the Cartesian product of closed intervals of $\mathbb{R}$, that is, if there exist $a_i, b_i \in \mathbb{R}$ for $i = 1, \dots, n$ such that $a_i \leq b_i$ and $\mathcal{Z} = \prod_{i=1}^n [a_i, b_i]_{\mathbb{R}_{\geq 0}}$ where $[\cdot, \cdot]_{\mathbb{R}_{\geq 0}}$ denotes the usual interval on $\mathbb{R}$.

We let $x^+ = F(x, d)$ describe a discrete-time dynamical system where the state $x^+$ at the next time step is a function of the current state $x$ and a disturbance input $d$. We denote the $i$th coordinate mapping of $F$ by $F^i$, that is, $(x^i)^+ = F^i(x, d)$ and $F(x, d) = (F^1(x, d), \dots, F^n(x, d))$.

## 3. Mixed monotone systems

### 3.1. Problem statement

We first consider discrete-time dynamical systems of the form

$$x^+ = F(x, d) \tag{2}$$

with state $x \in \mathcal{X} \subset \mathbb{R}^n$, disturbance input $d \in \mathcal{D} \subset \mathbb{R}^p$, and a continuous map $F : \mathcal{X} \times \mathcal{D} \to \mathcal{X}$. We present a technique for efficiently computing a finite state abstraction of (2) when $F$ is *mixed monotone* as defined below. The resulting symbolic model is amenable to standard formal methods techniques to verify desirable properties, as demonstrated in the case study in Section 6.1.

Next, we consider the problem of controlling the switched discrete-time dynamical system

$$x^+ = F_m(x, d) \tag{3}$$

for $m \in \mathcal{M}$ where $\mathcal{M}$ is a finite set of modes and each $F_m : \mathcal{X} \times \mathcal{D} \to \mathcal{X}$ is continuous. For switched systems of the form (3), the control input is the mode $m$ at each time step. When each $F_m$ satisfies a mixed monotonicity property, we propose an efficient algorithm for obtaining a finite state abstraction. As demonstrated in the case study of Section 6.2, this abstraction is amenable to synthesis algorithms to meet complex control objectives expressible in, *e.g.*, *Linear Temporal Logic* (LTL). In Section 5, we extend this model to allow continuous inputs rather than a finite set of modes.

Please cite this article in press as: S. Coogan, M. Arcak, Finite abstraction of mixed monotone systems with discrete and continuous inputs, Nonlinear Analysis: Hybrid Systems (2016), http://dx.doi.org/10.1016/j.nahs.2016.04.005

### 3.2. Basic definitions and results

For systems of the form (2), we let $\leq_{\mathcal{X}}$ and $\leq_{\mathcal{D}}$ denote order relations on $\mathcal{X} \subset \mathbb{R}^n$ and $\mathcal{D} \subset \mathbb{R}^p$, respectively, induced by positive cones. The notation $[\cdot,\cdot]_{\mathcal{X}}$ (resp. $[\cdot,\cdot]_{\mathcal{D}}$) denotes an interval with respect to $\leq_{\mathcal{X}}$ (resp. $\leq_{\mathcal{D}}$). For systems of the form (3), we wish to allow potentially different order relations on $\mathcal{X}$, and thus consider a set $\{\leq_m\}_{m\in\mathcal{M}}$ of order relations on $\mathcal{X}$ and $\leq_{\mathcal{D}}$, a fixed order relation on $\mathcal{D}$. The notation $[\cdot,\cdot]_m$ denotes an interval with respect to $\leq_m$. For notational convenience, we assume that the same partial order on $\mathcal{D}$ holds for all modes; however different partial orders on $\mathcal{D}$ for each mode are possible with suitable alterations to the development below.

We begin with the well-known class of monotone dynamical systems:

**Definition 1** (*Monotonicity*)**.** The system (2) is *monotone with respect to* $\leq_{\mathcal{X}}$ *and* $\leq_{\mathcal{D}}$, or simply *monotone*, if

$$x_1 \leq_{\mathcal{X}} x_2 \quad \text{and} \quad d_1 \leq_{\mathcal{D}} d_2 \Longrightarrow F(x_1, d_1) \leq_{\mathcal{X}} F(x_2, d_2). \tag{4}$$

The switched system (3) is *monotone with respect to* $\{\leq_m\}_{m\in\mathcal{M}}$ *and* $\leq_{\mathcal{D}}$, or simply *monotone*, if each mode $m$ is monotone with respect to $\leq_m$ and $\leq_{\mathcal{D}}$.

We next provide a significant generalization of Definition 1:

**Definition 2** (*Mixed Monotonicity*)**.** The system (2) is *mixed monotone with respect to* $\leq_{\mathcal{X}}$ *and* $\leq_{\mathcal{D}}$, or simply *mixed monotone* [6], if there exists a function $f : \mathcal{X} \times \mathcal{D} \times \mathcal{X} \times \mathcal{D} \to \mathcal{X}$ satisfying:

(C1) $\forall x \in \mathcal{X},\ \forall d \in \mathcal{D}$: $F(x, d) = f(x, d, x, d)$
(C2) $\forall x_1, x_2, y \in \mathcal{X},\ \forall d_1, d_2, e \in \mathcal{D}$: $x_1 \leq_{\mathcal{X}} x_2$ and $d_1 \leq_{\mathcal{D}} d_2$ implies $f(x_1, d_1, y, e) \leq_{\mathcal{X}} f(x_2, d_2, y, e)$
(C3) $\forall x, y_1, y_2 \in \mathcal{X},\ \forall d, e_1, e_2 \in \mathcal{D}$: $y_1 \leq_{\mathcal{X}} y_2$ and $e_1 \leq_{\mathcal{D}} e_2$ implies $f(x, d, y_2, e_2) \leq_{\mathcal{X}} f(x, d, y_1, e_1)$.

We then call $F(x, d)$ a *mixed monotone map*. We say that the switched system (3) is *mixed monotone with respect to* $\{\leq_m\}_{m\in\mathcal{M}}$ *and* $\leq_{\mathcal{D}}$, or simply *mixed monotone*, if each mode $x^+ = F_m(x, d)$ is mixed monotone with respect to $\leq_m$.

The function $f$ is nondecreasing in the first pair of variables and nonincreasing in the second pair of variables, and is henceforth called a *decomposition function*:

**Definition 3** (*Decomposition Function*)**.** A function $f$ satisfying C1–C3 above is a *decomposition function* for $F(x, d)$.

Clearly every monotone system is mixed monotone with $f(x, d, y, e) \triangleq F(x, d)$. In the case of a switched system (3), we denote by $f_m$ a corresponding decomposition function for each mode $m \in \mathcal{M}$.

**Example 1.** Consider the system

$$x^+ = G(x, d) - H(x, d) \tag{5}$$

for $x \in \mathcal{X} \subset \mathbb{R}^n$, $d \in \mathcal{D} \subset \mathbb{R}^p$, and $G, H : \mathcal{X} \times \mathcal{D} \to \mathcal{X}$ such that $x^+ = G(x, d)$ and $x^+ = H(x, d)$ are monotone systems for $\leq_{\mathcal{X}} = \leq_+$ and $\leq_{\mathcal{D}} = \leq_+$. Then (5) is mixed monotone for $\leq_{\mathcal{X}} = \leq_+$ and $\leq_{\mathcal{D}} = \leq_+$ and $f(x, d, y, e) \triangleq G(x, d) - H(y, e)$ is a decomposition function.

**Example 2.** Consider the system

$$x^+ = A(x, d)x + B(x, d)d =: F(x, d) \tag{6}$$

for $x \in \mathcal{X} \subset \mathbb{R}^n_{\geq 0}$, $d \in \mathcal{D} \subset \mathbb{R}^p_{\geq 0}$, such that:

- $0 \leq_+ A(x, d)$ and $0 \leq_+ B(x, d)$ for all $x \in \mathcal{X}$ for all $d \in \mathcal{D}$,
- $x_1 \leq_+ x_2$ and $d_1 \leq_+ d_2 \Longrightarrow A(x_2, d_2) \leq_+ A(x_1, d_1)$ and $B(x_2, d_2) \leq_+ B(x_1, d_1)$.

Equations of the form (6) arise in the study of population dynamics, [27]. With $f(x, d, y, e) = A(y, e)x + B(y, e)d$, system (6) is mixed monotone for $\leq_{\mathcal{X}} = \leq_+$ and $\leq_{\mathcal{D}} = \leq_+$.

We now characterize a special class of mixed monotone systems in terms of the sign of the entries in $\partial F/\partial x$ and $\partial F/\partial d$, the Jacobians of $F$ with respect to $x$ and $d$.

**Proposition 1.** *Consider the system (2) where $x \in \mathcal{X} \subset \mathbb{R}^n$, $d \in \mathcal{D} \subset \mathbb{R}^p$, $\mathcal{X}$ and $\mathcal{D}$ are boxes, and $F$ is continuously differentiable. If for all $i \in \{1, \ldots, n\}$,*

$$\forall j \in \{1, \ldots, n\}\ \exists s_j \in \{0, 1\} : (-1)^{s_j} \frac{\partial F^i}{\partial x^j}(x, d) \geq 0 \quad \forall x, d \tag{7}$$

*and*

$$\forall j \in \{1, \ldots, p\}\ \exists \sigma_j \in \{0, 1\} : (-1)^{\sigma_j} \frac{\partial F^i}{\partial d^j}(x, d) \geq 0 \quad \forall x, d \tag{8}$$

*then (2) is mixed monotone with respect to any orthant order on $\mathcal{X}$ and $\mathcal{D}$.*

Please cite this article in press as: S. Coogan, M. Arcak, Finite abstraction of mixed monotone systems with discrete and continuous inputs, Nonlinear Analysis: Hybrid Systems (2016), http://dx.doi.org/10.1016/j.nahs.2016.04.005

**Proof.** Let $\nu \in \{0,1\}^n$ and $\mu \in \{0,1\}^p$ characterize arbitrary orthant orders $\leq_{K_\nu}$ and $\leq_{K_\mu}$ on $\mathcal{X}$ and $\mathcal{D}$, respectively. Define

$$f^i(x,d,y,e) \triangleq F^i(z^i, w^i) \tag{9}$$

where $z^i = (z^{i,1}, \ldots, z^{i,n})$, $w^i = (w^{i,1}, \ldots, w^{i,p})$, and

$$z^{i,j} \triangleq \begin{cases} x^j & \text{if } (-1)^{\nu_i+\nu_j}\partial F^i/\partial x^j \geq 0 \; \forall x \in \mathcal{X}, d \in \mathcal{D} \\ y^j & \text{if } (-1)^{\nu_i+\nu_j}\partial F^i/\partial x^j \leq 0 \; \forall x \in \mathcal{X}, d \in \mathcal{D} \end{cases} \tag{10}$$

$$w^{i,j} \triangleq \begin{cases} d^j & \text{if } (-1)^{\nu_i+\mu_j}\partial F^i/\partial d^j \geq 0 \; \forall x \in \mathcal{X}, d \in \mathcal{D} \\ e^j & \text{if } (-1)^{\nu_i+\mu_j}\partial F^i/\partial d^j \leq 0 \; \forall x \in \mathcal{X}, d \in \mathcal{D}. \end{cases} \tag{11}$$

If $\partial F^i/\partial x^j = 0 \; \forall x, d$ for some $i, j$, then the assignment to $z^{i,j}$ is arbitrary, likewise for $w^{i,j}$ if $\partial F^i/\partial d^j = 0 \; \forall x, d$ for some $i, j$. Let $f(x,d,y,e) = (f^1(x,d,y,e), \ldots, f^n(x,d,y,e))$. Clearly $f(x,d,x,d) = F(x,d)$, and a straightforward modification of the well-known *Kamke* conditions for monotonicity [2, Section 3.1] proves that $f$ satisfies the remaining conditions of Definition 2. □

Proposition 1 states that if the partial derivatives of $F$ are *sign stable* over $\mathcal{X} \times \mathcal{D}$, then (2) is mixed monotone with respect to any orthant order on $\mathcal{X}$ and $\mathcal{D}$. The special class characterized in Proposition 1 plays an important role in the case study of Section 6.2; see [28] for a similar characterization that excludes disturbance inputs and [29] for a modified condition when the partial orders on $\mathcal{X}$ and $\mathcal{D}$ are taken to be standard orthant orders.

