

Formal Controller Synthesis from Specifications Given by Discrete-Time Hybrid Automata

Vladimir Sinyakov, Antoine Girard

► To cite this version:

Vladimir Sinyakov, Antoine Girard. Formal Controller Synthesis from Specifications Given by Discrete-Time Hybrid Automata. 2019. hal-02361404

HAL Id: hal-02361404

<https://hal.archives-ouvertes.fr/hal-02361404>

Preprint submitted on 13 Nov 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Formal Controller Synthesis from Specifications Given by Discrete-Time Hybrid Automata [★]

Vladimir Sinyakov ^a, Antoine Girard ^a,

^a*Laboratoire des Signaux et Systèmes (L2S), CNRS, CentraleSupélec, Université Paris-Sud, Université Paris-Saclay,
3, rue Joliot-Curie, 91192 Gif-sur-Yvette, cedex, France*

Abstract

This paper deals with formal controller synthesis for discrete-time dynamical systems. For a specification under the form of a discrete-time hybrid automaton, we aim at synthesizing controllers such that the trajectories of the closed-loop system are also trajectories of the hybrid automaton. We first show that the existence of an alternating simulation relation from the specification to the open-loop system is a necessary and sufficient condition for the existence of such controllers. Then, we propose an approach based on the use of symbolic (i.e. finite-state) abstractions of both the system and the specification. Effective computations are discussed for systems that are monotone and for specifications given by piecewise affine hybrid automata. We extend our approach to handle specifications with additional safety or reachability requirements. Finally, we illustrate our approach with examples from autonomous vehicle control.

Key words: Formal synthesis; alternating simulation; symbolic abstractions; hybrid automata

1 Introduction

Recent years saw a burst in research on formal methods in control theory (see e.g. [18,2] and the references therein). One of the key problems in this field is that of synthesizing automatically a controller for a dynamical system so that the closed-loop system has a certain desired behavior. The set of all accepted behaviors is referred to as the specification. The considered specifications can be complex and usually go beyond traditional stability properties. For example, they can be described by finite automata [5,14,6] or by linear temporal logic formulas [19,4,8,2]. In some other cases, the specification itself can be given under the form of a dynamical system: a trajectory is then accepted if it can be related in some sense to a trajectory of the specification system: see e.g. [18,13] where specifications are given by finite-state dynamical systems or [20,9] where both the system and the specification are given by linear systems.

When a dynamical system has an infinite number of states, a common approach in formal methods is to transition from the original infinite system to an approximating finite-state system called symbolic abstraction (see e.g. [15,21,3,11,7,16]). These two systems should be in a certain relation to justify rigorously this transition. Different kinds of relations, such as alternating simulation relations [18] or feedback refinement relations [16], were introduced to formalize whether one system can be controlled by the same or by a related control strategy as its symbolic abstraction.

This paper is mainly concerned with the problem of synthesizing controllers for discrete-time control systems from specifications given by discrete-time hybrid automata. The solution is provided under the form of a controllable region of the state space and a pair of controllers defining the control input and the active mode of the hybrid automata. The first contribution of the paper is to show that the existence of an alternating simulation relation from the specification to the open-loop system is not only a sufficient but also a necessary condition for the existence of a non-trivial solution. The result is constructive and controllers can be obtained from the alternating simulation relation.

We then present an approach based on the use of symbolic abstractions. Since both the system and the speci-

[★] This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 725144).

Email addresses:

Vladimir.Sinyakov@l2s.centralesupelec.fr (Vladimir Sinyakov), Antoine.Girard@l2s.centralesupelec.fr (Antoine Girard).

cation are given by infinite-state dynamical systems, our solution includes the computation of symbolic abstractions for both of them. While the proposed construction of the symbolic abstraction for the system resembles existing approaches in the literature [3,11,16], the proposed construction of a symbolic abstraction for the specification is, to the best of our knowledge, new. The effective computation of these symbolic abstractions is discussed for monotone systems and piecewise affine hybrid automata. We then show how controller synthesis can be performed by computing an alternating simulation relation between the two symbolic abstractions, using fixed-point computations. Afterwards, we propose two extensions of the main control problem with additional safety or reachability objectives. We show that similar abstraction-based approaches can be used, with only modifications to the discrete controller synthesis step. Finally, we illustrate our approach with examples taken from autonomous vehicle control applications: we first consider cruise control with collision avoidance, and then takeover maneuvers.

The paper is organized as follows. Section 2 formulates the main controller synthesis problem under consideration. Section 3 provides a characterization of the solution in terms of alternating simulation relations. Section 4 proposes a computational approach to synthesize automatically controllers using symbolic abstractions. Section 5 extends the main controller synthesis problem with additional safety or reachability requirements. Finally, Section 6 illustrates our approach with autonomous vehicle control examples.

Notations: For $x \in \mathbb{R}^n$, $\|x\|_\infty = \max_i |x_i|$ is the infinity norm; for a matrix $A \in \mathbb{R}^{n \times n}$, $\|A\|_\infty$ is the matrix norm induced by the infinity norm. The Hausdorff distance $h(A, B)$ between two sets $A, B \subseteq \mathbb{R}^n$ is defined by

$$h(A, B) = \max \left(\sup_{x \in A} \inf_{y \in B} \|x - y\|_\infty, \sup_{y \in B} \inf_{x \in A} \|x - y\|_\infty \right).$$

The inradius and the circumradius of a set $A \subseteq \mathbb{R}^n$ are defined respectively by

$$\begin{aligned} \rho^-(A) &= \sup_{x \in \mathbb{R}^n} (\sup \{r \mid x + rB \subseteq A\}), \\ \rho^+(A) &= \inf_{x \in \mathbb{R}^n} (\inf \{r \mid A \subseteq x + rB\}), \end{aligned}$$

where, in the previous expressions, B is the unit ball in the infinity norm. For two vectors $x, x' \in \mathbb{R}^n$, we denote $x \preceq x'$ if and only if $x_i \leq x'_i$, for all $1 \leq i \leq n$; given $\underline{x} \preceq \bar{x}$, we define the corresponding n -dimensional interval $[\underline{x}, \bar{x}] = \{x \in \mathbb{R}^n \mid \underline{x} \preceq x \preceq \bar{x}\}$. For a set-valued map $f: X \rightrightarrows Y$, its domain is defined as $\text{dom}(f) = \{x \in X \mid f(x) \neq \emptyset\}$.

2 Problem formulation

In this section, we start by introducing some preliminary definitions and then formulate the main controller synthesis problem under consideration.

2.1 Transition systems

In this paper, we consider transition systems to model in a common framework control systems and specifications. These are defined as follows [18]:

Definition 1 A transition system S is a tuple (X, U, Y, Δ, H) , where

- X is a set of states;
- U is a set of inputs;
- Y is a set of outputs;
- $\Delta: X \times U \rightrightarrows X$ is a set-valued transition map;
- $H: X \rightarrow Y$ is an output map.

An input $u \in U$ is called enabled at $x \in X$ if $\Delta(x, u) \neq \emptyset$. Let $\text{enab}_\Delta(x) \subseteq U$ denote the set of all inputs enabled at x . If $\text{enab}_\Delta(x) = \emptyset$ the state x is called blocking.

Definition 2 A sequence $(x_k, u_k)_{k=0}^K$, where $K \in \mathbb{N} \cup \{+\infty\}$, $x_k \in X$, $u_k \in U$, for $0 \leq k \leq K$, is called a trajectory of S if and only if $x_{k+1} \in \Delta(x_k, u_k)$ for $0 \leq k < K$. A trajectory $(x_k, u_k)_{k=0}^K$ is called maximal if either $K = +\infty$ or $\Delta(x_K, u_K) = \emptyset$, it is complete if $K = +\infty$.

2.2 Controller synthesis problem

In this paper, we consider a discrete-time control system S_1 of the form:

$$x_{k+1} \in F(x_k, u_k), \quad (S_1)$$

where state $x_k \in X \subseteq \mathbb{R}^{n_x}$, control input $u_k \in U \subseteq \mathbb{R}^{n_u}$ and $F: X \times U \rightrightarrows X$ is a set-valued map.

We consider a specification given under the form of a discrete-time hybrid automaton S_2 :

$$(x_{k+1}, p_{k+1}) \in G(x_k, p_k, v_k), \quad (S_2)$$

where $x_k \in X$, $p_k \in P$, $v_k \in V$ with P and V being finite sets of modes and external inputs, respectively, and $G: X \times P \times V \rightrightarrows X \times P$ is a set-valued map. Let us emphasize that the set of continuous states of S_2 coincides with the set of states of S_1 . Intuitively, the specification describes how the closed loop system should react to external inputs v_k , which can represent for instance a sequence of instructions received from a human user or from some other system.

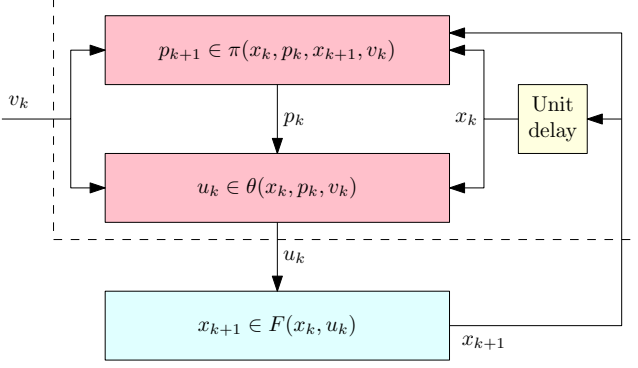


Fig. 1. Architecture of the closed-loop system S_{cl}

Formally, S_1 and S_2 can be written as transition systems

$$S_1 = (X, U, X, F, H_1),$$

$$S_2 = (X \times P, V, X, G, H_2),$$

where the output maps for S_1 and S_2 are given by $H_1(x) = x$ and $H_2(x, p) = x$ respectively.

