Safety Controller Synthesis for Incrementally Stable Switched Systems Using Multiscale Symbolic Models
Antoine Girard, Gregor Gössler, Sebti Mouelhi

To cite this version:

HAL Id: hal-01197426
https://hal.archives-ouvertes.fr/hal-01197426v2
Submitted on 19 Feb 2016

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.
Safety Controller Synthesis for Incrementally Stable Switched Systems using Multiscale Symbolic Models

Antoine Girard, Gregor Gössler and Sebti Mouelhi

Abstract—We propose an approach to the synthesis of safety controllers for a class of switched systems, based on the use of multiscale symbolic models that describe transitions of various durations and whose sets of states are given by a sequence of embedded lattices approximating the state-space, the finer lattices being accessible only by transitions of shorter duration. We prove that these multiscale symbolic models are approximately bisimilar to the original switched system provided it enjoys an incremental stability property attested by the existence of a common Lyapunov function or of multiple Lyapunov functions with a minimal dwell-time. Then, for specifications given by a safety automaton, we present a controller synthesis algorithm that exploits the specificities of multiscale symbolic models. We formalize the notion of maximal lazy safety controller which gives priority to transitions of longer durations; the shorter transitions and thus the finer scales of the symbolic model are effectively explored only when safety cannot be ensured at the coarser level and fast switching is needed. We propose a synthesis algorithm where symbolic models can be computed on the fly, this allows us to keep the number of symbolic states as low as possible. We provide computational evidence that shows drastic improvements of the complexity of controller synthesis using multiscale symbolic models instead of uniform ones.

I. INTRODUCTION

Symbolic control approaches, based on the use of discrete abstractions, have become quite popular in the area of hybrid systems (see e.g. [18] and the references therein). In symbolic control, continuous behaviors are abstracted over a finite set of symbols, each symbol representing infinitely many states. The main advantage of these approaches is that they offer the possibility to leverage controller synthesis techniques developed in the area of discrete-event dynamic systems (see e.g. [8]). Also, these approaches allow one to address specifications that are often different from traditional properties in control theory (e.g. stability, controllability, observability...): such specifications can for instance be given by some logic formula or by an automaton describing the acceptable temporal behaviors of the system.

A recent trend in symbolic control is to use discrete abstractions, also called symbolic models, that are related to the original system by some approximate equivalence relationship such as approximate bisimulation [9]. It has been shown that such abstractions are computable for several classes of control systems including incrementally stable switched systems [10], nonlinear systems with or without disturbances [14], [16], [20], time delay systems [15], networked control systems [5] and stochastic systems [22]. These approaches are based on sampling of time and space where the sampling parameters must satisfy some relation in order to obtain abstractions of a prescribed precision. Particularly, the faster the time sampling, the finer the lattice approximating the state-space has to be, resulting in symbolic models with a large number of states.

In this paper, we present a class of multiscale symbolic models for incrementally stable switched systems that allows us to deal with fast time sampling while keeping the number of symbolic states at a reasonable level. Following the self-triggered control paradigm [21], [2], we assume that the controller has to decide the control mode and the duration during which it will be applied. Then, it is natural to consider discrete abstractions where transitions have various durations. For transitions of longer duration, it is sufficient to consider abstract states on a coarse lattice. For transitions of shorter duration, it becomes necessary to use finer lattices. These finer lattices are effectively used only on a restricted area of the state-space where fast time sampling is needed. We prove that these multiscale symbolic models are approximately bisimilar to the original incrementally stable switched system under the existence of a common Lyapunov function or of multiple Lyapunov functions with a minimal dwell-time. Moreover, we show that any precision can be achieved. The concept of approximately bisimilar multiscale abstractions has also been explored in [19] where the multiscale feature is used for accommodating locally the precision of the abstraction while the time sampling period remains constant. On the contrary, our approach seeks for a uniform precision but varying time sampling periods.

In the second part of the paper, we propose to use these multiscale symbolic models for the synthesis of safety controllers for switched systems. For specifications given by safety automata, we introduce the notion of maximal lazy safety controller which exploits the specificities of multiscale symbolic models: it gives priority to transitions of longer durations; the faster transitions and thus the finer scales of the symbolic models are effectively explored only when safety cannot be ensured at the coarser level and fast switching is needed. We present an algorithm computing symbolic models on the fly during controller synthesis, and therefore dynamics at the finest scales are explored only when necessary. We provide experimental results, obtained by the toolbox CoSyMA [12] that show drastic improvements of the complexity of controller synthesis using multiscale models instead of uniform ones defined in [10]. Symbolic controller synthesis algorithms using on the fly computation of uniform symbolic models have also been considered in [13], for specifications described by a target deterministic transition system approximately simulating the observed behavior of the controlled system.

The results presented in this paper appeared partially and in a preliminary formulation in the conference papers [7], [6], [12]. We provide in the following a coherent and improved presentation of the main results of these papers with the following extensions. Firstly, the output of the symbolic models are continuous-time functions instead of sequences of states. While technically simple, this extension allows us to address specifications on continuous-time signals and to consider transitions of long duration without overlooking the behavior of the system between two samples; as far as we know, this is the first work on approximately bisimilar abstractions offering this possibility. Secondly, we address the case of switched systems with multiple Lyapunov functions and dwell-time when we previously
focused on the case of systems with common Lyapunov functions. Thirdly, instead of considering only simple safety specifications (i.e. keep the state of the system within a safe set), we extend our approach to handle more complex properties given by safety specification automata. Finally, we present improved experimental results.

Notations: \( \mathbb{N} \) and \( \mathbb{Z} \) denote the sets of nonnegative integers and of integers, respectively, \( \mathbb{R}, \mathbb{R}_+ \) and \( \mathbb{R}^+ \) denote the sets of real, of non-negative real and of positive real numbers, respectively. For \( x \in \mathbb{R}^n, x[i] \) denotes its \( i \)-th coordinate, \( i = 1, \ldots, n \); \( \|x\| = \sqrt{\sum_{i=1}^n x[i]^2} \) denotes the Euclidean norm of \( x \). Let \( I = [a, b] \) be a compact interval of \( \mathbb{R} \) with \( a < b \), we say that \( f \) is continuous on \( I \), if it is continuous on \( (a, b) \), right-continuous at \( a \) and left-continuous at \( b \). The set of continuous functions from \( I \) to \( \mathbb{R}^m \) is denoted \( C(I, \mathbb{R}^m) \). For \( f \in C(I, \mathbb{R}^m) \), we define \( \|f\| = \sup_{x \in I} \|f(x)\| \). Given a function \( f : \mathbb{R}_+^n \to \mathbb{R}^n \), and \( t \in \mathbb{R}_+ \), we denote by \( f_t \) the restriction of \( f \) to the interval \([0, t]\). A continuous function \( \gamma : \mathbb{R}^+_0 \to \mathbb{R}^+_0 \) is said to belong to class \( K \) if it is strictly increasing, and \( \gamma(0) = 0 \). It is of class \( K_{\infty} \) if it is of class \( K \) and \( \gamma(t) \to +\infty \) as \( t \to +\infty \). A continuous function \( \beta : \mathbb{R}^+_0 \times \mathbb{R}^+_0 \to \mathbb{R}^+_0 \) is said to belong to class \( KL \) if for all fixed \( s \), the map \( \beta(s, \cdot) \) belongs to class \( K \) and for all fixed \( r \) greater than \( s \), the map \( \beta(\cdot, r) \) is strictly decreasing and \( \beta(s, r) = 0 \) when \( s \to +\infty \). Given two sets \( S_1, S_2 \) and a relation \( R \subseteq S_1 \times S_2 \), we denote \( \mathcal{R}(s_1) = \{ s_2 \in S_2 | (s_1, s_2) \in R \} \) and \( R^{-1}(s_2) = \{ s_1 \in S_1 | (s_1, s_2) \in R \} \) for \( S_1 \subseteq S \), \( R(S_1) = \bigcup_{s_1 \in S_1} \mathcal{R}(s_1) \) and similarly for \( S_2 \subseteq S_2 \), \( R^{-1}(S_2) = \bigcup_{s_2 \in S_2} R^{-1}(s_2) \). Given a set \( S \) and a relation \( \leq \subseteq S \times S \) \( \leq \) is a total preorder if and only if:

(1) \( \text{for all } s_1, s_2, s_3 \in S, s_1 \leq s_2 \text{ and } s_2 \leq s_3 \text{ implies } s_1 \leq s_3 \);

(2) \( \text{for all } s_1, s_2 \in S, s_1 \leq s_2 \text{ or } s_2 \leq s_1 \).