**Example 3.** Let $\mathcal{X} = \mathbb{R}^2_{\geq 0}$, $\mathcal{D} = \mathbb{R}^2_{\geq 0}$, and consider the system

$$x^+ = F(x,d) = (F^1(x,d), F^2(x,d)) \tag{12}$$

$$= (5x_1 - x_2^3 + 5d_1^2, x_1^2 + 3x_2x_1 - 6d_1d_2) \tag{13}$$

where $x = (x_1, x_2) \in \mathbb{R}^2_{\geq 0}$ and $d = (d_1, d_2) \in \mathbb{R}^2_{\geq 0}$ (we momentarily abandon our superscript convention for notational convenience). For all $x \in \mathcal{X}$, $d \in \mathcal{D}$,

$$\partial F^1/\partial x_1 = 5 \geq 0 \qquad \partial F^1/\partial x_2 = -3x_2^2 \leq 0 \tag{14}$$

$$\partial F^2/\partial x_1 = 2x_1 + 3x_2 \geq 0 \qquad \partial F^2/\partial x_2 = 3x_1 \geq 0 \tag{15}$$

$$\partial F^1/\partial d_1 = 10d_1 \geq 0 \qquad \partial F^1/\partial d_2 = 0 \tag{16}$$

$$\partial F^2/\partial d_1 = -6d_2 \leq 0 \qquad \partial F^2/\partial d_2 = -6d_1 \leq 0. \tag{17}$$

Thus, the system is mixed monotone by Proposition 1. Taking $\leq_{\mathcal{X}} = \leq_+$ and $\leq_{\mathcal{D}} = \leq_+$, we have that

$$f(x,d,y,e) = (5x_1 - y_2^3 + 5d_1^2, x_1^2 + 3x_2x_1 - 6e_1e_2) \tag{18}$$

is a decomposition function where $y = (y_1, y_2)$, $e = (e_1, e_2)$.

Although while Proposition 1 assumed that $F$ is continuously differentiable, the results in fact hold if $F$ is continuous and *piecewise differentiable*, and thus nondifferentiable on a set of measure zero as in the case study of Section 6.2.

### 3.3. Reachable set computation

In this section, we show that an overapproximation of the reachable set from a box of initial states is efficiently computed by evaluating the decomposition function at only two points, regardless of the state space dimension. In the next section, we use this result to obtain finite state abstractions of mixed monotone systems.

We begin with the following key theorem:

**Theorem 1.** *Let (2) be a mixed monotone system with decomposition function $f(x,d,y,e)$. Given $x_1, x_2 \in \mathcal{X}$ and $d_1, d_2 \in \mathcal{D}$ with $x_1 \leq_{\mathcal{X}} x_2$ and $d_1 \leq_{\mathcal{D}} d_2$,*

$$f(x_1,d_1,x_2,d_2) \leq_{\mathcal{X}} F(x,d) \leq_{\mathcal{X}} f(x_2,d_2,x_1,d_1) \quad \forall\, x \in [x_1,x_2]_{\mathcal{X}} \; \forall\, d \in [d_1,d_2]_{\mathcal{D}}. \tag{19}$$

**Proof.** Consider $x, d, y, e$ satisfying

$$x_1 \leq_{\mathcal{X}} x \text{ and } d_1 \leq_{\mathcal{X}} d, \text{ and} \tag{20}$$

$$y \leq_{\mathcal{X}} x_2 \text{ and } e \leq_{\mathcal{X}} d_2. \tag{21}$$

It follows that

$$f(x_1,d_1,x_2,d_2) \leq_{\mathcal{X}} f(x,d,y,e), \qquad \text{and} \tag{22}$$

$$f(y,e,x,d) \leq_{\mathcal{X}} f(x_2,d_2,x_1,d_1). \tag{23}$$

Please cite this article in press as: S. Coogan, M. Arcak, Finite abstraction of mixed monotone systems with discrete and continuous inputs, Nonlinear Analysis: Hybrid Systems (2016), http://dx.doi.org/10.1016/j.nahs.2016.04.005

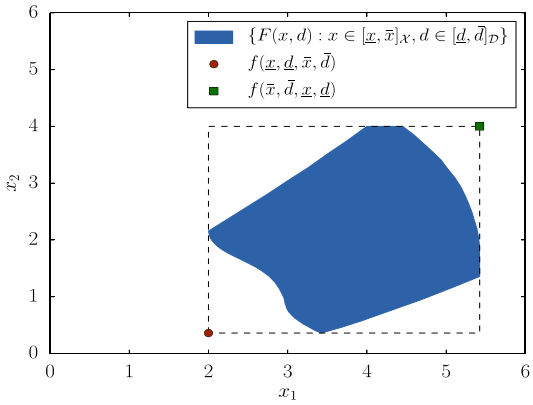


**Fig. 1.** The mixed monotone system in Examples 3 and 4. This system satisfies the conditions of Theorem 1, thus we bound $F(x, d)$ when $x$ and $d$ are confined to lie within a given rectangle by evaluating the decomposition function at two points, and the bounding is tight.

Restricting to the set $\{(x, d, y, e) \mid x = y \text{ and } d = e\}$, we obtain

$$\begin{aligned} f(x_1, d_1, x_2, d_2) &\leq_{\mathcal{X}} f(x, d, x, d) = F(x, d) \\ &\leq_{\mathcal{X}} f(x_2, d_2, x_1, d_1). \quad \square \end{aligned} \tag{24}$$

The analogous result for monotone systems is:

**Corollary 1.** *Given $x_1, x_2 \in \mathcal{X}$ and $d_1, d_2 \in \mathcal{D}$ with $x_1 \leq_{\mathcal{X}} x_2$ and $d_1 \leq_{\mathcal{D}} d_2$. If system (2) is monotone, then*

$$F(x_1, d_1) \leq_{\mathcal{X}} F(x, d) \leq_{\mathcal{X}} F(x_2, d_2) \quad \forall\, x \in [x_1, x_2]_{\mathcal{X}}\ \forall\, d \in [d_1, d_2]_{\mathcal{D}}. \tag{25}$$

The result in [24] is a special case of Corollary 1 restricted to systems with no disturbance input and $\leq_{\mathcal{X}} = \leq_{+}$.

**Definition 4.** For $\mathcal{X}' \subseteq \mathcal{X}$ and $\mathcal{D}' \subseteq \mathcal{D}$, the *one-step reachable set from $\mathcal{X}'$ and $\mathcal{D}'$* is

$$F(\mathcal{X}', \mathcal{D}') \triangleq \{F(x, d) \mid x \in \mathcal{X}' \text{ and } d \in \mathcal{D}'\}. \tag{26}$$

Then we respectively write (19) and (25) as

$$F([x_1, x_2]_{\mathcal{X}}, [d_1, d_2]_{\mathcal{D}}) \subseteq [f(x_1, d_1, x_2, d_2), f(x_2, d_2, x_1, d_1)]_{\mathcal{X}} \tag{27}$$

and

$$F([x_1, x_2]_{\mathcal{X}}, [d_1, d_2]_{\mathcal{D}}) \subseteq [F(x_1, d_1), F(x_2, d_2)]_{\mathcal{X}}. \tag{28}$$

**Example 4.** Consider again Example 3 and let $\underline{x} = (0.6, 0.3)$, $\overline{x} = (1, 1)$, $\underline{d} = (0, 0)$, $\overline{d} = (0.3, 0.3)$. From Theorem 1, it follows that

$$\begin{aligned} F([\underline{x}, \overline{x}]_{\mathcal{X}}, [\underline{d}, \overline{d}]_{\mathcal{D}}) &\subset [f(\underline{x}, \underline{d}, \overline{x}, \overline{d}), f(\overline{x}, \overline{d}, \underline{x}, \underline{d})]_{\mathcal{X}} \\ &= [(2, 0.36), (5.423, 4)]_{\mathcal{X}}. \end{aligned} \tag{29}$$

Fig. 1 shows the set $\{F(x, d) \mid x \in [\underline{x}, \overline{x}]_{\mathcal{X}}, d \in [\underline{d}, \overline{d}]_{\mathcal{D}}\}$ as a shaded region, and plots $f(\underline{x}, \underline{d}, \overline{x}, \overline{d})$ $f(\overline{x}, \overline{d}, \underline{x}, \underline{d})$ as two corners of a box that bounds this set.

For monotone systems, Corollary 1 provides tight bounds since the upper and lower bounds are achieved. For mixed monotone systems satisfying (7)–(8) of Proposition 1, the bounds given in Theorem 1 are also tight as suggested in Fig. 1 for Example 4. We make this precise in the following proposition, which follows immediately from the definition in (9):

**Proposition 2.** *Suppose $\leq_{\mathcal{X}} = \leq_{K_\nu}$ and $\leq_{\mathcal{D}} = \leq_{K_\mu}$ for some orthants $K_\nu$ and $K_\mu$. If (2) is mixed monotone by (7)–(8) of Proposition 1, and $f$ is the decomposition function as defined in (9)–(11), then for all $i \in \{1, \ldots, n\}$ there exist $\underline{z}^i, \overline{z}^i \in [x_1, x_2]_{\mathcal{X}}$ and $\underline{w}^i, \overline{w}^i \in [d_1, d_2]_{\mathcal{D}}$ such that*

$$f^i(x_1, d_1, x_2, d_2) = F^i(\underline{z}^i, \underline{w}^i), \quad \textit{and} \tag{30}$$

$$f^i(x_2, d_2, x_1, d_1) = F^i(\overline{z}^i, \overline{w}^i). \tag{31}$$

In particular, $\underline{z}^i$ as in (10) with $x = x_1$ and $y = x_2$, and $\underline{w}^i$ as in (11) with $d = d_1$ and $e = d_2$ satisfies (30). A symmetric result holds for (31) after interchanging $x_1, x_2$ and $d_1, d_2$.

Please cite this article in press as: S. Coogan, M. Arcak, Finite abstraction of mixed monotone systems with discrete and continuous inputs, Nonlinear Analysis: Hybrid Systems (2016), http://dx.doi.org/10.1016/j.nahs.2016.04.005

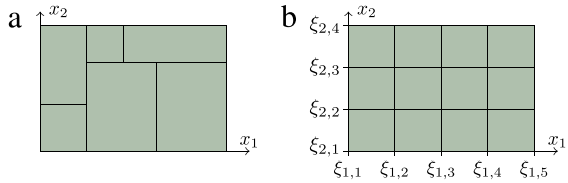


**Fig. 2.** Stylized depiction of (a) an interval partition, and (b) a gridded partition.

## 4. Abstraction of mixed monotone systems

We have seen that for mixed monotone systems, an overapproximation of the one-step reachable set from the set $[x_1, x_2]_{\mathcal{X}}$ under a disturbance input from the set $[d_1, d_2]_{\mathcal{D}}$ can be computed by evaluating the decomposition function $f$ at only two particular points. We now exploit Theorem 1 and Corollary 1 and present an efficient algorithm for computing a *symbolic model*, or *finite state abstraction* of a mixed monotone system. For systems of the form (2), we wish to *verify* that a certain property, usually given in a temporal logic, holds under all possible disturbance inputs. For switched systems of the form (3), we wish to *synthesize* a mode selection policy such that the resulting system satisfies a given property. In Section 5, we extend this idea to systems where the input takes values in a continuous set.

### 4.1. Finite state abstraction

Now we introduce a partition of the domain $\mathcal{X}$ by intervals and construct a finite state abstraction from the partition.[1] We discuss systems of the form (3), since (2) is a special case.

Assume system (3) is mixed monotone with respect to $\{\leq_m\}_{m\in\mathcal{M}}$ and $\leq_{\mathcal{D}}$. Furthermore, assume $\mathcal{D}$ is representable as the union of intervals:

$$\mathcal{D} = \bigcup_{\ell=1}^{L} \mathcal{D}^{\ell} \tag{32}$$

where $\mathcal{D}^{\ell} \triangleq [d_1^{\ell}, d_2^{\ell}]_{\mathcal{D}}$ for $d_1^{\ell} \leq_{\mathcal{D}} d_2^{\ell}$.

**Definition 5** (*Interval Partition*)**.** The collection $\{\mathcal{I}_q\}_{q\in\mathcal{Q}}$ for finite set $\mathcal{Q}$ with $\mathcal{I}_q \subseteq \mathcal{X}$ for all $q \in \mathcal{Q}$ is an *interval partition* of $\mathcal{X}$ if:

1. For all $m \in \mathcal{M}$ and for all $q \in \mathcal{Q}$, there exist $x_1^{q,m}, x_2^{q,m} \in \mathcal{X}$ satisfying $x_1^{q,m} \leq_m x_2^{q,m}$ and $\mathcal{I}_q = [x_1^{q,m}, x_2^{q,m}]_m$,
2. $\bigcup_{q\in\mathcal{Q}} \mathcal{I}_q = \mathcal{X}$,
3. $\mathrm{int}(\mathcal{I}_q) \cap \mathrm{int}(\mathcal{I}_{q'}) = \emptyset$ for all $q, q' \in \mathcal{Q}$, $q \neq q'$.