We consider a *controller* given by a pair of set-valued maps $\theta: X \times P \times V \rightrightarrows U$ and $\pi: X \times P \times X \times V \rightrightarrows P$. Controller (θ, π) is said to be *compatible* with S_1 if for all $x \in X, p \in P, v \in V$,

$$\begin{aligned} \theta(x, p, v) &\subseteq \text{enab}_F(x) \text{ and} \\ \forall x' \in F(x, \theta(x, p, v)), \pi(x, p, x', v) &\neq \emptyset. \end{aligned} \quad (1)$$

Then, the dynamics of the closed-loop system S_{cl} , shown in Figure 1, is given by:

$$\begin{cases} x_{k+1} \in F(x_k, \theta(x_k, p_k, v_k)), \\ p_{k+1} \in \pi(x_k, p_k, x_{k+1}, v_k). \end{cases} \quad (S_{cl})$$

and formally described by the transition system

$$S_{cl} = (X \times P, V, X, \Delta_{cl}, H_2)$$

where for all $x \in X, p \in P, v \in V$,

$$\Delta_{cl}(x, p, v) = \left\{ (x', p') \mid \begin{array}{l} x' \in F(x, \theta(x, p, v)) \\ p' \in \pi(x, p, x', v) \end{array} \right\}. \quad (2)$$

Let us remark that if (θ, π) is compatible with S_1 then $v \in \text{enab}_{\Delta_{cl}}(x, p)$ if and only if $\theta(x, p, v) \neq \emptyset$.

The main problem considered in this paper is to synthesize a controller (θ, π) and a set of controllable states $Z_c \subseteq X \times P$ such that every trajectory of the closed-loop system S_{cl} initialized in Z_c is a trajectory of the specification S_2 .

Problem 3 Find a controller (θ, π) compatible with S_1 and a controllable set $Z_c \subseteq X \times P$ such that for every $(x_0, p_0) \in Z_c$, every maximal trajectory $(x_k, p_k, v_k)_{k=0}^K$ of S_{cl} is also a maximal trajectory of S_2 .

Remark 4 If Problem 3 is solved, then for every $(x_0, p_0) \in Z_c$, the following holds:

- every trajectory $(x_k, p_k, v_k)_{k=0}^K$ of S_{cl} is also a trajectory of S_2 ;
- every maximal trajectory $(x_k, p_k, v_k)_{k=0}^K$ of S_{cl} , where $v_k \in \text{enab}_G(x_k, p_k)$ for all $0 \leq k \leq K$, is complete (i.e. $K = +\infty$).

3 Characterization using alternating simulation relations

In this section, we establish that the solution of Problem 3 is characterized by the existence of an alternating simulation relation [18] from the specification S_2 to the system S_1 .

Definition 5 Let $S_a = (X_a, U_a, Y_a, \Delta_a, H_a)$ and $S_b = (X_b, U_b, Y_b, \Delta_b, H_b)$ be two transition systems with $Y_a = Y_b$. A relation $R \subseteq X_a \times X_b$ is an alternating simulation relation from S_a to S_b if the following conditions are satisfied:

- (1) for every $(x_a, x_b) \in R$ we have $H_a(x_a) = H_b(x_b)$;
- (2) for every $(x_a, x_b) \in R$ and for every $u_a \in \text{enab}_{\Delta_a}(x_a)$ there exists $u_b \in \text{enab}_{\Delta_b}(x_b)$ such that for every $x'_b \in \Delta_b(x_b, u_b)$ there exists $x'_a \in \Delta_a(x_a, u_a)$ satisfying $(x'_a, x'_b) \in R$.

It is said that S_b alternatingly simulates S_a , denoted by $S_a \preceq_{AS} S_b$, if there exists an alternating simulation relation $R \neq \emptyset$ from S_a to S_b .

Remark 6 Let R be an alternating simulation relation from S_a to S_b and consider $(x_a, x_b) \in R$. If x_a is a blocking state of S_a then condition 2) holds automatically. If x_b is blocking for S_b then x_a must be blocking for S_a as well.

We describe in the following the connections between Problem 3 and alternating simulation relations.

3.1 Sufficiency

Let us assume that $R \subseteq X \times P \times X$ is an alternating simulation relation from S_2 to S_1 , then let us define the set $Z_c \subseteq X \times P$ and the controller (θ, π) as follows:

$$Z_c = \{(x, p) \in X \times P \mid ((x, p), x) \in R\}, \quad (3)$$

$$\theta(x, p, v) = \left\{ u \in \text{enab}_F(x) \mid \begin{array}{l} \forall x' \in F(x, u), \exists p' \in P : \\ (x', p') \in G(x, p, v) \cap Z_c \end{array} \right\}, \quad (4)$$

$$\pi(x, p, x', v) = \{p' \in P \mid (x', p') \in G(x, p, v) \cap Z_c\}. \quad (5)$$

Theorem 7 *Let R be an alternating simulation relation from S_2 to S_1 , let the set Z_c and the controller (θ, π) be defined by (3), (4), (5). Then, (θ, π) and Z_c solve Problem 3.*

PROOF. We start by proving that (θ, π) is compatible with S_1 . By (4) it is clear that $\theta(x, p, v) \subseteq \text{enab}_F(x)$. Moreover, by (4) and (5), we get that for all $x' \in F(x, \theta(x, p, v))$, $\pi(x, p, x', v) \neq \emptyset$.

Then, let us consider $(x_0, p_0) \in Z_c$, and a maximal trajectory $(x_k, p_k, v_k)_{k=0}^K$ of S_{cl} . Then, by (2), we get that $p_{k+1} \in \pi(x_k, p_k, x_{k+1}, v_k)$ for $0 \leq k < K$. By (5), we get that $(x_{k+1}, p_{k+1}) \in G(x_k, p_k, v_k) \cap Z_c$ for $0 \leq k < K$. Hence, it follows that $(x_k, p_k, v_k)_{k=0}^K$ is a trajectory of S_2 and that $(x_k, p_k) \in Z_c$, for $0 \leq k \leq K$.

Let us assume that $(x_k, p_k, v_k)_{k=0}^K$ is not maximal for S_2 , then $v_K \in \text{enab}_G(x_K, p_K)$. Since $(x_K, p_K) \in Z_c$, we get $((x_K, p_K), x_K) \in R$, which is an alternating simulation relation from S_2 to S_1 . Therefore, from the condition 2) in Definition 5, there exists $u \in \text{enab}_F(x_K)$ such that for all $x' \in F(x_K, u)$, there exists $(x'', p'') \in G(x_K, p_K, v_K)$ such that $((x'', p''), x') \in R$. Condition 1) of Definition 5 gives that $x'' = x'$ and therefore $(x', p'') \in Z_c$. Hence, by (4), $u \in \theta(x_K, p_K, v_K)$, which is therefore non-empty. Since (θ, π) is compatible with S_1 , we obtain that $v_K \in \text{enab}_{\Delta_{cl}}(x_K, p_K)$. It follows that $(x_k, p_k, v_k)_{k=0}^K$ is not maximal for S_{cl} , which leads to a contradiction. \square

3.2 Necessity

While the previous theorem shows that solutions of Problem 3 can be obtained from alternating simulation relations, the following theorem shows the converse result:

Theorem 8 *Let controller (θ, π) and set Z_c solve Problem 3. Let us define a relation R by the following: $((x, p), x') \in R$ if and only if $x = x'$ and there exists $(x_0, p_0) \in Z_c$ and a trajectory $(x_k, p_k, v_k)_{k=0}^K$ of S_{cl} , with $K \in \mathbb{N}$, such that $x_K = x$, $p_K = p$. Then R is an alternating simulation relation from S_2 to S_1 .*

PROOF. Condition 1) of Definition 5 does obviously hold. Now, let $((x, p), x) \in R$ and consider the corresponding trajectory $(x_k, p_k, v_k)_{k=0}^K$ of S_{cl} , such that $(x_0, p_0) \in Z_c$ and $x_K = x$, $p_K = p$. If (x_K, p_K) is blocking for S_2 , then by Remark 6, condition 2) of Definition 5 holds. Otherwise, let $v \in \text{enab}_G(x_K, p_K)$. Then let $v'_k = v_k$ for $0 \leq k \leq K-1$ and $v'_K = v$. Then, $(x_k, p_k, v'_k)_{k=0}^K$ is a trajectory of S_{cl} and also of S_2 by Remark 4. Moreover, it is not maximal for

S_2 , which implies by Problem 3 that it is not maximal for S_{cl} . Hence, there exists $u \in \theta(x_K, p_K, v'_K)$. Let $x' \in F(x_K, u)$, since (θ, π) is compatible with S_1 we get that there exists $p' \in \pi(x_K, p_K, x', v'_K)$. By (2), we get that $(x', p') \in \Delta_{cl}(x_K, p_K, v'_K)$. Hence, $(x_k, p_k, v'_k)_{k=0}^{K+1}$ where $x_{K+1} = x'$, $p_{K+1} = p'$ and $v'_{K+1} \in V$ is a trajectory of S_{cl} . Then, $((x', p'), x') \in R$. Moreover by Remark 4, we get that $(x_k, p_k, v'_k)_{k=0}^{K+1}$ is also a trajectory of S_2 , which implies that $(x', p') \in G(x_K, p_K, v'_K)$. Hence, condition 2) of Definition 5 holds. \square

4 Abstraction based approach

In the previous section, we have shown that solving Problem 3 is actually equivalent to computing an alternating simulation relation from S_2 to S_1 . In this section, we present an approach, based on finite abstractions, for computing such a relation.

