Synthesis of Switching Controllers using Approximately Bisimilar Multiscale Abstractions*

Javier Cámara
POP ART project
INRIA Grenoble Rhône-Alpes
38334 Saint-Ismier cedex, France

Antoine Girard
Laboratoire Jean Kuntzmann
Université de Grenoble
B.P. 53, 38041 Grenoble, France

Gregor Gössler
POP ART project
INRIA Grenoble Rhône-Alpes
38334 Saint-Ismier cedex, France

ABSTRACT

When available, discrete abstractions provide an appealing approach to controller synthesis. Recently, an approach for computing discrete abstractions of incrementally stable switched systems has been proposed, using the notion of approximate bisimulation. This approach is based on sampling of time and space where the sampling parameters must satisfy some relation in order to achieve a certain precision. Particularly, the smaller the sampling period, the finer the lattice approximating the state-space and the larger the number of states in the abstraction. This renders the use of these abstractions for synthesis of fast switching controllers computationally prohibitive. In this paper, we present a novel class of multiscale discrete abstractions for switched systems that allows us to deal with fast switching while keeping the number of states in the abstraction at a reasonable level. The transitions of our abstractions 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 the fast switching occurs. We show how to use these abstractions for multiscale synthesis of self-triggered switching controllers for reachability specifications under time optimization. We illustrate the merits of our approach by applying it to the boost DC-DC converter.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods and Search—*Control theory*

General Terms

Design, Performance

*This work was supported by the Agence Nationale de la Recherche (VEDECY project - ANR 2009 SEGI 015 01).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Copyright 200X ACM X-XXXXX-XX-X/XX/XX ...$10.00.

Keywords

Switched systems, Multiscale abstractions, Approximate bisimulation, Optimal control, Self-triggered controllers.

1. INTRODUCTION

The use of discrete abstractions for continuous dynamics has become standard in hybrid systems design (see e.g. [15] and the references therein). The main advantage of this approach is that it offers the possibility to leverage controller synthesis techniques developed in the areas of supervisory control of discrete-event systems [13] or algorithmic game theory [3]. Historically, the first attempts to compute discrete abstractions for hybrid systems were based on traditional systems behavioral relationships such as simulation or bisimulation [10], initially proposed for discrete systems most notably in the area of formal methods. These notions require inclusion or equivalence of observed behaviors which is often too restrictive when dealing with systems observed over metric spaces. For such systems, a more natural abstraction requirement is to ask for closeness of observed behaviors. This leads to the notions of approximate simulation and bisimulation introduced in [7].

These notions enabled the computation of approximately equivalent discrete abstractions for several classes of dynamical systems, including nonlinear control systems with or without disturbances (see [12] and [11], respectively) and switched systems [8]. 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, it should be noticed that the smaller the time sampling parameter, the finer the lattice used for approximating the state-space; this may result in abstractions with a very large number of states when the sampling period is small. However, there are a number of applications where sampling has to be fast; though this is generally necessary only on a small part of the state-space. For instance, in sliding mode control of switched systems (see e.g. [14]), the sampling has to be fast only in the neighborhood of a sliding surface.

In this paper, we present a novel class of multiscale discrete abstractions for incrementally stable switched systems that allows us to deal with fast switching while keeping the number of states in the abstraction at a reasonable level. Following the self-triggered control paradigm [17, 18, 2], we assume that the controller of the switched system has to decide the control input and the time period during which it will be applied before the controller executes again. In this context, it is natural to consider 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 the fast switching occurs.

These abstractions allow us to use multiscale iterative approaches for controller synthesis as follows. An initial controller is synthesized based on the dynamics of the abstraction at the coarsest scale where only transitions of longer duration are enabled. An analysis of this initial controller allows us to identify regions of the state-space where transitions of shorter duration may be useful (e.g. to improve the performance of the controller). Then, the controller is refined by enabling transitions of shorter duration in the identified regions. The last two steps can be repeated until we are satisfied with the obtained controller.

The concept of approximately bisimilar multiscale discrete abstractions has already been explored in [16]. In this work, the multiscale feature was used for accomodating locally the precision of the abstraction while the time sampling period remained constant. On the contrary, our approach seeks for a uniform precision but varying time sampling periods.

The paper is organized as follows. In section 2, we present previous results on switched systems and approximate bisimulation. In section 3, we introduce our multiscale discrete abstractions for a class of incrementally stable switched systems. In section 4, we show how to use them for multiscale synthesis of controllers for reachability specifications under time optimization. Finally, we illustrate the merits of our approach by applying it to the boost DC-DC converter.

2. PRELIMINARIES

2.1 Incrementally stable switched systems

In this paper, we shall consider the class of switched systems formalized in the following definition.

Definition 2.1. *A switched system is a quadruple* $\Sigma = (\mathbb{R}^n, P, \mathcal{P}, F)$, *where:*

- $\mathbb{R}^n$ *is the state space;*
- $P = \{1, \dots, m\}$ *is the finite set of modes;*
- $\mathcal{P}$ *is the set of piecewise constant functions from* $\mathbb{R}^+$ *to* P*, continuous from the right and with a finite number of discontinuities on every bounded interval of* $\mathbb{R}^+$*;*
- $F = \{f_1, \dots, f_m\}$ *is a collection of smooth vector fields indexed by* P*.*

A *switching signal* of Σ is a function $\mathbf{p} \in \mathcal{P}$, the discontinuities of $\mathbf{p}$ are called *switching times.* A piecewise $\mathcal{C}^1$ function $\mathbf{x} : \mathbb{R}^+ \to \mathbb{R}^n$ is said to be a *trajectory* of Σ if it is continuous and there exists a switching signal $\mathbf{p} \in \mathcal{P}$ such that, at each $t \in \mathbb{R}^+$ where the function $\mathbf{p}$ is continuous, $\mathbf{x}$ is continuously differentiable and satisfies:

$$\dot{\mathbf{x}}(t) = f_{\mathbf{p}(t)}(\mathbf{x}(t)).$$

We assume that the vector fields $f_1, \dots, f_m$ are such that for all initial conditions and switching signals, there is existence and uniqueness of the trajectory of Σ. We will denote $\mathbf{x}(t, x, \mathbf{p})$ the point reached at time $t \in \mathbb{R}^+$ from the initial condition x under the switching signal $\mathbf{p}$. We will denote $\mathbf{x}(t, x, p)$ the point reached by Σ at time $t \in \mathbb{R}^+$ from the initial condition x under the constant switching signal $\mathbf{p}(t) = p$, for all $t \in \mathbb{R}^+$.

The results presented in this paper rely on some notion of incremental stability (i.e. δ-GUAS [1, 8]). Essentially, incremental stability of a switched system means that all the trajectories associated with the same switching signal converge asymptotically to the same reference trajectory independently of their initial condition. Let us remark that, unless all vector fields share a common equilibrium, this does not imply stability. In the following, $\|.\|$ denotes the usual Euclidean norm over $\mathbb{R}^n$. Incremental stability of a switched system can be characterized using Lyapunov functions:

Definition 2.2. *A smooth function* $V : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^+$ *is a common* δ*-GUAS Lyapunov function for* Σ *if there exist* $\mathcal{K}_\infty$ *functions*[1] $\underline{\alpha}$, $\overline{\alpha}$ *and* $\kappa > 0$ *such that for all* $x, y \in \mathbb{R}^n$, *for all* $p \in P$*:*

$$\underline{\alpha}(\|x - y\|) \le V(x, y) \le \overline{\alpha}(\|x - y\|); \quad (1)$$

$$\frac{\partial V}{\partial x}(x, y) f_p(x) + \frac{\partial V}{\partial y}(x, y) f_p(y) \le -\kappa V(x, y). \quad (2)$$

The computation of a δ-GUAS Lyapunov function is generally hard and out of the scope of the paper. However, if all vector fields are affine one can look for a quadratic δ-GUAS Lyapunov function by solving a set of linear matrix inequalities. As in [8], we will make in the following the supplementary assumption on the δ-GUAS Lyapunov function that there exists a $\mathcal{K}_\infty$ function γ such that

$$\forall x, y, z \in \mathbb{R}^n, \ |V(x, y) - V(x, z)| \le \gamma(\|y - z\|). \quad (3)$$

This assumption was shown to be not restrictive provided V is smooth and we are interested in the dynamics of Σ on a compact subset of $\mathbb{R}^n$, which is often the case in practice.

In [8], it was shown that under the existence of common δ-GUAS Lyapunov function and equation (3) it is possible to compute discrete abstractions that are approximately equivalent to a switched system.

2.2 Approximate bisimulation

In this section, we present a notion of approximate equivalence which will relate a switched system to the discrete systems that we construct. We start by introducing the class of transition systems which allows us to model switched and discrete systems in a common framework.