We can define the associated equivalence relation \( \equiv \) and strict weak order \( \prec \) given by \( s_1 \equiv s_2 \text{ if and only if } s_1 \leq s_2 \text{ and } s_2 \leq s_1 \).

A. Incrementally stable switched systems

We briefly introduce the class of systems that we consider in this paper.

**Definition 2.1:** A switched system is a quadruple \( \Sigma = (\mathbb{R}^n, P, \mathcal{F}, \alpha) \), where \( \mathbb{R}^n \) is the state space; \( P = \{1, \ldots, m\} \) is the finite set of modes; \( \mathcal{F} \) is a subset of \( \mathbb{R}^n \times \mathbb{R}^m \) which denotes the set of piecewise constant functions from \( \mathbb{R}^+_0 \) to \( \mathbb{R}^n \), continuous from the right and with a finite number of discontinuities on every bounded interval of \( \mathbb{R}^+_0 \); \( \alpha = \{f_1, \ldots, f_m\} \) is a collection of smooth vector fields indexed by \( P \).

A switching signal of \( \Sigma \) is a function \( p \in P \), the discontinuities of \( p \) are called switching times. A continuous function \( x : \mathbb{R}^+_0 \to \mathbb{R}^n \) is said to be a trajectory of \( \Sigma \) if there exists a switching signal \( p \in P \) such that, at each \( t \in \mathbb{R}^+_0 \), the function \( p(t) \) is continuous, \( x \) is continuously differentiable and satisfies:

\[
\dot{x}(t) = f_{p(t)}(x(t)).
\]

We make the assumption that the vector fields \( f_p \) are such that for all initial conditions \( x \in \mathbb{R}^n \), for all switching signals \( p \in P \), there exists a unique trajectory of \( \Sigma \), denoted \( x(x, p) \) or \( x(t, x, p) \) if \( p \) is constant equal to \( p \in P \). Necessary and sufficient conditions to be satisfied by \( f_p \) can be found in [1].

The results presented in this paper apply to switched systems satisfying the incremental stability property (i.e. \( \delta \)-GUAS [10]).

**Definition 2.2:** A switched system \( \Sigma \) is incrementally globally uniformly asymptotically stable (\( \delta \)-GUAS) if there exists a \( KL \) function \( \beta \) such that for all \( t \in \mathbb{R}^+_0 \), for all \( x_1, x_2 \in \mathbb{R}^n \), for all switching signals \( p \in P \), the following condition is satisfied:

\[
\|x(t, x_1, p) - x(t, x_2, p)\| \leq \beta(\|x_1 - x_2\|, t).
\]
outputs $Y$ is equipped with a metric $d$, symbolic if $X$ and $U$ are finite or countable sets.

The transition $(x, u, x', y) \in \Delta$ will be denoted $(x', y) \in \Delta(x, u)$; this means that the system can evolve from state $x$ to state $x'$ under the input $u$, while producing output $y$. An input $u \in U$ belongs to the set of enabled inputs at state $x$, denoted $\text{enab}(x)$, if $\Delta(x, u) \neq \emptyset$. If $\text{enab}(x) = \emptyset$, then $x$ is said to be blocking, otherwise it is said to be non-blocking. The transition system is said to be deterministic if for all $x \in X$ and $u \in \text{enab}(x)$, $\Delta(x, u)$ has only one element; in that case, we shall write with a slight abuse of notation $(x', y) = \Delta(x, u)$.

A trajectory of the transition system is a finite or infinite sequence of transitions $\sigma = (x_0, u_0, y_0)(x_1, u_1, y_1)(x_2, u_2, y_2) \cdots$ where $(x_i+1, y_i) \in \Delta(x_i, u_i)$, for all $i \geq 0$. It is initialized if $x_0 \in X_0$, it is maximal if it is infinite or it is finite and ends in a blocking state. A state $x \in X$ is reachable if there exists an initialized trajectory reaching $x$. The transition system is said to be non-blocking if all initialized maximal trajectories are infinite or equivalent if all reachable states are non-blocking. The output behavior associated to the trajectory $\sigma$ is the sequence of outputs $y_0, y_1, y_2, \ldots$.

In the following, we will consider approximation relationships between transition systems in the sense of approximate bisimulation [9].

**Definition 2.6:** Let $T_1 = (X_1, U, Y, \Delta_1, X_0)$, with $i = 1, 2$ be metric transition systems with the same sets of inputs $U$ and outputs $Y$ equipped with the metric $d$. Let $\epsilon \in \mathbb{R}^+$, $R \subseteq X_1 \times X_2$ is said to be an $\epsilon$-approximate bisimulation relation between $T_1$ and $T_2$ if for all $(x_1, x_2) \in R$, $u \in U$:

$$\forall (x_1', y_1) \in \Delta_1(x_1, u), \exists (x_2', y_2) \in \Delta_2(x_2, u), d(y_1, y_2) \leq \epsilon \text{ and } (x_1', x_2') \in R;$$

$$\forall (x_2', y_2) \in \Delta_2(x_2, u), \exists (x_1', y_1) \in \Delta_1(x_1, u), d(y_1, y_2) \leq \epsilon \text{ and } (x_1', x_2') \in R. \quad (9)$$

**Theorem 3:** Let $\Sigma = (\mathbb{R}_+, P, F)$ be a switched system where $P = \mathcal{S}(\mathbb{R}_+, P)$. We assume that the switching in $\Sigma$ is determined by a self-triggered controller (see e.g. [21], [2]), which not only selects the mode of the switched system but also the duration during which the mode remains active. We should assume that the controller can choose durations from a finite set

$$\Theta_\Sigma^N = \{ \theta_s = 2^{-s} \tau \mid s = 0, \ldots, N \} \quad (12)$$

consisting of dyadic fractions of a time sampling parameter $\tau \in \mathbb{R}^+$ up to some scale parameter $N \in \mathbb{N}$. Let us remark that the shortest control cycle is $2^{-N} \tau$ while the longest one is $\tau$.

The dynamics of the switched system $\Sigma$ is then naturally described by the transition system $T_\Sigma^N = (X, U, Y, \Delta, X_0)$ where

- the set of states is $X = \mathbb{R}^n$;
- the set of inputs consists of pairs of mode and duration $U = \mathcal{P} \times \Theta_\Sigma^N$;
- the set of outputs is the set of continuous functions $Y = \bigcup_{s=0}^N \mathcal{C}([0, \theta_s], \mathbb{R}^m)$;
- the transition relation is given for $x \in X$ and $u = (p, \theta_s) \in U$ by $(x', y) = \Delta(x, u)$ if and only if $x' = x(\theta_s, x, p)$ and $y = x|_{\theta_s}(.., x, p)$.
i.e. the switched system moves from state $x$ to state $x'$ by applying the constant mode $p$ for a duration $\theta_s$; $y$ is the continuous-time trajectory of the switched system connecting $x$ to $x'$;

- the set of initial states is $X^0 = \mathbb{R}^n$.