In other words, $\{\mathcal{I}_q\}_{q\in\mathcal{Q}}$ is an *interval partition* of $\mathcal{X}$ if the sets $\mathcal{I}_q$, $q \in \mathcal{Q}$ partition $\mathcal{X}$ and each $\mathcal{I}_q$ is representable as an interval of $\mathcal{X}$ with respect to each order $\leq_m$. For notational convenience, in defining a partition we ignore the set of measure zero where intervals overlap as is done in, *e.g.*, [21]. However, as noted in [21,22], if the dynamics are such that trajectories remain within the boundaries after a certain time, one should account for such sets. Grid-based abstractions are common in the literature; see, for example, [24,30–32]. Here we provide a novel technique in Theorem 2 for computing reachable states using mixed monotonicity. Additionally, we consider a nonuniform interval partition characterized by a general partial order on the domain, and thus the partition does not necessarily arise by gridding along each dimension. We consider grid-based partitions as a special case.

If each $\leq_m$ is an orthant order, then a partition $\{\mathcal{I}_q\}_{q\in\mathcal{Q}}$ with each $\mathcal{I}_q$ a box constitutes an interval partition of $\mathcal{X} \subset \mathbb{R}^n$, motivating the following special case.

**Definition 6** (*Gridded Interval Partition*)**.** An interval partition is a *gridded* interval partition, or simply *gridded partition*, if each $\leq_m$ is an orthant order and for each $i \in \{1, \ldots, n\}$ there exist $N_i > 0$ and $\{\xi_{i,1}, \ldots, \xi_{i,N_i+1}\}$ such that $\mathcal{Q} = \prod_{i=1}^{n}\{1, \ldots, N_i\}$ and for each $q = (\iota_1, \ldots, \iota_n) \in \mathcal{Q}$, we have $\mathcal{I}_q = \prod_{i=1}^{n}[\xi_{i,\iota_i}, \xi_{i,\iota_i+1}]_{\mathbb{R}_{\geq 0}}$.

Fig. 2 shows schematic depictions of two interval partitions, one of which is a gridded partition.

When clear from context, we refer to the index set $\mathcal{Q}$ itself as an interval partition with the associated notation as above. From such a partition, we readily construct a finite state abstraction of the resulting dynamics.

[1] If we wish to consider unbounded domains $\mathcal{X}$ for the system (72), we may then construct a finite state abstraction on a bounded subset and include an *overflow* symbol in the finite state abstraction as in [20,24].

Please cite this article in press as: S. Coogan, M. Arcak, Finite abstraction of mixed monotone systems with discrete and continuous inputs, Nonlinear Analysis: Hybrid Systems (2016), http://dx.doi.org/10.1016/j.nahs.2016.04.005

Consider a map $\delta : \mathcal{Q} \times \mathcal{M} \to 2^{\mathcal{Q}}$ that satisfies the following property:

$$\begin{aligned} &\text{If } \exists x \in \mathcal{I}_q,\ \exists d \in \mathcal{D} \quad \text{such that } F_m(x, d) \in \mathcal{I}_{q'} \\ &\text{Then } q' \in \delta(q, m). \end{aligned} \tag{33}$$

The map $\delta$ includes a transition from $q$ to $q'$ whenever it is possible for the state $x$ to transition from the interval $\mathcal{I}_q$ to $\mathcal{I}_{q'}$.

**Definition 7** (*Interval Finite State Abstraction*)**.** An *interval finite state abstraction* or simply *abstraction* of system (3) is a tuple $\mathcal{T} = (\mathcal{Q}, \mathcal{M}, \delta)$ where $\mathcal{Q}$ is an interval partition of $\mathcal{X}$ and $\delta$ satisfies (33) for all $m \in \mathcal{M}$. We call $\delta$ a *transition function* and say $q' \in \mathcal{Q}$ is a *successor* of $q$ in mode $m$ if $q' \in \delta(q, m)$.

$\mathcal{T}$ is a nondeterministic transition system, *i.e.*, $\delta(q, m)$ is, in general, not a singleton set. The nondeterminism arises because $\mathcal{T}$ abstracts an entire set of states into one state or *symbol*, and the transitions account for all possible states in the symbol as well as the disturbance. Nonetheless, $\mathcal{T}$ is a transition system that *overapproximates* the dynamics (3), that is, for every trajectory $x[t]$ satisfying $x[t+1] = F_{m[t]}(x[t], d[t])$ such that $m[t] \in \mathcal{M}$ and $d[t] \in \mathcal{D}$ for all $t$, there exists $q[t]$ such that $x[t] \in \mathcal{I}_{q[t]}$ and $q[t+1] \in \delta(q[t], m[t])$ for all $t$.

Computing a transition function $\delta$ that is useful in practice is a serious difficulty for standard abstraction approaches. Many existing results apply only to linear or piecewise linear systems, and even in this case, scale poorly with the state space. For example, the polytope-based computations suggested in [21] require computing $F_m$ at a number of points that scales exponentially with the dimension of the state space and disturbance space. By exploiting the mixed monotonicity properties of system (3), we propose an efficient method for computing an abstraction that requires evaluating $f_m$ at only $2L$ points for each $q \in Q$ and $m \in \mathcal{M}$.

**Theorem 2.** *Consider a mixed monotone system (3) with interval partition $\mathcal{Q}$. Let $\delta : \mathcal{Q} \times \mathcal{M} \to 2^{\mathcal{Q}}$ be given by $q' \in \delta(q, m)$ if and only if*

$$\exists \ell : [f_m(x_1^{q,m}, d_1^\ell, x_2^{q,m}, d_2^\ell), f_m(x_2^{q,m}, d_2^\ell, x_1^{q,m}, d_1^\ell)]_{\mathcal{X}} \cap [x_1^{q',m}, x_2^{q',m}]_{\mathcal{X}} \neq \emptyset. \tag{34}$$

*Then $\mathcal{T} = (\mathcal{Q}, \mathcal{M}, \delta)$ is a finite state abstraction of (3).*

**Proof.** Consider $x \in \mathcal{I}_q$ and $d \in \mathcal{D}$ such that $x' = F_m(x, d) \in \mathcal{I}_{q'} = [x_1^{q',m}, x_2^{q',m}]_m$. Let $\ell \in \{1, \ldots, L\}$ be such that $d \in \mathcal{D}^\ell$. From Theorem 1, it holds that also

$$x' \in [f_m(x_1^{q,m}, d_1^\ell, x_2^{q,m}, d_2^\ell), f_m(x_2^{q,m}, d_2^\ell, x_1^{q,m}, d_1^\ell)]_m,$$

which implies $q' \in \delta(q, m)$, thus $\delta$ satisfies (33). □

**Corollary 2.** *Consider a monotone system (3) with interval partition $\mathcal{Q}$. Let $\delta : \mathcal{Q} \times \mathcal{M} \to 2^{\mathcal{Q}}$ be given by $q' \in \delta(q, m)$ if and only if*

$$\exists \ell : [F_m(x_1^{q,m}, d_1^\ell), F_m(x_2^{q,m}, d_2^\ell)] \cap [x_1^{q',m}, x_2^{q',m}] \neq \emptyset. \tag{35}$$

*Then $\mathcal{T} = (\mathcal{Q}, \mathcal{M}, \delta)$ is a finite state abstraction of (3).*

We summarize the algorithm implied by Theorem 2 in Algorithm 1. For systems of the form (2), we interpret $\mathcal{M}$ as a singleton and proceed as above. We then notationally omit $\mathcal{M}$ and instead write $\mathcal{T} = (\mathcal{Q}, \delta)$, and $\delta(q) \subset \mathcal{Q}$.

### 4.2. Computing successor states

Theorem 2 and Corollary 2 provided a method for overapproximating the one-step reachable set of an interval. How do we identify the successor states from this overapproximation? Lemma 1 provides an efficient method for determining if two intervals overlap.

**Lemma 1.** *Consider $[\alpha_1, \beta_1]_{\mathcal{X}}$ and $[\alpha_2, \beta_2]_{\mathcal{X}}$ for $\alpha_1, \beta_1 \in \mathcal{X}$ and $\alpha_2, \beta_2 \in \mathcal{X}$. Then $[\alpha_1, \beta_1]_{\mathcal{X}} \cap [\alpha_2, \beta_2]_{\mathcal{X}} \neq \emptyset$ implies $\alpha_1 \leq_{\mathcal{X}} \beta_2$ and $\alpha_2 \leq_{\mathcal{X}} \beta_1$.*

**Proof.** Choosing $x \in [\alpha_1, \beta_1]_{\mathcal{X}} \cap [\alpha_2, \beta_2]_{\mathcal{X}}$, the lemma follows from transitivity of $\leq_{\mathcal{X}}$. □

For special types of partitions, however, more efficient methods exist for computing the successor states. In particular, when each $\leq_m$ is an orthant order and $\mathcal{Q}$ is a gridded partition, computing successor states is accomplished by considering each coordinate separately. This algorithm scales linearly with $\sum_{i=1}^n N_i$. When all $N_i$ are approximately the same, the algorithm scales approximately linearly with $n$.

Please cite this article in press as: S. Coogan, M. Arcak, Finite abstraction of mixed monotone systems with discrete and continuous inputs, Nonlinear Analysis: Hybrid Systems (2016), http://dx.doi.org/10.1016/j.nahs.2016.04.005

**function** FiniteStateAbtraction(*system*, $\mathcal{D}$, $\mathcal{Q}$) **returns** $\mathcal{T}$
  **inputs:** *system*, a mixed monotone system (3) with domain $\mathcal{X}$, modes $\mathcal{M}$ and decomposition functions $\{f_m\}_{m\in\mathcal{M}}$
    $\mathcal{D}$, the disturbance set $\mathcal{D} = \cup_{\ell=1}^{L}\mathcal{D}^\ell$ with $\mathcal{D}^\ell \triangleq [d_1^\ell, d_2^\ell]_{\mathcal{D}}$ for $d_1^\ell \preceq_{\mathcal{D}} d_2^\ell$
    $\mathcal{Q}$, an interval partition $\mathcal{X}$
  **for each** $m \in \mathcal{M}$ **do**
    **for each** $q \in \mathcal{Q}$ **do**
      $\delta(q, m) := \emptyset$
      **for** $\ell := 1$ **to** $L$ **do**
        $y_1 := f_m(x_1^{q,m}, d_1^\ell, x_2^{q,m}, d_2^\ell)$
        $y_2 := f_m(x_2^{q,m}, d_2^\ell, x_1^{q,m}, d_1^\ell)$
        $\mathcal{Q}' :=$ ComputeSuccessors$(y_1, y_2, \mathcal{Q})$
        $\delta(q, m) := \delta(q, m) \cup \mathcal{Q}'$
      **end for**
    **end for**
  **end for**
  **return** $\mathcal{T} := (\mathcal{Q}, \mathcal{M}, \delta)$ ▷ abstraction of (3)
**end function**

Algorithm 1: Computing an interval finite state abstraction of (3).

### 4.3. Computational requirements

We now address the computational requirements of the proposed algorithms. Determining $\delta(q, m)$ requires first evaluating the decomposition function $f_m$ at $2L$ points where $L$ is the number of boxes constituting the disturbance set $\mathcal{D}$. For each $\ell = 1, \ldots, L$, the corresponding pair of evaluations of $f_m$ is then used to determine successor states representing an overapproximation of the reachable set from $q$. For general interval partitions, this requires $2|\mathcal{Q}|$ order comparisons of vectors in $\mathbb{R}^n$, and each comparison scales linearly with $n$. For gridded partitions, determining successor states requires $\sum_{i=1}^{n} N_i$ scalar order comparisons.

Thus, computing $\delta$ scales linearly with $|\mathcal{M}|$ and linearly with $L$. For general interval partitions, the computation further scales quadratically with $|\mathcal{Q}|$ and linearly with $n$, and for gridded partitions, it scales linearly with $|\mathcal{Q}|$ and linearly with $\sum_{i=1}^{n} N_i$. In contrast, computing successor states from a polyhedral region as in, *e.g.*, [21] requires polyhedral computations that scale exponentially in both $n$ and $p$ [33]. The approximation of reachable sets in [20] also relies on polyhedral operations and scales exponentially with $n$. Above, we have assumed that $f_m$ requires constant computation time. This is reasonable in some cases, such as the case study in Section 6.2 where intrinsic sparsity of traffic networks implies that the required computation time of $f_m$ does not increase with $n$ or $p$. However, in other cases, the complexity of evaluating $f_m$ must be taken into account.

Several abstraction approaches in the literature require a single evaluation of the system dynamics per discrete partition per input. These approaches rely on some known relationship between nearby trajectories in the form of, *e.g.*, incremental stability [19], incremental forward completeness [30], a discrepancy function [34], or a growth bound [35]. Often, such relationships are conservative as they must hold between any pair of trajectories throughout the space and thus require a fine gridding of the state space. Our proposed method, which also requires a constant number of evaluations per discrete state, exploits additional structure in the dynamics to reduce the conservatism.

