Stability Controllers for Sampled Switched Systems

Laurent Fribourg and Romain Soulat

LSV, ENS de Cachan & CNRS, France

Abstract. We consider in this paper switched systems, a class of hybrid systems recently used with success in various domains such as automotive industry and power electonics. We propose a state-dependent control strategy which makes the trajectories of the analyzed system converge to finite cyclic sequences of points. Our method relies on a technique of decomposition of the state space into local regions where the control is uniform. We have implemented the procedure using zonotopes, and applied it successfully to several examples of the literature.

1 Introduction

Switched systems are now widely used in industrial applications in domains such as power electronics or automative industry. A switched system can be viewed as a family of continuous-time subsystems with a rule that orchestrates the switching between them. A suitable switching rule allows to steer the system to interesting operating regions which are not accessible using a single subsystem. However, it becomes impossible to stabilize the system around a unique equilibrium point, as in classical systems. The stabilization problem is relaxed as a problem of "practical stabilization", as follows: given a region R of the state space, find a switching rule that makes the system converge to a region, located inside R, as small as possible. In practice, the controlled trajectories of switched systems often converge to *limit cycles* (see, e.g., [His01]). We present here a forward-oriented method that performs a *decomposition* of the region R, and induces a state-dependent control which, under certain conditions, makes the system converge to a cyclic trajectory.

Related work

To the best of of our knowledge, applying a process of state space decomposition in order to stabilize system dynamics is original, at least in the context of switched systems. The method presents some similarities with the method of *box invariance* of [ATS09] which exhibits rectangular invariant subregions of affine hybrid systems containing an equilibrium point, and with the method of *bisection* used in [JKDW01] for the purpose of "set inversion".

The classical methods that are used for proving the existence and stability of limit cycles are based on various techniques such as Lyapunov functions (see, e.g., [BRC05,RRL00]), Poincaré map (see, e.g., [Gon03,His01]), sensibility functions [FRL06], or describing functions [San93]. We give here a couple of conditions, called (A1)-(A2), from which the existence of stable limit cycles follows in an elementary way.

Outline of this paper

We first present the decomposition method in Section 2. We then show that the decomposition induces a state-dependent control in Section 3. We explain that, under certain conditions, the controlled trajectories converge to limit cycles (Section 4). Experimental results are described in Section 5. We conclude in Section 6.

2 State Space Decomposition

A *switched system* Σ is defined by a finite family of differential equations of the form $\{\dot{x} = f_u(x)\}_{u \in U}$ where U is a finite set of *modes* (see, e.g., [GPT10,Tab09]). In the following, we consider that the dynamics of the subsystems are *affine* (i.e., $f_u(x)$ is of the form $A_u x + b_u$ with $A_u \in \mathbb{R}^{n \times n}$ and b_u a vector of $\mathbb{R}^n$). The control problem for a switched system Σ is to find a piecewise constant law $\mathbf{u} : \mathbb{R}_{\geq 0} \to U$ in order to achieve some pertained goals. The *switching instants* are the times at which $\mathbf{u}$ changes its value. A *sampled switched system* is a switched system for which the switching instants occur at integer multiples of τ (called *sampling parameter*). We will use $\mathbf{x}(t, x, u)$ to denote the point reached by Σ at time t under mode u from the initial condition x. This gives a transition relation $\to_u^\tau$ defined for x and x' in $\mathbb{R}^n$ by: $x \to_u^\tau x'$ iff $\mathbf{x}(\tau, x, u) = x'$. Given a set $X \subseteq \mathbb{R}^n$, we define:

$$Post_u(X) = \{x' \mid x \to_u^\tau x' \text{ for some } x \in X\}.$$

It can be seen that $Post_u(X)$ is the result of an affine transformation of the form $C_u X + d_u$ with $C_u \in \mathbb{R}^{n \times n}$ and d_u a vector of $\mathbb{R}^n$.