Definition 2.3. *A transition system is a tuple*

$$T = (Q, L, \longrightarrow, O, H, I)$$

consisting of:

- *a set of states* Q*;*
- *a set of labels* L*;*
- *a transition relation* $\longrightarrow \subseteq Q \times L \times Q$*;*
- *an output set* O*;*
- *an output function* $H : Q \to O$*;*
- *a set of initial states* $I \subseteq Q$*.*

[1] A continuous function $\gamma : \mathbb{R}^+ \to \mathbb{R}^+$ is said to belong to class $\mathcal{K}_\infty$ if it is strictly increasing, $\gamma(0) = 0$ and $\gamma(r) \to \infty$ when $r \to \infty$.

T is said to be metric if the output set O is equipped with a metric d, discrete if Q and L are finite or countable sets.

The transition $(q, l, q') \in \longrightarrow$ will be denoted $q \xrightarrow{l} q'$, or alternatively $q' \in \text{Succ}(q, l)$; this means that the system can evolve from state q to state q' under the action l. An action $l \in L$ belongs to the set of *enabled actions* at state q, denoted $\text{Enab}(q)$, if $\text{Succ}(q, l) \neq \emptyset$. If $\text{Enab}(q) = \emptyset$, then q is said to be a *blocking* state; otherwise it is said to be *non-blocking*. If all states are non-blocking, we say that the transition system T is non-blocking. The transition system is said to be *deterministic* if for all $q \in Q$ and $l \in \text{Enab}(q)$, $\text{Succ}(q, l)$ has only one element. A *trajectory* of the transition system is a finite sequence of transitions

$$\sigma = q_0 \xrightarrow{l_0} q_1 \xrightarrow{l_1} q_2 \xrightarrow{l_2} \dots \xrightarrow{l_{N-1}} q_N.$$

$N \in \mathbb{N}$ is referred to as the *length* of the trajectory. The associated *observed behavior* is the finite sequence of outputs $o_0 o_1 o_2 \dots o_N$ where $o_i = H(q_i)$, for all $i \in \{0, \dots, N\}$.

Transition systems can serve as models for switched systems. Given a switched system $\Sigma = (\mathbb{R}^n, P, \mathcal{P}, F)$, we define the transition system $T(\Sigma) = (Q, L, \longrightarrow, O, H, I)$, where the set of states is $Q = \mathbb{R}^n$; the set of labels is $L = P \times \mathbb{R}^+$; the transition relation is given by

$$x \xrightarrow{p,\tau} x' \text{ iff } \mathbf{x}(\tau, x, p) = x',$$

i.e. the switched system Σ goes from state x to state x' by applying the constant mode p for a duration τ; the set of outputs is $O = \mathbb{R}^n$; the observation map H is the identity map over $\mathbb{R}^n$; the set of initial states is $I = \mathbb{R}^n$. The transition system $T(\Sigma)$ is non-blocking and deterministc, it is metric when the set of outputs $O = \mathbb{R}^n$ is equipped with the metric $d(x, x') = \|x - x'\|$. Note that the state space of $T(\Sigma)$ is uncountable.

Traditional equivalence relationships for transition systems rely on equality of observed behaviors. One of the most common notions is that of bisimulation equivalence [10, 15]. For metric transition systems, requiring strict equivalence of observed behaviors is often quite restrictive. A natural relaxation is to ask for closeness of observed behaviors where closeness is measured with respect to the metric on the output set. This leads to the notion of approximate bisimulation introduced in [7].

Definition 2.4. *Let $T_i = (Q_i, L, \underset{i}{\longrightarrow}, O, H_i, I_i)$, with $i = 1, 2$ be metric transition systems with the same sets of labels L and outputs O equipped with the metric d. Let $\varepsilon \geq 0$ be a given precision. A relation $R \subseteq Q_1 \times Q_2$ is said to be an ε-approximate bisimulation relation between T_1 and T_2 if for all $(q_1, q_2) \in R$:*

- *$d(H_1(q_1), H_2(q_2)) \leq \varepsilon$;*
- *$\forall q_1 \xrightarrow[1]{l} q_1'$, $\exists q_2 \xrightarrow[2]{l} q_2'$, such that $(q_1', q_2') \in R$;*
- *$\forall q_2 \xrightarrow[2]{l} q_2'$, $\exists q_1 \xrightarrow[1]{l} q_1'$, such that $(q_1', q_2') \in R$.*

The transition systems T_1 and T_2 are said to be approximately bisimilar with precision ε, denoted $T_1 \sim_\varepsilon T_2$, if:

- *$\forall q_1 \in I_1$, $\exists q_2 \in I_2$, such that $(q_1, q_2) \in R$;*
- *$\forall q_2 \in I_2$, $\exists q_1 \in I_1$, such that $(q_1, q_2) \in R$.*

If T_1 is a system we want to control and T_2 is a simpler system that we want to use for controller synthesis, then T_2 is called an *approximately bisimilar abstraction* of T_1.

3. MULTISCALE ABSTRACTIONS FOR SWITCHED SYSTEMS

Let Σ be a switched system and $\tau > 0$ a time-sampling parameter, $T_\tau(\Sigma)$ is the sub-transition system of $T(\Sigma)$ obtained by selecting the transitions of $T(\Sigma)$ that describe trajectories of duration τ. This is a natural assumption when the switching in Σ is determined by a time-triggered controller with period τ. In [8], an approach to compute approximately bisimilar abstractions of $T_\tau(\Sigma)$ was presented. It is based on a quantization of the state-space $\mathbb{R}^n$ which is approximated by the lattice:

$$[\mathbb{R}^n]_\eta = \left\{ q \in \mathbb{R}^n \;\middle|\; q[i] = k_i \frac{2\eta}{\sqrt{n}},\ k_i \in \mathbb{Z},\ i = 1, ..., n \right\}$$

where $q[i]$ is the i-th coordinate of q and $\eta > 0$ is a state space discretization parameter. The resulting abstraction $T_{\tau,\eta}(\Sigma)$ is discrete since its set of states $[\mathbb{R}^n]_\eta$ and its set of actions P are respectively countable and finite. It is shown in [8] that under the existence of a common δ-GUAS Lyapunov function and equation (3), $T_\tau(\Sigma)$ and $T_{\tau,\eta}(\Sigma)$ are approximately bisimilar:

Theorem 3.1. *[8] Consider a switched system Σ, time and state space sampling parameters $\tau, \eta > 0$ and a desired precision $\varepsilon > 0$. If there exists a common δ-GUAS Lyapunov function V for Σ such that equation (3) holds and*

$$\eta \leq \min \left\{ \gamma^{-1} \left((1 - e^{-\kappa\tau}) \underline{\alpha}(\varepsilon) \right), \overline{\alpha}^{-1} \left(\underline{\alpha}(\varepsilon) \right) \right\} \qquad (4)$$

then, $T_\tau(\Sigma) \sim_\varepsilon T_{\tau,\eta}(\Sigma)$.

Particularly, it should be noted that given a time sampling parameter $\tau > 0$ and a desired precision $\varepsilon > 0$, it is always possible to choose $\eta > 0$ such that equation (4) holds. This essentially means that approximately bisimilar discrete abstractions of arbitrary precision can be computed for $T_\tau(\Sigma)$. However, the smaller τ or ε, the smaller η must be to satisfy equation (4). In practice, for a small time sampling parameter τ, the ratio ε/η can be very large and discrete abstractions with an acceptable precision may have a very large number of states (see e.g. [8]).

Unfortunately, there are a number of applications where the switching has to be fast; though this fast switching is generally necessary only on a restricted part of the state space. For instance, in sliding mode control (see e.g. [14]), the switching is fast only in the neighborhood of a sliding surface. In order to enable fast switching while dealing with abstractions with a reasonable number of states, one may consider discrete abstractions enabling transitions of different durations. For transitions of long duration, it is sufficient to consider abstract states on a coarse lattice to meet the desired precision ε. As we consider transitions of shorter durations, it becomes necessary to use finer lattices for the abstract state-space. These finer lattices are effectively used only on a restricted area of the state space, where the fast switching occurs. This makes it possible to keep the number of states in the abstraction at a reasonable level. This results naturally in a notion of multiscale discrete abstraction presented next.