$T^N_\Sigma$ is deterministic and metric when the set of outputs $Y$ is equipped with the following metric $d$: for $y \in C([0, \theta_s], \mathbb{R}^n)$, $y' \in C([0, \theta_s], \mathbb{R}^n)$

$$d(y, y') = \begin{cases} \|y - y'\| & \text{if } \theta_s = \theta_{s'}, \\
+\infty & \text{if } \theta_s \neq \theta_{s'}, \end{cases}$$ (13)

Note that the state space of $T^N_\Sigma$ is uncountable. Let us remark that an output behavior of $T^N_\Sigma$ is a sequence of continuous functions $y^1, y^2, \ldots$. The concatenation of these functions is itself a continuous function which is a trajectory of switched system $\Sigma$.

The computation of a multiscale symbolic model approximating $T^N_\Sigma$ can be done using the following approach. We approximate the set of states $\mathbb{R}^n$ by a sequence of embedded multiscale lattices $X^s = [\mathbb{R}^n]_{2^{-s} n}$, $s = 0, \ldots, N$ where

$$[\mathbb{R}^n]_{2^{-s} n} = \left\{ q \in \mathbb{R}^n \mid q[i] = k_i 2^{-s-i} n, k_i \in \mathbb{Z}, i = 1, \ldots, n \right\}$$

where $n \in \mathbb{Z}^+$ is a state space sampling parameter. Let us remark that we have $X^0 \subseteq X^1 \subseteq \cdots \subseteq X^N$. We associate a multiscale quantizer $Q^s_n : \mathbb{R}^n \rightarrow [\mathbb{R}^n]_{2^{-s} n}$ such that $Q^s_n(x) = q$ if and only if $q[i] = 2^{-s-i} n \leq x[i] < q[i] + 2^{-s-i} n$, $i = 1, \ldots, n$.

By simple geometrical considerations, we can check that for all $x \in \mathbb{R}^n$ and $s = 0, \ldots, N$, $\|x - Q^s_n(x)\| \leq 2^{-s} n$.

Then, let us define the transition system $T^N_{\tau, \theta_s}(\Sigma) = (X^N_\tau, U, Y, \Delta_\tau, X^0_\tau)$, where

- the set of states is $X^N_\tau = [\mathbb{R}^n]_{2^{-\tau} n}$;
- the set of inputs consists of pairs of mode and duration $U = P \times \Theta^N_\tau$;
- the set of outputs is the set of continuous functions $Y = \bigcup_{s=0}^{N} C([0, \theta_s], \mathbb{R}^n)$;
- the transition relation is given for $q \in X^N_\tau$ and $u = (p, \theta_s) \in U$ by $(q', y) = \Delta_\tau(q, u)$ if and only if $q' = Q^s_n(x(\theta_s, q, p))$ and $y = x|_{\theta_s} (\ldots, q, p)$;
- the set of initial states is $X^0_\tau = [\mathbb{R}^n]_n$.

$T^N_{\tau, \theta_s}(\Sigma)$ is deterministic and metric when the set of outputs $Y$ is equipped with metric defined in (13). Note that it is symbolic since its sets of states and inputs are respectively countable and finite. An output behavior of $T^N_{\tau, \theta_s}(\Sigma)$ is a sequence of continuous functions $y^1, y^2, \ldots$. It should be noted that the concatenation of these functions may not be a continuous function. Hence, the trajectories of $\Sigma$ are approximated by piecewise continuous functions.

The approximation principle is illustrated in Figure 1. It is important to remark that all the transitions of duration $\theta_s$ end in states belonging to $X^s$. This means that the states on the finer lattices are only accessible by transitions of shorter duration. Also, if we only consider transitions of duration $\tau$, that is if $N = 0$, $T^0_{\tau, \theta_s}(\Sigma)$ is similar to the uniform abstraction defined in [10].

Remark 3.1: In [10], the proposed abstraction approach may produce non-deterministic symbolic models. In this paper, the transition relation is defined using the quantizers $Q^s_n$, $s = 0, \ldots, N$ that ensure determinism of the symbolic model. This is important as the controller synthesis algorithm presented in the next section assumes determinism of the transition system.

Remark 3.2: The computation of the transition relation of the symbolic models involves the resolution of the differential equation (1). The exact computation of the solutions may not be possible but these can be approached arbitrarily close using standard numerical algorithms. In the following, we omit the error due to numerical computations for the sake of simplicity; though, these could be easily taken into account.

Theorem 4: Let us assume that the switched system $\Sigma$ admits a common $\delta$-GUAS Lyapunov function $V$ satisfying (7). Let us consider time and state space sampling parameters $\tau, \eta \in \mathbb{R}^+$, scale parameter $N \in \mathbb{N}$, and a desired precision $\varepsilon \in \mathbb{R}^+$. If

$$\eta \leq \min\left\{ \frac{\pi N}{\min_{s=0}^{2 \gamma^{-1} \left(1 - e^{-\gamma \theta_s}\right) \varepsilon} \min_{s=0}^{\pi^{-1} \varepsilon} (2 \gamma^{-1})}, \frac{\pi^{-1} \varepsilon}{\eta} \right\}$$ (14)

then $T^N_{\tau, \eta}(\Sigma) \approx T^N_{\tau, \theta_s}(\Sigma)$.

Proof: We start by showing that the relation $R$ defined by

$$R = \left\{ (x, q) \in X \times X^N_\eta \mid V(x, q) \leq \alpha(\varepsilon) \right\}$$

is an $\varepsilon$-approximate bisimulation relation between $T^N_{\tau, \eta}(\Sigma)$ and $T^N_{\tau, \theta_s}(\Sigma)$. We start by remarking that the transition systems are deterministic. Therefore Lemma 2.7 applies. Let $(x, q) \in R$, we have $\text{enab}(x) = \text{enab}(q) = U$; then let $u = (p, \theta_s) \in U$, $(x', y) = \Delta_\tau(x, u)$ and $(q', z) = \Delta_\tau(q, u)$. From (3), it holds for all $t \in [0, \theta_s]$.

$$V(x(t, x, p), x(t, q, p)) \leq e^{-\eta t} V(x, q) \leq V(x, q) \leq \alpha(\varepsilon).$$

Then, (2) gives for all $t \in [0, \theta_s]$, $\varepsilon$.

$$\|y(t) - z(t)\| = \|x(t, x, p) - x(t, q, p)\|$$

$$\leq \alpha^{-1} (V(x(t, x, p), x(t, q, p))) \leq \varepsilon.$$

It follows that $d(y, z) \leq \varepsilon$. Since $q' = Q^s_n(x(\theta_s, q, p))$, equation (7) yields

$$V(x', q') - V(x', x(\theta_s, q, p)) \leq \gamma(\|q' - x(\theta_s, q, p)\|) \leq \gamma(2^{-s} \eta).$$

Then, it holds

$$V(x', q') \leq V(x', x(\theta_s, q, p)) + \gamma(2^{-s} \eta) \leq V(x(\theta_s, x, p), x(\theta_s, q, p)) + \gamma(2^{-s} \eta) \leq e^{-\eta \theta_s} V(x, q) + \gamma(2^{-s} \eta).$$

Then, it holds

$$V(x', q') \leq e^{-\eta \theta_s} \alpha(\varepsilon) + \gamma(2^{-s} \eta) \leq \alpha(\varepsilon).$$
because of equation (14). Hence, \((x', q') \in R\) and \(R\) is an \(\varepsilon\)-approximate bisimulation relation between \(T^N_c(\Sigma)\) and \(T^N_{\tau q}(\Sigma)\).