We say that a subset X of $\mathbb{R}^n$ is *controlled invariant* if:

$$\forall x \in X \ \exists u \in U \ \exists x' \in X : \ x \to_u^\tau x'.$$

A *pattern* π is defined as a finite sequence of modes. A k-pattern is a pattern of length at most k. Given a pattern π of the form $(u_1 \cdots u_n)$ and a subset X of $\mathbb{R}^n$, we define:

$$Post_\pi(X) = \{x' \mid x \to_{u_1}^\tau x_1 \ \wedge x_1 \to_{u_2}^\tau x_2 \ \wedge \cdots \ \wedge x_{m-1} \to_{u_m}^\tau x' \text{ for some } x \in X \text{ and } x_1, \ldots, x_{m-1} \in \mathbb{R}^n\}.$$

Given a pattern π of the form $(u_1 \cdots u_m)$, and a set $X \subseteq \mathbb{R}^n$, the *unfolding of X via* π, denoted by $Unf_\pi(X)$, is the set $\bigcup_{i=0}^m X_i$ with:

- $X_0 = X$,
- $X_{i+1} = Post_{u_{i+1}}(X_i)$, for all $0 \leq i \leq m-1$.

Definition 1. *Given a set $R \subseteq \mathbb{R}^n$, a* k-invariant decomposition *of R is a set Δ of the form $\{V_i, \pi_i\}_{i \in I}$, where I is a finite set of indices, $V_i s$ are subsets of R, $\pi_i s$ are k-patterns, such that:*

- $\bigcup_{i \in I} V_i = R$, *and*
- *for all $i \in I$: $Post_{\pi_i}(V_i) \subseteq R$.*

Given a set $R \subseteq \mathbb{R}^n$ and a k-invariant decomposition $\Delta = \{(V_i, \pi_i)\}_{i \in I}$ of R, the *Δ-unfolding* of a subset X of R is defined by $\bigcup_{i \in I} Unf_{\pi_i}(V_i \cap X)$. The operator $Post_\Delta$ is defined, for all subset X of R by:

$$Post_\Delta(X) = \bigcup_{i \in I} Post_{\pi_i}(X \cap V_i).$$

We have:

Proposition 1. *Suppose that a set R has a k-invariant decomposition Δ. Then we have: $Post_\Delta(R) \subseteq R$.*

We now give a simple algorithm, called Decomposition algorithm, which, given a set R, outputs, when it succeeds, a k-invariant decomposition Δ of the form $\{V_i, \pi_i\}_{i \in I}$ for R. The input set R is given under the form of a *box* of $\mathbb{R}^n$, that is a cartesian product of n closed intervals. The subsets V_is of R are boxes that are obtained by bisection. Two adjacent boxes thus share a common border.

The Decomposition procedure first calls sub-procedure Find_Pattern in order to get a k-pattern π such that $Post_\pi(R) \subseteq R$. If it succeeds, then it is done. Otherwise, it divides R into 2^n sub-boxes $V_1, \dots, V_{2^n}$ of equal size. If for each V_i, Find_Pattern gets a k-pattern π_i such that $Post_{\pi_i}(V_i) \subseteq R$, it is done. If, for some V_j, no such pattern exists, the procedure is recursively applied to V_j. It ends with success when a k-invariant decomposition of R is found, or failure when the maximal degree d of decomposition is reached. The algorithmic form of the procedure is given in Algorithms 1 and 2. (For the sake of simplicity, we consider the case of dimension $n = 2$, but the extension to $n > 2$ is straightforward.) The main procedure Decomposition(W,R,D,K) is called with R as input value for W, d for input value for D, and k as input value for K; it returns either $\langle \{(V_i, \pi_i)\}_i, True \rangle$ with $\bigcup_i V_i = W$ and $\bigcup_i Post_{\pi_i}(V_i) \subseteq R$, or $\langle \_, False \rangle$. Procedure Find_Pattern(W,R,K) looks for a K-pattern π for which $Post_\pi(W) \subseteq R$: it selects all the K-patterns by non-decreasing length order until either it finds such a pattern π (output: $\langle \pi, True \rangle$), or none exists (output: $\langle \_, False \rangle$). The correctness of the procedure is stated as follows.