Our approach is based on the computation of an abstract system \hat{S}_1 and an abstract specification \hat{S}_2 such that the following relations hold:

$$S_2 \preceq_{AS} \hat{S}_2, \quad (6)$$

$$\hat{S}_2 \preceq_{AS} \hat{S}_1, \quad (7)$$

$$\hat{S}_1 \preceq_{AS} S_1. \quad (8)$$

Then, by transitivity of alternating simulation relations (see Proposition 4.23 in [18]), we get that $S_2 \preceq_{AS} S_1$ and the associated alternating simulation relation may be obtained from those involved in (6), (7), (8).

In the following, let the state space X be covered by a finite partition $(X_q)_{q \in Q}$: $X = \cup_{q \in Q} X_q$, $X_q \cap X_{q'} = \emptyset$, for $q \neq q'$. Let us derive the conditions for relations (6), (7), (8) to hold.

4.1 Abstraction for the control system

We first consider relation (8). Let $\hat{U} \subseteq U$ be a finite subset of control inputs, and let us define an abstract system $\hat{S}_1 = (X, \hat{U}, X, \hat{F}, H_1)$ where the transition relation \hat{F} is given by:

$$x' \in \hat{F}(x, \hat{u}) \iff x \in X_q, x' \in X_{q'}, q' \in \Delta_1(q, \hat{u}) \quad (9)$$

with the finite transition map $\Delta_1 : Q \times \hat{U} \rightrightarrows Q$ satisfying for all $q \in Q$

$$\text{enab}_{\Delta_1}(q) \subseteq \bigcap_{x \in X_q} \text{enab}_F(x) \quad (10)$$

and for all $q, q' \in Q$, $\hat{u} \in \text{enab}_{\Delta_1}(q)$

$$X_{q'} \cap F(X_q, \hat{u}) \neq \emptyset \implies q' \in \Delta_1(q, \hat{u}). \quad (11)$$

Proposition 9 For transition system S_1 and abstract system \hat{S}_1 satisfying (9), (10), (11), we have $\hat{S}_1 \preceq_{AS} S_1$.

PROOF. Consider the identity relation on X and let us prove that it is an alternating simulation relation. Condition 1) of Definition 5 does obviously hold. Condition 2) reads: for every $x \in X$ and every $u_1 \in \text{enab}_{\hat{F}}(x)$ there exists $u_2 \in \text{enab}_F(x)$ such that $F(x, u_2) \subseteq \hat{F}(x, u_1)$. Let us verify a stricter statement when $u_2 = u_1 = \hat{u} \in \hat{U}$. Consider an arbitrary $x \in X$ and $\hat{u} \in \text{enab}_{\hat{F}}(x)$, then by (9) and (10), $\hat{u} \in \text{enab}_F(x)$. Then, let $x' \in F(x, \hat{u})$, and let q and q' be such that $x \in X_q$, $x' \in X_{q'}$. Then $x' \in X_{q'} \cap F(X_q, \hat{u})$. Therefore by (9) and (11), $x' \in \hat{F}(x, \hat{u})$. \square

The previous construction resembles several approaches in the literature (see e.g. [3,11,16]) where condition (11) is used. However, in these works, condition (10) does not appear since it is assumed that $\text{enab}_F(x) = U$, for all $x \in X$. In that case, condition (10) is satisfied automatically.

The finite transition map Δ_1 where inclusion (10) and implication (11) are replaced by equality and equivalence, has the “maximal” number of enabled inputs at state q and the “minimal” number of transitions from a state q with input \hat{u} , and leads to the least conservative abstraction. Using this particular abstraction it is easier to synthesize a controller which solves our problem. We now discuss practical computations in a particular case.

4.1.1 Monotone systems

Consider a system of the following type:

$$x_{k+1} = f(x_k, u_k, w_k),$$

where disturbance $w_k \in W \subseteq \mathbb{R}^{n_w}$ and $f : X \times U \times W \rightarrow \mathbb{R}^{n_x}$ is a map. This system is restated in the form of S_1 where the set-valued map F is defined as follows:

$$F(x, u) = \begin{cases} f(x, u, W) & \text{if } f(x, u, W) \subseteq X, \\ \emptyset & \text{otherwise.} \end{cases}$$

Let this system be monotone [1] on $X \times W$, i.e. for all control input $u \in U$,

$$x \preceq x', w \preceq w' \implies f(x, u, w) \preceq f(x', u, w').$$

Let us assume $X_q \subseteq [\underline{x}_q, \bar{x}_q]$, $q \in Q$ and $W \subseteq [\underline{w}, \bar{w}]$. The monotonicity implies that

$$f(X_q, u, W) \subseteq [f(\underline{x}_q, u, \underline{w}), f(\bar{x}_q, u, \bar{w})].$$

Then, condition (10) can be replaced by the following condition, for $q \in Q$, $\hat{u} \in \hat{U}$:

$$\hat{u} \in \text{enab}_{\Delta_1}(q) \implies [f(\underline{x}_q, \hat{u}, \underline{w}), f(\bar{x}_q, \hat{u}, \bar{w})] \subseteq X.$$

Similarly, condition (11) can be replaced by, for $q, q' \in Q$, $\hat{u} \in \text{enab}_{\Delta_1}(q)$:

$$X_{q'} \cap [f(\underline{x}_q, \hat{u}, \underline{w}), f(\bar{x}_q, \hat{u}, \bar{w})] \neq \emptyset \implies q' \in \Delta_1(q, \hat{u}).$$

Let us remark that when the system is not monotone, one can apply a similar approach as long as one can compute interval over-approximations of the sets $f(X_q, u, W)$ using e.g. techniques in [3,16,10].

4.2 Abstraction for the specification

We now consider relation (6). Let us define an abstract specification $\hat{S}_2 = (X \times P, V, X, \hat{G}, H_2)$ where the transition relation \hat{G} is given by

$$(x', p') \in \hat{G}(x, p, v) \iff x \in X_q, x' \in X_{q'}, (q', p') \in \Delta_2(q, p, v) \quad (12)$$

with the finite transition map $\Delta_2 : Q \times P \times V \rightrightarrows Q \times P$ satisfying for all $q, q' \in Q$, $p, p' \in P$, $v \in V$,

$$(q', p') \in \Delta_2(q, p, v) \iff X_q \times X_{q'} \subseteq \{(x, x') \mid (x', p') \in G(x, p, v)\}. \quad (13)$$

For subsequent discussions, let us represent, without loss of generality, the transition relation G in the following form:

$$G(x, p, v) = \bigcup_{p' \in P} G_{p, p'}^v(x) \times \{p'\}$$

where $G_{p, p'}^v : X \rightrightarrows X$, for all $p, p' \in P$, $v \in V$. Let us make the following assumption:

Assumption 10 The transition map G satisfies the following conditions for some $L > 0$, $\delta > 0$:

- (1) for all $p, p' \in P$, $v \in V$, $G_{p, p'}^v$ is L -Lipschitz on its domain: for all $x_1, x_2 \in \text{dom}(G_{p, p'}^v)$,

$$h(G_{p, p'}^v(x_1), G_{p, p'}^v(x_2)) \leq L \|x_1 - x_2\|_\infty;$$

- (2) for all $p, p' \in P$, $v \in V$, $x \in \text{dom}(G_{p, p'}^v)$ the inradius of $G_{p, p'}^v(x)$ is greater than or equal to $\delta > 0$:

$$\rho^-(G_{p, p'}^v(x)) \geq \delta.$$

We also make the following assumption on the finite partition $(X_q)_{q \in Q}$:

Assumption 11 The partition $(X_q)_{q \in Q}$ satisfies:

(1) for all $q \in Q, p \in P, v \in V$,

$$\begin{aligned} \exists x \in X_q, G(x, p, v) \neq \emptyset &\implies \\ \exists p' \in P, X_q \subseteq \text{dom}(G_{p,p'}^v); \end{aligned}$$

(2) for all $q \in Q$, the circumradius of all sets X_q is less than $\delta/(2+L)$:

$$\rho^+(X_q) < \delta/(2+L)$$

where L and δ are defined in Assumption 10.

Remark 12 Let us point out that Condition 1) in Assumption 11 can be replaced by the stronger (but easier to check) condition that for all $q \in Q, p, p' \in P, v \in V$,

$$X_q \cap \text{dom}(G_{p,p'}^v) \neq \emptyset \implies X_q \subseteq \text{dom}(G_{p,p'}^v).$$

We now establish the following instrumental result:

Lemma 13 Under Assumptions 10 and 11, we have for all $x \in X, p \in P, \text{enab}_G(x, p) = \text{enab}_{\hat{G}}(x, p)$.