For all \(x \in X^0 \subseteq \mathbb{R}^n\), \(q = Q^n_0(x) \in X^0\) satisfies \(\|x-q\| \leq \eta\). Then,
\[
V(x,q) \leq \mathcal{H}(\|x-q\|) \leq \mathcal{H}(\eta) \leq \Theta(\varepsilon)
\]
because of equation (14). Hence, \(X^0 \subseteq R^{-1}(X^0)\). Conversely, for all \(q \in X^0\), \(x = q \in X^0\) satisfies \(V(x,q) = 0\). Hence, \(X^0 \subseteq R(X^0)\).

Therefore, \(T^N_c(\Sigma)\) and \(T^N_{\tau q}(\Sigma)\) are \(\varepsilon\)-approximately bisimilar.

It is interesting to note that given a time sampling parameter \(\tau > 0\) and a scale parameter \(N \in \mathbb{N}\), for any desired precision \(\varepsilon > 0\), there always exists \(\eta > 0\) such that equation (14) holds. This essentially means that approximately bisimilar multiscale symbolic models of arbitrary precision can be computed for \(T^N_c(\Sigma)\). Let us remark that for \(N = 0\) we obtain a result that is similar to that in [10] with the difference that the symbolic models here have continuous-time outputs.

B. Switched systems with dwell-time

We now consider the case of switched systems with minimum dwell-time. Let \(\Sigma_{\tau d} = (\mathbb{R}^n, P, \mathcal{P}, F)\) be a switched system where \(\mathcal{P} = \mathcal{S}_{\tau d}(\mathbb{R}^n, P)\). Let \(\Theta^N_c\) be given as before by (12) where the time sampling parameter \(\tau \in \mathbb{R}^+\) and the scale parameter \(N \in \mathbb{N}\). For simplicity, we make the assumption that there exists \(N_d \in \{0, \ldots, N\}\) such that the minimum dwell-time \(\tau_{d} = \theta_{N_d} = 2^{-N_d}\). Then, let \(\Theta^N_{\tau d}\) be defined as in (12); we have \(\Theta^N_{\tau d} \subseteq \Theta^N_c\).

The dynamics of the switched system \(\Sigma_{\tau d}\) can then be described by the transition system \(T^N_{\tau d}(\Sigma_{\tau d}) = (X, U, Y, \Delta, X^0)\) where

- the set of states is \(X = \mathbb{R}^n \times P\), for \(z = (x,p) \in X\), \(x \in \mathbb{R}^n\) and \(p \in P\) represent the state and the active mode of the switched system respectively;
- the set of inputs consists of pairs of mode and duration \(U = P \times \Theta^N_{\tau d}\);
- the set of outputs is the set of continuous functions \(Y = \bigcup_{\nu=0}^{\infty} C([0,\theta_{\nu}],\mathbb{R})\);
- the transition relation is given for \(z = (x,p) \in X\) and \(u = (p', \theta_{\nu}) \in \text{enab}(z)\) by \(\Delta(x,u)\) if and only if \(x' = (x',p')\) and \(x' = x|_{\theta_{\nu}}(x,p')\) and \(y = y|_{\theta_{\nu}}(x,p')\).

The set of enabled inputs is \(\text{enab}(z) = \{p\} \times \Theta^N_{\tau d} \cup (P \setminus \{p\}) \times \Theta^N_{\tau d}\), i.e., if the current mode is \(p\) and a mode \(p' \neq p\) is applied, it needs to be held for a period of at least \(\theta_{N_d} = \tau_d\); the set of initial states is \(X^0 = \mathbb{R}^n \times P\).

\(T^N_c(\Sigma_{\tau d})\) is deterministic and metric when the set of outputs \(Y\) is equipped with the metric defined in (13). It is symbolic since its sets of states and inputs are respectively countable and finite.

**Theorem 5:** Let us assume that the switched system \(\Sigma_{\tau d}\) admits multiple \(\delta\)-GUAS Lyapunov functions \(V_{p}, p \in P\), satisfying (8). Let us consider time and state space sampling parameters \(\tau, \eta \in \mathbb{R}^+\), scale parameter \(N \in \mathbb{N}\), and a desired precision \(\varepsilon \in \mathbb{R}^+\). If \(\tau_d > \frac{\log \mu}{\kappa}\) and
\[
\eta \leq \min \left\{ \sum_{s=0}^{N_d} \left[ 2^s \gamma^{-1}(\left( \frac{1}{\mu} - e^{-\kappa \theta_{s}} \right) \alpha(\varepsilon)) \right]^{1/2}, \mu^{-1} \right\}
\]
then \(T^N_{\tau d}(\Sigma_{\tau d}) \sim_{\varepsilon} T^N_{\tau q}(\Sigma_{\tau d})\).

**Proof:** We start by showing that the relation \(R\) defined by
\[
R = \left\{ (z,r) \in X \times X^N_c \mid z = (x,p), r = (q,p), V_p(x,q) \leq \frac{\mu}{\kappa} \alpha(\varepsilon) \right\}
\]
is an \(\varepsilon\)-approximate bisimulation relation between \(T^N_c(\Sigma_{\tau d})\) and \(T^N_{\tau q}(\Sigma_{\tau d})\). The transition systems are deterministic, therefore Lemma 2.7 applies. Let \(z, r \in R\), \(z = (x,p), r = (q,p)\), we have \(\text{enab}(z) = \text{enab}(r)\). Hence, approximately bisimilar multiscale symbolic models of arbitrary precision can be computed for \(T^N_c(\Sigma)\). Let us remark that for \(N = 0\) we obtain a result that is similar to that in [10] with the difference that the symbolic models here have continuous-time outputs.
multiscale symbolic models of arbitrary precision can be computed for $T^N_\gamma(\Sigma_{eq})$.

IV. SAFETY CONTROLLER SYNTHESIS USING MULTISCALE SYMBOLIC MODELS

In this section, we present controller synthesis techniques based on the use of the multiscale symbolic models defined in the previous section. We focus on a class of safety specifications. The principle of the proposed solution is to exploit the specific properties of multiscale abstractions: i.e. we give higher priority to transitions of longer duration in order to keep the state of the system as much as possible at the coarser scale. Then, the finer scales are explored only when the specification cannot be met at the coarser level. The symbolic models need not be computed prior to controller synthesis. Their computation can be handled on the fly, thus keeping the effective number of symbolic states at a reasonable level.

In the following, let us consider a transition system $T = (X, U, Y, \Delta, X^0)$ which can be, for instance, one of the multiscale symbolic models defined in the previous section.

A. Safety controllers

In this section, we formalize the problem of safety controller synthesis. We make the following assumption on $T$:

Assumption 1: We assume that $T$ is symbolic and deterministic. We assume that the set of outputs $Y$ consists of continuous functions $y \in C([0, \theta_y], \mathbb{R}^n)$ for some $\theta_y \in \mathbb{R}^+$.

Assumption 1 is satisfied by the multiscale symbolic models defined in the previous section.

Definition 4.1: A safety specification automaton is a tuple $S = (Q, E, I, G, Q^0)$ consisting of a finite set of states $Q$; a transition relation $E \subseteq Q \times Q$; a set of invariants $I = \{I_q \subseteq \mathbb{R}^n, q \in Q\}$; a set of guards $G = \{G_e \subseteq \mathbb{R}^n, e \in E\}$; a set of initial states $Q^0 \subseteq Q$.