Theorem 1. *If Decomposition(R,R,d,k) returns $\langle \Delta, True \rangle$, then Δ is a k-invariant decomposition of R.*

Example 1. (**Boost DC-DC Converter**). This example is taken from [BPM05] (see also, e.g., [BRC05,GPT10,SEK03]). This is a boost DC-DC converter with one switching cell (see Figure 1). There are two operation modes depending on the position of the switching cell. An example of pattern of length 4 is illustrated

Algorithm 1: Decomposition(W,R,D,K)

```
Input: A box W, a box R, a degree D of decomposition, a length K of pattern
Output: ⟨{(V_i, π_i)}_i, True⟩ with ⋃_i V_i = W and ⋃_i Post_{π_i}(V_i) ⊆ R, or
        ⟨_, False⟩
(π, b) := Find_Pattern(W, R, K)
if b = True then
    return ⟨{(W, π)}, True⟩
else
    if D = 0 then
        return ⟨_, False⟩
    else
        Divide equally W into (W_1, W_2, W_3, W_4)    /* (case n = 2)    */
        (Δ_1, b_1) := Decomposition(W_1,R,D − 1,K)
        (Δ_2, b_2) := Decomposition(W_2,R,D − 1,K)
        (Δ_3, b_3) := Decomposition(W_3,R,D − 1,K)
        (Δ_4, b_4) := Decomposition(W_4,R,D − 1,K)
        return (Δ_1 ∪ Δ_2 ∪ Δ_3 ∪ Δ_4, b_1 ∧ b_2 ∧ b_3 ∧ b_4)
```

Algorithm 2: Find_Pattern(W,R,K)

```
Input: A box W, a box R, a length K of pattern
Output: ⟨π, True⟩ with Post_π(W) ⊆ R, or ⟨_, False⟩ when no pattern maps W
        into R
for i = 1 . . . K do
    Π := set of patterns of length i
    while Π is non empty do
        Select π in Π
        Π := Π \ {π}
        if Post_π(W) ⊆ R then
            return ⟨π, True⟩
return ⟨_, False⟩
```

in Figure 2: it corresponds to the application of mode 2 on $[0, \tau)$ and mode 1 on $[\tau, 4\tau)$. The state of the system is $x(t) = [i_l(t)\ v_c(t)]^T$ where i_l is the current intensity in inductor, and $v_c(t)$ the voltage of capacitor. The aim of the control is to maintain the system inside a given zone R while the output voltage stabilizes around a desired value. The dynamics associated with mode u is of the form $\dot{x}(t) = A_u x(t) + b_u$ $(u = 1, 2)$ with

$$A_1 = \begin{pmatrix} -\frac{r_l}{x_l} & 0 \\ 0 & -\frac{1}{x_c}\frac{1}{r_0+r_c} \end{pmatrix} \quad b_1 = \begin{pmatrix} \frac{v_s}{x_l} \\ 0 \end{pmatrix}$$

$$A_2 = \begin{pmatrix} -\frac{1}{x_l}(r_l + \frac{r_0.r_c}{r_0+r_c}) & -\frac{1}{x_l}\frac{r_0}{r_0+r_c} \\ \frac{1}{x_c}\frac{r_0}{r_0+r_c} & -\frac{1}{x_c}\frac{1}{r_0+r_c} \end{pmatrix} \quad b_2 = \begin{pmatrix} \frac{v_s}{x_l} \\ 0 \end{pmatrix}$$

We will use the numerical values of [BPM05], expressed in the per unit system: $x_c = 70$, $x_l = 3$, $r_c = 0.005$, $r_l = 0.05$, $r_0 = 1$, $v_s = 1$. The sampling period is $\tau = 0.5$. For $R = [1.55, 2.15] \times [1.0, 1.4]$, the Decomposition algorithm yields