PROOF. Consider an arbitrary (x, p, v) such that $G(x, p, v) \neq \emptyset$. Let q be such that $x \in X_q$, then from condition 1) of Assumption 11, there exists $p' \in P$, such that $X_q \subseteq \text{dom}(G_{p,p'}^v)$. From conditions 1) and 2) in Assumption 10, it follows that

$$\rho^-\left(\bigcap_{x \in X_q} G_{p,p'}^v(x)\right) \geq \delta - L\rho^+(X_q).$$

Then, by condition 2) of Assumption 11, we have for all $q' \in Q$,

$$\rho^-\left(\bigcap_{x \in X_q} G_{p,p'}^v(x)\right) > 2\rho^+(X_{q'}),$$

which implies since $(X_q)_{q \in Q}$ is a partition that there exists $q' \in Q$ such that

$$X_{q'} \subseteq \bigcap_{x \in X_q} G_{p,p'}^v(x).$$

Then, $X_q \times X_{q'} \subseteq \{(x, x') \mid (x', p') \in G(x, p, v)\}$ and $\hat{G}(x, p, v) \neq \emptyset$. Thus, $\text{enab}_G(x, p) \subseteq \text{enab}_{\hat{G}}(x, p)$. The inclusion $\text{enab}_{\hat{G}}(x, p) \subseteq \text{enab}_G(x, p)$, is a direct consequence of (12) and (13). \square

Proposition 14 Under Assumptions 10 and 11, for transition system S_2 and abstract system \hat{S}_2 satisfying (12), (13), we have $S_2 \preceq_{AS} \hat{S}_2$.

PROOF. Consider the identity relation on $X \times P$ and let us prove that it is an alternating simulation relation. Condition 1) of Definition 5 does obviously hold. Condition 2) reads: for every $(x, p) \in X \times P$ and every $v_1 \in \text{enab}_G(x, p)$ there exists $v_2 \in \text{enab}_{\hat{G}}(x, p)$ such that $\hat{G}(x, p, v_2) \subseteq G(x, p, v_1)$. Let us verify a stricter statement when $v_2 = v_1 = v$. Consider an arbitrary $(x, p) \in X \times P, v \in \text{enab}_G(x, p)$, then from Lemma 13, $v \in \text{enab}_{\hat{G}}(x, p)$. Let $(x', p') \in \hat{G}(x, p, v)$, let $q, q' \in Q$ be such that $x \in X_q, x' \in X_{q'}$. Then $(x, x') \in \{(z, z') \mid (z', p') \in G(z, p, v)\}$. Therefore, $(x', p') \in G(x, p, v)$. \square

The use of Assumptions 10 and 11 in Lemma 13 and Proposition 14 shows that the partition $(X_q)_{q \in Q}$ needs to be chosen carefully in order to abstract the specification. In particular, condition 2) in Assumption 11 states that the regions X_q should be small enough.

Let us now specify a class of specification systems S_2 for which we are able to efficiently compute the abstractions above.

4.2.1 Piecewise affine hybrid automata

We consider specifications given by piecewise affine hybrid automata where for all $p, p' \in P, v \in V$,

$$\text{dom}(G_{p,p'}^v) = D_{p,p'}^v,$$

and for all $x \in \text{dom}(G_{p,p'}^v)$

$$G_{p,p'}^v(x) = A_{p,p'}^v x + W_{p,p'}^v,$$

where $A_{p,p'}^v$ is a matrix in $\mathbb{R}^{n_x \times n_x}$, $D_{p,p'}^v, W_{p,p'}^v$ are closed convex polytopes in \mathbb{R}^{n_x} .

For Assumption 10 to hold, it is sufficient that the sets $W_{p,p'}^v$ have non-empty interior for all $p, p' \in P, v \in V$. Then, L and δ satisfying conditions 1) and 2) of Assumption 10 are given by

$$L = \max_{p,p' \in P, v \in V} \|A_{p,p'}^v\|, \quad \delta = \min_{p,p' \in P, v \in V} \rho^-(W_{p,p'}^v).$$

Let us assume that the partition $(X_q)_{q \in Q}$ is chosen so as to satisfy Assumption 11. Let us remark that if X is a polytope it is always possible to chose such a partition where for all $q \in Q, X_q$ is a polytope.

Then according to (13), for a transition $(q', p') \in \Delta_2(q, p, v)$ to be enabled it is necessary and sufficient that

$$X_q \subseteq D_{p,p'}^v \text{ and } X_{q'} - A_{p,p'}^v X_q \subseteq W_{p,p'}^v.$$

These inclusions can be checked effectively by verifying that

$$x \in D_{p,p'}^v \text{ and } x' \in A_{p,p'}^v, x \in W_{p,p'}^v.$$

hold when x and x' belong to the set of vertices of X_q and $X_{q'}$, respectively.

4.3 Controller synthesis

To synthesize a controller which solves Problem 3 we first define an alternating simulation relation R which ensures relation (7). We construct R as follows:

$$R = \left\{ ((x, p), x') \in X \times P \times X \mid \begin{array}{l} x = x' \in X_q \\ (q, p) \in \hat{Z}^\infty \end{array} \right\}, \quad (14)$$

where \hat{Z}^∞ denotes the limit of the sequence $(\hat{Z}^k)_{k \in \mathbb{N}}$ defined by:

$$\begin{aligned} \hat{Z}^0 &= Q \times P, \\ \hat{Z}^{k+1} &= \left\{ (q, p) \in \hat{Z}^k \mid \begin{array}{l} \forall v \in \text{enab}_{\Delta_2}(q, p), \\ \exists u \in \text{enab}_{\Delta_1}(q) : \\ \forall q' \in \Delta_1(q, u), \exists p' \in P : \\ (q', p') \in \Delta_2(q, p, v) \cap \hat{Z}^k \end{array} \right\}. \end{aligned} \quad (15)$$

Proposition 15 *The sequence $(\hat{Z}^k)_{k \in \mathbb{N}}$ defined by (15) reaches the fixed point \hat{Z}^∞ in finite time. The relation R defined by (14) is an alternating simulation relation from \hat{S}_2 to \hat{S}_1 .*

PROOF. Since $Q \times P$ and thus $\hat{Z}^k \subseteq Q \times P$ is finite for all $k \in \mathbb{N}$, the first statement follows directly from the inclusion $\hat{Z}^{k+1} \subseteq \hat{Z}^k$.

Now, let us prove that R is an alternating simulation relation from \hat{S}_2 to \hat{S}_1 . Condition 1) of Definition 5 is obviously satisfied. Let $((x, p), x) \in R$, then from (14), $(q, p) \in \hat{Z}^\infty$ with $x \in X_q$. Let $v \in \text{enab}_{\hat{G}}(x, p)$, then from (12), $v \in \text{enab}_{\Delta_2}(q, p)$. Since \hat{Z}^∞ is the fixed point of (15), we have that there exists $u \in \text{enab}_{\Delta_1}(q)$ such that for all $q' \in \Delta_1(q, u)$, there exists $p' \in P$ with $(q', p') \in \Delta_2(q, p, v) \cap \hat{Z}^\infty$. From (9), $\text{enab}_{\Delta_1}(q) = \text{enab}_{\hat{F}}(x)$, then $u \in \text{enab}_{\hat{F}}(x)$. Let $x' \in \hat{F}(x, u)$, and let $q' \in Q$ such that $x' \in X_{q'}$, then from (9), $q' \in \Delta_1(q, u)$. Hence, there exists $p' \in P$ such that $(q', p') \in \Delta_2(q, p, v) \cap \hat{Z}^\infty$. Then, from (12), $(x', p') \in \hat{G}(x, p, v)$, and from (14), $((x', p'), x') \in R$. Thus, condition 2) of Definition 5 holds. \square

Under Assumptions 10 and 11, it follows from Propositions 9, 14 and 15 and Proposition 4.23 in [18] that R

is also an alternating simulation relation from S_2 to S_1 . Then, Theorem 7 allows us to find a solution to Problem 3. However, the controller given by (4) and (5) is not practical to compute, since it involves the maps F and G . For that reason, we propose in the following, an alternative controller, which also solves Problem 3 and can be computed using the symbolic abstractions only.

Let the relation R be given by (14) and let us define the set Z_c and the controller (θ, π) as follows:

$$Z_c = \{(x, p) \mid ((x, p), x) \in R\}, \quad (16)$$

$$\theta(x, p, v) = \left\{ u \in \text{enab}_{\hat{F}}(x) \mid \begin{array}{l} \forall x' \in \hat{F}(x, u), \exists p' \in P : \\ (x', p') \in \hat{G}(x, p, v) \cap Z_c \end{array} \right\}, \quad (17)$$

$$\pi(x, p, x', v) = \{p' \in P \mid (x', p') \in \hat{G}(x, p, v) \cap Z_c\}. \quad (18)$$

Theorem 16 *Under Assumptions 10 and 11, let the set Z_c and the controller (θ, π) be defined by (16), (17), (18). Then, (θ, π) and Z_c solve Problem 3.*

PROOF. First, by Proposition 15 and Theorem 7, we can check that (θ, π) and Z_c solve Problem 3 for system \hat{S}_1 and specification \hat{S}_2 . Let us denote $\hat{S}_{cl} = (X \times P, V, X, \hat{\Delta}_{cl}, H_2)$ the transition system describing the closed-loop dynamics of \hat{S}_1 with controller (θ, π) . Then, any maximal trajectory of \hat{S}_{cl} , initialized in Z_c , is also a maximal trajectory of \hat{S}_2 .