We further remark that $|\mathcal{Q}|$ typically increases exponentially with $n$. However, this dependence can be mitigated *via* various techniques such as interval partitions that incorporate domain specific knowledge. For example, the authors of [24] consider monotone systems that converge to a low-dimensional manifold, and suggest a methodology for abstracting the low dimensional manifold while retaining the intrinsic high dimensional dynamics. Future research will investigate related techniques for mixed monotone systems.

### 4.4. Control synthesis from abstraction

Finding a controller for a finite state abstraction amounts to finding a sequence of modes such that, regardless of how the nondeterminism in the transition map $\delta$ is resolved, every possible execution of the finite state transition system satisfies a given objective. When this objective is given in, *e.g.*, linear temporal logic (LTL), the synthesis algorithm may be interpreted as a game between the adversarial nondeterminism and the controller [36,37]. Central to these approaches are fixed point algorithms which determine states that are guaranteed to be reached and states that are guaranteed to be avoided. Similar fixed point algorithms for safety and reachability games appear in [11]. In these cases, the result is a state feedback controller with memory. That is, we interpret the controller itself as another finite transition system with a set of memory states. At

each time step $t$, the controller prescribes a particular mode $m[t] \in \mathcal{M}$ as a function of the controller's memory state and the current finite state $q \in \mathcal{Q}$.

The finite state abstraction $\mathcal{T}$ overapproximates the behavior of the original system in the sense that, for any mode $m$, the transitions of $\mathcal{T}$ account for at least all possible executions of the underlying dynamical system. This overapproximating property is sufficient for control synthesis for properties that must hold for any possible execution of the closed loop system (e.g., properties given in LTL [15]) including safety and reachability.

Various relationships have been proposed in the literature to capture this sufficiency for control synthesis. For example, it is straightforward to show that the finite state abstraction computed above is *alternatingly simulated* [11] by the underlying dynamical system (3). Furthermore, since the set of modes are the same for the abstraction and the underlying system (3), it is additionally the case that the abstraction is a *feedback refinement* [35] of the system (3). For our problem setup, both notions capture the fact that for each mode $m \in \mathcal{M}$, each possible transition of (3) from some state $x \in \mathcal{I}_q$ is matched by a transition from state $q$ in the abstraction. Alternating simulations and feedback refinements are both sufficient for control synthesis as shown in [11, Proposition 8.7] and [35, Theorem V.4], that is, a controller synthesized for the abstraction can be suitably applied to the original system with the same guarantees.

The final step is specifying how to *refine* the controller developed for the abstraction so that it is applicable to the original system. For our problem setup, this is straightforward; at each time $t$, if the state of the system is $x[t] \in \mathcal{I}_q$, then whichever mode $m$ is prescribed at finite state $q$ by the controller is instead applied to the original system. The above mentioned properties of the abstraction ensure that the closed loop trajectories of the dynamical system satisfy the control objective.

We remark that controller refinement from abstractions differs substantially depending on the abstraction approach; see, *e.g.*, [38,39,32] for alternative approaches in the literature and [35] for a discussion of the intricate relationship between feedback refinements and alternating simulation relations.

## 5. Continuous inputs

In this section, we extend finite state abstractions to the case when the system is driven by a continuous input rather than by a set of discrete input modes. Instead of (3), we consider

$$x^+ = \mathcal{F}(x, d, u) \tag{36}$$

where, as before, $x \in \mathcal{X} \subset \mathbb{R}^n$, $d \in \mathcal{D} \subset \mathbb{R}^p$, and $\mathcal{F} : \mathcal{X} \times \mathcal{D} \times \mathbb{R}^r \to \mathcal{X}$. In the present setting, $u \in \mathcal{U}(x) \subseteq \mathbb{R}^r$ is the control input with the state-based input constraint $\mathcal{U}(x)$, that is, at state $x$ only inputs from the set $\mathcal{U}(x)$ are admissible.

In Section 5.1, we first extend the notion of finite state abstraction to this continuous input case; the development proceeds abstractly and does not require mixed monotonicity assumptions on $\mathcal{F}$. In Section 5.2 we propose a computational procedure for the case when $\mathcal{F}(x, d, u) = F(x, d) + Bu$ with $F(x, d)$ mixed monotone and $B$ a matrix.

### 5.1. Modified definition of finite state abstraction

Let $\{\mathcal{I}_q\}_{q\in\mathcal{Q}}$ be an interval partition of $\mathcal{X}$. We again seek a finite state abstraction that suitably overapproximates the dynamics of (36) based on the partition $\mathcal{Q}$. One approach is to grid the input space in order to obtain a discrete number of inputs. Instead, we follow the line of reasoning in, *e.g.*, [21], which suggests partitioning the set of inputs to obtain a finite state abstraction where we use the underlying dynamics to obtain the partition. Here, we compute a finite collection of subsets of the input space at each $q \in \mathcal{Q}$ such that each subset induces a fixed set of possible transitions in $\mathcal{Q}$. To this end, let

$$\mathcal{U}^q = \{u \in \mathbb{R}^r \mid u \in \mathcal{U}(x)\ \forall x \in \mathcal{I}_q\} \tag{37}$$

be the set of inputs admissible for any state $x$ in interval $\mathcal{I}_q$. For each $\mathcal{Q}' \subseteq \mathcal{Q}$, let

$$\mathcal{U}^{q\to\mathcal{Q}'} \triangleq \{u \in \mathcal{U}^q \mid \forall x \in \mathcal{I}_q, \forall d \in \mathcal{D}\ \mathcal{F}(x, d, u) \in \cup_{q'\in\mathcal{Q}'} \mathcal{I}_{q'}\} \tag{38}$$

denote the set of inputs under which the system (36) is guaranteed to only make a transition to some partition $q' \in \mathcal{Q}'$ when initialized anywhere in partition $q$. In general, $\mathcal{U}^{q\to\mathcal{Q}'}$ may be empty, however we assume $\mathcal{U}^{q\to\mathcal{Q}} \neq \emptyset$ to ensure the finite abstraction defined below is nonblocking, that is, there always exists some input inducing some transition.

From (38) we construct a finite state abstraction as follows. For each $q \in \mathcal{Q}$, let $\mathbb{Q}_q \subseteq 2^{\mathcal{Q}}$ be some collection of subsets of interval indices such that for each $\mathcal{Q}' \in \mathbb{Q}_q$, we have nonempty $\underline{\mathcal{U}}^{q\to\mathcal{Q}'}$ satisfying $\underline{\mathcal{U}}^{q\to\mathcal{Q}'} \subseteq \mathcal{U}^{q\to\mathcal{Q}'}$. We allow $\underline{\mathcal{U}}^{q\to\mathcal{Q}'}$ to be a subset of $\mathcal{U}^{q\to\mathcal{Q}'}$ as the latter may be difficult to compute where an underapproximation is not. Define

$$\mathbb{U}_q = \{\underline{\mathcal{U}}^{q\to\mathcal{Q}'} \mid \mathcal{Q}' \in \mathbb{Q}_q\} \tag{39}$$

as the input *symbols*[2] available at state $q$. Construct the transition map $\delta^c : \{(q, \underline{\mathcal{U}}) \mid q \in \mathcal{Q},\ \underline{\mathcal{U}} \in \mathbb{U}_q\} \to 2^{\mathcal{Q}}$ by defining

$$\delta^c(q, \underline{\mathcal{U}}^{q\to\mathcal{Q}'}) \triangleq \mathcal{Q}' \quad \forall q \in \mathcal{Q}, \mathcal{Q}' \subseteq \mathcal{Q} \text{ s.t. } \underline{\mathcal{U}}^{q\to\mathcal{Q}'} \in \mathbb{U}_q. \tag{40}$$

[2] We freely conflate notation for subsets of $\mathcal{U}$ and input symbols (that is, inputs to the finite state abstraction) by using, *e.g.*, $\underline{\mathcal{U}}^{q\to\mathcal{Q}'}$ for both.

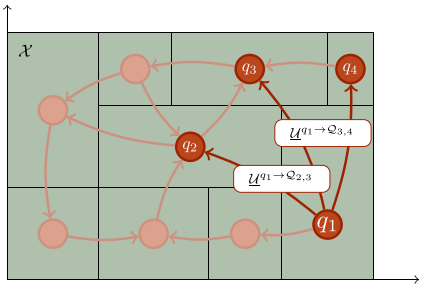


**Fig. 3.** Input symbols of the finite state abstraction are subsets of the continuous input space $\mathbb{R}^r$ that induce a certain set of transitions. In the figure, $\mathcal{Q}_{2,3} = \{q_2, q_3\}$ and $\mathcal{Q}_{3,4} = \{q_3, q_4\}$, and $\underline{\mathcal{U}}^{q_1\to\mathcal{Q}_{2,3}}$, $\underline{\mathcal{U}}^{q_1\to\mathcal{Q}_{3,4}}$ are input symbols to the finite state abstraction such that $\delta^c(q_1, \underline{\mathcal{U}}^{q_1\to\mathcal{Q}_{2,3}}) = \mathcal{Q}_{2,3}$ and $\delta^c(q_1, \underline{\mathcal{U}}^{q_1\to\mathcal{Q}_{3,4}}) = \mathcal{Q}_{3,4}$.

Let $\mathbb{U} = \cup_{q\in\mathcal{Q}} \mathbb{U}_q$. Then $\delta^c$ satisfies the following property, directly analogous to (33):

$$\begin{aligned} \forall \underline{\mathcal{U}}^{q\to\mathcal{Q}'} \in \mathbb{U}, \quad & \forall u \in \underline{\mathcal{U}}^{q\to\mathcal{Q}'} : \\ & \text{If } \exists x \in \mathcal{I}_q,\ \exists d \in \mathcal{D} \text{ such that } \mathcal{F}(x, d, u) \in \mathcal{I}_{q'} \\ & \text{Then } q' \in \delta^c(q, \underline{\mathcal{U}}^{q\to\mathcal{Q}'}) = \mathcal{Q}'. \end{aligned} \tag{41}$$

**Definition 7b** (*Interval Finite State Abstraction, Modified*)**.** An *interval finite state abstraction* or simply *abstraction* of system (36) is a tuple $\mathcal{T} = (\mathcal{Q}, \mathbb{U}, \delta^c)$ where $\mathcal{Q}$ is an interval partition of $\mathcal{X}$ and $\delta^c$ satisfies (41) for all $\underline{\mathcal{U}}^{q\to\mathcal{Q}'} \in \mathbb{U}_q$ with $\mathbb{U} = \cup_{q\in\mathcal{Q}} \mathbb{U}_q$ a finite set of input symbols.

We first note that input symbols to the finite abstraction are identified with subsets of $\mathbb{R}^r$, the input space. This is because if we apply as input to the original system (36) any $u \in \underline{\mathcal{U}}^{q\to\mathcal{Q}'}$ for some $\underline{\mathcal{U}}^{q\to\mathcal{Q}'}$, we have that necessarily $x^+ \in \cup_{q'\in\mathcal{Q}'} \mathcal{I}_{q'}$ for any $x \in \mathcal{I}_q$ by (41), thus we can choose any input $u$ from the set (*i.e.*, symbol) $\mathcal{U}^{q\to\mathcal{Q}'}$ to realize a transition to some state in $\mathcal{Q}'$ as illustrated in Fig. 3. This freedom may be exploited in, *e.g.*, control algorithms that minimize some cost subject to an abstraction-based synthesis procedure. Furthermore, by considering explicitly the transitions induced by the underlying dynamics in computing $\mathcal{U}^{q\to\mathcal{Q}'}$, we reduce spurious transitions as compared to, *e.g.*, a naive gridding of the input space.

Additionally, when the finite state abstraction is to be used in synthesis for satisfying a linear time property (*e.g.*, a linear temporal logic (LTL) formula), we need not compute all possible inputs $\mathcal{U}^{q\to\mathcal{Q}'}$. For example, if $\mathcal{U}^{q\to\mathcal{Q}'} \neq \emptyset$ for some $\mathcal{Q}'$, then we need not consider $\mathcal{U}^{q\to\mathcal{Q}''}$ for any $\mathcal{Q}'' \supseteq \mathcal{Q}'$ for the purposes of control synthesis since $\mathcal{U}^{q\to\mathcal{Q}''} \supseteq \mathcal{U}^{q\to\mathcal{Q}'}$. As an alternative interpretation from the automata theoretic perspective, in obtaining a successor state $q'$ from $\delta^c(q, \mathcal{U}^{q\to\mathcal{Q}''}) = \mathcal{Q}''$, the (assumed adversarial) nondeterminism may always choose $q' \in \mathcal{Q}'$ and no additional control authority is obtained by choosing $\mathcal{U}^{q\to\mathcal{Q}''}$ as the input symbol.