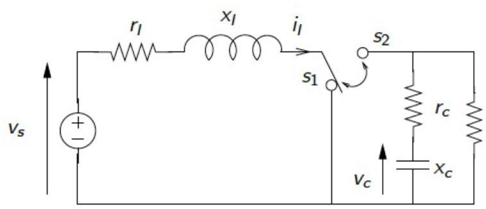

Fig. 1. Scheme of the boost DC-DC converter

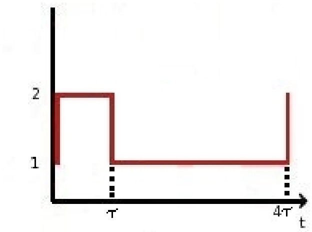

Fig. 2. Cell switching for pattern (2.1.1.1)

a decomposition $\Delta = \{(V_i, \pi_i)\}_{i=1,\ldots,4}$, which is depicted in the left part of Figure 3: the sub-region $V_1 = [1.55, 1.85] \times [1.0, 1.2]$ is associated pattern $\pi_1 = (1{\cdot}1{\cdot}2{\cdot}2{\cdot}2)$, $V_2 = [1.85, 2.15] \times [1.0, 1.2]$ with $\pi_2 = (2)$, $V_3 = [1.85, 2.15] \times [1.2, 1.4]$ with $\pi_3 = (2 \cdot 1 \cdot 2)$, and $V_4 = [1.55, 1.85] \times [1.2, 1.4]$ with $\pi_4 = (1)$. For all $1 \leq i \leq 4$, we have: $Post_\Delta(V_i) = Post_{\pi_i}(V_i) \subseteq R$. This is visualized in the right part of Figure 3.

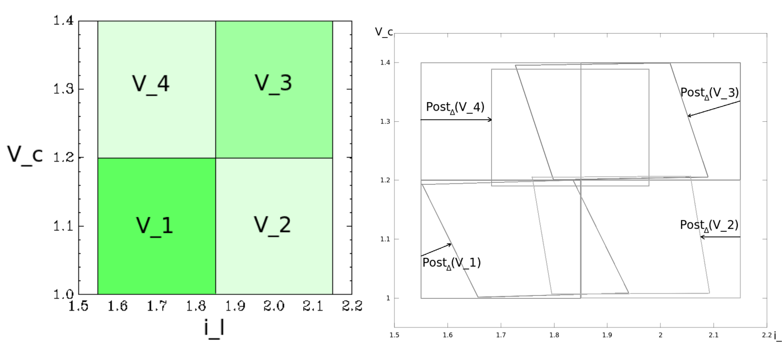

Fig. 3. Decomposition Δ of $R = [1.55, 2.15] \times [1.0, 1.4]$ for the boost DC-DC converter example (left), and visualization of $Post_\Delta(V_i) \subseteq R$, $i = 1, \ldots, 4$ (right)

3 Δ-trajectories

A k-invariant decomposition Δ of R induces a state-dependent control that makes any trajectory starting fom R go back to R within at most k steps:

given a starting state x_0 in R, we know that $x_0 \in V_i$ for some $i \in I$ (since $R = \bigcup_{i \in I} V_i$); one thus applies π_i to x_0, which gives a new state x_1 that belongs to R (since $Post_\pi(V_i) \subseteq R$); the process is repeated on x_1, and so on iteratively. Given a point $x \in R$, we will denote by $succ_\Delta(x)$ the point of R obtained by applying π_i to x when x is in V_i. [1]

Definition 2. *Suppose that Δ is a k-invariant decomposition of a given set R. A* discrete trajectory induced by Δ*, or more simply, a* Δ-trajectory*, is a sequence of points $\{x_i\}_{i \geq 0}$ of R, with $x_{i+1} = succ_\Delta(x_i)$ for all $i \geq 0$.* [2] *A* Δ-cycle *is a Δ-trajectory of R of the form $\{x_0, x_1, \ldots, x_{m-1}\}$ with $x_0 = succ_\Delta(x_{m-1})$.*

Example 2. Consider the boost example 1. Figure 4 depicts a Δ-trajectory starting from the left upper corner of $R = [1.55, 2.15] \times [1.0, 1.4]$, together with its Δ-unfolding.