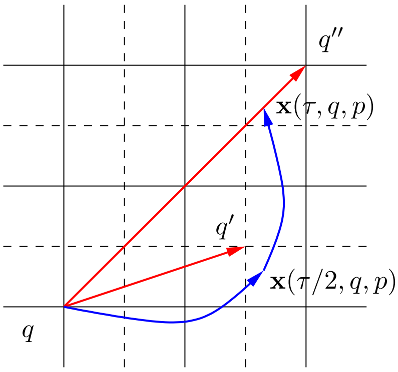

Figure 1: Principle for the computation of the discrete abstraction: $q' = \mathrm{Succ}_2(q,(p,\tau/2))$ **where** $q' = \arg\min_{r\in Q_2^1}(\|\mathbf{x}(\tau/2,q,p)-r\|)$ **and** $q'' = \mathrm{Succ}_2(q,(p,\tau))$ **where** $q'' = \arg\min_{r\in Q_2^0}(\|\mathbf{x}(\tau,q,p)-r\|)$**.**

We work with a sub-transition system of $T(\Sigma)$ obtained by selecting the transitions of $T(\Sigma)$ that describe trajectories of duration that are dyadic fractions of a time sampling parameter $\tau > 0$. The duration of the trajectories are elements of the finite set $\Theta_\tau^N = \{2^{-s}\tau \mid s = 0,\dots,N\}$ for some scale parameter $N \in \mathbb{N}$. Intuitively, one can think that the switched system Σ has to be controlled by a controller with time period $2^{-N}\tau$. The controller chooses to apply a selected mode for a given duration in the set Θ_τ^N: the controller may not need to take a decision at each period $2^{-N}\tau$. This enables the design of self-triggered controllers [17, 18, 2], where the controller not only chooses the control action but also the next instant where it should be executed.

Given a switched system $\Sigma = (\mathbb{R}^n, P, \mathcal{P}, F)$, a time sampling parameter $\tau > 0$, and a scale parameter $N \in \mathbb{N}$, we define the transition system $T_\tau^N(\Sigma) = (Q_1, L, \underset{1}{\longrightarrow}, O, H_1, I_1)$ where the set of states is $Q_1 = \mathbb{R}^n$; the set of labels is $L = P \times \Theta_\tau^N$; the transition relation is given by

$$x \xrightarrow[1]{p,2^{-s}\tau} x' \text{ iff } \mathbf{x}(2^{-s}\tau, x, p) = x';$$

the set of outputs is $O = \mathbb{R}^n$; the observation map H_1 is the identity map over $\mathbb{R}^n$; the set of initial states is $I_1 = \mathbb{R}^n$.

The computation of a discrete abstraction of $T_\tau^N(\Sigma)$ can be done by the following approach. We approximate the set of states $Q_1 = \mathbb{R}^n$ by a sequence of embedded lattices: for $s = 0,\dots,N$, let $Q_2^s = [\mathbb{R}^n]_{2^{-s}\eta}$, i.e.

$$Q_2^s = \left\{ q \in \mathbb{R}^n \;\middle|\; q[i] = k_i \frac{2^{-s+1}\eta}{\sqrt{n}},\ k_i \in \mathbb{Z},\ i = 1, ..., n \right\}.$$

where $\eta > 0$ is a state space discretization parameter. Let us remark that we have the inclusions $Q_2^0 \subseteq Q_2^1 \subseteq \dots \subseteq Q_2^N$. By simple geometrical considerations, we can check that for all $x \in \mathbb{R}^n$, for all $s = 0,\dots,N$, there exists $q \in Q_2^s$ such that $\|x - q\| \le 2^{-s}\eta$.

We now define the abstraction of $T_\tau^N(\Sigma)$ as the transition system $T_{\tau,\eta}^N(\Sigma) = (Q_2, L, \underset{2}{\longrightarrow}, O, H_2, I_2)$, where the set of states is $Q_2 = Q_2^N$; the set of actions remains the same $L = P \times \Theta_\tau^N$; the transition relation is given by

$$q \xrightarrow[2]{p,2^{-s}\tau} q' \text{ iff } q' = \arg\min_{r\in Q_2^s}(\|\mathbf{x}(2^{-s}\tau, q, p) - r\|).$$

If the minimizer $r \in Q_2^s$ is not unique, then one can choose arbitrarily one of them. By definition of the set $Q_2^s = [\mathbb{R}^n]_{2^{-s}\eta}$, if $q \xrightarrow[2]{p,2^{-s}\tau} q'$ then we have $\|\mathbf{x}(2^{-s}\tau, q, p) - q'\| \le 2^{-s}\eta$. The approximation principle is illustrated in Figure 1. The set of outputs remains the same $O = \mathbb{R}^n$; the observation map H_2 is the natural inclusion map from Q_2^N to $\mathbb{R}^n$, i.e. $H_2(q) = q$; the set of initial states is $I_2 = Q_2^0$.

It is important to note that all the transitions of duration $2^{-s}\tau$ end in states belonging to Q_2^s. This means that the states on the finer lattices are only accessible by transitions of shorter duration. Note that the transition system $T_{\tau,\eta}^N(\Sigma)$ is discrete since its sets of states and actions are respectively countable and finite. Also, if we only consider transitions of duration τ, the dynamics of $T_{\tau,\eta}^N(\Sigma)$ coincides with that of the transition $T_{\tau,\eta}(\Sigma)$ defined in [8]. Both transition systems $T_\tau^N(\Sigma)$ and $T_{\tau,\eta}^N(\Sigma)$ are non-blocking and deterministic.

Theorem 3.2. *Consider a switched system Σ, time and state space sampling parameters $\tau, \eta > 0$, scale parameter $N \in \mathbb{N}$, and a desired precision $\varepsilon > 0$. Let us assume that there exists a common δ-GUAS Lyapunov function V for Σ such that equation (3) holds. If*

$$\eta \le \min\left\{ \min_{s=0\dots N} \left[2^s \gamma^{-1}\left((1 - e^{-\kappa 2^{-s}\tau}) \underline{\alpha}(\varepsilon) \right) \right], \overline{\alpha}^{-1}\left(\underline{\alpha}(\varepsilon)\right) \right\} \tag{5}$$

then $R = \{(x,q) \in Q_1 \times Q_2 \mid V(x,q) \le \underline{\alpha}(\varepsilon)\}$ is an ε-approximate bisimulation relation between $T_\tau^N(\Sigma)$ and $T_{\tau,\eta}^N(\Sigma)$. Moreover, $T_\tau^N(\Sigma) \sim_\varepsilon T_{\tau,\eta}^N(\Sigma)$.

Proof. We start by showing that the relation R is an ε-approximate bisimulation relation. Let $(x,q) \in R$, then $V(x,q) \le \underline{\alpha}(\varepsilon)$ and we have from equation (1) that

$$\|x - q\| \le \underline{\alpha}^{-1}(V(x,q)) \le \varepsilon.$$

Thus, the first condition of Definition 2.4 holds.

Let $x \xrightarrow[1]{p,2^{-s}\tau} x'$, then $x' = \mathbf{x}(2^{-s}\tau, x, p)$. Let

$$q' = \arg\min_{r\in Q_2^s}(\|\mathbf{x}(2^{-s}\tau, q, p) - r\|),$$

then $q \xrightarrow[2]{p,2^{-s}\tau} q'$ and $\|\mathbf{x}(2^{-s}\tau, q, p) - q'\| \le 2^{-s}\eta$. Let us check that $(x', q') \in R$. From equation (3),

$$\begin{aligned} |V(x',q') - V(x', \mathbf{x}(2^{-s}\tau, q, p))| &\le \gamma(\|q' - \mathbf{x}(2^{-s}\tau, q, p))\|) \\ &\le \gamma(2^{-s}\eta). \end{aligned}$$

It follows that

$$\begin{aligned} V(x',q') &\le V(x', \mathbf{x}(2^{-s}\tau, q, p)) + \gamma(2^{-s}\eta) \\ &\le V(\mathbf{x}(2^{-s}\tau, x, p), \mathbf{x}(2^{-s}\tau, q, p)) + \gamma(2^{-s}\eta) \\ &\le e^{-\kappa 2^{-s}\tau} V(x,q) + \gamma(2^{-s}\eta) \end{aligned}$$

because V is a δ-GUAS Lyapunov function for Σ. Then,

$$V(x',q') \le e^{-\kappa 2^{-s}\tau}\underline{\alpha}(\varepsilon) + \gamma(2^{-s}\eta) \le \underline{\alpha}(\varepsilon)$$

because of equation (5) and γ is a $\mathcal{K}_\infty$ function, Hence, $(x', q') \in R$. In a similar way, we can prove that, for all $q \xrightarrow[2]{p,2^{-s}\tau} q'$, there is $x \xrightarrow[1]{p,2^{-s}\tau} x'$ such that $(x', q') \in R$. Hence R is an ε-approximate bisimulation relation between $T_\tau^N(\Sigma)$ and $T_{\tau,\eta}^N(\Sigma)$.