From the proof of Proposition 9, we get that

$$\begin{aligned} \forall x \in X, \text{enab}_{\hat{F}}(x) &\subseteq \text{enab}_F(x) \\ \forall x \in X, u \in \text{enab}_{\hat{F}}(x), F(x, u) &\subseteq \hat{F}(x, u). \end{aligned} \quad (19)$$

Then, it follows that (θ, π) being compatible with \hat{S}_1 implies that it is also compatible with S_1 . Moreover, for all $x \in X$, $p \in P$, $\text{enab}_{\hat{\Delta}_{cl}}(x, p) = \text{enab}_{\Delta_{cl}}(x, p)$ and consists of inputs $v \in V$ such that $\theta(x, p, v) \neq \emptyset$. Then, from (19), we get that for all $v \in \text{enab}_{\Delta_{cl}}(x, p)$, $\Delta_{cl}(x, p, v) \subseteq \hat{\Delta}_{cl}(x, p, v)$. Hence, any maximal trajectory of S_{cl} is also a maximal trajectory of \hat{S}_{cl} .

From Lemma 13 and from the proof of Proposition 14, we get that

$$\begin{aligned} \forall x \in X, p \in P, \text{enab}_G(x, p) &= \text{enab}_{\hat{G}}(x, p) \\ \forall x \in X, p \in P, v \in \text{enab}_{\hat{G}}(x, p), \hat{G}(x, p, v) &\subseteq G(x, p, v). \end{aligned} \quad (20)$$

Hence, any maximal trajectory of \hat{S}_2 is also a maximal trajectory of S_2 .

Therefore, we get that any maximal trajectory of S_{cl} is also a maximal trajectory of S_2 . \square

Remark 17 Z_c defined by (16) can also be defined directly from \hat{Z}^∞ as

$$Z_c = \left\{ (x, p) \in X \times P \mid x \in X_q, (q, p) \in \hat{Z}^\infty \right\}. \quad (21)$$

Similarly, θ and π defined in (17), (18) can be easily defined using the transition relations Δ_1 and Δ_2 instead of \hat{F} and \hat{G} thanks to the relations (9), (12).

5 Safety and reachability requirements

In this section, we extend our approach to solve synthesis problems similar to Problem 3 with additional safety and reachability requirements. The motivation for considering such additional objectives is as follows. Firstly, in Problem 3 there is no mechanism preventing the system from reaching a blocking state of the specification, which would also be a blocking state of the closed-loop system. Secondly, we would like to make it possible to define terminal states to specify when the task described by the specification has been accomplished. Depending on whether the terminal states have to be reached or not leads to safety or reachability requirements.

Formally, let us consider a set of terminal states given by $Z_f \subseteq X \times P$. We define the following extensions of Problem 3:

Problem 18 (Safety requirements) Find a controller (θ, π) compatible with S_1 and a controllable set $Z_c \subseteq X \times P$ such that for every $(x_0, p_0) \in Z_c$, for every maximal trajectory $(x_k, p_k, v_k)_{k=0}^K$ of S_{cl} , one of the following condition holds:

- (1) $(x_k, p_k, v_k)_{k=0}^K$ is a trajectory of S_2 , $K \in \mathbb{N}$ and $(x_K, p_K) \in Z_f$;
- (2) $(x_k, p_k, v_k)_{k=0}^K$ is a maximal trajectory of S_2 , and either $K = +\infty$, or $K \in \mathbb{N}$ and $\text{enab}_G(x_K, p_K) \neq \emptyset$.

Remark 19 If Problem 18 is solved then, for every $(x_0, p_0) \in Z_c$, the following holds:

- every trajectory $(x_k, p_k, v_k)_{k=0}^K$ of S_{cl} is also a trajectory of S_2 , and for all $0 \leq k \leq K$, such that $(x_k, p_k) \notin Z_f$, $\text{enab}_G(x_k, p_k) \neq \emptyset$;
- for every maximal trajectory $(x_k, p_k, v_k)_{k=0}^K$ of S_{cl} , where for all $0 \leq k \leq K$, such that $(x_k, p_k) \notin Z_f$, $v_k \in \text{enab}_G(x_k, p_k)$, we have either $K = +\infty$, or $K \in \mathbb{N}$ and $(x_K, p_K) \in Z_f$.

We refer to Problem 18 as a safety problem because the controller has to avoid reaching a set of “unsafe” states, which consists of blocking states of the specification S_2

outside Z_f . Let us remark that in Problem 18, it is acceptable that the set of terminal states is not reached, provided the closed-loop system can behave as the specification for an unlimited amount of time. Let us remark that Problem 3 is a particular case of Problem 18 where Z_f coincides with the set of blocking states of S_2 .

Problem 20 (Reachability requirements) Find a controller (θ, π) compatible with S_1 and a controllable set $Z_c \subseteq X \times P$ such that for every $(x_0, p_0) \in Z_c$, for every maximal trajectory $(x_k, p_k, v_k)_{k=0}^K$ of S_{cl} , one of the following condition holds:

- (1) $(x_k, p_k, v_k)_{k=0}^K$ is a trajectory of S_2 , $K \in \mathbb{N}$ and $(x_K, p_K) \in Z_f$;
- (2) $(x_k, p_k, v_k)_{k=0}^K$ is a maximal trajectory of S_2 , and $K \in \mathbb{N}$ and $\text{enab}_G(x_K, p_K) \neq \emptyset$.

Remark 21 If Problem 20 is solved then, for every $(x_0, p_0) \in Z_c$, the following holds:

- every trajectory $(x_k, p_k, v_k)_{k=0}^K$ of S_{cl} is also a trajectory of S_2 , and for all $0 \leq k \leq K$, such that $(x_k, p_k) \notin Z_f$, $\text{enab}_G(x_k, p_k) \neq \emptyset$;
- for every maximal trajectory $(x_k, p_k, v_k)_{k=0}^K$ of S_{cl} , where for all $0 \leq k \leq K$, such that $(x_k, p_k) \notin Z_f$, $v_k \in \text{enab}_G(x_k, p_k)$, we have $K \in \mathbb{N}$ and $(x_K, p_K) \in Z_f$.

We refer to Problem 20 as a reachability problem because the controller has to reach the set of terminal states Z_f . Let us remark that in Problem 20, it is necessary that the set of terminal states is reached in finite time.

In the following, we show how to modify the approach presented in Section 4 in order to solve Problems 18 and 20. Let us assume that symbolic abstractions \hat{S}_1 and \hat{S}_2 for the system and the specification have been computed according to the procedure described in Sections 4.1 and 4.2. Let us define the set

$$\hat{Z}_f = \{(q, p) \in Q \times P \mid X_q \times \{p\} \subseteq Z_f\}.$$

5.1 Safety requirements

We start by describing a solution approach to Problem 18. Let \hat{Z}^∞ denote the limit of the sequence $(\hat{Z}^k)_{k \in \mathbb{N}}$ defined by:

$$\begin{aligned} \hat{Z}^0 &= Q \times P, \\ \hat{Z}^{k+1} &= \hat{Z}_f \cup \left\{ (q, p) \in \hat{Z}^k \mid \begin{array}{l} \text{enab}_{\Delta_2}(q, p) \neq \emptyset, \text{ and} \\ \forall v \in \text{enab}_{\Delta_2}(q, p), \\ \exists u \in \text{enab}_{\Delta_1}(q) : \\ \forall q' \in \Delta_1(q, u), \exists p' \in P : \\ (q', p') \in \Delta_2(q, p, v) \cap \hat{Z}^k \end{array} \right\}. \end{aligned} \quad (22)$$

Since for all $k \in \mathbb{N}$, \hat{Z}^k is finite and $\hat{Z}^{k+1} \subseteq \hat{Z}^k$, it follows that the fixed point \hat{Z}^∞ is reached in finite time.

Theorem 22 *Under Assumptions 10 and 11, let the set Z_c and the controller (θ, π) be defined by (21), (17), (18). Then, (θ, π) and Z_c solve Problem 18.*

PROOF. Using the same arguments as in the first two paragraphs of the proof of Theorem 7, we get that (θ, π) is compatible with \hat{S}_1 and that any maximal trajectory $(x_k, p_k, v_k)_{k=0}^K$ of \hat{S}_{cl} with $(x_0, p_0) \in Z_c$ is a trajectory of \hat{S}_2 and that $(x_k, p_k) \in Z_c$ for all $0 \leq k \leq K$. If $K = +\infty$ or if $K \in \mathbb{N}$ and $(x_K, p_K) \in Z_f$, the requirements of Problem 18 are met for \hat{S}_1 and \hat{S}_2 . Then, let us assume that $K \in \mathbb{N}$ and $(x_K, p_K) \notin Z_f$. Since $(x_K, p_K) \in Z_c \setminus Z_f$, it follows by (21), (22) and (12) that $\text{enab}_{\hat{G}}(x_K, p_K) \neq \emptyset$. Let us assume that $(x_k, p_k, v_k)_{k=0}^K$ is not maximal for \hat{S}_2 , then $v_K \in \text{enab}_{\hat{G}}(x_K, p_K)$ and from (21), (22), (12), (9), we get that there exists $u \in \text{enab}_{\hat{F}}(x_K)$ such that for all $x' \in \hat{F}(x_K, u)$, there exists $p' \in P$ such that $(x', p') \in \hat{G}(x, p, v) \cap Z_c$. Then, from (17), $u \in \theta(x_K, p_K, v_K)$, which is therefore non-empty. Since (θ, π) is compatible with \hat{S}_1 , we obtain that $v_K \in \text{enab}_{\hat{\Delta}_{cl}}(x_K, p_K)$. It follows that $(x_k, p_k, v_k)_{k=0}^K$ is not maximal for \hat{S}_{cl} , which leads to a contradiction. Hence, it follows that (θ, π) and Z_c solve Problem 18 for \hat{S}_1 and \hat{S}_2 .