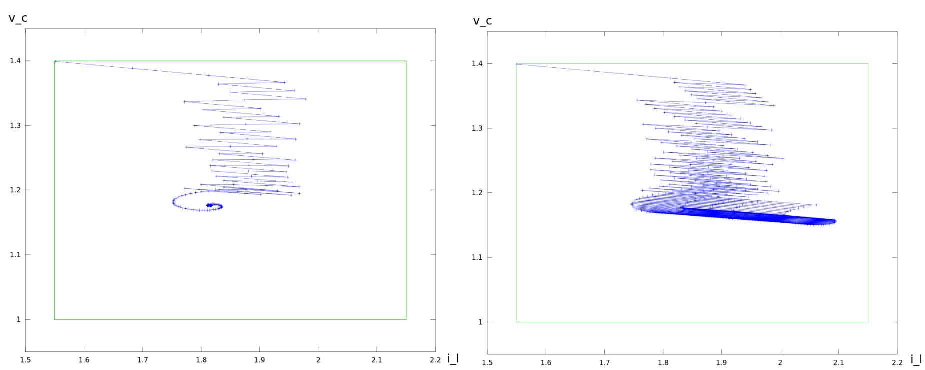

Fig. 4. Δ-trajectory for the boost example (left), and its Δ-unfolding (right)

In Figure 4, we can see that the Δ-trajectory and its Δ-unfolding seem to converge to cycles. We now formally state that, under certain assumptions, this is actually the case.

4 Limit cycles

We suppose that we are given a region $R \subseteq \mathbb{R}^n$ and a k-invariant decomposition $\Delta = \{(V_i, \pi_i)\}_{i \in I}$ of R (produced, e.g., by the Decomposition algorithm of Section 2).

[1] A nondeterministic choice has to be done when a point x belongs to more than one subset V_i. When x belongs to a single subset V_i, then $succ_\Delta(x) = Post_\Delta(x)$.

[2] We will sometimes denote such a trajectory under the form: $x_0 \to_{\pi_{i_1}} x_1 \to_{\pi_{i_2}} \cdots$ with $i_1, i_2, \cdots \in I$.

Proposition 2. *Consider a k-invariant decomposition* $\Delta = \{(V_i, \pi_i)\}_{i \in I}$ *of R. Let* R^j_Δ *be defined by* $R^0_\Delta = R$*, and* $R^j_\Delta = Post_\Delta(R^{j-1}_\Delta)$ *for* $j > 0$*. The sequence* $\{R^j_\Delta\}_{j \geq 0}$ *is a decreasing nested sequence and the set* $R^*_\Delta = \bigcap_{j \geq 0} R^j_\Delta$ *is well-defined. Furthermore,* R^*_Δ *is an* attractor set *of R, i.e.:*

- $Post_\Delta(R^*_\Delta) = R^*_\Delta$ *(*invariance property*)*
- $\forall x \in R,\ d(succ^j_\Delta(x), R^*_\Delta) \to 0$ *as j tends to* ∞ *(*attraction property*).*[3]

Furthermore we have:

Proposition 3. *For all* $i \geq 0$*, the set* R^i_Δ *is a finite union of polyhedra.*

We now make the following assumption:

(A1): There exists $N > 0$ such that R^N_Δ is a finite union of polyhedra $P_1, \ldots, P_q$ (with $q \in \mathbb{N}$) such that:

$$\forall j \in \{1, \ldots, q\}\ \exists! i \in I:\ P_j \cap V_i \neq \emptyset.$$

Assumption $(A1)$ states that every polyhedral component of R^N_Δ shares common points with a single subset V of R. In particular no polyhedron can cross a common intersection ("border") of two distinct subsets V and V' of R. This implies that operator $Post_\Delta$ applied to any polyhedron of R^N_Δ is deterministic: $\forall j \in \{1, \ldots, q\}\ \exists! i \in I\ Post_\Delta(P_j) = Post_{\pi_i}(P_j)$. Furthermore, we have:

$$\forall j \in \{1, \ldots, q\}\ \exists! j' \in \{1, \ldots, q\}:\ Post_\Delta(P_j) \subseteq P_{j'}.$$