By definition of $I_2 = Q_{2,0} = [\mathbb{R}^n]_\eta$, for all $x \in I_1 = \mathbb{R}^n$, there exists $q \in I_2$ such that $\|x - q\| \le \eta$. Then,

$$V(x,q) \le \overline{\alpha}(\|x - q\|) \le \overline{\alpha}(\eta) \le \underline{\alpha}(\varepsilon)$$

because of equation (5) and $\overline{\alpha}$ is a $\mathcal{K}_\infty$ function. Hence, $(x,q) \in R$. Conversely, for all $q \in I_2$, $x = q \in \mathbb{R}^n = I_1$, then $V(x,q) = 0$ and $(x,q) \in R$. Therefore, $T_\tau^N(\Sigma)$ and $T_{\tau,\eta}^N(\Sigma)$ are approximately bisimilar with precision ε. $\square$

It is interesting to note that given a time sampling parameter $\tau > 0$ and a scale parameter $N \in \mathbb{N}$, for all desired precisions $\varepsilon > 0$, there exists $\eta > 0$ such that equation (4) holds. This essentially means that approximately bisimilar multiscale abstractions of arbitrary precision can be computed for $T_\tau^N(\Sigma)$.

4. CONTROLLER SYNTHESIS USING MULTISCALE ABSTRACTIONS

We illustrate the use of multiscale abstractions for synthesizing sub-optimal reachability controllers. This problem was considered in [9, 6] based on the use of uniform discrete abstractions. We extend the synthesis algorithm to multiscale abstractions that are computed on-the-fly, so as to provide a scalable trade-off between precision and cost, while guaranteeing a lower bound on the performance of the closed-loop system.

4.1 Problem formulation

Let us consider a system $T = (Q, L, \longrightarrow, O, H, I)$. For simplicity, we assume that T is deterministic. This is satisfied by the transition systems $T_\tau^N(\Sigma)$ and $T_{\tau,\eta}^N(\Sigma)$ defined in the previous section. In the following, we consider deterministic static state-feedback controllers. However, we just use the term controller for brevity.

Definition 4.1. *A controller for T is a map $\mathcal{S} : Q \to L \cup \{\o\}$ where $\o$ is the dummy symbol. It is well-defined if for all $q \in Q$, $\mathcal{S}(q) \in \mathrm{Enab}(q) \cup \{\o\}$. The dynamics of the controlled system is described by the transition system $T_\mathcal{S} = (Q, L, \underset{\mathcal{S}}{\longrightarrow\!\!\triangleright}, O, H, I)$ where the transition relation is given by*

$$q \underset{\mathcal{S}}{\xrightarrow{l}\!\!\triangleright} q' \iff \left[(l = \mathcal{S}(q)) \wedge (q \xrightarrow{l} q')\right].$$

$\mathcal{S}(q) = \o$ essentially means that the controller is not defined at q. q is a blocking state of $T_\mathcal{S}$ if and only if $\mathcal{S}(q) = \o$. Since we assumed that T is deterministic, the system $T_\mathcal{S}$ is deterministic as well. Moreover, since $\mathcal{S}$ enables at most one action at each state, it follows that for any non-blocking state $q \in Q$ there exists a unique transition starting from q. We assume that for all transitions $q \xrightarrow{l} q'$, the time needed by T for moving from state q to state q' is given by $\delta(l)$ where $\delta : L \to \mathbb{R}^+$. Then, for all trajectories of T,

$$\sigma = q_0 \xrightarrow{l_0} q_1 \xrightarrow{l_1} \dots \xrightarrow{l_{N-1}} q_N$$

we define its *duration* as $\Delta(\sigma) = \delta(l_0) + \delta(l_1) + \dots + \delta(l_{N-1})$. For instance, for the transition systems $T_\tau^N(\Sigma)$ and $T_{\tau,\eta}^N(\Sigma)$, we have that $L = P \times \Theta_{\tau,N}$ and for all $l = (p, 2^{-s}\tau) \in L$, $\delta(l) = 2^{-s}\tau$. The time-optimal reachability problem consists in steering the state of a system to a desired target and keeping the system safe along the way while minimizing the duration of the trajectory.

Definition 4.2. *Let T be a transition system and $\mathcal{S}$ a controller. Let $O_S \subseteq O$ and $O_T \subseteq O_S$ be sets of outputs associated with safe states and target states, respectively. The entry time of $T_\mathcal{S}$ from $q_0 \in Q$ for specification (O_S, O_T) is the smallest $\theta \in \mathbb{R}^+$ such that there exists a trajectory of $T_\mathcal{S}$ starting from q_0, $\sigma = q_0 \underset{\mathcal{S}}{\xrightarrow{l_0}\!\!\triangleright} \dots \underset{\mathcal{S}}{\xrightarrow{l_{N-1}}\!\!\triangleright} q_N$ with $\Delta(\sigma) = \theta$, and such that:*

$$\forall i \in \{0, \dots, N\},\ H(q_i) \in O_S \text{ and } H(q_N) \in O_T.$$

The entry time is denoted by $J(T_\mathcal{S}, O_S, O_T, q_0)$. If such a θ does not exist, then we define $J(T_\mathcal{S}, O_S, O_T, q_0) = +\infty$.

We now define the notion of time-optimal controller:

Definition 4.3. *We say that a controller $\mathcal{S}^*$ for T is time-optimal for specification (O_S, O_T) if for all controllers $\mathcal{S}$ the following holds:*

$$\forall q \in I,\ J(T_{\mathcal{S}^*}, O_S, O_T, q) \le J(T_\mathcal{S}, O_S, O_T, q).$$

4.2 Hierarchical synthesis using abstractions

Let us assume that we want to compute a controller for the switched system $T_\tau^N(\Sigma)$ for a reachability specification (O_S, O_T). An approach to compute a sub-optimal controller using an approximately bisimilar abstraction was proposed in [6]. It consists in computing a controller for the discrete abstraction for a modified specification where the safe and target sets are given by contractions of O_S and O_T.

Let $O' \subseteq O$ and $\varepsilon \ge 0$, the ε-contraction of O' is the subset of O defined as follows

$$C_\varepsilon(O') = \left\{o' \in O' \mid \forall o \in O, d(o,o') \le \varepsilon \implies o \in O'\right\}.$$

We start by synthesizing a controller $\tilde{\mathcal{S}}$ for the discrete abstraction $T_{\tau,\eta}^N(\Sigma)$ for the modified reachability specification $(C_\varepsilon(O_S), C_\varepsilon(O_T))$ where parameters ε, τ, η and N are related according to Theorem 3.2. The synthesis of the controller $\tilde{\mathcal{S}}$ will be the main focus of the following section.

The use of ε-contractions of the safe and target sets ensures that the controller $\tilde{\mathcal{S}}$ for the abstraction $T_{\tau,\eta}^N(\Sigma)$ can be used to derive a controller $\mathcal{S}$ for $T_\tau^N(\Sigma)$ that meets the specification (O_S, O_T). Such a controller $\mathcal{S}$, with guaranteed performance, can be synthesized using the following result that can be easily adapted from [6]:

Proposition 4.4. *Let R denote the ε-approximate bisimulation relation between $T_\tau^N(\Sigma)$ and $T_{\tau,\eta}^N(\Sigma)$ given by Theorem 3.2. Let $\tilde{\mathcal{S}}$ be a controller for abstraction $T_{\tau,\eta}^N(\Sigma)$ for specification $(C_\varepsilon(O_S), C_\varepsilon(O_T))$. Let us define the controller for the switched system $T_\tau^N(\Sigma)$*

$$\mathcal{S}(x) = \tilde{\mathcal{S}}\left(\arg\min_{q \in R(x)} J(T_{\tau,\eta,\tilde{\mathcal{S}}}^N(\Sigma), C_\varepsilon(O_S), C_\varepsilon(O_T), q)\right)$$

where $q \in R(x)$ stands for $(x,q) \in R$. Then, for all $x \in \mathbb{R}^n$, the entry time of $T_{\tau,\mathcal{S}}^N(\Sigma)$ for specification (O_S, O_T) satisfies:

$$J(T_{\tau,\mathcal{S}}^N(\Sigma), O_S, O_T, x) \le \min_{q \in R(x)} J(T_{\tau,\eta,\tilde{\mathcal{S}}}^N(\Sigma), C_\varepsilon(O_S), C_\varepsilon(O_T), q).$$

4.3 Discrete controller synthesis for multiscale abstractions

In principle, computing the optimal controller for the discrete abstraction $T_{\tau,\eta}^N(\Sigma) = (Q, L, \longrightarrow, O, H, I)$ for specification $(C_\varepsilon(O_S), C_\varepsilon(O_T))$ can be done using dynamic programming (see e.g. [5]). Termination of the dynamic programming algorithm is guaranteed provided $H^{-1}(C_\varepsilon(O_S))$

contains a finite number of states. Though, in practice, this number can be very large and the convergence of the algorithm very slow. For this reason, it may be desirable to compute only a sub-optimal controller $\tilde{S}$ but in a more reasonable time. We propose to exploit the multiscale structure of our abstraction for that purpose. The main idea is to compute a sequence of sub-optimal controllers $\tilde{S}_k$ for $T^N_{\tau,\eta}(\Sigma)$ with increasing performance. The sub-optimal controllers are obtained by restraining the sets of enabled transitions. Having less potential transitions to explore results in faster controller synthesis. Essentially, our approach (see Figure 2, detailed in Algorithm 1), can be summarized as follows.