We shall make the following assumptions on the specification automaton:

Assumption 2: There is a minimum separation between guards of successive transitions: there exists $\varepsilon > 0$ such that, for all $e = (q, q'), e' = (q'', q'') \in E$,

$$\inf\{|x - x'| \mid x \in G_e, x' \in G_{e'}\} \geq \varepsilon.$$

We now define the composition of a transition system with a safety specification automaton:

Definition 4.2: Under Assumptions 1 and 2, the composition of the transition system $T = (X, U, \Delta, X^0)$ with the safety automaton $S = (Q, E, I, G, Q^0)$ is the transition system $T||S = (X_S, U_S, Y, \Delta_S, X^0_S)$ where the set of states $X_S = Q \times X$; the set of inputs is $U_S = Q \times U$; the set of outputs is $Y_S$; the set of initial states is $X^0_S = Q^0 \times X^0$; the transition relation is given by $z = (q, x), z' = (q', x') \in X_S, v = (q'', u) \in U_S$ and $y \in Y$ by $(z', y) \in \Delta_S(z, v)$ if and only if $u \in \text{enab}(z), (x', y) = \Delta(x, u)$, $q' = q''$ and one of the following conditions holds:

- $q = q'$ and $\forall t \in [0, \theta_y], y(t) \in I_q$;
- there exists $0 = t_0 \leq t_1 \leq \cdots \leq t_N \leq t_{N+1} = \theta_y$ and $e_i = (q_i, q_{i+1}) \in E, i = 0, \ldots, N$ with $q_0 = q, q_N = q'$ and

$$\left\{ \begin{array}{ll} \forall t \in [t_i, t_{i+1}], y(t) \in I_{q_i}, & i = 0, \ldots, N; \\ y(t_{i+1}) \in C_{e_i}, & i = 0, \ldots, N - 1. \end{array} \right.$$ 

If $u \in \text{enab}(z), (x', y) = \Delta(x, u)$ and $q' = q''$, or if $u \notin \text{enab}(z)$, then $\Delta_S(z, v) = \emptyset$.

Remark 4.3: For the second type of transitions defined above, we would like to point out that $y$ is continuous on the compact set $[0, \theta_y]$, therefore it is uniformly continuous, then Assumption 2 on minimum separation of guards of successive transitions implies that there exists $h > 0$ such that $t_{i+1} - t_i \geq h$ for $i = 1, \ldots, N - 1$. Hence, $N$ is bounded above by $1 + \theta_y/h$ thus ruling out potential Zeno behaviors (see e.g. [23]).

Remark 4.4: The specification formalism of safety specification automata is derived from the classical notion of hybrid automata [11], also used as a specification formalism in [3]. It can be shown that any regular safety property over state predicates (i.e. whose set of bad prefixes is recognized by a finite state automaton, as defined in [41]) can be written under the form of a safety specification automaton.

We claim the following properties of the transition system $T||S$:

Lemma 4.5: $T||S$ is a symbolic and deterministic transition system.

Proof: The fact that $T||S$ is symbolic is a consequence of $T$ being symbolic and $Q$ finite. To show that it is deterministic, let $z = (q, x), v = (q'', u) \in U_S$ such that $\Delta_S(z, v) \neq \emptyset$ then let $(z', y) \in \Delta(z, v)$ with $z' = (q', x')$. From the Definition 4.2, it follows that $q' = q''$ and $(x', y) = \Delta(x, u)$. Thus, $(z', y)$ is uniquely determined and $T||S$ is deterministic.

A trajectory of $T$, $\sigma = (x^0, u^0, y^0), (x^1, u^1, y^1), (x^2, u^2, y^2), \ldots$ is safe according to the safety specification automaton $S$ if it is infinite and there exists a trajectory of $T||S$, $\sigma_S = (z^0, v^0, y^0), (z^1, v^1, y^1), (z^2, v^2, y^2), \ldots$ with $z^i = (q', x')$, $v^i = (q', u')$, for all $i \geq 0$. A safety controller is then a controller that prevents $T||S$ from reaching a blocking state.

Definition 4.6: A safety controller for $T||S = (X_S, U_S, Y, \Delta_S, X^0_S)$ is a relation $C \subseteq X_S \times U_S$ such that for all $z \in X_S$:

- $C(z) \subseteq \text{enab}(z)$;
- if $C(z) \neq \emptyset$, then for all $u \in C(z)$ with $\Delta_S(z, v) = (z', y)$, it holds that $C(z') \neq \emptyset$.

We denote the domain of $C$ as $\text{dom}(C) = \{z \in X_S \mid C(z) \neq \emptyset\}$. The controller is given by the transition system $T||S/C = (X_S, U_S, Y, \Delta_S/C, X^0_S)$ where the transition relation is given for $z \in X_S, v \in U_S$ by $(z, y) \in \Delta_S/C(z, v)$ if and only if $u \in C(z)$ and $(z', y) = \Delta_S(z, v)$; the set of initial states is $X^0_{S/C} = X^0_S \cap \text{dom}(C)$.

Lemma 4.7: $T||S/C$ is a symbolic, deterministic, and non-blocking transition system.

Proof: The fact that it is symbolic and deterministic comes directly from Lemma 4.5. From the first point of Definition 4.6 and by definition of $\Delta_S/C$, a state $z \in X_S$ of $T||S/C$ is non-blocking if and only if $C(z) \neq \emptyset$. Then, by definition of $X^0_{S/C}$, all initial states of $T||S/C$ are non-blocking. Also, by the second point of Definition 4.6 and by definition of the transition relation $\Delta_S/C$, all reachable states of $T||S/C$ are non-blocking.

Thus, the previous lemma shows that all maximal trajectories of $T||S/C$ are infinite trajectories of $T||S$ and thus generates safe trajectories of the transition system $T$. There are in general several safety controllers, however, one can show that there exists one that is maximal:

Lemma 4.8: There exists a unique maximal safety controller $C^* \subseteq X_S \times U_S$ such that for all safety controllers $C, C \subseteq C^*$.

Proof: We can check from Definition 4.6 that the union of safety controllers is again a safety controller. Then, the union of all safety controllers is a safety controller which contains all the others, it is clearly unique.

The previous result justifies the following notion of controllability:
Definition 4.9: A state $z \in X_S$ of $T||S$ is safety controllable if and only if $z \in \text{dom}(C^*)$. The set of safety controllable states is denoted $\text{cont}(T||S)$. Of course from the previous definition, it follows that $\text{cont}(T||S) = \text{dom}(C^*)$. Also, it can be easily established (see e.g. [18]) that for all $z \in X_S$:

$$C^*(z) = \left\{ v \in \text{enab}(z) \mid z' \in \text{cont}(T||S) \text{ with } (z', y) = \Delta_S(z, v) \right\}.$$  \hfill (16)

Remark 4.10: A classical safety specification consists in keeping the output values of $T$ in a given set $I_S \subseteq \mathbb{R}^n$. This is a special case of Definition 4.1 corresponding to the safety specification automaton $S = (Q, E, I, G, Q^0)$ with a single state $Q = \{ q \}$, no transitions $E = \emptyset$, the invariant $I = \{ I_S = I \}$, $G = \emptyset$, and $Q^0 = Q$. o

Remark 4.11: Continuous-time outputs do not play a role in the characterization of a safety controller given by Definition 4.6. Then, as far as controller synthesis is concerned, the transition system $T||S$ can be considered as a purely discrete transition system (without outputs) and any discrete controller synthesis technique can be used. In the following, we propose an approach exploiting the properties of multi-scale symbolic models.

B. Maximal lazy safety controller

Let us remark that the maximal safety controller $C^*$ is computable using a simple fixed point algorithm (see e.g. [17, 18]). Termination of the algorithm is guaranteed if the number of non-blocking states of $T||S$ is finite. This is the case for the symbolic models defined in Section III provided that invariants and guards of the safety specification automaton are bounded sets. However, the larger the number of states in the symbolic models, the more expensive the computation.

For that reason, we want to exploit multiscale symbolic models to propose a more efficient algorithm for the synthesis of safety controllers. The lazy safety synthesis problem consists in controlling the system $T||S$ so as to prevent trajectories from reaching a blocking state, while applying at each step a transition of the longest duration for which safety can be guaranteed.