Then, from the proof of Theorem 16, we get that any maximal trajectory of S_{cl} is also a maximal trajectory of \hat{S}_{cl} and that any (maximal) trajectory of \hat{S}_2 is a (maximal) trajectory of S_2 . Therefore, (θ, π) and Z_c solve Problem 18 for S_1 and S_2 . \square

5.2 Reachability requirements

We now present a solution approach to Problem 20. Let \hat{Z}^∞ denote the limit of the sequence $(\hat{Z}^k)_{k \in \mathbb{N}}$ defined by:

$$\begin{aligned} \hat{Z}^0 &= \hat{Z}_f, \\ \hat{Z}^{k+1} &= \hat{Z}_f \cup \left\{ (q, p) \in Q \times P \left| \begin{array}{l} \text{enab}_{\Delta_2}(q, p) \neq \emptyset, \text{ and} \\ \forall v \in \text{enab}_{\Delta_2}(q, p), \\ \exists u \in \text{enab}_{\Delta_1}(q) : \\ \forall q' \in \Delta_1(q, u), \exists p' \in P : \\ (q', p') \in \Delta_2(q, p, v) \cap \hat{Z}^k \end{array} \right. \right\}. \end{aligned} \quad (23)$$

Since $Q \times P$ is finite and for all $k \in \mathbb{N}$, $\hat{Z}^k \subseteq \hat{Z}^{k+1} \subseteq Q \times P$, it follows that the fixed point \hat{Z}^∞ is reached in

finite time. Let Z_c be defined by (21), and for $k \in \mathbb{N}$, let

$$Z^k = \left\{ (x, p) \in X \times P \mid x \in X_q, (q, p) \in \hat{Z}^k \right\}.$$

Then, for $(x, p) \in Z_c$ we define

$$\kappa(x, p) = \min \{k \in \mathbb{N} \mid (x, p) \in Z^k\}. \quad (24)$$

Then, let (θ, π) be given, for $(x, p) \in Z^0$, by $\theta(x, p, v) = \emptyset$, $\pi(x, p, x', v) = \emptyset$, and for $(x, p) \in Z_c \setminus Z^0$, by

$$\theta(x, p, v) = \left\{ u \in \text{enab}_{\hat{F}}(x) \left| \begin{array}{l} \forall x' \in \hat{F}(x, u), \exists p' \in P : \\ (x', p') \in \hat{G}(x, p, v) \cap \\ Z^{\kappa(x, p)-1} \end{array} \right. \right\}, \quad (25)$$

$$\pi(x, p, x', v) = \{p' \in P \mid (x', p') \in \hat{G}(x, p, v) \cap Z^{\kappa(x, p)-1}\}. \quad (26)$$

Theorem 23 *Under Assumptions 10 and 11, let the set Z_c and the controller (θ, π) be defined by (21), (25), (26). Then, (θ, π) and Z_c solve Problem 20.*

PROOF. Using the same arguments as in the first two paragraphs of the proof of Theorem 7, we get that (θ, π) is compatible with \hat{S}_1 and that any maximal trajectory $(x_k, p_k, v_k)_{k=0}^K$ of \hat{S}_{cl} with $(x_0, p_0) \in Z_c$ is a trajectory of \hat{S}_2 and that $\kappa(x_k, p_k)$ is strictly decreasing and non-negative. Then, necessarily $K \in \mathbb{N}$. If $(x_K, p_K) \in Z_f$, the requirements of Problem 18 are met for \hat{S}_1 and \hat{S}_2 . Then, let us assume that $(x_K, p_K) \notin Z_f$. Then, $(x_K, p_K) \notin Z^0$ and $\kappa(x_K, p_K) > 0$. Then, it follows by (24), (23) and (12) that $\text{enab}_{\hat{G}}(x_K, p_K) \neq \emptyset$. Let us assume that $(x_k, p_k, v_k)_{k=0}^K$ is not maximal for \hat{S}_2 , then $v_K \in \text{enab}_{\hat{G}}(x_K, p_K)$ and from (24), (23), (12), (9), we get that there exists $u \in \text{enab}_{\hat{F}}(x_K)$ such that for all $x' \in \hat{F}(x_K, u)$, there exists $p' \in P$ such that $(x', p') \in \hat{G}(x, p, v) \cap Z^{\kappa(x_K, p_K)-1}$. Then, from (25), $u \in \theta(x_K, p_K, v_K)$, which is therefore non-empty. Since (θ, π) is compatible with \hat{S}_1 , we obtain that $v_K \in \text{enab}_{\hat{\Delta}_{cl}}(x_K, p_K)$. It follows that $(x_k, p_k, v_k)_{k=0}^K$ is not maximal for \hat{S}_{cl} , which leads to a contradiction. Hence, it follows that (θ, π) and Z_c solve Problem 20 for \hat{S}_1 and \hat{S}_2 .

Then, from the proof of Theorem 16, we get that any maximal trajectory of S_{cl} is also a maximal trajectory of \hat{S}_{cl} and that any (maximal) trajectory of \hat{S}_2 is a (maximal) trajectory of S_2 . Therefore, (θ, π) and Z_c solve Problem 20 for S_1 and S_2 . \square

6 Autonomous vehicle examples

In this section, we provide illustrations of our approach by showing instances of Problems 18 and 20 in the context of autonomous vehicle control. We first consider cruise control with collision avoidance and then takeover maneuvers.

6.1 Safety: cruise control with collision avoidance

Let us consider a model with two vehicles moving in one lane on an infinite straight road. The leader (vehicle 2) is uncontrollable while the follower (vehicle 1) is controllable. A discrete-time approximation of this model is given by equations:

$$\begin{aligned} x_{k+1}^1 &= x_k^1 + v_k^1 T_0, \\ v_{k+1}^1 &= \chi(v_k^1 + \alpha(u_k, v_k^1) T_0, [0, v_{\max}^1]), \\ x_{k+1}^2 &= x_k^2 + v_k^2 T_0, \\ v_{k+1}^2 &= \chi(v_k^2 + \alpha(w_k, v_k^2) T_0, [0, v_{\max}^2]). \end{aligned}$$

Here u_k and w_k are the torques applied to the wheels, $u_k \in [u_{\min}, u_{\max}]$ is the control input, $w_k \in [w_{\min}, w_{\max}]$ is the uncertainty and

$$\begin{aligned} \alpha(u, v) &= u - M^{-1}(f_0 + f_1 v + f_2 v^2), \\ \chi(v, [v_1, v_2]) &= \min\{\max\{v, v_1\}, v_2\}. \end{aligned}$$

The vector of parameters $f = (f_0, f_1, f_2) \in \mathbb{R}_+^3$ describes road friction and vehicle aerodynamics whose numerical values are taken from [12]: $f_0 = 51.0709 \text{ N}$, $f_1 = 0.3494 \text{ N s/m}$, $f_2 = 0.4161 \text{ N s}^2/\text{m}^2$.

We rewrite this as a 3-dimensional system with $d = x^1 - x^2$ being the signed distance between the vehicles:

$$\begin{aligned} d_{k+1} &= d_k + (v_k^1 - v_k^2) T_0, \\ v_{k+1}^1 &= \chi(v_k^1 + \alpha(u_k, v_k^1) T_0, [0, v_{\max}^1]), \\ v_{k+1}^2 &= \chi(v_k^2 + \alpha(w_k, v_k^2) T_0, [0, v_{\max}^2]). \end{aligned}$$

A model of this type was also considered in [17]. We consider this model on the following state space:

$$X = (-\infty, 0) \times [0, v_{\max}^1] \times [0, v_{\max}^2].$$

For the simulations below we choose the following parameters: $v_{\max}^1 = 30 \text{ m/s}$, $v_{\max}^2 = 30 \text{ m/s}$, $M = 1370 \text{ kg}$, $u_{\max} = w_{\max} = 10 \text{ m/s}^2$, $u_{\min} = w_{\min} = -20 \text{ m/s}^2$, $T_0 = 0.1 \text{ s}$. The system is monotone in the variable d , v^1 , $-v^2$, $-w$ and has the same form as in Section 4.1.1.