Let $k = 0$, initially, we only enable the transitions of longer duration τ between states on the coarsest lattice. More precisely, let us define the initial map of enabled transitions Enab_0 such that for all $q \in Q$,

$$\mathrm{Enab}_0(q) = \begin{cases} P \times \{\tau\} & \text{if } q \in [\mathbb{R}^n]_\eta \\ \emptyset & \text{otherwise .} \end{cases}$$

Then, we repeat the following procedure:

1. **Controller synthesis**: We compute the optimal controller $\tilde{\mathcal{S}}_k$ for transition system $T^N_{\tau,\eta}(\Sigma)$ and specification $(C_\varepsilon(O_S),\ C_\varepsilon(O_T))$ with the additional constraint that for all $q \in Q$, $\tilde{\mathcal{S}}_k(q) \in \mathrm{Enab}_k(q) \cup \{\varnothing\}$.
2. **Refinement area identification**: We analyze the controller $\tilde{\mathcal{S}}_k$ to identify a region of the state-space O_{Bk} where the performance of the controller should be improved by enabling transitions of shorter duration.
3. **Abstraction refinement**: We define a new map of enabled transitions Enab_{k+1} by extending Enab_k with transitions of duration $2^{-(k+1)}\tau$ within the identified region. We stop when $k = N$ or the performance of the controller $\tilde{\mathcal{S}}_k$ stops improving beyond some threshold.

Algorithm 1 **comp_controller_ms**. Iterative computation of controller $\tilde{\mathcal{S}}$ for multiscale abstraction $T^N_{\tau,\eta}(\Sigma)$.

inputs abstraction $T^N_{\tau,\eta}(\Sigma)$, specification $(C_\varepsilon(O_S), C_\varepsilon(O_T))$, initial map of enabled transitions Enab_0, initial refinement area detection parameters μ,ζ
output Controller $\tilde{\mathcal{S}}$

```
(μ_0, ζ_0) := (μ, ζ)
S̃_0 := comp_controller(T^N_{τ,η}(Σ), C_ε(O_S), C_ε(O_T), Enab_0)
k := 0
while ¬stop_criterion do
    O_Bk := detect_area(S̃_k, μ_k, ζ_k)
    Enab_{k+1} := refine_area(S̃_k, μ_k, ζ_k, Enab_k, O_Bk)
    S̃_{k+1} := comp_controller(T^N_{τ,η}(Σ), C_ε(O_S), C_ε(O_T), Enab_{k+1})
    (μ_{k+1}, ζ_{k+1}) := update_params(μ_k, ζ_k)
    k := k + 1
end while
S̃ := S̃_k
return S̃
```

In the following we will detail each of the steps of this iterative procedure. Let us remark that initially it is sufficient to generate the abstraction only the coarsest level. The dynamics at the finer scales is computed only locally and on the fly during the abstraction refinement step.

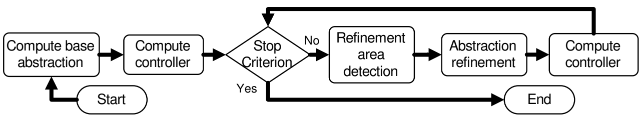

Figure 2: Iterative controller synthesis overview.

4.3.1 Controller synthesis

The optimal controller $\tilde{\mathcal{S}}_k$ for $T^N_{\tau,\eta}(\Sigma)$ and specification $(C_\varepsilon(O_S),\ C_\varepsilon(O_T))$ with the additional constraint that for all $q \in Q$, $\tilde{\mathcal{S}}_k(q) \in \mathrm{Enab}_k(q) \cup \{\varnothing\}$ can be computed using the following standard dynamic programming algorithm [5]:

Algorithm 2 **comp_controller**. Computation of controller $\tilde{\mathcal{S}}_k$

inputs abstraction $T^N_{\tau,\eta}(\Sigma)$, specification $(C_\varepsilon(O_S), C_\varepsilon(O_T))$, map of enabled transitions Enab_k
output Controller $\tilde{\mathcal{S}}_k$

$Q_S := H^{-1}(C_\varepsilon(O_S))$;
$Q_T = H^{-1}(C_\varepsilon(O_T))$;
$Q_k := \{q \in Q \mid \mathrm{Enab}_k(q) \neq \emptyset\}$;
$\forall q \in Q_T,\ J^0_k(q) := 0$;
$\forall q \in Q \setminus Q_T,\ J^0_k(q) := +\infty$;
$i := 0$;
repeat
  $\forall q \in Q_T,\ J^{i+1}_k(q) := 0$;
  $\forall q \in Q \setminus (Q_S \cap Q_k),\ J^{i+1}_k(q) := +\infty$;
  $\forall q \in Q_S \cap Q_k$,

$$J^{i+1}_k(q) := \min_{l \in Enab_k(q)} \Big(\delta(l) + J^i_k(\mathrm{Succ}(q,l))\Big);$$

  $i := i + 1$;
until $\forall q \in Q,\ J^i_k(q) = J^{i-1}_k(q)$
$\forall q \in Q\ ,\ J^*_k(q) := J^i_k(q)$;
$\forall q \in Q$, if $J^*_k(q) = +\infty$, $\tilde{\mathcal{S}}_k(q) := \varnothing$;
$\forall q \in Q$, if $J^*_k(q) \neq +\infty$,

$$\tilde{\mathcal{S}}_k(q) := \arg\min_{l \in \mathrm{Enab}_k(q)} \left(\delta(l) + J^*_k(\mathrm{Succ}(q,l))\right);$$

Algorithm 2 is guaranteed to terminate in a finite number of steps if $H^{-1}(C_\varepsilon(O_S))$ contains a finite number of states which is the case if O_S is a bounded subset of $\mathbb{R}^n$. It is simple to verify that for all $q \in Q$,

$$J(T^N_{\tau,\eta,\tilde{\mathcal{S}}_k}(\Sigma), C_\varepsilon(O_S), C_\varepsilon(O_T), q) = J^*_k(q).$$

It is important to note that it is not necessary to compute the full abstraction $T^N_{\tau,\eta}(\Sigma)$ for synthesizing the controller $\tilde{\mathcal{S}}_k$. It is sufficient to compute the transitions that are enabled by the map Enab_k. This actually renders the synthesis of $\tilde{\mathcal{S}}_k$ quite efficient.

REMARK 4.5. *For $k = 0$, the controller $\tilde{\mathcal{S}}_0$ we obtain is actually the time optimal controller for the uniform abstraction $T_{\tau,\eta}(\Sigma)$ presented in section 3.1.*

REMARK 4.6. *If for all $q \in Q$, $Enab_k(q) \subseteq Enab_{k+1}(q)$, then it follows that for all $q \in Q$, $J^*_k(q) \geq J^*_{k+1}(q)$. This provides a simple criterium to be satisfied when choosing the map of enabled transitions $Enab_{k+1}$ from $Enab_k$ to ensure that the successive controllers we compute will have increasing performance.*

4.3.2 Refinement area identification

The performance of the controller $\tilde{S}_k$ is constrained by the fact that only transitions of duration greater than $2^{-k}\tau$ are enabled. Coarse-grained resolutions (i.e. small values of k)

tend to show higher discrepancies between the ideal optimal trajectories that the controlled system $T^N_{\tau,\eta}(\Sigma)$ should follow, and those yielded by the application of the controller $\tilde{S}_k$. This situation can be mitigated by enabling transitions of shorter duration $2^{-(k+1)}\tau$. However, enabling these transitions at all states of $T^N_{\tau,\eta}(\Sigma)$ results in an exponential growth in the number of transitions to be considered, making the computation of a controller very expensive, or even impossible, depending on the case. It is therefore more efficient to determine a subset of states where enabling transitions of duration $2^{-(k+1)}\tau$ will help improve the performance of the controller.