### 5.2. Computing a finite set of inputs

In this section, we address computing approximations $\underline{\mathcal{U}}^{q\to\mathcal{Q}'}$ of the sets $\mathcal{U}^{q\to\mathcal{Q}'}$ defined in (38) in order to obtain a finite state abstraction as in Definition 7b. We specialize to the case

$$x^+ = F(x, d) + Bu \tag{42}$$

where $F(x, d)$ is a mixed monotone map and $B$ is a $n \times r$ matrix. We assume all partial orders are orthant orders, and then without loss of generality, we take $\preceq_{\mathcal{X}} = \preceq_+$ and $\preceq_{\mathcal{D}} = \preceq_+$ and equip the input space $\mathbb{R}^r$ with the partial order $\preceq_{\mathcal{U}} = \preceq_+$. For convenience, we consider $\mathcal{D} = [d_1, d_2]_+$ for some $d_1 \preceq_+ d_2$; that is, $L = 1$ in (32).

We assume that each $\mathcal{U}^q$ as in (37) is a polytope, that is, there exist $M > 0$, $H_q \in \mathbb{R}^{M\times r}$ and $K_q \in \mathbb{R}^M$ such that

$$\mathcal{U}^q = \{u \mid H_q u \preceq_+ K_q\}. \tag{43}$$

For example, suppose that for each $x \in \mathcal{X}$, $\mathcal{U}(x) \subset \mathbb{R}^r$ is a polytope dependent on $x$, specifically, we assume there exist $M > 0$ and $H \in \mathbb{R}^{M\times r}$, $G \in \mathbb{R}^{M\times n}$, $K \in \mathbb{R}^M$ such that

$$\mathcal{U}(x) = \{u \in \mathbb{R}^r \mid Hu \preceq_+ Gx + K\}. \tag{44}$$

The affine state-based constraint (44) is relevant for a number of physically-motivated applications, *e.g.*, ramp metering control as described in [40]. For this case, the set $\mathcal{U}^q$ in (37) is efficiently computed from $H, G, K$ and the extremal points of $\mathcal{I}_q$ and has the form (43). Specifically, let $G_+, G_- \in \mathbb{R}^{M\times n}$ be given elementwise as

$$G_+[i,j] \triangleq \begin{cases} G[i,j] & \text{if } G[i,j] \geq 0 \\ 0 & \text{else} \end{cases} \qquad G_-[i,j] \triangleq \begin{cases} G[i,j] & \text{if } G[i,j] \leq 0 \\ 0 & \text{else.} \end{cases} \tag{45}$$

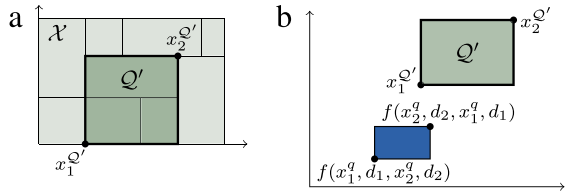


**Fig. 4.** (a) A contiguous block $\mathcal{Q}'$ of boxes is represented as one interval defined by the extreme points $x_1^{\mathcal{Q}'}, x_2^{\mathcal{Q}'}$. In this case, the input symbol of the finite state abstraction for transitioning to $\mathcal{Q}'$ is computed using the control remainder (50)–(51). Otherwise, the input symbol is computed by removing inputs that may cause transitions outside $\mathcal{Q}'$ as in (60). (b) The additive control input updates the state of the system beyond $[f_1(x_1^q, d_1, x_2^q, d_2), f_2(x_1^q, d_1, x_2^q, d_2)]_+$, the overapproximated one-step reachable set in the absence of input. The control remainder guarantees a transition within $\mathcal{Q}'$ while the set difference (54) considers the possibility of a transition.

Then $\mathcal{U}^q$ is given by

$$\mathcal{U}^q \triangleq \{u \mid Hu \leq_+ G_+x_1^q + G_-x_2^q + K\} \tag{46}$$
$$= \{u \mid H_q u \leq_+ K_q\} \tag{47}$$

for $H_q \triangleq H$ and $K_q \triangleq G_+x_1^q + G_-x_2^q + K$.

The assumption (43) is motivated in particular by systems for which $r$ is small and thus polyhedral operations are not excessively expensive in $\mathbb{R}^r$; mixed monotonicity of $F(x, d)$ allows us to still exploit the computational efficiencies developed in Section 4 for reachability computations in $\mathbb{R}^n$.

We now compute subsets of $\mathcal{U}^q$ that induce a certain set of transitions. Consider $q \in \mathcal{Q}$ with $\mathcal{I}_q = [x_1^q, x_2^q]_+$ and $\mathcal{Q}' \subseteq \mathcal{Q}$ and let $\mathcal{I}_{\mathcal{Q}'} = [x_1^{\mathcal{Q}'}, x_2^{\mathcal{Q}'}]_+$ be an interval such that

$$\mathcal{I}_{\mathcal{Q}'} \subseteq \cup_{q' \in \mathcal{Q}'} \mathcal{I}_{q'} \tag{48}$$

as illustrated in Fig. 4(a) where $\mathcal{I}_{\mathcal{Q}'}$ equals $\cup_{q' \in \mathcal{Q}'} \mathcal{I}_{q'}$ because $\mathcal{Q}'$ is a contiguous block of boxes that is itself a box. Equality in (48) can always be ensured for such subsets of gridded partitions as defined in Definition 6 with $\mathcal{Q} = \prod_{i=1}^n \{1, \ldots, N_i\}$. In particular, consider

$$\mathcal{Q}' = \prod_{i=1}^n \{\underline{\iota}_i, \underline{\iota}_i + 1, \ldots, \bar{\iota}_i\} \subseteq \mathcal{Q} \tag{49}$$

with $\underline{\iota}_i, \bar{\iota}_i \in \{1, \ldots, N_i\}$, $\underline{\iota}_i \leq \bar{\iota}_i$ for all $i = 1, \ldots, n$. Then we take $x_1^{\mathcal{Q}'} = (\underline{\iota}_1, \ldots, \underline{\iota}_n)$ and $x_2^{\mathcal{Q}'} = (\bar{\iota}_1, \ldots, \bar{\iota}_n)$ so that $\mathcal{I}_{\mathcal{Q}'} = \cup_{q' \in \mathcal{Q}'} \mathcal{I}_{q'}$.

Define the *control remainder*

$$\mathcal{X}^{q \to \mathcal{Q}'} \triangleq [x_1^{\mathcal{Q}'} - f(x_1^q, d_1, x_2^q, d_2), x_2^{\mathcal{Q}'} - f(x_2^q, d_2, x_1^q, d_1)]_+ \tag{50}$$
$$\subseteq \{x' \in \mathbb{R}^n \mid \forall x \in \mathcal{I}_q, \forall d \in \mathcal{D}\ F(x, d) + x' \in \mathcal{I}_{\mathcal{Q}'}\}. \tag{51}$$

The set $\mathcal{X}^{q \to \mathcal{Q}'}$ may be empty, in particular, $\mathcal{X}^{q \to \mathcal{Q}'} = \emptyset$ if $x_2^{\mathcal{Q}'} - x_1^{\mathcal{Q}'} \leq_+ f(x_2^q, d_2, x_1^q, d_1) - f(x_1^q, d_1, x_2^q, d_2)$. The control remainder is illustrated in Fig. 4(b). We let

$$\underline{\mathcal{U}}^{q \to \mathcal{Q}'} \triangleq \{u \in \mathcal{U}^q \mid Bu \in \mathcal{X}^{q \to \mathcal{Q}'}\} \tag{52}$$
$$= \{u \in \mathbb{R}^p \mid H_q u \leq_+ K_q \text{ and } Bu \leq_+ (x_2^{\mathcal{Q}'} - f(x_2^q, d_2, x_1^q, d_1))$$
$$\text{and } -Bu \leq_+ -(x_1^{\mathcal{Q}'} - f(x_1^q, d_1, x_2^q, d_2))\}. \tag{53}$$

The set $\underline{\mathcal{U}}^{q \to \mathcal{Q}'}$ as defined in (53) is a polytope and is efficiently computed using polyhedral operations [41]. Moreover, $\underline{\mathcal{U}}^{q \to \mathcal{Q}'} \subseteq \mathcal{U}^{q \to \mathcal{Q}'}$ and thus is suitable as an allowable input symbol to the abstraction according to (39).

Thus far, we have assumed availability of an interval $\mathcal{I}_{\mathcal{Q}'}$ satisfying (48), motivated by $\mathcal{Q}'$ that is a block of adjacent boxes. The advantages of doing so is the efficient computation (53).

We now adapt a technique from [21] that does not require $\mathcal{I}_{\mathcal{Q}'}$ satisfying (48). To that end, consider $q, q' \in \mathcal{Q}$ and define[3]

$$\mathcal{X}^{q' \ominus q} \triangleq [x_1^{q'} - f(x_2^q, d_2, x_1^q, d_1), x_2^{q'} - f(x_1^q, d_1, x_2^q, d_2)]_+ \tag{54}$$
$$\supseteq \{x' \in \mathbb{R}^n \mid \exists x \in \mathcal{I}_q, \exists d \in \mathcal{D} \text{ s.t. } F(x, d) + x' \in \mathcal{I}_{q'}\}. \tag{55}$$

[3] The set $\mathcal{X}^{q' \ominus q}$ is the Minkowski difference of $\mathcal{I}_{q'}$ and $[f(x_1^q, d_1, x_2^q, d_2), f(x_2^q, d_2, x_1^q, d_1)]_+$.

Please cite this article in press as: S. Coogan, M. Arcak, Finite abstraction of mixed monotone systems with discrete and continuous inputs, Nonlinear Analysis: Hybrid Systems (2016), http://dx.doi.org/10.1016/j.nahs.2016.04.005

Note in particular the exchange of universal quantifiers for existential quantifiers and the exchange of $\subseteq$ for $\supseteq$ in comparing (50)–(51) with (54)–(55). Further define

$$\mathcal{U}^{q'\ominus q} \triangleq \{u \in \mathcal{U}^q \mid Bu \in \mathcal{X}^{q'\ominus q}\} \tag{56}$$

so that $\mathcal{U}^{q'\ominus q}$ contains the set of inputs for which the system may transition to partition $q'$ from some state in partition $q$. Analogous to (52), we have

$$\begin{aligned}\underline{\mathcal{U}}^{q'\ominus q} = \{u \in \mathbb{R}^p \mid & H_q u \leq_+ K_q \text{ and } Bu \leq_+ (x_2^{q'} - f(x_1^q, d_1, x_2^q, d_2)) \\ & \text{and } -Bu \leq_+ -(x_1^{q'} - f(x_2^q, d_2, x_1^q, d_1))\}.\end{aligned} \tag{57}$$

Consider $\underline{\mathcal{U}}^{q\to\mathcal{Q}}$ satisfying

$$\underline{\mathcal{U}}^{q\to\mathcal{Q}} \subseteq \{u \in \mathcal{U}^q \mid x' + Bu \in \mathcal{X}\ \forall x' \in [f(x_1^q, d_1, x_2^q, d_2), f(x_2^q, d_2, x_1^q, d_1)]_+\} \tag{58}$$

$$\subseteq \mathcal{U}^{q\to\mathcal{Q}} \tag{59}$$

so that $\underline{\mathcal{U}}^{q\to\mathcal{Q}}$ is a subset of inputs that ensures the state of the system does not leave the domain from partition $q$. For example, we may compute $\underline{\mathcal{U}}^{q\to\mathcal{Q}}$ according to the approximation (52)–(53) with $\mathcal{Q}' = \mathcal{Q}$. We assume $\underline{\mathcal{U}}^{q\to\mathcal{Q}}$ is not empty, *i.e.*, there exists some input that is guaranteed to keep the system within the domain $\mathcal{X}$. By removing from $\underline{\mathcal{U}}^{q\to\mathcal{Q}}$ possible inputs that may take the system outside $\mathcal{Q}'$, we obtain a set of allowable inputs:

**Proposition 3.**

$$\widehat{\mathcal{U}}^{q\to\mathcal{Q}'} \triangleq \underline{\mathcal{U}}^{q\to\mathcal{Q}} \setminus \bigcup_{q''\in\mathcal{Q}\setminus\mathcal{Q}'} \underline{\mathcal{U}}^{q''\ominus q} \tag{60}$$

$$\subseteq \mathcal{U}^{q\to\mathcal{Q}'}. \tag{61}$$

**Proof.** Consider $u \in \widehat{\mathcal{U}}^{q\to\mathcal{Q}'}$, $x \in \mathcal{I}_q$, $d \in \mathcal{D}$, and let $x' = F(x, d) + Bu$. It follows that $x \in \mathcal{X}$ by (58). Let $q'$ be such that $x' \in \mathcal{I}_{q'}$. Suppose $q' \in \mathcal{Q}\setminus\mathcal{Q}'$. But then $Bu \in \mathcal{X}^{q'\ominus q}$ by (54)–(55) so that $u \in \underline{\mathcal{U}}^{q'\ominus q}$, contradicting (60). Therefore $q' \in \mathcal{Q}'$ and thus $u \in \mathcal{U}^{q\to\mathcal{Q}'}$. □