Let us introduce a partition for each coordinate:

$$\begin{aligned} &(-\infty, -d_{\max}), [-d_{\max}, -\frac{N_d - 1}{N_d} d_{\max}), \dots, \\ &[-\frac{2}{N_d} d_{\max}, -\frac{1}{N_d} d_{\max}), [-\frac{1}{N_d} d_{\max}, 0) \end{aligned}$$

for the signed distance,

$$\{0\}, (0, \frac{1}{N_{v^1}} v_{\max}^1], \dots, (\frac{N_{v^1} - 1}{N_{v^1}} v_{\max}^1, v_{\max}^1]$$

for the velocity v^1 of the controllable vehicle and

$$\{0\}, (0, \frac{1}{N_{v^2}} v_{\max}^2], \dots, (\frac{N_{v^2} - 1}{N_{v^2}} v_{\max}^2, v_{\max}^2]$$

for the velocity v^2 of the uncontrollable vehicle. Based on this partition we construct regions X_q as Cartesian products of the elements from those partitions. The set of all such X_q covers the whole state space X . We also utilize

$$\begin{aligned} \hat{U} &= \{u_{\min}, u_{\min} + \frac{1}{N_u}(u_{\max} - u_{\min}), \dots, \\ &u_{\min} + \frac{N_u - 1}{N_u}(u_{\max} - u_{\min}), u_{\max}\} \end{aligned}$$

as a finite approximation of U . In the following, we use $d_{\max} = 100 \text{ m}$, $N_d = 50$, $N_{v^1} = 100$, $N_{v^2} = 15$, $N_u = 6$.

The specification is given by a hybrid automaton with 2 modes $\{p^0, p^1\}$. In the mode p^0 , vehicle 1 has to track the desired velocity v^* , which is provided by an external user. If the distance between the two vehicles is too short, mode p^1 is activated in order to avoid collision. When a longer distance is restored, mode p^0 is reactivated. The discrete dynamics of the hybrid automaton is shown in Figure 2.

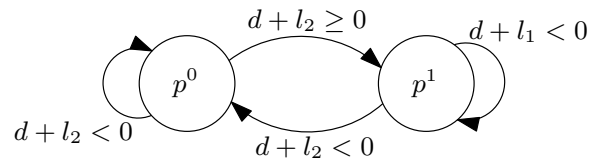


Fig. 2. Discrete dynamics of the hybrid automaton.

The associated continuous dynamics is then given by

$$\begin{cases} v_{k+1}^1 \geq v_k^1 + C, & \text{if } v_k^1 \leq v_k^* - \varepsilon, \\ v_{k+1}^1 \leq v_k^1 - C, & \text{if } v_k^1 \geq v_k^* + \varepsilon \end{cases}$$

for $p_{k+1} = p^0$ and

$$v_{k+1}^1 \leq v_k^1 - C, \text{ if } v_k^1 \geq v_k^* + \varepsilon$$

for $p_{k+1} = p^1$. Input parameter v^* in the specification belongs to the following set:

$$V = \left\{ \frac{1}{N_m} v_{\max}^1, \dots, \frac{N_m - 1}{N_m} v_{\max}^1 \right\}.$$

The previous hybrid automaton is not piecewise affine but has an equivalent piecewise affine representation with 5 modes, 3 associated to p^0 and 2 associated to p^1 . This hybrid automaton may reach blocking states so we add a safety requirement with an empty set of final states (i.e. the task should run forever). For the simulation we choose the following numerical values for the parameters of this specification:

$$N_m = 6, \quad \varepsilon = 3 \text{ m/s}, \quad C = 0.1 \text{ m/s}, \\ l_1 = 2 \text{ m}, \quad l_2 = 50 \text{ m}.$$

One may observe that the chosen partition contains unbounded elements which were not previously considered. However, when partition elements X_q and sets

$$\{(z, z') \mid (z', p') \in G(z, p, v)\}$$

are all Cartesian products of the matching dimensions, condition (13) splits into the set of respective conditions for the factors of these products. In the currently considered example, it is therefore sufficient to check the conditions of Lemma 13 only for v^1 as the inclusion in (13) holds along other dimensions. Thus, the specification is piecewise affine and satisfies Assumption 10 along v^1 with $L = 1$ and $\delta = 0.5 \times (\frac{v_{\max}^1}{N_m} + \varepsilon - C)$. The state space partition along v^1 satisfies Assumption 11 since

$$0.15 = \frac{v_{\max}^1}{2N_{v^1}} < \frac{\delta}{2 + L} = 3.95/3 \approx 1.31.$$

A controller has been synthesized by the approach presented in Section 5. Figures 3 shows the projection of the controllable set onto X .

Next, for the computed controller we simulate a trajectory of the closed-loop system. Figure 4 depicts signed distance d_k , velocities v_k^1 , v_k^2 and v_k^* , and control u_k generated by the feedback controller. One can check that initially the follower tracks the desired velocity until the distance between the two vehicles drops lower than the safe distance and the controller moves to mode p^1 , reducing the velocity of the follower to avoid collision.

6.2 Reachability: takeover maneuvers

Consider now a similar setting with two vehicles where the controllable vehicle is able to perform a takeover maneuver (see Fig. 5). The corresponding dynamics can

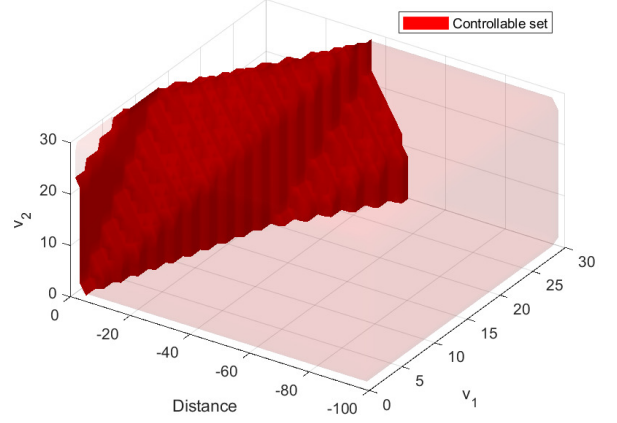


Fig. 3. The projection of the controllable region.

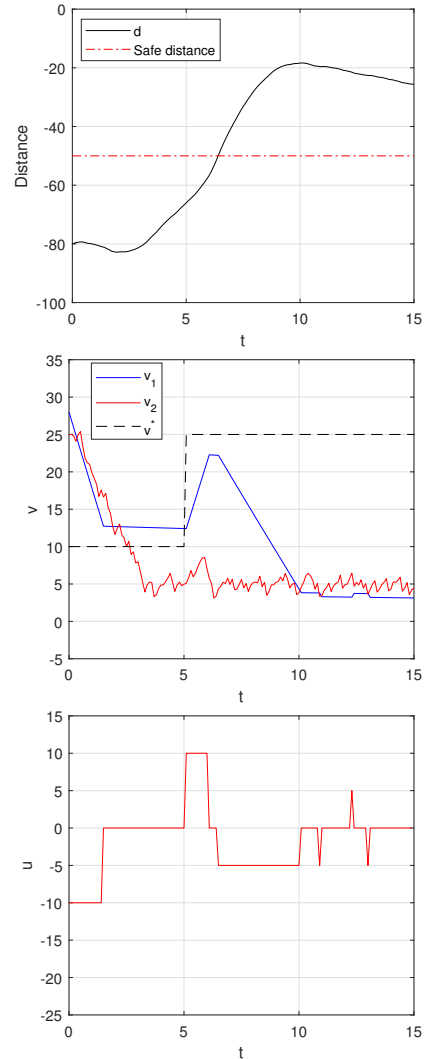


Fig. 4. Signed distance d_k , velocities v_k^1 , v_k^2 and v_k^* , and control u_k .

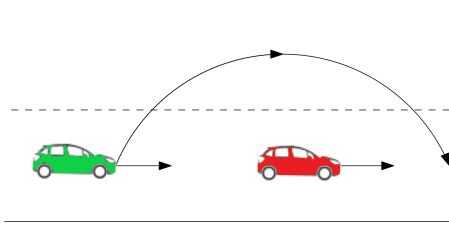


Fig. 5. Illustration of the vehicle takeover problem.

be approximated by the following discrete-time system:

$$\begin{aligned} d_{k+1} &= d_k + (v_k^1 - v_k^2)T_0, \\ v_{k+1}^1 &= \chi(v_k^1 + \alpha(u_k, v_k^1)T_0, [0, v_{\max}^1]), \\ v_{k+1}^2 &= \chi(v_k^2 + \alpha(w_k, v_k^2)T_0, [0, v_{\max}^2]), \\ h_{k+1} &= u_k^2 \end{aligned} \quad (27)$$

if $u_k^2 = h_k$ and

$$\begin{aligned} d_{k+1} &= d_k + \beta(v_k^1, v_k^2)T_0, \\ v_{k+1}^1 &= v_k^1, \\ v_{k+1}^2 &= \chi(v_k^2 + \alpha(w_k, v_k^2)T_0, [0, v_{\max}^2]), \\ h_{k+1} &= u_k^2 \end{aligned} \quad (28)$$

if $u_k^2 = 2 - h_k$, $v_k^1 \geq v_* \geq r/T_0$. Here h_k is the lane number of the controllable vehicle at time instant k ; u and w are torques applied to the wheels as before. In this system $u_k^1 \in [u_{\min}, u_{\max}]$ and $u_k^2 \in \{0.5, 1.5\}$ are the controls, $w_k \in [w_{\min}, w_{\max}]$ is the uncertainty.

Equation (28) is a discrete time approximation of the lane-changing maneuver. For simplicity, we assume that this maneuver takes the same time T_0 and function β is defined as follows

$$\beta(v^1, v^2) = \frac{1}{T_0} \sqrt{(v^1 T_0)^2 - r^2} - v^2.$$

Here r is the distance between lanes.

The control u_k^2 corresponds to the lane number at the new time instant. We assume that the uncontrollable vehicle moves in lane 1.