In time-optimal control, the states where the mode of consecutive transitions does not change represent little opportunity for performance improvement, since they tend to better approximate the optimal trajectories. In contrast, a good strategy to detect parts of the state space where we can gain some performance improvement is checking for trajectories where the number of mode changes exceeds a threshold ζ during a time lapse of duration μ. In order to detect those parts of the state space, we define the function MSC (Mode Switch Count) that obtains the number of mode changes within a trajectory in $T^N_{\tau,\eta,\tilde{\mathcal{S}}_k}(\Sigma)$:

$$\mathrm{MSC}(q_0 \xrightarrow[\tilde{\mathcal{S}}_k]{l_0} \ldots \xrightarrow[\tilde{\mathcal{S}}_k]{l_{K-1}} q_K) = |\{i \mid l_i \neq l_{i+1}\}|$$

Moreover, we define function $\mathrm{Sw}(\tilde{\mathcal{S}}_k, \mu, \zeta)$, which computes all the states that are included in a trajectory of $T^N_{\tau,\eta,\tilde{\mathcal{S}}_k}(\Sigma)$ of duration μ and that contains more than ζ mode changes. Both μ and ζ are user-defined parameters. In particular, the length of the trajectories of $T^N_{\tau,\eta,\tilde{\mathcal{S}}_k}(\Sigma)$ that are analyzed is the maximum $K \in \mathbb{N}$, such that $\delta(l_0) + \delta(l_1) + \cdots + \delta(l_{K-1}) \leqslant \mu$.

$$\mathrm{Sw}(\tilde{\mathcal{S}}_k, \mu, \zeta) = \{\ q \in \{q_0 \ldots q_K\} | \exists \sigma = q_0 \xrightarrow[\tilde{\mathcal{S}}_k]{l_0} \ldots \xrightarrow[\tilde{\mathcal{S}}_k]{l_{K-1}} q_K : \Delta(\sigma) \leqslant \mu \wedge \mathrm{MSC}(\sigma) \geqslant \zeta\}$$

Using this measure, Algorithm 3 determines the region of the state-space that needs to be refined for the computation of $\tilde{\mathcal{S}}_{k+1}$. Specifically, it determines the range of each of the variables in $\mathbb{R}^n$ for the refinement area (i.e., the bounding hyperrectangle of states in $\mathrm{Sw}(\tilde{\mathcal{S}}_k, \mu, \zeta)$), returning them as a list of couples with the minimum and maximum values for each variable.

Algorithm 3 **detect_area**. Controller refinement area detection.

inputs controller $\tilde{\mathcal{S}}_k$, refinement area detection parameters μ,ζ
output refinement area O_{Bk}

$Q' := \mathrm{Sw}(\tilde{\mathcal{S}}_k, \mu, \zeta)$
for all $i \in \{1, \ldots, n\}$ **do**
  $O_{Bk}[i] := (\min_{q \in Q'}(q[i]), \max_{q \in Q'}(q[i]))$
end for
return O_{Bk}

4.3.3 Abstraction refinement

Once the refinement area of the state-space has been detected, we proceed to extend the map of enabled transitions Enab_k with transitions of duration $2^{-(k+1)}\tau$. Let us note as O_{Bk} the set of outputs associated to the refinement area detected by Algorithm 3 for the controller $\tilde{\mathcal{S}}_k$. Algorithm 4 computes the map Enab_{k+1} by: (i) computing the set of states Q' included in trajectories of $T^N_{\tau,\eta,\tilde{\mathcal{S}}_k}(\Sigma)$ of duration $\leqslant \mu$ and with ζ or more mode switches, (ii) enabling transitions of duration $2^{-(k+1)}\tau$ for those states. The algorithm keeps on enabling transitions of the same duration for all the subsequently computed successor states within the refinement area O_{Bk}.

Algorithm 4 **refine_area**. Abstraction area refinement.

inputs controller $\tilde{\mathcal{S}}_k$, map of enabled transitions Enab_k, refinement area detection parameters μ,ζ, detected refinement area O_{Bk}
output extended map of enabled transitions Enab_{k+1}

$\mathrm{Enab}_{k+1} := \mathrm{Enab}_k$
$Q' := \mathrm{Sw}(\tilde{\mathcal{S}}_k, \mu, \zeta)$
$V := \emptyset$
while $\exists q \in Q'$ **do**
  $Q' := Q' \backslash \{q\}$
  $V := V \cup \{q\}$
  for all $l \in (P \times \{2^{-(k+1)}\tau\})$ **do**
    $Q_{\mathrm{Succ}} := \mathrm{Succ}(q, l) \cap H^{-1}(O_{Bk})$
    if $Q_{\mathrm{Succ}} \neq \emptyset$ **then**
      $\mathrm{Enab}_{k+1}(q) := \mathrm{Enab}_{k+1}(q) \cup \{l\}$
      $Q' := Q' \cup (Q_{\mathrm{Succ}} \backslash V)$
    end if
  end for
end while
return Enab_{k+1}

Algorithm 4 is guaranteed to terminate in a finite number of steps, since the number of successor states to refine is finite (bounded by refinement area O_{Bk}), and the algorithm processes each state only once, keeping track of already traversed states. In the worst case, the algorithm traverses once all the states in $H^{-1}(O_{Bk})$.

As an alternative to Algorithm 4, we may directly compute Enab_{k+1} by refining the full set of states in $H^{-1}(O_{Bk})$:

$$\forall (q \in [\mathbb{R}^n]_{2^{-k}\eta} \cap H^{-1}(O_{Bk}), l \in P \times \{2^{-(k+1)}\tau\})$$

$$\mathrm{Enab}_{k+1}(q) = \begin{cases} \mathrm{Enab}_k(q) \cup \{l\} & \text{if } \mathrm{Succ}(q,l) \cap H^{-1}(O_{Bk}) \neq \emptyset \\ \mathrm{Enab}_k(q) & \text{otherwise} . \end{cases}$$

This alternative (referred to as full-area refinement in the following), is aimed at optimizing the performance of the resulting controller, although at the cost of a partial reduction of the efficiency in the overall controller synthesis process (we provide experimental results for both alternatives in Section 5).

5. EXPERIMENTAL RESULTS

We show an application of our multiscale synthesis approach for a concrete switched system: the boost DC-DC converter (see Figure 3). The boost converter has two operation modes depending on the position of the switch. The state of the system is $x(t) = [i_l(t)\ v_c(t)]^T$ where $i_l(t)$ is the inductor current and $v_c(t)$ the capacitor voltage. The dynamics associated with both modes are affine of the form $\dot{x}(t) = A_p x(t) + b$ $(p = 1, 2)$ with

$$A_1 = \begin{bmatrix} -\frac{r_l}{x_l} & 0 \\ 0 & -\frac{1}{x_c}\frac{1}{r_0+r_c} \end{bmatrix},\ b = \begin{bmatrix} \frac{v_s}{x_l} \\ 0 \end{bmatrix},$$

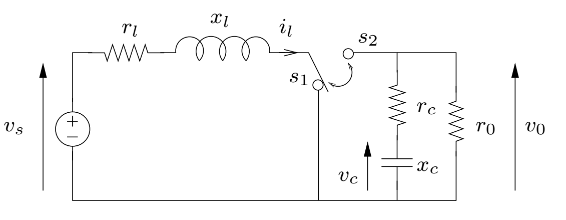

Figure 3: Boost DC-DC converter.

$$A_2 = \begin{bmatrix} -\frac{1}{x_l}(r_l + \frac{r_0 r_c}{r_0 + r_c}) & -\frac{1}{x_l}\frac{r_0}{r_0+r_c} \\ \frac{1}{x_c}\frac{r_0}{r_0+r_c} & -\frac{1}{x_c}\frac{1}{r_0+r_c} \end{bmatrix}.$$

It is clear that the boost DC-DC converter is an example of a switched system. In the following, we use the numerical values from [4], that are, in the per unit system, $x_c = 70$ p.u., $x_l = 3$ p.u., $r_c = 0.005$ p.u., $r_l = 0.05$ p.u., $r_0 = 1$ p.u. and $v_s = 1$ p.u.. We consider the problem of steering in minimal time the state of the system in a desired region of operation while respecting some safety constraints. This is a time-optimal control problem.

For a better numerical conditioning, we rescaled the second variable of the system (i.e. the state of the system becomes $x(t) = [i_l(t)\ 5v_c(t)]^T$; the matrices A_1, A_2 and vector b are modified accordingly). It can be shown that the switched system has a common δ-GUAS Lyapunov function of the form $V(x,y) = \sqrt{(x-y)^T M (x-y)}$, with

$$M = \begin{bmatrix} 1.0224 & 0.0084 \\ 0.0084 & 1.0031 \end{bmatrix}.$$

This δ-GUAS Lyapunov function has the following characteristics: $\underline{\alpha}(s) = s$, $\overline{\alpha}(s) = 1.0127s$, $\kappa = 0.014$. Let us remark that (3) holds as well with $\gamma(s) = 1.0127s$. The specification of the time-optimal control problem is given by the safe set $O_S = [0.65, 1.65] \times [4.95, 5.95]$ and the target $O_T = [1.1, 1.6] \times [5.4, 5.9]$.