Therefore, R^N_Δ can be seen as a directed graph of vertices $P_1, \ldots, P_q$, with an edge from P_j to $P_{j'}$ iff $Post_\Delta(P_j) \subseteq P_{j'}$. The vertices of this graph have a single outgoing edge. The sets R^i_Δ for $i \geq N$ are generated by further application of $Post_\Delta$. The polyhedral components of R^i_Δ which have no incoming edge will disppear at iteration $i + 1$. After a finite number of iterations, the graph of the polyhedral components of R^i_Δ corresponds to the strongly connected components of R^N_Δ. Furthermore, these strongly connected components correspond to disjoint cycles, since the vertices of the graphs have only one outgoing edge. This is formally stated as follows.

Theorem 2. *Under assumptions (A1), we have:*

1. R^*_Δ *is a finite union of disjoint cycles of polyhedra.*
2. *The* Δ*-unfolding of each cycle of* R^*_Δ *is a controlled invariant set.*

Let $\mathcal{C}_1, \ldots, \mathcal{C}_r$ denote the cycles of polyhedra of R^*_Δ. Each cycle $\mathcal{C}_i$ $(1 \leq i \leq r)$ is made of a finite set of polyhedra. Each polyhedron P of a cycle $\mathcal{C}$ is associated with a pattern π such that $Post_\pi(P) = P$. Let us now consider the additional assumption

[3] $d(y, Z)$ denotes the smallest distance between a point y and any point of Z, and $succ^j_\Delta(x)$ the point obtained from x after j applications of $succ_\Delta$.

(A2): For each pattern π associated with a polyhedron P of a cycle $\mathcal{C}$, π is locally contractive in R, i.e.:

$$\forall x, y \in R \ \ \|Post_\pi(x) - Post_\pi(y)\| < \|x - y\|$$

for some norm $\|\cdot\|$ of $\mathbb{R}^n$.

Then we have:

Theorem 3. *Under assumptions (A1) and (A2), we have:*

1. *R^*_Δ is a finite union of disjoint cycles of points of R.*
2. *The Δ-unfolding of each cycle of R^*_Δ is a controlled invariant set.*
3. *Each Δ-trajectory $\{x_0, x_1, \dots\}$* converges to a cycle *of the form $\{y_0, y_1, \dots, y_{m-1}\}$ in the following sense:*

$$\exists M \in \mathbb{N}\ \forall \ell = 0, \dots, m-1 \ \ \lim_{i\to\infty} x_{M+i\cdot m+\ell} = y_\ell.$$

for all $\ell = 0, \dots, m-1$.

The proof of a variant of Theorem 3 is given in [FS13].

We now illustrate the convergence of R^k_Δ to a cyclic set of points as k tends to infinity, on the boost example.

Example 3. (**Boost DC-DC Converter**). One can check that the modes of the boost converter are locally contractive in $R = [1.55, 1.85] \times [1.0, 1.2]$, hence (A2) is satisfied. Likewise, (A1) is satisfied: for $N = 100$, all the polyhedral components of R^N_Δ belong to a single box (viz., V_1) of the decomposition Δ. This is shown in Figure 5, which depicts the iterated images R^k_Δ for $k = 0, 20, 40, 60, 80, 100$. The limit set R^*_Δ is here composed of a unique limit cycle that is made of a single point $y_0 \in V_1$. We have: $y_0 \to_{\pi_1} y_1 = y_0$, with $\pi_1 = (1 \cdot 1 \cdot 2 \cdot 2 \cdot 2)$. The Δ-unfolding of this limit cycle is thus made of 5 points (corresponding to the composing modes of π_1) and is depicted in Figure 6.

5 Implementation

The implementation of the method is made of two basic procedures: a procedure Decomposition (described in Section 2), which outputs Δ, and a procedure, called Iteration which constructs R^i_Δ for $i \geq 0$. The Decomposition procedure makes use of zonotopes [K$\ddot{9}$8], and has been written in Octave [oct], except for the multilevel examples which have been implemented using PLECS [ple]. The procedure Iteration does not use the data structure zonotopes because it involves the intersection operator which does not preserve the structure of zonotopes. It has been written in Ocaml [oca], using the PPL library [ppl] of polyhedra. The Iteration procedure receives Δ from module Decomposition and outputs the successive iterations of $Post_\Delta$. The sequence of post sets can also be visualized as an animation (see Figure 5).