The state space in this example is given by

$$X = (-\infty, +\infty) \times [0, v_{\max}^1] \times [0, v_{\max}^2] \times [0, 2].$$

We define the target set as

$$X^1 = \{z = (d, v^1, v^2, h) \in X \mid d > 0, h < 1\}.$$

As in previous subsection, in this example we use very similar partition for d , v^1 and v^2 coordinates and

$$\hat{U} = \{u_{\min}, 0, u_{\max}\} \times \{0.5, 1.5\}$$

as a finite approximation of $[u_{\min}, u_{\max}] \times \{0.5, 1.5\}$. For the simulation below we choose the following parameters: $d_{\max} = 60$ m, $v_{\max}^1 = 30$ m/s, $v_{\max}^2 = 20$ m/s, $M = 1370$ Kg, $u_{\max} = w_{\max} = 10$ m/s², $u_{\min} = w_{\min} = -20$ m/s², $T_0 = 0.2$ s, $r = 2$ m, $v_* = 20$ m/s, $N_d = 80$, $N_{v^1} = 60$, $N_{v^2} = 20$.

The specification system $(z', p') \in G(z, p)$ here is defined by the following constraints on the continuous dynamics:

$$\begin{cases} d_{k+1} < 0 & \text{if } p_{k+1} = p^0, \\ d_{k+1} \geq 0 & \text{if } p_{k+1} = p^1, \\ h_{k+1} \geq 1 & \text{if } p_k = p^0, p_{k+1} = p^1 \end{cases}$$

and by the discrete variable evolution which is depicted on Fig. 6.

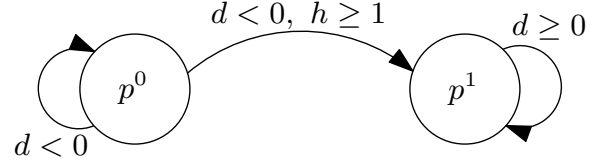


Fig. 6. Graphical representation of the discrete variable p evolution in the specification.

Assumptions 10 and 11 are obviously satisfied.

A controller has been synthesized by the approach presented in Section 5. Figure 7 depicts a part of the controllable set which corresponds to $h < 1$, $d < 0$, $p = p^0$.

In this example we also simulate a trajectory of the closed-loop system using the computed reachability con-

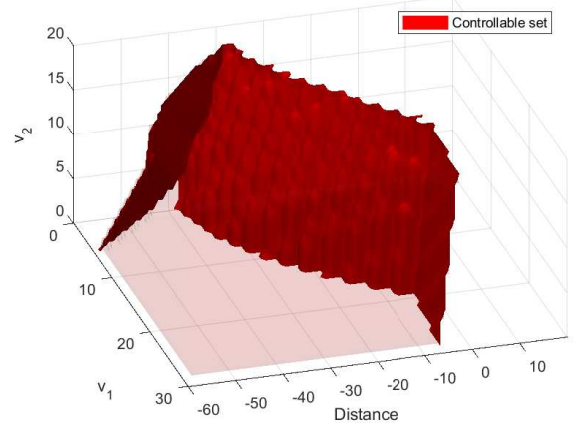


Fig. 7. The projection of the controllable region.

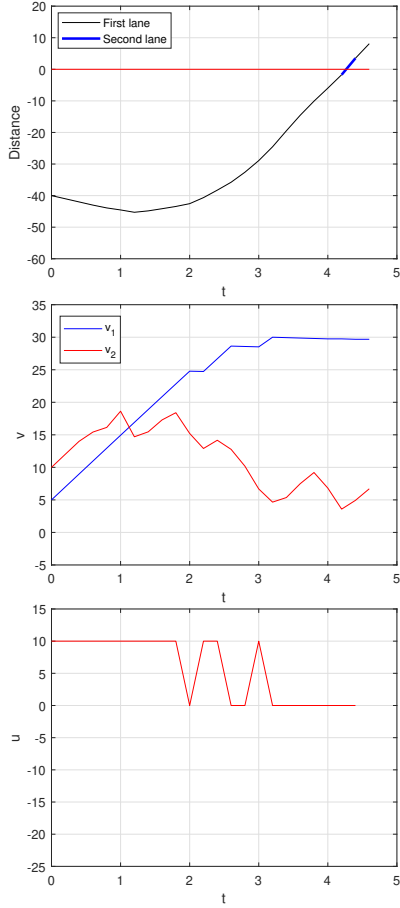


Fig. 8. Signed distance d_k , velocities v_k^1 and v_k^2 , and control u_k .

troller. Figure 8 depicts signed distance d_k , velocities v_k^1 and v_k^2 , and control u_k .

7 Conclusion and future work

In this paper we presented a control synthesis algorithm which enforces a discrete-time dynamical system to satisfy a specification given by a discrete-time hybrid system. Furthermore, we considered such specifications along with terminal conditions to pose safety and reachability synthesis problems. In our future work we plan to use these two problems as building blocks for more complicated specifications.

The synthesis algorithm involves computing symbolic abstractions of the system and the specification and applying a fixed-point type procedure to compute the feedback control and the controllable set. The computation of finite symbolic abstraction is challenging in general but for certain types of systems we are able to point out efficient abstraction algorithms. We mention two such cases (one for dynamical system and one for specifications): monotone systems and piecewise linear hybrid

systems. The examples in the end illustrate the proposed approach.

References

- [1] David Angeli and Eduardo D Sontag. Monotone control systems. *IEEE Transactions on automatic control*, 48(10):1684–1698, 2003.
- [2] Calin Belta, Boyan Yordanov, and Ebru Aydin Gol. *Formal Methods for Discrete-Time Dynamical Systems*. Springer, 2017.
- [3] Samuel Coogan and Murat 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.
- [4] Georgios E Fainekos, Antoine Girard, Hadas Kress-Gazit, and George J Pappas. Temporal logic motion planning for dynamic robots. *Automatica*, 45(2):343–352, 2009.
- [5] Antoine Girard, Gregor Gössler, and Sebti Mouelhi. Safety controller synthesis for incrementally stable switched systems using multiscale symbolic models. *IEEE Transactions on Automatic Control*, 61(6):1537–1549, 2015.
- [6] E. A. Gol, M. Lazar, and C. Belta. Language-guided controller synthesis for linear systems. *IEEE Transactions on Automatic Control*, 59(5):1163–1176, 2014.
- [7] Jun Liu and Necmiye Ozay. Finite abstractions with robustness margins for temporal logic-based control synthesis. *Nonlinear Analysis: Hybrid Systems*, 22:1–15, 2016.
- [8] Jun Liu, Necmiye Ozay, Ufuk Topcu, and Richard M Murray. Synthesis of reactive switching protocols from temporal logic specifications. *IEEE Transactions on Automatic Control*, 58(7):1771–1785, 2013.
- [9] Noorma Yulia Megawati and Arjan van der Schaft. Abstraction and control by interconnection of linear systems: A geometric approach. *Systems & Control Letters*, 105:27–33, 2017.
- [10] Pierre-Jean Meyer, Alex Devonport, and Murat Arcak. TIRA: Toolbox for interval reachability analysis. In *22nd ACM International Conference on Hybrid Systems: Computation and Control*, 2019.
- [11] Pierre-Jean Meyer, Antoine Girard, and Emmanuel Witrant. Safety control with performance guarantees of cooperative systems using compositional abstractions. *IFAC-PapersOnLine*, 48(27):317–322, 2015.
- [12] Petter Nilsson, Omar Hussien, Yuxiao Chen, Ayca Balkan, Matthias Rungger, Aaron D. Ames, Jessy W. Grizzle, Necmiye Ozay, Huei Peng, and Paulo Tabuada. Preliminary results on correct-by-construction control software synthesis for adaptive cruise control. *53rd IEEE Conference on Decision and Control*, pages 816–823, 2014.
- [13] Giordano Pola, Alessandro Borri, and Maria Domenica Di Benedetto. Integrated design of symbolic controllers for nonlinear systems. *IEEE Transactions on Automatic Control*, 57(2):534–539, 2011.
- [14] Giordano Pola and Maria Domenica Di Benedetto. Control of cyber-physical-systems with logic specifications: A formal methods approach. *Annual Reviews in Control*, 2019.
- [15] Gunther Reifig. Computing abstractions of nonlinear systems. *IEEE Transactions on Automatic Control*, 56(11):2583–2598, 2011.

- [16] Gunther Reissig, Alexander Weber, and Matthias Rungger. Feedback refinement relations for the synthesis of symbolic controllers. *IEEE Transactions on Automatic Control*, 62(4):1781–1796, 2016.
- [17] Adnane Saoud, Antoine Girard, and Laurent Fribourg. Contract based design of symbolic controllers for interconnected multiperiodic sampled-data systems. In *57th IEEE Conference on Decision and Control*, 2018.
- [18] Paulo Tabuada. *Verification and Control of Hybrid Systems: a symbolic approach*. Springer, 2008.
- [19] Paulo Tabuada and George J Pappas. Linear time logic control of discrete-time linear systems. *IEEE Transactions on Automatic Control*, 51(12):1862–1877, 2006.
- [20] Harsh Vinjamoor and Arjan J van der Schaft. The achievable dynamics via control by interconnection. *IEEE Transactions on Automatic Control*, 56(5):1110–1117, 2010.
- [21] Majid Zamani, Giordano Pola, Manuel Mazo, and Paulo Tabuada. Symbolic models for nonlinear control systems without stability assumptions. *IEEE Transactions on Automatic Control*, 57(7):1804–1809, 2012.