Assumption 3: We assume that the set of inputs $U$ is finite and equipped with a priority relation given by a total preorder $\leq U \times U$.

For the multiscale symbolic models where $U = P \times \Theta^N$, for $u = (p, \theta_i)$, $u' = (p', \theta'_i) \in U$, we give priority to transitions of longer duration by defining $u \preceq u'$ if and only if $\theta_i \leq \theta'_i$. The associated equivalence relation $\sim$ and strict weak order $\prec$ are then given by $u \sim u'$ if and only if $\theta_i = \theta'_i$ and $u \prec u'$ if and only if $\theta_i < \theta'_i$.

We lift the preorder $\leq \subseteq U \times U$ to the set $U_\beta = Q \times U$ as follows, for $v = (q, u), v' = (q', u') \in U_\beta$, $v \preceq v'$ if and only if $u \preceq u'$. The associated equivalence relation $\sim$ and strict weak order $\prec$ are lifted similarly. Since $U_\beta$ is finite, we can define for any subset $V \subseteq U_\beta$, $\max_\sim(V) = \{ v \in V \mid \forall v' \in V, v' \preceq v \}$.

We can now formalize the notion of maximal lazy safety controller:

Definition 4.12: A maximal lazy safety (MLS) controller for $T||S = (X_S, U, Y, \Delta_S, X^0_S)$ is a safety controller $C \subseteq X_S \times U_S$ such that:

- all safety controllable states are in $\text{dom}(C)$:
  \[ X^0_S \cap \text{cont}(T||S) \subseteq \text{dom}(C); \]
- all states $z \in \text{dom}(C)$ are reachable in $T||S/C$;
- for all states $z \in \text{dom}(C)$:
  \[ 1) \text{ if } v \in C(z), \text{ then for all } v' \in \text{enab}(z) \text{ with } v \simeq v', (z', y) = \Delta_S(z, v'), \text{ it holds that } v' \notin C(z) \text{ if and only if } z' \notin \text{cont}(T||S); \]
  \[ 2) \text{ if } v \notin C(z), \text{ then for all } v' \in \text{enab}(z) \text{ with } v \prec v', (z', y) = \Delta_S(z, v'), \text{ it holds that } z' \notin \text{cont}(T||S). \]

The term maximal comes from the fact that all safety controllable initial states are in $\text{dom}(C)$, and if the controller enables an input, it also enables all inputs with the same priority and which preserve safety. The term lazy refers to the fact that when several inputs can preserve safety, the controller enables only inputs with highest priority (i.e. with longer duration for our multiscale symbolic models).

Hence, the maximal lazy safety controller $C$ represents a trade-off between maximal permissiveness and efficiency.

Theorem 6: There exists a unique MLS controller for $T||S$.

Proof: We first prove existence and then uniqueness.

Existence: Let $C^*$ be the maximal (non-lazy) safety controller for $T||S$. Let $C^*$ be the controller defined from $C^*$ as follows: for all $z \in X_S$, $C^*(z) = \max_\prec C^*(z)$. Then, we have $\text{dom}(C^*) = \text{cont}(T||S)$.

Let us show that $C^*$ is a safety controller for $T||S$. Let $z \in X_S$, we have $C^*(z) \subseteq C^*(z) \subseteq \text{enab}(z)$; hence the first condition of Definition 4.6 holds. If $C^*(z) \neq \emptyset$, then for all $v \in C^*(z)$, we also have $v \in C^*(z)$ and therefore for $(z', y) = \Delta_S(z, v)$, it holds $C^*(z') \neq \emptyset$ which yields $C^*(z') \neq \emptyset$; hence the second condition of Definition 4.6 holds as well. Now let us show that $C^*$ satisfies conditions (1) and (2) of Definition 4.12. Let $z \in \text{dom}(C^*)$, and $v \in \text{enab}(z)$ with $v \prec v'$, $(z', y) = \Delta_S(z, v')$. If $v' \in C^*(z)$ then since $C^*$ is a safety controller, we have $z' \in \text{dom}(C^*) = \text{cont}(T||S)$.

If $z' \notin \text{cont}(T||S)$, then from (16) that $v' \in C^*(z)$. Since $v \in \max_\prec C^*(z)$ and $v \prec v'$, it follows that $v' \in \max_\prec C^*(z)$. Hence, $v' \in C^*(z)$ and condition (1) of Definition 4.12 holds. Let $z \in \text{dom}(C^*)$, $v \in C^*(z)$, and $v' \in \text{enab}(z)$ with $v \prec v'$, $(z', y) = \Delta_S(z, v')$. Since $C^*(z) = \max_\prec C^*(z)$, it follows since $v \prec v'$ that $v' \notin C^*(z)$. From (16), we have $z' \notin \text{cont}(T||S)$. Hence, condition (2) of Definition 4.12 holds as well.

Now let $C$ be the controller defined from $C^*$ by $C(z) = C^*(z)$ if $z$ is reachable in $T||S/C^*$ and $C(z) = \emptyset$ otherwise. It is clear that the reachable states in $T||S/C^*$ and $T||S/C$ are the same. Hence, for all $z \in \text{dom}(C) \subseteq T||S/C$. Moreover, it follows from the properties of $C^*$ that $C$ is a safety controller for $T||S$, and that C satisfies conditions (1) and (2) of Definition 4.12. Let $z \in X^0_S \cap \text{cont}(T||S) = X^0_S \cap \text{dom}(C^*) = X^0_S/C^*$. Since any initial state is reachable, $z$ is reachable in $T||S/C^*$. Therefore, we have $C(z) = C^*(z) \neq \emptyset$ and $z \in \text{dom}(C)$.

Uniqueness: Let $C_1$ and $C_2$ be two MLS controllers and assume that there exists $z \in X_S$ such that $C_1(z) \neq C_2(z)$. If both $C_1(z)$ and $C_2(z)$ are not empty, we can assume without loss of generality that there exists $v_1 \in C_1(z)$ such that $v_1 \notin C_2(z)$. Then, let $v_2 \in C_2(z)$. If $v_1 \prec v_2$ then condition (2) of Definition 4.12 does not hold for $C_1$ since $\Delta_S(z, v_1) \in \text{dom}(C_1) \subseteq \text{cont}(T||S)$. If $v_2 \prec v_1$ then condition (2) of Definition 4.12 does not hold for $C_2$ since $\Delta_S(z, v_1) \in \text{dom}(C_1) \subseteq \text{cont}(T||S)$. If $v_1 \simeq v_2$, then condition (1) of Definition 4.12 does not hold for $C_2$ since $\Delta_S(z, v_1) \in \text{dom}(C_1) \subseteq \text{cont}(T||S)$. In all the cases one of the controllers is not a MLS controller.
Algorithm 1: MLS controller synthesis

Input: Transition system $T|\{S = (x, u, y, \Delta, x_0^u)}$, priority relation $\preceq \subseteq u_S \times u_S$

Output: MLS controller $C \subseteq X_S \times u_S$

1. **Global variables**: controllable states $X_c \subseteq X_S$, uncontrollable states $X_u \subseteq X_S$

2. **begin**

3. \[ (x, x_u, C) := (\emptyset, \emptyset, \emptyset) ; \]

4. \[ \text{for } z \in x_0^u \text{ do} \]

5. \[ \text{explore} (z, \emptyset) ; \]

6. **return** $C$

$z^0 \in \text{dom}(C_1)$ is a controllable initial states and therefore $C_2(z^0) \neq \emptyset$. Then, there exists $i \in \{0, \ldots, N-1\}$ such that $C_2(z^i) \neq \emptyset$ and $v_i \notin C_2(z^i)$ otherwise we would have $C_2(z) \neq \emptyset$. Therefore, there exists $z' \in X_S$, such that $C_1(z') \neq C_2(z')$ and both $C_1(z')$ and $C_2(z')$ are not empty. We have already proved that in this case one of the controllers is not a MLS controller.