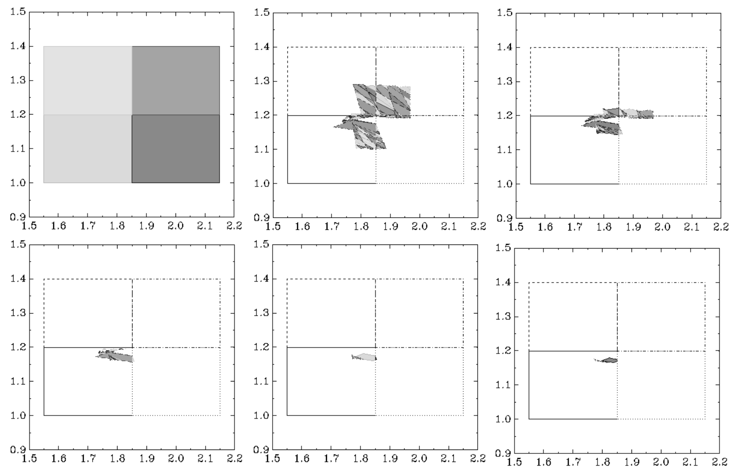

Fig. 5. Visualization of R^k_Δ for $k = 0, 20, 40, 60, 80, 100$ for the boost example

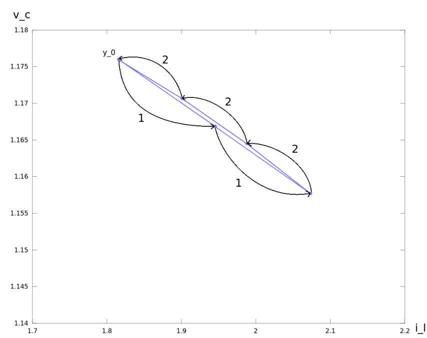

Fig. 6. Δ-unfolding of the limit cycle $\{y_0\}$ for the boost example

The examples have run on a machine equipped with an Intel Core2 CPU X6800 at 2.93GHz and with 2GB of Ram memory. Some figures of the experiments are listed in the following table.

Example	Running time	# patterns	$\|U\|$	k	d	n	(A1)	(A2)	cycle
Boost [BPM05]	150 seconds	12113	2	5	1	2	yes	yes	yes
Two-tank [His01]	4 seconds	1423	4	3	1	2	yes	yes	yes
Heating [Gir12]	1 second	134	2	2	4	2	yes	yes	yes
Helicopter [DLHT11]	$\approx$ 2 hours	$\approx$ 1.5 million	9	6	4	2	yes	no	yes
5-level [FFL$^+$12]	3 minutes	-	16	8	1	3	yes	no	yes
7-level [FFL$^+$12]	35 minutes	-	64	32	1	5	yes	no	yes
9-level [FFL$^+$12]	$\approx$ 5 hours	-	256	128	1	7	yes	no	yes

The first column indicates the name of the example together with its reference. The second column indicates the running time to obtain a decomposition, and the third one the numbers of patterns generated to obtain this decomposition[4]. The subsequent columns labeled by $|U|$, k, d and n indicate the number of modes, the input parameter of maximal pattern length, the input parameter of decomposition depth and the space dimension respectively. Finally, the column '(A1)' (resp. '(A2)') indicates if (A1) (resp. (A2)) is satisfied, and the column 'cycle' if the controlled trajectories converge to a limit cycle.

6 Final Remarks

We have presented an original technique to synthesize stability controllers for switched systems. We have implemented the procedure, and applied it successfully to several examples of the literature. The method can also be used for synthesizing *safety controllers* in order to guarantee safety properties of the controlled system (see [FS13]). A sufficient condition for the existence of a k-invariant decomposition of a given box R is also given in [FS13].