We now have two methods for obtaining a set of inputs that guarantees a transition to $\mathcal{Q}'$. We conclude with the following:

**Proposition 4.** *Consider $\mathcal{Q}' \subseteq \mathcal{Q}$ and $\mathcal{I}_{\mathcal{Q}'} = [x_1^{\mathcal{Q}'}, x_2^{\mathcal{Q}'}]_+$ such that $\mathcal{I}_{\mathcal{Q}'} = \cup_{q'\in\mathcal{Q}'} \mathcal{I}_{q'}$. Then*

$$\widehat{\mathcal{U}}^{q\to\mathcal{Q}'} \subseteq \underline{\mathcal{U}}^{q\to\mathcal{Q}'}. \tag{62}$$

**Proof.** Let $u \in \widehat{\mathcal{U}}^{q\to\mathcal{Q}'}$. Trivially $u \in \mathcal{U}^q$ so that $H_q u \leq_+ K_q$. We consider the possibility that $u \notin \underline{\mathcal{U}}^{q\to\mathcal{Q}'}$ and, in light of (53), first suppose $Bu \not\leq_+ x_2^{\mathcal{Q}'} - f(x_2^q, d_2, x_1^q, d_1)$. Let $\tilde{x} = Bu + f(x_2^q, d_2, x_1^q, d_1)$. It follows that $\tilde{x} \notin \mathcal{I}_{\mathcal{Q}'}$ since $\tilde{x} \not\leq_+ x_2^{\mathcal{Q}'}$. Since $u \in \underline{\mathcal{U}}^{q\to\mathcal{Q}}$, we have $\tilde{x} \in \mathcal{X}$ from (58), and thus there exists $q'' \notin \mathcal{Q}'$ such that $\tilde{x} \in \mathcal{I}_{q''}$, *i.e.*, let $q''$ satisfy $x_1^{q''} \leq_+ \tilde{x} \leq_+ x_2^{q''}$. Then

$$x_1^{q''} - f(x_2^q, d_2, x_1^q, d_1) \leq_+ Bu \leq_+ x_2^{q''} - f(x_2^q, d_2, x_1^q, d_1), \tag{63}$$

and since $-f(x_2^q, d_2, x_1^q, d_1) \leq_+ -f(x_1^q, d_1, x_2^q, d_2)$, we have $Bu \in \mathcal{X}^{q''\ominus q}$ and thus $u \in \mathcal{U}^{q''\ominus q}$, a contradiction. A symmetric argument demonstrates a contradiction for the case $-Bu \not\leq_+ -(x_1^{\mathcal{Q}'} - f(x_1^q, d_1, x_2^q, d_2))$, thus $u \in \underline{\mathcal{U}}^{q\to\mathcal{Q}'}$. □

From Proposition 4, we see that whenever $\mathcal{Q}'$ is a block of contiguous boxes equivalent to a larger box $\mathcal{I}_{\mathcal{Q}'}$, we should approximate $\mathcal{U}^{q\to\mathcal{Q}'}$ with $\underline{\mathcal{U}}^{q\to\mathcal{Q}'}$ as it is simpler to compute and is less conservative compared to $\widehat{\mathcal{U}}^{q\to\mathcal{Q}'}$. In practice, we need only consider $\mathcal{Q}'$ such that

$$\mathcal{Q}' \subseteq \textsc{Post}(\mathcal{I}_q, \mathcal{U}^q) \tag{64}$$

$$\triangleq \{q' \mid \exists x' \in [f(x_1^q, d_1, x_2^q, d_2), f(x_2^q, d_2, x_1^q, d_1)]_+ \exists u \in \mathcal{U}^q \text{ s.t. } x' + u \in \mathcal{I}_{q'}\}, \tag{65}$$

that is, $\mathcal{Q}'$ is a subset of the states that are reachable from $\mathcal{I}_q$ under an input from $\mathcal{U}^q$. The set $\textsc{Post}(\mathcal{I}_q, \mathcal{U}^q)$ is easily computed using polyhedral operations or efficiently overapproximated. For example, if $y_1, y_2 \in \mathbb{R}^n$ satisfy $\{Bu \mid \exists x \text{ s.t. } u \in \mathcal{U}(x)\} \subseteq [y_1, y_2]_+$, then $\textsc{Post}(\mathcal{I}_q, \mathcal{U}^q) \subseteq [f(x_1^q, d_1, x_2^q, d_2) + y_1, f(x_2^q, d_2, x_1^q, d_1) + y_2]_+$.

Computing $\underline{\mathcal{U}}^{q\to\mathcal{Q}'}$ and $\underline{\mathcal{U}}^{q'\ominus q}$ require polyhedral operations that scale exponentially in $r$, the dimension of the input space. Computing $\underline{\mathcal{U}}^{q\to\mathcal{Q}'}$ requires only one such polyhedral operation per $\mathcal{Q}'$ satisfying (64). To compute $\widehat{\mathcal{U}}^{q\to\mathcal{Q}'}$ requires computing $\underline{\mathcal{U}}^{q'\ominus q}$ for each $q'' \in \mathcal{Q}\setminus\mathcal{Q}'$. In practice, we need only consider $q'' \in \textsc{Post}(\mathcal{I}_q, \mathcal{U}^q)$ in (60).

Please cite this article in press as: S. Coogan, M. Arcak, Finite abstraction of mixed monotone systems with discrete and continuous inputs, Nonlinear Analysis: Hybrid Systems (2016), http://dx.doi.org/10.1016/j.nahs.2016.04.005

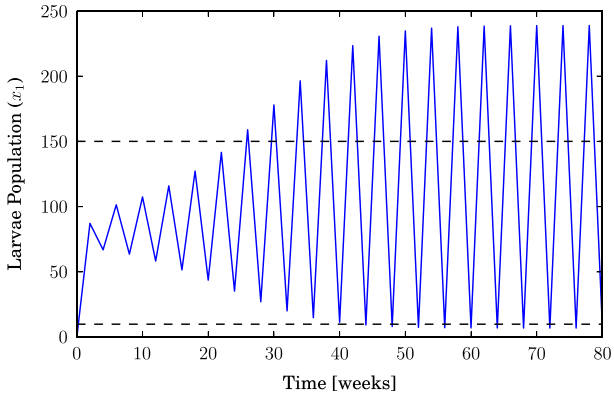


**Fig. 5.** Sample trajectory of the insect population model (66)–(67), plotting $x_1$ over time when the system is initialized at $(x_1, x_2, x_3) = (0, 0, 300)$. The trajectory satisfies (69).

## 6. Case studies

### 6.1. Verifying oscillations in insect population dynamics

We consider the following model from [42] for the population dynamics of the flour beetle *Tribolium castaneum*:

$$x^+ = A(x)x, \quad x = (x_1, x_2, x_3) \in \mathbb{R}^3, \tag{66}$$

$$A(x) = \begin{pmatrix} 0 & 0 & b\exp(-c_{e\ell}x_1 - c_{ea}x_3) \\ p & 0 & 0 \\ 0 & \exp(-c_{pa}x_3) & q \end{pmatrix}, \tag{67}$$

where $x_1, x_2$, and $x_3$ represent populations of the insect at various stages of life (larvae, pupae, and adults, respectively), and $p, q \in (0, 1]$ are probabilities of survival. The exponential nonlinearities are the result of cannibalism of eggs and pupae. The dynamics are mixed monotone with $f(x, d, y, e) = A(y)x$ where $\leq_{\mathcal{X}} = \leq_+$.

Using parameters from [42], we let $b = 7.88$, $c_{ea} = 0.011$, $c_{e\ell} = 0.014$, $p = 0.839$, $q = 0.5$, and $c_{pa} = 0.0047$, with a time step of 2 weeks. We first note that the domain

$$\mathcal{X} = [0, (265, 225, 450)]_+ \tag{68}$$

is invariant. This follows because $bx_3\exp(-c_{ea}x_3) \leq 265$ for all $x_3 \geq 0$ and, thus, $x_1 \leq 265$ is invariant, from which $x_2 \leq p \cdot 265 \leq 225$. Since $x_3^+ \leq x_2 + qx_3$, we conclude that $x_3 \leq 225/(1-q) = 450$ is invariant.

For certain sets of parameters, the dynamics (66)–(67) induce oscillations in the number of larvae—a phenomenon documented in controlled laboratory experiments [42]. We wish to verify the following LTL formula which is a consequence of this oscillatory behavior:

$$\Box\Big(\big((x_1 \leq 10) \wedge (x_3 \geq 40)\big) \rightarrow \Diamond(x_1 \geq 150)\Big). \tag{69}$$

In words, "if the larvae population ($x_1$) reduces to a small number or zero and the adult population ($x_3$) is not too small, then the larvae population will eventually reach a large population size".

We partition the state space into 2376 intervals using a gridded partition. The grid points along the $x_1$, $x_2$, and $x_3$ dimensions are, respectively,

$$\{0, 10, 20, 40, 50, 60, 80, 100, 125, 150, 175, 200, 275\},$$
$$\{0, 20, 40, 50, 60, 80, 100, 125, 150, 175, 200, 225\}, \quad \text{and}$$
$$\{0, 10, 20, 40, 50, 60, 80, 100, 125, 150, 175, 200, 225, 250, 275, 300, 325, 350, 450\}.$$

Computing the finite state abstraction takes less than one second on a standard personal computer. We remove 14 self transitions that are *spurious*, *i.e.*, do not correspond to actual trajectories of the system [21,43]; see [23, Algorithm 4]. Checking the model with SPIN [44] took 103 s, and we verify that (69) is satisfied. Fig. 5 shows a sample trajectory of the population dynamics initialized at $(x_1, x_2, x_3) = (0, 0, 300)$. We see that the larvae population does not reach the desired population 150 immediately, but it does so eventually around week 26.

Please cite this article in press as: S. Coogan, M. Arcak, Finite abstraction of mixed monotone systems with discrete and continuous inputs, Nonlinear Analysis: Hybrid Systems (2016), http://dx.doi.org/10.1016/j.nahs.2016.04.005

### 6.2. Synthesizing control laws for traffic networks

We next synthesize a traffic signal control policy for a network of signalized intersections. We consider a discrete-time model of traffic flow where each road *link* contains a queue of vehicles waiting to proceed through an intersection. Each intersection signal actuates a subset of its queues at a given time step, and the vehicles in actuated queues are allowed to flow to downstream links if there is available space. In [29], we considered a similar model for which the dynamics are piecewise affine. Here, we build on this prior result and consider a more general nonlinear model with the help of the theory developed in Sections 3 and 4.

We consider a network of $\mathcal{L}$ *links* and a set $\mathcal{V}$ of *signalized intersections*. We assume each link $\ell \in \mathcal{L}$ has a queue of size $x_\ell \in [0, x_\ell^{\text{crit}}]$ representing the number of vehicles on the link where $x_\ell^{\text{crit}} > 0$ is the capacity of link $\ell \in \mathcal{L}$ so that $\mathcal{X} = \prod_{\ell\in\mathcal{L}}[0, x_\ell^{\text{crit}}]$. By allowing $x_\ell$ to be continuous, we adopt a *fluid* model of traffic flow.