In the proof of Theorem 6, we give a construction of the MLS controller. Intuitively, the MLS controller for $T|\{S$ can be obtained from the maximal safety controller by selecting the enabled inputs of higher priority and by removing the states that are not reachable.

Of course, this construction does not lead to an efficient algorithm for the computation of the MLS since it needs first to compute the maximal safety controller. In the following section, we address the problem of computing effectively the MLS controller.

C. Controller synthesis

In this section, we present an algorithm for synthesizing the maximal lazy safety controller. It is based on a depth first search exploration of the trajectories, starting from initial states and exploring transitions of higher priority first.

More precisely, the MLS controller is computed by Algorithm 1 which calls the function $\text{explore}(z, \emptyset)$ on each initial state $z \in X_0^S$; the second argument $X_v$ of the function $\text{explore}$ is the set of states already visited by the current trajectory. The global variables are $X_c$, $X_u$, and $C$ for the sets of controllable and uncontrollable states and the controller, respectively. Function $\text{explore}(z, X_v)$ returns whether $z$ is controllable. This is done by recursively exploring the path starting from $z$ until either a controllable or an uncontrollable state is reached; or a state already visited by the current trajectory is reached, which means that a circular path containing $z$ has been found, and therefore the state $z$ is controllable. The outer loop explores inputs of decreasing priority as long as no controllable successor of $z$ has been found.

**Theorem 7**: Let $C$ be computed by Algorithm 1. Then, $C$ is the MLS controller for $T|\{S$.

**Proof**: The set of visited states $X_v$ contains, at each call to $\text{explore}$, the set of states visited along one path from some initial state $z^0 \in X_0^S$ to (excluding) the current state $z$. A state $z$ is determined controllable if it:

- has one immediate controllable successor (line 19), in that case $z$ and all states in $X_v$ are added recursively to the set of controllable states $X_c$;
- has already been visited along the current path $X_v$ (i.e. $z \in X_v$) (line 8) — hence, lying on a cyclic path of safe states, in that case all states in $X_v$ (including $z$) are added recursively to the set of controllable states $X_c$.

In contrast, $z$ is determined controllable if all its successors are uncontrollable (line 22), in that case $z$ is added to the set of uncontrollable states $X_u$. Hence, $X_c$ and $X_u$ contains all the controllable and uncontrollable states that have been explored, respectively. Let us also remark that $C(z) \neq \emptyset$ if and only if $z \in X_c$.

Let us verify that $C$ is a safety controller. By construction (line 9), we have $C(z) \subseteq \text{enab}(z)$. Then, assume $C(z) \neq \emptyset$, and let $v \in C(z)$, $(z', y) = \Delta_S(z, v)$, then it holds that $z'$ is controllable (line 14). Thus $z' \in X_c$ and $C(z') \neq \emptyset$.

We now prove that $C$ is the MLS controller. In Algorithm 1, the function $\text{explore}$ is called for all initial states in $X_0^S$. Then, it follows that all controllable initial states are in $X_0^S$, the first point of Definition 4.12 holds. Since Algorithm 1 explores trajectories starting from initial states, all states in dom$(C) = X_c$ are reachable in $T|\{S/C$, the second point of Definition 4.12 holds. As for the third point, let $z \in \text{dom}(C)$ and $v \in C(z)$. The outer loop of the function $\text{explore}$ (lines 10 to 20) explores trajectories using inputs of higher priority first. Hence, if $v \in C(z)$ then all inputs $v' \in \text{enab}(z)$ with $v \prec v'$ have been explored and none of them leads to a controllable state. Hence, 2) in the third point of Definition 4.12 holds. The inner loop of the function $\text{explore}$ (lines 12 to 16) explores trajectories using all inputs with the same level of priority. Hence, if $v \in C(z)$ then all inputs $v' \in \text{enab}(z)$ with $v \approx v'$ have been explored and $v' \in C(z)$ if an only if it leads to a controllable state. Hence, 1) in the third point of Definition 4.12 holds. Thus, $C$ is the MLS controller for $T|\{S$.

In Algorithm 1, each transition initiating from a non-blocking state is explored at most once. Hence, termination of Algorithm 1 is guaranteed if the sets of inputs $U_S$ and of non-blocking states are finite: note that this is the case for our multiscale symbolic models when invariants and guards of the safety specification automaton are bounded sets. In the worst case, (when all non-blocking states are reachable but none is controllable), all the transitions initiating from a non-blocking state need to be explored. This provides us with a
worst-case (time and space) complexity given by $|X'_S| \times |I_S|$ where $X'_S$ denote the set of non-blocking states. However, in practice this upper-bound is not attained.

V. Computational results

In this section, we show some computational results of our approach obtained using CoSyMA [12], a tool for automatic controller synthesis for incrementally stable switched systems based on multiscale symbolic models. It is written in OCaml and provides an implementation of Algorithm 1 for symbolic models of switched systems (with or without dwell time) and simple safety specifications (see Remark 4.10). The results reported in the following have been obtained on a laptop with i7 processor and 4 GB RAM.

A. DC-DC Converter

As a first case study, we apply our approach to a boost DC-DC converter. It is a switched system with two modes, the two dimensional dynamics associated with both modes are affine of the form $\dot{x}(t) = A_{p(t)}x(t) + b$ for $p(t) \in \{1, 2\}$ (see [10] for numerical values). It can be shown that it has a common $\delta$-GUAS Lyapunov function and thus approximately bisimilar symbolic models can be computed. We consider the problem of keeping the state of the system in a desired region of operation given by the safe set $I_S = [1.15, 1.55] \times [5.45, 5.85]$.

We use approximately bisimilar symbolic models to synthesize MLS controllers for the DC-DC converter. We compare the cost of controller synthesis for the uniform symbolic model $T^{\alpha}_{T_{1,2}}(\Sigma)$ for parameters $\tau_1 = 0.5$ and $\eta_1 = 3 \times 10^{-4}$ (containing only transitions of duration $0.5s$) and the multi-scale symbolic model $T^{\alpha}_{T_{2,2}}(\Sigma)$ for parameters $\tau_2 = 64\tau_1$ and $\eta_2 = 64\eta_1$ (containing transitions of durations in $\Theta^\varepsilon = \{32, 16, 8, 4, 2, 1, 0.5\}$). The set of states of $T^{\alpha}_{T_{2,2}}(\Sigma)$ consists of a set of 7 embedded lattices whose finest one coincides with the set of states of $T^{\alpha}_{T_{1,2}}(\Sigma)$. These two symbolic models have the same precision $\varepsilon = 0.05$, according to Theorem 4.

Table I details the experimental results obtained for the synthesis of the $T^{\alpha}_{T_{1,2}}(\Sigma)$ and $T^{\alpha}_{T_{2,2}}(\Sigma)$. We can see that there is a noteworthy reduction of the time used to compute the controller using a multiscale symbolic model instead of using a uniform one (up to a 93% improvement between $T^{\alpha}_{T_{1,2}}(\Sigma)$ and $T^{\alpha}_{T_{2,2}}(\Sigma)$). This is due to the fact that the size of uniform symbolic models grows exponentially with higher resolutions, whereas multiscale symbolic models are refined only when we get closer to unsafe regions (size reduced by more than 99% between $T^{\alpha}_{T_{1,2}}(\Sigma)$ and $T^{\alpha}_{T_{2,2}}(\Sigma)$). Interestingly, this reduction in computation time and size does not affect the performance of the multi-scale controllers, which yield a ratio of controllable initial states (defined as $|\text{dom}(C)\cap X^S_S|/|X^S_S\cap I_S|$) comparable to that of their uniform counterparts. Figure 2 depicts the maximal lazy safety controller for $T^{\alpha}_{T_{2,2}}(\Sigma)$ and a trajectory of the controlled switched system.