References

[ATS09] A. Abate, A. Tiwari, and S. Sastry. Box invariance in biologically-inspired dynamical systems. *Automatica*, 45(7):1601–1610, 2009.

[BPM05] A.G. Beccuti, G. Papafotiou, and M. Morari. Optimal control of the boost dc-dc converter. In *Proc. 44th IEEE Conference on Decision and Control European Control Conference (CDC-ECC '05)*, pages 4457 – 4462, dec. 2005.

[BRC05] J. Buisson, P.-Y. Richard, and H. Cormerais. On the stabilisation of switching electrical power converters. In *HSCC*, volume 3414 of *LNCS*, pages 184–197. Springer, 2005.

[DLHT11] J. Ding, E. Li, H. Huang, and C.J. Tomlin. Reachability-based synthesis of feedback policies for motion planning under bounded disturbances. In *IEEE International Conference on Robotics and Automation (ICRA'11)*, pages 2160–2165, 2011.

[FFL$^+$12] Gilles Feld, Laurent Fribourg, Denis Labrousse, Bertrand Revol, and Romain Soulat. Correct by design control of 5-level and 7-level converters. Research Report LSV-12-25, LSV, ENS Cachan, France, Dec 2012.

[FRL06] D. Flieller, P. Riedinger, and J.-P. Louis. Computation and stability of limit cycles in hybrid systems. *Nonlinear Analysis*, 64(2):352–367, 2006.

[FS13] L. Fribourg and R. Soulat. Finite controlled invariants for sampled switched systems. Research Report LSV-13-09, Laboratoire Spécification et Vérification, ENS Cachan, France, April 2013. 27 pages.

[Gir12] Antoine Girard. Low-complexity switching controllers for safety using symbolic models. In Maurice Heemels, Bart De Schutter, and Mircea Lazar, editors, *4th IFAC conference on Analysis and Design of Hybrid Systems, June, 2012*, Eindhoven, Pays-Bas, 2012.

[Gon03] J. M. Gonçalves. Region of stability for limit cycles of piecewise linear systems. In *IEEE Conference on Decision and Control*, 2003.

[4] This figure is not available for the multilevel converters because they have been implemented using PLECS, rather than Octave.

[GPT10] A. Girard, G. Pola, and P. Tabuada. Approximately bisimilar symbolic models for incrementally stable switched systems. *IEEE Trans. on Automatic Control*, 55:116–126, 2010.

[His01] I.A. Hiskens. Stability of limit cycles in hybrid systems. In *34th Annual Hawaii International Conference on System Sciences (HICSS-34)*. IEEE Computer Society, January 3-6, 2001.

[JKDW01] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis: with examples in parameter and state estimation, Robust Control and Robotics*. Springer, 2001.

[K$\ddot{9}$8] W. Kühn. Zonotope dynamics in numerical quality control. *Mathematical Visualization*, pages 125–134, 1998.

[oca] OCaml Web page. http://caml.inria.fr/ocaml/index.en.html.

[oct] Octave Web page. http://www.gnu.org/software/octave/.

[ple] PLECS Web page. http://www.plexim.com.

[ppl] PPL Web page. http://bugseng.com/products/ppl/.

[RRL00] M. Rubensson, M. Rubensson, and B. Lennartson. Stability of limit cycles in hybrid systems using discrete-time lyapunov techniques. In *Proceedings of the 39th IEEE Conference on Decision and Control*, 2000.

[San93] S.R. Sanders. On limit cycles and the describing function method in periodically switched circuits. *IEEE Trans. Circuits and Systems*, 40(9):564–572, 1993.

[SEK03] M. Senesky, G. Eirea, and T.-K. Koo. Hybrid modelling and control of power electronics. In *HSCC*, volume 2623 of *Lecture Notes in Computer Science*, pages 450–465, 2003.

[Tab09] Paulo Tabuada. *Verification and Control of Hybrid Systems: A Symbolic Approach*. Springer Publishing Company, Incorporated, 1st edition, 2009.