For $\ell \in \mathcal{L}$, let $\eta(\ell) \in \mathcal{V}$ denote the head node of link $\ell$ and let $\tau(\ell) \in \mathcal{V} \cup \emptyset$ denote the tail node. A link $\ell$ with $\tau(\ell) = \emptyset$ serves as an entry-point into the network, and we assume $\eta(\ell) \neq \tau(\ell)$ for all $\ell \in \mathcal{L}$ (*i.e.*, no self-loops). Link $k \neq \ell$ is *upstream* of link $\ell$ if $\eta(k) = \tau(\ell)$, *downstream* of link $\ell$ if $\tau(k) = \eta(\ell)$, and *adjacent to* link $\ell$ if $\tau(k) = \tau(\ell)$. Roads exiting the traffic network are not modeled explicitly. For each $v \in \mathcal{V}$, define

$$\mathcal{L}_v^{\text{in}} = \{\ell \mid \eta(\ell) = v\}, \qquad \mathcal{L}_v^{\text{out}} = \{\ell \mid \tau(\ell) = v\}. \tag{70}$$

For simplicity of notation, we assume each intersection $v \in \mathcal{V}$ has two possible states actuating either "East–West" (EW) incoming links or "North–South" (NS) incoming links. Thus, we have the partition $\mathcal{L} = \mathcal{L}^{\text{EW}} \cup \mathcal{L}^{\text{NS}}$, $\mathcal{L}^{\text{EW}} \cap \mathcal{L}^{\text{NS}} = \emptyset$. At each junction $v \in \mathcal{V}$, we define the signal variable $m_v \in \{0, 1\}$ as follows:

$$m_v = \begin{cases} 1 & \text{if links } \mathcal{L}_v^{\text{in}} \cap \mathcal{L}^{\text{EW}} \text{ are actuated} \\ 0 & \text{if links } \mathcal{L}_v^{\text{in}} \cap \mathcal{L}^{\text{NS}} \text{ are actuated.} \end{cases} \tag{71}$$

Let $m = \{m_v\}_{v\in\mathcal{V}}$ so that $\mathcal{M} = \{0, 1\}^{\mathcal{V}}$. When a link $\ell$ is actuated, the *turn ratio* $\beta_{\ell k}$ denotes the fraction of vehicles exiting link $\ell$ that is routed to link $k$. It follows that $\beta_{\ell k} \neq 0$ only if $\eta(\ell) = \tau(k)$ and

$$\sum_{k\in\mathcal{L}_{\eta(\ell)}^{\text{out}}} \beta_{\ell k} \leq 1. \tag{72}$$

Strict inequality in (72) implies that a fraction of vehicles on link $\ell$ are routed off the network via unmodeled roads.

Each link $\ell \in \mathcal{L}$ possesses a *demand* function $\Phi_\ell^{\text{out}} : [0, x_\ell^{\text{crit}}] \to \mathbb{R}$ that gives the number of vehicles wishing to flow downstream in one time step and a *supply* function $\Phi_\ell^{\text{in}} : [0, x_\ell^{\text{crit}}] \to \mathbb{R}$ that gives the available road space for incoming upstream vehicles in one time step. Thus, $\Phi_\ell^{\text{out}}$ is an increasing function and $\Phi_\ell^{\text{in}}$ is a decreasing function of queue length. In this example, we let

$$\Phi_\ell^{\text{out}}(x_\ell) = c_\ell(1 - \exp(-x_\ell/c_\ell)) \tag{73}$$

$$\Phi_\ell^{\text{in}}(x_\ell) = w_\ell(x_\ell^{\text{crit}} - x_\ell) \tag{74}$$

where $c_\ell > 0$ is a *saturation rate* and $0 < w_\ell < 1$ scales the available queue capacity to account for, *e.g.*, vehicles still traveling on the link and not enqueue. This demand–supply approach to vehicular traffic flow is rooted in the Cell Transmission Model [45].

Movement of vehicles among link queues is governed by mass-conservation laws and the state of the signalized intersections. When a link is *actuated*, a maximum of $\Phi_\ell^{\text{out}}(x_\ell)$ vehicles are allowed to flow from link $\ell$ to links $\mathcal{L}_{\eta(\ell)}^{\text{out}}$ per time step. We let $\alpha_{\ell k}$ denote the fraction of link $k$'s supply available to link $\ell$. Since only incoming EW or NS links are actuated in each time step, we have

$$\sum_{\ell\in\mathcal{L}_{\tau(k)}^{\text{in}}\cap\mathcal{L}^{\text{EW}}} \alpha_{\ell k} = \sum_{\ell\in\mathcal{L}_{\tau(k)}^{\text{in}}\cap\mathcal{L}^{\text{NS}}} \alpha_{\ell k} = 1 \tag{75}$$

for all $k \in \mathcal{L}$. It then follows that the dynamics on link $\ell$ are given by

$$x_\ell^+ = F_m^\ell(x, d) \tag{76}$$

$$\triangleq x_\ell - f_\ell^{\text{out}}(x, m) + \sum_{j\in\mathcal{L}_{\tau(\ell)}^{\text{in}}} \beta_{j\ell} f_j^{\text{out}}(x, m) + d_\ell \tag{77}$$

where

$$f_\ell^{\text{out}}(x, m) = s_\ell(m) \cdot \min\left\{\Phi_\ell^{\text{out}}(x_\ell),\ \min_{\substack{k \text{ s.t.} \\ \beta_{\ell k}\neq 0}} \frac{\alpha_{\ell k}}{\beta_{\ell k}} \Phi_k^{\text{in}}(x_k)\right\} \tag{78}$$

$$s_\ell(m) = \begin{cases} m_{\eta(\ell)} & \text{if } \ell \in \mathcal{L}^{\text{EW}} \\ 1 - m_{\eta(\ell)} & \text{if } \ell \in \mathcal{L}^{\text{NS}}. \end{cases} \tag{79}$$

**Assumption 1.** For all $\ell \in \mathcal{L}$ and all $k$ upstream of $\ell$,

$$\exp\left(\frac{-1}{c_\ell}\left(x_\ell^{\text{crit}} - \frac{\beta_{k\ell}}{w_\ell \alpha_{k\ell}} c_k\right)\right) \leq 1 - w_\ell. \tag{80}$$

Assumption 1 ensures that an increase in $x_i$ does not lead to a decrease in $x_i^+$. This assumption is mild because (80) is satisfied for small enough $c_\ell$ and $c_k$, and these parameters decrease for shorter time steps; indeed, violation of the assumption would indicate that the chosen time step is too large to accurately capture the queue dynamics.

**Lemma 2.** *Assumption 1 ensures that* $\frac{\partial F_m^\ell}{\partial x_\ell}(x, d) \geq 0$ *for all m whenever the partial derivative exists.*[4]

**Proof.** We have

$$\frac{\partial F_m^\ell}{\partial x_\ell}(x, d) = 1 - \frac{\partial f_\ell^{\text{out}}}{\partial x_\ell}(x_\ell, m) + \sum_{j \in \mathcal{L}_{\tau(\ell)}^{\text{in}}} \beta_{j\ell} \frac{\partial f_j^{\text{out}}}{\partial x_\ell}(x, m). \tag{81}$$

Note that $\frac{\partial f_\ell^{\text{out}}}{\partial x_\ell}(x_\ell, m) \leq 1$. Furthermore, $\frac{\partial f_j^{\text{out}}}{\partial x_\ell}(x, m) \neq 0$ only if $s_j(m) = 1$ and $\frac{\alpha_{j\ell}}{\beta_{j\ell}} \Phi_\ell^{\text{in}}(x_\ell)$ is the minimizer in (78). As $\Phi_j^{\text{out}}(x_j) \leq c_j$, the latter condition can only occur if

$$c_j \geq \frac{\alpha_{j\ell}}{\beta_{j\ell}} w_\ell (x_\ell^{\text{crit}} - x_\ell) \iff x_\ell \geq x_\ell^{\text{crit}} - \frac{\beta_{j\ell}}{w_\ell \alpha_{j\ell}} c_j. \tag{82}$$

It then follows that $\sum_{j \in \mathcal{L}_{\tau(\ell)}^{\text{in}}} \beta_{j\ell} \frac{\partial f_j^{\text{out}}}{\partial x_\ell}(x, m) < 0$ only if there exists $j \in \mathcal{L}_{\tau(\ell)}^{\text{in}}$ such that the inequalities in (82) hold. But this implies $\frac{\partial f_\ell^{\text{out}}}{\partial x_\ell}(x_\ell, m) \leq 1 - w_\ell$ by Assumption 1 and the fact that $\exp(-\frac{1}{c_\ell} x_\ell)$ decreases in $x_\ell$. Furthermore, $\sum_{j \in \mathcal{L}_{\tau(\ell)}^{\text{in}}} \beta_{j\ell} \frac{\partial f_j^{\text{out}}}{\partial x_\ell}(x, m) \geq -w_\ell$, and we thus conclude that $\frac{\partial F_m^\ell}{\partial x_\ell}(x, d) \geq 0$. □

We make the natural choice $\leq_{\mathcal{X}} = \leq_+$ and $\leq_{\mathcal{D}} = \leq_+$.

**Proposition 5.** *The traffic model (76)–(77) is mixed monotone.*

In the proof below, we show that (76)–(77) satisfies the conditions of Proposition 1. In [29], we refer to this stronger condition as *componentwise monotonicity*.

**Proof.** We show for all $\ell, k \in \mathcal{L}$, and all $m \in \mathcal{M}$,

$$\frac{\partial F_m^\ell}{\partial x_k}(x, d) \leq 0 \quad \text{if } \tau(k) = \tau(\ell), k \neq \ell \tag{83}$$

$$\frac{\partial F_m^\ell}{\partial x_k}(x, d) \geq 0 \quad \text{if } k = \ell \text{ or } \tau(k) \neq \tau(\ell) \tag{84}$$

$$\frac{\partial F_m^\ell}{\partial d_k}(x, d) = \begin{cases} 1 & \text{if } k = \ell \\ 0 & \text{if } k \neq \ell. \end{cases} \tag{85}$$

This implies that the conditions of Propositions 1 are satisfied, specifically, we take $s_j = 1$ if and only if $\tau(j) = \tau(i), j \neq i$ and $\sigma_j = 0$ for all $j$ in (7)–(8). Note that (85) follows immediately from (77). We now show (83)–(84) by considering 4 exhaustive cases:

(Case 1, $\tau(k) = \tau(\ell), k \neq \ell$). We have that

$$\frac{\partial F_m^\ell}{\partial x_k}(x, d) = \sum_{j \in \mathcal{L}_{\tau(\ell)}^{\text{in}}} \beta_{j\ell} \frac{\partial f_j^{\text{out}}}{\partial x_k}(x, m). \tag{86}$$

Since $\frac{\partial f_j^{\text{out}}}{\partial x_k} \in \{0, \frac{\alpha_{jk}}{\beta_{jk}} \frac{d\Phi_k^{\text{in}}}{dx_k}\}$ and $\Phi_k^{\text{in}}$ is decreasing, we have $\frac{\partial F_m^\ell}{\partial x_k}(x, d) \leq 0$.

(Case 2, $\eta(k) = \tau(\ell)$ or $\tau(k) = \eta(\ell)$). We have that $\frac{\partial F_m^\ell}{\partial x_k}(x, d) \in \{0, -\frac{\partial f_\ell^{\text{out}}}{\partial x_k}, \beta_{k\ell} \frac{\partial f_k^{\text{out}}}{\partial x_k}, -\frac{\partial f_\ell^{\text{out}}}{\partial x_k} + \beta_{k\ell} \frac{\partial f_k^{\text{out}}}{\partial x_k}\}$ where the second possibility occurs only if $\tau(k) = \eta(\ell)$, the third occurs only if $\eta(k) = \tau(\ell)$, and the fourth occurs only if $\tau(k) = \eta(\ell)$ and

[4] The minimization in (78) implies that some partial derivatives do not exist on a set of measure zero. However, as noted above, the results developed in this paper still apply.

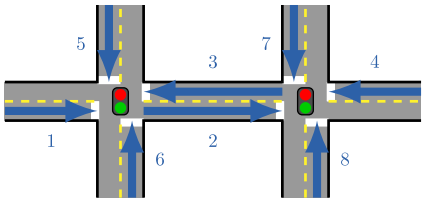


**Fig. 6.** A traffic network with two signalized intersections and 8 links. The blue links represent queues of vehicles. The leftmost signal actuates links 1 and 3 simultaneously, or links 5 and 6 simultaneously. Likewise, the rightmost signal actuates links 2 and 4 or 7 and 8 simultaneously. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$\eta(k) = \tau(\ell)$. We have $\partial f_k^{\text{out}}/\partial x_k \geq 0$ since $\Phi_k^{\text{out}}$ is increasing and $\frac{\partial f_\ell^{\text{out}}}{\partial x_k} \in \{0, \frac{\alpha_{\ell k}}{\beta_{\ell k}} \frac{d\Phi_k^{\text{in}}}{dx_k}\} \leq 0$ since $\Phi_k^{\text{in}}$ is decreasing, thus $\frac{\partial F_m^\ell}{\partial x_\ell}(x, d) \geq 0$.

(Case 3, $k = \ell$). $\frac{\partial F_m^\ell}{\partial x_\ell}(x, d) \geq 0$ by Lemma 2.

(Case 4, else). Trivially, $\frac{\partial F_m^\ell}{\partial x_\ell}(x, d) = 0$. □

The decomposition function is as constructed in the proof of Proposition 1.