<table>
<thead>
<tr>
<th>Uniform symbolic model</th>
<th>Multiscale symbolic model</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N = 0, \tau = 0.5$</td>
<td>$N = 6, \tau = 32$</td>
</tr>
<tr>
<td>$\eta = 0.0003, \varepsilon = 0.05$</td>
<td>$\eta = 0.018, \varepsilon = 0.05$</td>
</tr>
</tbody>
</table>

**Table I**

<table>
<thead>
<tr>
<th>Time</th>
<th>Size ($10^5$)</th>
<th>Durations</th>
<th>Cont. Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.2s</td>
<td>936</td>
<td>0.5 (100%)</td>
<td>93%</td>
</tr>
</tbody>
</table>

**Experimental results for the MLS controller synthesis for the boost DC-DC converter**

![Figure 2: MLS controller for $T^{\alpha}_{T_{2,2}}(\Sigma)$ and $I_S$. Top: mode 1 is enabled (light gray); mode 2 is enabled (dark gray); modes 1 and 2 are enabled (medium gray); and a trajectory. Bottom: duration 4 (light gray), 2 (medium gray), 1 (dark gray), 0.5 (black) are enabled.**

B. Temperature regulation in an n-room building

The second case study deals with temperature regulation in a circular building with $n$ rooms. Each room is equipped with a heater and at a given instant at most one heater is switched on. The temperature $T_i(t)$ of the room $i, 1 \leq i \leq n$, is defined by the differential equation

$$T_i(t) = \alpha(T_{i+1}(t) + T_{i-1}(t) - 2T_i(t)) + \beta(t_e - T_i(t)) + \gamma(t_h - T_i(t))u_i(t)$$

where $T_{i-1}(t)$ is the temperature of the room $i-1$; $T_{i+1}(t)$ the temperature of the room $i+1$ (with the convention that $T_0(t) = T_n(t)$ and $T_{n+1}(t) = T_1(t)$); $t_e$ is the temperature of the external environment of the building; $t_h$ is the temperature of the heater; $\alpha$ is the conduction factor between the rooms $i \pm 1$ and the room $i$; $\beta$ is the conduction factor between the external environment and the room $i$; $\gamma$ is the conduction factor between the heater and the room $i$; $u_i(t)$ equals to 1 if the room $i$ is heated, or 0 otherwise.

Given a number $n \geq 2$ of rooms, we distinguish $n+1$ switching modes. For $1 \leq i \leq n$, the mode $p_i$ represents the mode of activating the heater of room $i$. The mode $p_{n+1}$ represents that no heater is activated. The values of $\alpha$, $\beta$, $\gamma$, $t_e$, and $t_h$ are respectively...
19.5
19.5
20.0
21.5
21.5
20.0
20.5
20.5
21.5
21.5

Fig. 3. Partial view of the MLS controller for $T^3_{\tau,\eta}((\Sigma_{\tau})_d)$ and $I_S$ for the 4 room building in the plane $(T_1, T_2)$ with $T_3 = T_4 = 19$. Left: mode 1 is enabled (light gray); mode 2 is enabled (black); modes 1 and 2 are enabled (gray); other modes are not displayed. Right: duration 20 (light gray), 10 (gray), 5 (black) are enabled.

Fig. 4. Mode map of the MLS controller for $T^3_{\tau,\eta}((\Sigma_{\tau})_d)$ and $I_S \setminus I_I$ for a switched system with dwell-time. Top: mode 1 is active. Bottom: mode 2 is active. Mode 1 is enabled (black), mode 2 is enabled (light gray), modes 1 and 2 are enabled (gray).

Following the approach in [10] (extended to deal with continuous-time outputs), a controller is computed using a uniform symbolic model with dwell-time for time and state sampling parameters $\tau_1 = 0.5$ and $\eta_1 = 1/(100\sqrt{2})$. Let us remark that the uniform abstraction with dwell time can not be defined as described in Section III-B since we have $\tau_d > \tau_1$. The computation of the controller using a Matlab script takes 160 seconds. The resulting controller contains 5228091 states.

We computed the MLS controller for the multiscale symbolic model $T^3_{\tau_2,\eta_2}((\Sigma_{\tau_d})_d)$ for parameters $\tau_2 = 8\tau_1$, $\eta_2 = 8\eta_1$ (containing transitions of durations in $\Theta^3_{\tau_d} = \{4, 2, 1, 0.5\}$). The set of states of $T^3_{\tau_2,\eta_2}((\Sigma_{\tau_d})_d)$ consists of a set of 4 embedded lattices whose finest one coincides with the set of states of the uniform abstraction. The two symbolic models have the same precision $\varepsilon = 0.4$. The computation of the MLS controller took 7.3 seconds. The resulting controller contains 33826 states, a notable reduction compared to the uniform symbolic model.

C. Switched system with dwell-time

The third case study is taken from [10] and illustrates the case of switched systems without a common $\delta$-GUAS Lyapunov function. The system has two modes and the state space is $\mathbb{R}^2$. The dynamics associated with both modes are affine of the form $x(t) = A_p(t)x(t) + b_p(t)$ for $p(t) \in \{1, 2\}$ with

$$A_1 = \begin{bmatrix} -0.25 & 1 \\ -2 & 0.25 \end{bmatrix}, \quad A_2 = \begin{bmatrix} -0.25 & 2 \\ -1 & 0.25 \end{bmatrix},$$

$$b_1 = \begin{bmatrix} -0.25 & -2 \end{bmatrix}^T, \quad b_2 = \begin{bmatrix} 0.25 & 1 \end{bmatrix}^T.$$ 

The system does not have a common $\delta$-GUAS Lyapunov function but admits multiple $\delta$-GUAS Lyapunov functions (see [10] for details). Then, restricting the set of switching signals to those having dwell time $\tau_d = 2$, the switched system is incrementally stable and admits approximately bisimilar uniform and multiscale symbolic models. We consider the same safety specification as in [10] which consists in keeping the state in $I_S = [-6, 6] \times [-4, 4]$ while avoiding $I_u = [-1.5, 1.5] \times [-1, 1]$.

VI. CONCLUSION

In this paper we have proposed the use of multiscale symbolic models for the computation of controllers for switched systems, by applying them to the specific case of safety problems. We have proposed a construction of multiscale symbolic models and proved
that they are approximately bisimilar to the original switched systems under the existence of (common or multiple) 6-GUAS Lyapunov functions. We have considered the use of these models for synthesizing controllers enforcing properties given by safety specification automata. An appropriate formulation of the synthesis problem and an algorithm have been proposed, exploiting the characteristics of the multiscale symbolic models. Experimental results show a major improvement of computational complexity when using multiscale symbolic models instead of uniform ones.

REFERENCES

Sebti Mouelhi received the M.S. degree in computer science from the University of Lorraine, Nancy, France, in 2007 and the Ph.D. in computer science from the University of Franche-Comté, Besançon, France, in 2011. From October 2011 to September 2012, he was hired as a postdoctoral researcher at INRIA Grenoble, France. From October 2012 to May 2015, he was research and development engineer at SafeRiver, Montrouge, France. During Summer 2015, he was engineer in safety assurance at ALSTOM Transport, Saint-Ouen, France. Since September 2015, he is teacher and researcher at École Centrale d'Électronique (ECE Paris), member of Laureate International Universities.

His activities are mainly in the topic of the formal design and verification of component-based, real-time, and safety-critical embedded systems, especially railway and automotive systems. He is also interested in the use of formal methods for the control of hybrid and cyber-physical systems.