In the following, we use approximately bisimilar abstractions to synthesize sub-optimal switching controllers. We set the desired precision of abstractions to 0.1. Then, the contracted safe set and target are $C_\varepsilon(O_S) = [0.75, 1.55] \times [5.05, 5.85]$ and $C_\varepsilon(O_T) = [1.2, 1.5] \times [5.5, 5.8]$. For the sake of comparison, we choose to work both with uniform and multiscale abstractions.

The uniform abstractions $T_{\tau_i,\eta_i}(\Sigma)$ are computed according to [8] for time sampling parameters $\tau_0 = 1$, $\tau_1 = 0.5$ and $\tau_2 = 0.25$. The state-space sampling parameters are chosen according to Theorem 3.1, that is $\eta_0 = 9.7 \times 10^{-4}\sqrt{2}$, $\eta_1 = 4.85 \times 10^{-4}\sqrt{2}$ and $\eta_2 = 2.425 \times 10^{-4}\sqrt{2}$ respectively.

We also use multiscale abstractions $T^N_{\tau,\eta}$ for parameters $\tau = 1$, $\eta = 9.7 \times 10^{-4}\sqrt{2}$ and $N \in \{1, 2\}$. This corresponds to transitions of possible duration $\Theta^1_\tau = \{1, 0.5\}$ and $\Theta^2_\tau = \{1, 0.5, 0.25\}$. Hence, the controllers synthesized using $T^1_{\tau,\eta}$ and $T^2_{\tau,\eta}$ are to be compared with those of T_{τ_1,η_1} and T_{τ_2,η_2} respectively. For the detection of refinement regions in the multiscale controllers, we set as initial parameters $\mu_0 = 8$ and $\zeta_0 = 4$. For each subsequent refinement step, we set $\mu_{k+1} = \mu_k/2$, whereas we kept ζ constant. This essentially means that at scale k we refine the trajectories that have at least 4 switches in less that 8×2^{-k} time units.

The initial controller $\tilde{S}_0$ for multiscale abstractions $T^1_{\tau,\eta}$ and $T^2_{\tau,\eta}$ computed in Algorithm 1 is shown in Figure 4. This corresponds to the time-optimal controller for $T_{\tau_0,\eta_0}(\Sigma)$. In particular, we can observe that the path from an arbitrarily chosen state to $C_\varepsilon(O_T)$ (black rectangle) switches many times in a zigzagging pattern within the region enclosed in the green rectangle, which is the region of the state-space detected by the application of Algorithm 3 for a subsequent refinement ($O_{B0} = [1.12, 1.54] \times [5.04, 5.78]$). This is clearly a region where faster switching would reduce the time to reach the target state. Figure 5 depicts the optimal controllers for T_{τ_1,η_1} and T_{τ_2,η_2} and the sub-optimal controllers computed by Algorithm 1 for $T^1_{\tau,\eta}$ and $T^2_{\tau,\eta}$, using point-wise (Algorithm 4) or full area refinement.

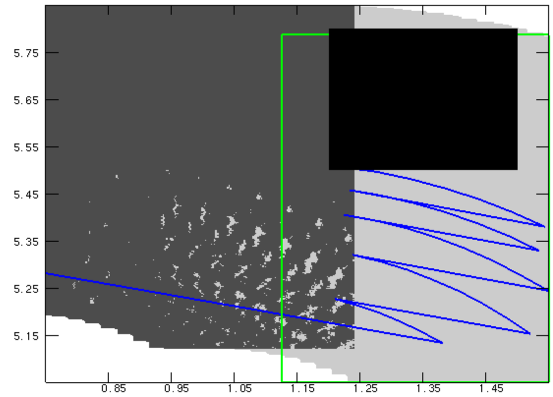

Figure 4: Initial controller $\tilde{S}_0$ for multiscale abstractions $T^1_{\tau,\eta}$ and $T^2_{\tau,\eta}$ (dark gray: mode 1, light gray: mode 2). The area inside the green box is the region detected for subsequent refinement.

Table 1 details the experimental results obtained for the synthesis of the aforementioned set of controllers. In particular, we summarize the results of multiscale controllers for one and two refinement steps, both using point-wise refinement (Algorithm 4), as well as full-area refinement. Moreover, we compare them with the controllers obtained from uniform abstractions with time and state sampling parameters similar to those present at the highest level of refinement in their multiscale counterparts.

Looking at the results, it is worth emphasizing that in general, there is a remarkable reduction in the overall time used to compute the controller using multiscale abstractions with respect to the use of uniform ones (up to a 79% improvement for two refinement steps using Algorithm 4). However, this reduction in computation time is obtained in all cases during the second iteration of the process (in the range of 70-80% versus negligible variations for the first iteration). This is due to the fact that the size of uniform abstractions grows exponentially with higher resolutions, whereas the refinement process that we use with multiscale abstractions bounds this growth by progressively reducing the region of the state-space to refine. In particular, we may observe in Figure 5 (center and right) how the refinement region is reduced in the second iteration (bottom), compared to the original refinement area in the first iteration (top).

Regarding controller performance, in order to minimize the bias in our measures of worst and average entry times, we have only considered the states at the coarsest level of resolution in the multiscale controllers, as well as the subsets of these same states on their uniform counterparts in order to compute our measures. In this respect, despite the afore-described reduction in computation time, the general levels of performance of the multiscale controllers show only small

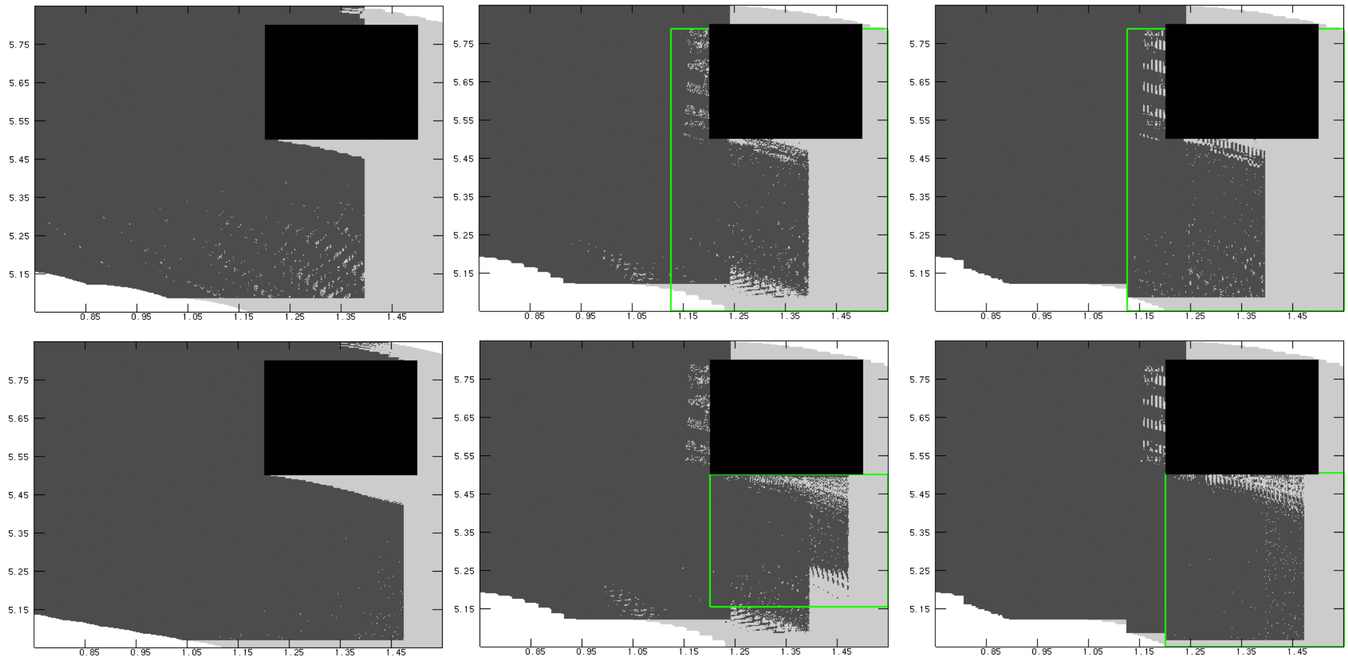

Figure 5: Left: optimal controllers for uniform abstractions $T_{\tau_1,\eta_1}(\Sigma)$ (top) and $T_{\tau_2,\eta_2}(\Sigma)$ (bottom). Center: sub-optimal controllers for multiscale abstractions $T^1_{\tau,\eta}(\Sigma)$ (top) and $T^2_{\tau,\eta}(\Sigma)$ (bottom) computed using point-wise refinement (Algorithm 4). Right: sub-optimal controllers for multiscale abstractions $T^1_{\tau,\eta}(\Sigma)$ (top) and $T^2_{\tau,\eta}(\Sigma)$ (bottom) computed using full-area refinement.