Consider the traffic network shown in Fig. 6 consisting of two signalized intersections and eight links. We have $\mathcal{L}^{\text{EW}} = \{1, 2, 3, 4\}$ and $\mathcal{L}^{\text{NS}} = \{5, 6, 7, 8\}$. The leftmost signal actuates the EW links 1 and 3 simultaneously, or the NS links 5 and 6 simultaneously, and similarly for the rightmost signal. We take the time step to be 15 s and assume $c_1 = c_2 = c_3 = c_4 = 20$, $c_5 = c_6 = c_7 = c_8 = 5$, $x_1^{\text{crit}} = x_4^{\text{crit}} = 50$, $x_2^{\text{crit}} = x_3^{\text{crit}} = 60$, $x_5^{\text{crit}} = x_6^{\text{crit}} = x_7^{\text{crit}} = x_8^{\text{crit}} = 40$, $w_\ell = 0.75$ for all $\ell$, $\beta_{12} = \beta_{43} = \beta_{52} = \beta_{62} = \beta_{73} = \beta_{83} = 0.5$, $\alpha_{52} = \alpha_{62} = \alpha_{73} = \alpha_{83} = 0.5$, $\alpha_{12} = 1$, and $\alpha_{43} = 1$. For the disturbance input, we assume that at each time step, up to 7 vehicles join each of the queues on links 1 and 3, or up to 8 vehicles join each of the queues on links 5 and 6, or up to 8 vehicles join each of the queues on links 7 and 8, thus $\mathcal{D}$ is the union of three hyperrectangles.

We partition the domain of the traffic network, representing the state of all queues, into 3600 boxes using a gridded partition; the grid points for links 1 and 4 are {0, 30, 40, 50}, the grid points for links 2 and 3 are {0, 20, 30, 40, 50, 60}, and the grid points for links 5, 6, 7, and 8 are {0, 20, 40}. Using the mixed monotonicity properties of the dynamics, we obtain a finite state abstraction of the dynamics in 43.8 s on a Macbook Pro laptop using software written in Python. To allow specifications on the sequence of inputs, we augment the transition system with the current and previous control inputs, see [29] for details. The abstraction thus contains a total of 57,600 states and 16.9 million transitions.

Next, we wish to find a controller that satisfies the specification:

> "Infinitely often, the cross streets on links 5 and 6 are actuated, AND infinitely often, the cross streets on links 7 and 8 are actuated, AND eventually, the queue lengths on links 2 and 3 are each less than 40 vehicles and remain so for all future time, AND whenever the queue on link 1 exceeds 40 vehicles, it eventually is less than 30 vehicles, AND whenever the queue on link 4 exceeds 40 vehicles, it eventually is less than 30 vehicles".

The above specification can be expressed in linear temporal logic and encoded in a deterministic *Rabin automaton* [15] with 46 states. By solving a Rabin game, we construct a controller that is guaranteed to satisfy the specification. For synthesis, we use a modified version of the `CONPAS2` software package [21] which implements in MATLAB the algorithm proposed in [37] and required 2.04 h for computation. In Fig. 7, we plot an example trajectory of the system where we assume the maximum number of allowed vehicles enters the network in each time step. We see in the figure that the trajectory satisfies the above specification.

## 7. Conclusions

We have efficiently computed finite state abstractions for mixed monotone discrete-time systems. Mixed monotonicity is a general property encompassing many practical systems and provides a powerful tool for analysis and control. The primary feature that permits efficient abstraction is overapproximation of reachable sets by evaluating a decomposition function at two points. Future research will investigate using mixed monotonicity to reduce the number of intervals required to establish an effective partition of the state space.

## Acknowledgment

This work was supported in part by the National Science Foundation under Award 1446145.

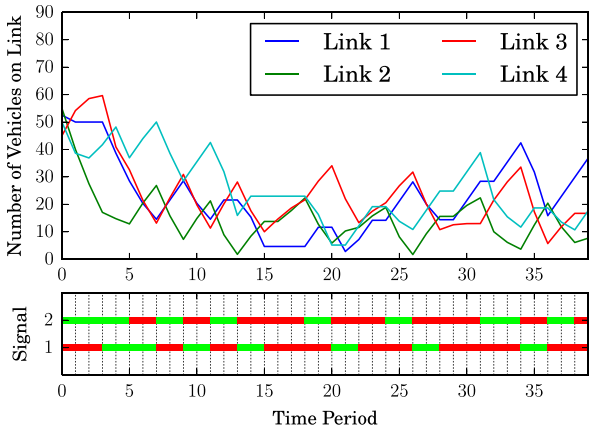


**Fig. 7.** An example trajectory of the traffic network for links 1, 2, 3, and 4. Signal 1 (resp. 2) is the leftmost (resp. rightmost) signal in Fig. 6. The trajectory satisfies the given specification. In the lower plot, green (resp. red) indicates that EW (resp. NS) links are actuated. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

## References

[1] M.W. Hirsch, Systems of differential equations that are competitive or cooperative II: Convergence almost everywhere, SIAM J. Math. Anal. 16 (3) (1985) 423–439.
[2] H.L. Smith, Monotone Dynamical Systems: An Introduction to the Theory of Competitive and Cooperative Systems, American Mathematical Society, 1995.
[3] M. Hirsch, H. Smith, Monotone maps: a review, J. Difference Equ. Appl. 11 (4–5) (2005) 379–398.
[4] D. Angeli, E. Sontag, Monotone control systems, IEEE Trans. Automat. Control 48 (10) (2003) 1684–1698.
[5] J. Gouzé, K. Hadeler, Monotone flows and order intervals, Nonlinear World 1 (1994) 23–34.
[6] H. Smith, The discrete dynamics of monotonically decomposable maps, J. Math. Biol. 53 (4) (2006) 747–758.
[7] H. Smith, Global stability for mixed monotone systems, J. Difference Equ. Appl. 14 (10–11) (2008) 1159–1164.
[8] G. Enciso, H. Smith, E. Sontag, Nonmonotone systems decomposable into monotone systems with negative feedback, J. Differential Equations 224 (1) (2006) 205–227.
[9] R. Alur, T.A. Henzinger, G. Lafferriere, G.J. Pappas, Discrete abstractions of hybrid systems, Proc. IEEE 88 (7) (2000) 971–984.
[10] A. Tiwari, G. Khanna, Series of abstractions for hybrid automata, in: Hybrid Systems: Computation and Control, Springer, 2002, pp. 465–478.
[11] P. Tabuada, Verification and Control of Hybrid Systems: A Symbolic Approach, Springer, 2009.
[12] T. Wongpiromsarn, U. Topcu, R. Murray, Receding horizon temporal logic planning, IEEE Trans. Automat. Control 57 (2012) 2817–2830.
[13] J. Liu, N. Ozay, U. Topcu, R. Murray, Synthesis of reactive switching protocols from temporal logic specifications, IEEE Trans. Automat. Control 58 (2013) 1771–1785.
[14] E.M. Clarke, O. Grumberg, D.A. Peled, Model Checking, MIT Press, 1999.
[15] C. Baier, J. Katoen, Principles of Model Checking, MIT Press, 2008.
[16] T.A. Henzinger, P.W. Kopke, A. Puri, P. Varaiya, What's decidable about hybrid automata? J. Comput. System Sci. 57 (1) (1998) 94–124.
[17] P. Tabuada, G. Pappas, Linear time logic control of discrete-time linear systems, IEEE Trans. Automat. Control 51 (12) (2006) 1862–1877.
[18] A. Girard, G.J. Pappas, Approximation metrics for discrete and continuous systems, IEEE Trans. Automat. Control 52 (5) (2007) 782–798.
[19] A. Girard, G. Pola, P. Tabuada, Approximately bisimilar symbolic models for incrementally stable switched systems, IEEE Trans. Automat. Control 55 (1) (2010) 116–126.
[20] G. Reissig, Computing abstractions of nonlinear systems, IEEE Trans. Automat. Control 56 (11) (2011) 2583–2598.
[21] B. Yordanov, J. Tůmová, I. Černá, J. Barnat, C. Belta, Temporal logic control of discrete-time piecewise affine systems, IEEE Trans. Automat. Control 57 (6) (2012) 1491–1504.
[22] D. Adzkiya, B. De Schutter, A. Abate, Finite abstractions of max-plus-linear systems, IEEE Trans. Automat. Control 58 (2013) 3039–3053.
[23] S. Coogan, M. Arcak, Efficient finite abstraction of mixed monotone systems, in: Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control, 2015, pp. 58–67.
[24] T. Moor, J. Raisch, Abstraction based supervisory controller synthesis for high order monotone continuous systems, in: Modelling, Analysis, and Design of Hybrid Systems, Springer, 2002, pp. 247–265.
[25] D. Gromov, J. Raisch, Detecting and enforcing monotonicity for hybrid control systems synthesis, in: Proc. 2nd IFAC Conf. on Analysis and Design of Hybrid Systems, 2006, pp. 7–9.
[26] N. Ramdani, N. Meslem, Y. Candau, Computing reachable sets for uncertain nonlinear monotone systems, Nonlinear Anal. Hybrid Syst. 4 (2) (2010) 263–278.
[27] J.M. Cushing, An Introduction to Structured Population Dynamics, SIAM, 1998.
[28] M. Kulenovic, O. Merino, A global attractivity result for maps with invariant boxes, Discrete Contin. Dyn. Syst. Ser. B 6 (1) (2006) 97.
[29] S. Coogan, E.A. Gol, M. Arcak, C. Belta, Traffic network control from temporal logic specifications, IEEE Trans. Control Netw. Syst. (2015) http://dx.doi.org/10.1109/TCNS.2015.2428471. in press. arXiv:1408.1437.
[30] M. Zamani, G. Pola, M. Mazo, P. Tabuada, Symbolic models for nonlinear control systems without stability assumptions, IEEE Trans. Automat. Control 57 (2012) 1804–1809.
[31] G. Reissig, M. Rungger, Abstraction-based solution of optimal stopping problems under uncertainty, in: 2013 IEEE 52nd Annual Conference on Decision and Control, CDC, IEEE, 2013, pp. 3190–3196.
[32] J. Liu, N. Ozay, Abstraction, discretization, and robustness in temporal logic control of dynamical systems, in: Proceedings of the 17th International Conference on Hybrid Systems: Computation and Control, ACM, 2014, pp. 293–302.
[33] A. Kurzhanskiy, P. Varaiya, Computation of reach sets for dynamical systems, in: The Control Systems Handbook, second ed., CRC Press, 2010, Chapter 29.
[34] P. Duggirala, S. Mitra, M. Viswanathan, Verification of annotated models from executions, in: 2013 Proceedings of the International Conference on Embedded Software, EMSOFT, 2013, pp. 1–10.
[35] G. Reissig, A. Weber, M. Rungger, Feedback refinement relations for the synthesis of symbolic controllers, arXiv preprint, 2015. arXiv:1503.03715.
[36] N. Piterman, A. Pnueli, Faster solutions of Rabin and Streett games, in: 21st Annual IEEE Symposium on Logic in Computer Science, 2006, pp. 275–284.
[37] F. Horn, Streett games on finite graphs, in: Proc. 2nd Workshop Games in Design Verification, GDV, 2005.

Please cite this article in press as: S. Coogan, M. Arcak, Finite abstraction of mixed monotone systems with discrete and continuous inputs, Nonlinear Analysis: Hybrid Systems (2016), http://dx.doi.org/10.1016/j.nahs.2016.04.005

[38] E. Dallal, A. Colombo, D. DelVecchio, S. Lafortune, Supervisory control for collision avoidance in vehicular networks with imperfect measurements, in: 2013 IEEE 52nd Annual Conference on Decision and Control, CDC, IEEE, 2013, pp. 6298–6303.
[39] A. Girard, Low-complexity quantized switching controllers using approximate bisimulation, Nonlinear Anal. Hybrid Syst. 10 (2013) 34–44.
[40] S. Coogan, M. Arcak, Freeway traffic control from linear temporal logic specifications, in: Proceedings of the 5th ACM/IEEE International Conference on Cyber-Physical Systems,2014, pp. 36–47.
[41] M. Herceg, M. Kvasnica, C. Jones, M. Morari, Multi-Parametric Toolbox 3.0, in: Proceedings of the European Control Conference, Zürich, Switzerland, July 17–19 2013, pp. 502–510, http://control.ee.ethz.ch/~mpt.
[42] R. Costantino, J. Cushing, B. Dennis, R.A. Desharnais, Experimentally induced transitions in the dynamic behaviour of insect populations, Nature 375 (6528) (1995) 227–230.
[43] L. Grüne, F. Müller, Global optimal control of quantized systems, in: Proceedings of the 18th International Symposium on Mathematical Theory of Networks and Systems—MTNS2010, Budapest, Hungary, 2010, pp. 1231–1237.
[44] G.J. Holzmann, The model checker SPIN, IEEE Trans. Softw. Eng. 23 (1997) 279–295.
[45] C.F. Daganzo, The cell transmission model, part II: Network traffic, Transp. Res. B 29 (2) (1995) 79–93.

Please cite this article in press as: S. Coogan, M. Arcak, Finite abstraction of mixed monotone systems with discrete and continuous inputs, Nonlinear Analysis: Hybrid Systems (2016), http://dx.doi.org/10.1016/j.nahs.2016.04.005