	Uniform Abstractions - $T_{\tau,\eta}(\Sigma)$			Multiscale Abstractions - $T^N_{\tau,\eta}(\Sigma)$ (point-wise / full-area refinement)	
	$\tau = 1s$ $\eta = 13.7 \times 10^{-4}$	$\tau = 0.5s$ $\eta = 6.85 \times 10^{-4}$	$\tau = 0.25s$ $\eta = 3.42 \times 10^{-4}$	$N = 1, \tau = 1s$ $\eta = 13.7 \times 10^{-4}$	$N = 2, \tau = 1s$ $\eta = 13.7 \times 10^{-4}$
Computation Time (s)	60.99	436.01	4167.23	425.75/437.37	846.31/1138.2
Average Entry Time (s)	16.52	10.58	9.21	10.73/10.71	9.69/9.43
Worst Entry Time (s)	54	34.5	29.25	40/35.5	38/31.5
Abstraction Size (# States)	170569	680625	2719201	486606/520668	1250205/1511404
Worst Entry Time Imp. (%)	-	36.11	45.83	25.92/34.25	29.62/41.66
Average Entry Time Imp. (%)	-	35.95	44.24	35.04/35.16	41.34/42.91
Rel. Computation Time Imp. (%)	-	-	-	2.35/-0.31	79.69/72.68
Rel. Worst Entry Time Imp. (%)	-	-	-	-10.18/-1.85	-16.2/-4.16
Rel. Average Entry Time Imp. (%)	-	-	-	-0.9/-0.78	-2.9/-1.33
Rel. Abstraction Size Imp. (%)	-	-	-	28.6/23.6	54.1/44.5

Table 1: Experimental results comparing controller synthesis for time-optimal control of the Boost DC-DC Converter using both uniform and multiscale abstractions.

variations with respect to the uniform controllers with similar resolutions. In the particular case of Algorithm 4, the average entry time shows a slight degradation (-0.9%) with respect to the equivalent uniform controller ($\tau = 0.5$) for 1 iteration. For 2 iterations, the performance further degrades up to -2.9%. However, this does not constitute a remarkable performance loss, especially if we consider that the reduction in the time used to compute the controller is more than 79% in this case. For full-area refinement, the performance of the controller slightly improves, with a degradation on average entry time of up to -1.33% in the controller obtained after 2 iterations of the process, with a computation time reduction still above 70%.

In the case of worst entry time, there is a stronger performance reduction on the multiscale controllers, which yield worst entry times between 1.85% and 16.2% longer. In any case, it is worth considering that the set of controllable regions of the state-space in the controllers obtained from multiscale abstractions does not exactly match with those in the uniform versions, which are slightly more extensive. In particular, we have observed that the highest entry times in the uniform controllers always correspond to the additional set of controllable states in regions of the state-space not covered by the multiscale controllers. However, these additional states are not considered in our measures to provide an even comparison with the multiscale controllers. Hence, this measure provides a less representative indicator of the real controller's performance compared to average entry time.

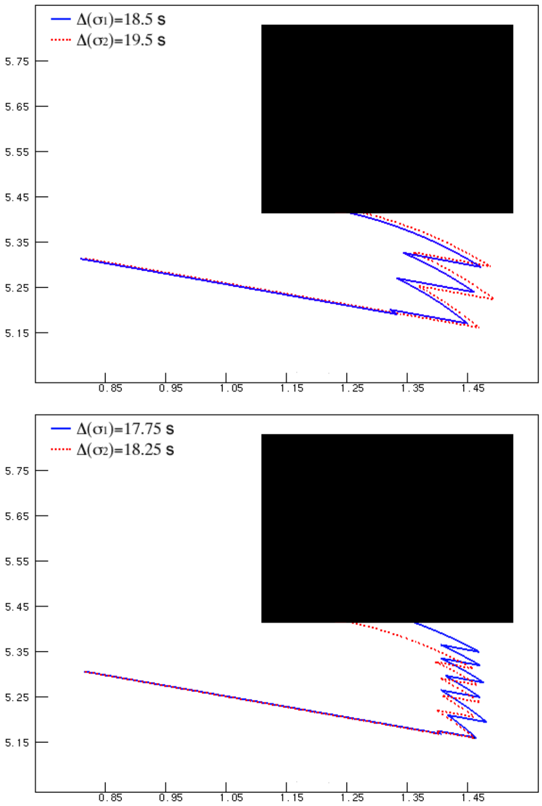

Figure 6: Trajectories of the switched system driven by controllers derived using Proposition 4.4: (top) based on controller of uniform abstraction T_{τ_1,η_1} (solid line) and multiscale abstraction $T^1_{\tau,\eta}$ (dashed line); (bottom) based on controller of uniform abstraction T_{τ_2,η_2} (solid line) and multiscale abstraction $T^2_{\tau,\eta}$ (dashed line).

In relation with the size of abstractions required to compute multiscale controllers, we can observe that there is a reduction with respect to uniform abstractions in the range 23-29% and 44-54% for 1 and 2 iterations, respectively.

Finally, Figure 6 shows trajectories of the actual switched system driven by controllers derived using Proposition 4.4 based on controllers of uniform abstractions T_{τ_1,η_1}, T_{τ_2,η_2} and multiscale abstractions $T^1_{\tau,\eta}$, $T^2_{\tau,\eta}$. First, we can observe that the specification is met: all trajectories reach the target without leaving the safe set. Regarding the performance of the controllers, the ones synthesized using uniform abstractions are more efficient than those synthesized using multiscale abstractions. This was expected, although it should be noticed that their performance levels are still comparable and tend to become closer in finer scales (18.5 vs 19.5 at the first scale, 17.75 vs 18.25 at the second).

6. CONCLUSION

In this paper, we have proposed the use of multiscale, approximately bisimilar discrete abstractions for the computation of controllers, applying them to the specific case of time-optimal control problems. In particular, our experimental results have shown that we can achieve a remarkable reduction in the computation time of such controllers in comparison with the use of uniform abstractions, while preserving similar levels of performance. Future work will deal with the application of multiscale abstractions to other kinds of control problems.

7. REFERENCES

[1] D. Angeli. A Lyapunov approach to incremental stability properties. *IEEE Trans. on Automatic Control*, 47(3):410–421, March 2002.

[2] A. Anta and P. Tabuada. To sample or not to sample: Self-triggered control for nonlinear systems. *IEEE Transactions on Automatic Control*, 2010. To appear.

[3] A. Arnold, A. Vincent, and I. Walukiewicz. Games for synthesis of controllers with partial observation. *Theoretical Computer Science*, 28(1):7–34, 2003.

[4] A. Beccuti, G. Papafotiou, and M. Morari. Optimal control of the boost dc-dc converter. In *IEEE Conf. on Decision and Control*, pages 4457–4462, 2005.

[5] D. P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 2000.

[6] A. Girard. Synthesis using approximately bisimilar abstractions: time-optimal control problems. In *IEEE Conf. on Decision and Control*, 2010.

[7] A. Girard and G. Pappas. Approximation metrics for discrete and continuous systems. *IEEE Trans. on Automatic Control*, 52(5):782–798, 2007.

[8] A. Girard, G. Pola, and P. Tabuada. Approximately bisimilar symbolic models for incrementally stable switched systems. *IEEE Transactions on Automatic Control*, 55(1):116–126, 2010.

[9] M. Mazo Jr. and P. Tabuada. Approximate time-optimal control via approximate alternating simulations. In *ACC*, pages 10201–10206, 2010.

[10] R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.

[11] G. Pola, A. Girard, and P. Tabuada. Approximately bisimilar symbolic models for nonlinear control systems. *Automatica*, 44(10):2508–2516, 2008.

[12] G. Pola and P. Tabuada. Symbolic models for nonlinear control systems: Alternating approximate bisimulations. *SIAM J. on Con. and Opt.*, 48(2):719–733, 2009.

[13] P. Ramadge and W. Wonham. Supervisory control of a class of discrete event systems. *SIAM Journal on Control and Optimization*, 25(1):206–230, 1987.

[14] P. Richard, H. Cormerais, and J. Buisson. A generic design methodology for sliding mode control of switched systems. *Nonlinear Analysis: Hybrid Systems and Applications*, 65(9):1751–1772, 2006.

[15] P. Tabuada. *Verification and Control of Hybrid Systems - A Symbolic Approach*. Springer, 2009.

[16] Y. Tazaki and J. Imura. Approximately bisimilar discrete abstractions of nonlinear systems using variable-resolution quantizers. In *America Control Conference*, pages 1015–1020, 2010.

[17] M. Velasco, J. Fuertes, and P. Marti. The self triggered task model for real-time control systems. In *24th IEEE Real-Time Systems Symposium*, pages 67–70, 2003.

[18] X. Wang and M. Lemmon. State based self-triggered feedback control systems with l2 stability. In *17th IFAC world congress*, 2008.