Design Verification Methods for Switching Power Converters

Taylor T. Johnson, Zhihao Hong, and Akash Kapoor
Dept. of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
Urbana, IL 61801
Email: {johnso99, hong64, akapoor5}@illinois.edu

***Abstract*—In this paper, we present two methods for performing design verification of switching power converters. The first method can be used to compute the set of reachable states from an initial set of states with non-deterministic parameters. We demonstrate the method on a buck converter in an open-loop configuration. The method is automatic and uses the hybrid systems reachability analysis tool SpaceEx. The second method uses model checking to verify circuits that can naturally be modeled as timed automata. We demonstrate the method on an open-loop multilevel converter used to convert several DC inputs to one AC output. The method is also automatic and uses the timed automata model checker Uppaal. Finally, we mention that in contrast to simulation or testing based approaches—for instance, the standard Monte Carlo analysis used when analyzing component variation in circuit designs—the methods presented in this paper perform the verification for all runs of the circuits and all possible component parameter variations.**

***Index Terms*—hybrid systems, verification, buck-converter, multilevel converter**

I. Introduction

A traditional method used to validate that a circuit design is behaving according to its requirements is to first run a few simulations, then perhaps perform a Monte Carlo analysis varying the circuit elements within their tolerances. However, all such methods are incomplete, in the sense that they cannot try all possible combinations of parameter variations or all initial conditions. That is, each iteration of such a method corresponds to a single execution of the system, and since there are uncountably infinitely many such executions, any method would run forever. More recent techniques apply reachability and verification methods from hybrid systems research to circuits, such as [1], [2], [3], which may allow one to check all possible parameter variations and use sets of initial conditions.

These reachability techniques rely on computing an overapproximation of the *reach set* of a system, which is the set of all trajectories of the system from some set of initial conditions (sometimes allowing for variation in the parameters of the system). For many classes of systems, computation of the reach set for unbounded time is either undecidable or computational impractical, so frequently a *time-bounded reach set* is used, which is the reach set up to some finite time. More concretely, the reach set from time t_0 to t_f is

$$R_{t_0}^{t_f} = \{x \in \mathbb{R}^n : x = \int_{t_0}^{t} f(x(\tau))d\tau, x(t_0) \in \mathbf{x}_0, t \in [t_0, t_f]\},$$

for a system with dynamics $\dot{x} = f(x)$ and an initial set of states $\mathbf{x}_0 \subseteq \mathbb{R}^n$.

Most reachability methods compute an overapproximation of the set $R_{t_0}^{t_f}$. Since these methods compute an overapproximation of the reach set, one can then guarantee that some bad (or unsafe) property does not occur (in bounded time) by ensuring the intersection of the reach set and the bad set of states is empty. If this intersection is empty, then the system is safe, but if the intersection is nonempty, one generally may not conclude that the system is unsafe. This is due to the nature of overapproximation—additional trajectories are included in the reach set that do not actually exist in executions of the actual system, i.e., the trajectories intersecting the bad set of states may be spurious and arise from the overapproximation.

Concretely, in the DC-to-DC converter investigated in this paper for instance, the bad set of states would be those where the output voltage strays too far from the desired value due to voltage ripple. More concretely, if a desired output voltage is 5V, then the bad set of states could be those outside the interval $[4.9, 5.1]$V, or whatever the particular specification requires.

In this paper, we consider application of two verification methods to ensure circuit designs satisfy their specifications. The first method is a reachability analysis method that can be used to verify switched-mode power supplies like buck-converters, boost-converters, Cuk converters, push-pull converters, and other topologies. We present it as a case study on verifying that

the output voltage of a buck-converter are within a tolerance band due to voltage ripple. Specifically, we answer two questions for the buck converter. We first present a method to answer the question of whether a particular converter design and an open-loop control (in terms of a control period and duty cycle) will satisfy a specified output voltage range. For instance, does a given buck-converter design with a 12V input always maintain an output voltage between 4V and 6V? We do not report it here for space, but a similar method answers the same question of the buck-converter with a given closed-loop control strategy. We illustrate this method using the hybrid systems reachability tool SpaceEx [4].

The second method relies on using model checking for timed automata [5] to verify properties of circuits that can be modeled as timed automata. For reference, a timed automaton is a finite-state machine with additional clock variables that evolve monotonically with time, where potentially the clock rates vary in different states of the automaton. Concretely, timed automata allow dynamics of the form $\dot{x} = a$ (upon integrating with respect to time, one has $x = at$, which gives the reason as to why they are called clocks and timed automata). Specifically, we show that multilevel converters used to convert multiple DC inputs to a single AC output can naturally be modeled as timed automata. The contribution of this method is primarily the realization that some analog circuits can be naturally modeled as timed automata. We illustrate the method by verifying several properties of a particular multilevel converter using the timed automata model checker Uppaal [6].

Related Work: There are a variety of computational methods and tools for computing reach sets of different classes of systems, several of which have been applied to circuits [1], [2], [3], [7]. Researchers have been developing computational tools for computing reach sets and analyzing hybrid systems for some time [8], [9], [10], [11], [12], [13], [14], [15], [4], [16]. Many of these methods work by computing an overaproximation of the reach set by computing an overapproximation of the state transition function corresponding to the solution of the dynamical system (recall for the case of a linear system that the state transition function is the matrix exponential, so $x(t) = e^{At}x_0$).

For instance, several of these methods are inspired by the following idea. A bounded subset of the state space of the system is discretized into compact (frequently convex) regions. This discretization can occur either before computing the reach set, or on-the-fly during the computation (which may allow methods to discretize "'interesting states"' more finely). Over each

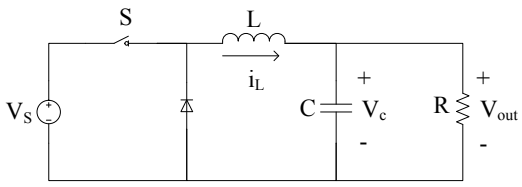

Fig. 1. Buck-converter circuit—a DC input V_s is decreased to a lower DC output V_{out}.

of these compact regions, the dynamics of the system are overapproximated. Since the regions are compact and the dynamics are usually continuous, the dynamics will take a minimum and maximum over the compact set. The resulting system is a hybrid system with dynamics defined in each compact region, and for this reason, the method is known as hybridization [15]. If the original system is hybrid, this idea can be used in each mode.

The Ellipsoidal Toolbox [14] was used in [3], and allows for calculation of reach sets of piecewise affine systems. One major issue in computing reach sets of systems is that the methods may scale very poorly in the number of dimensions (variables) of the system. The other obstacle to a useful reachability method is how many spurious states are reached—i.e., the overapproximation from the actual reach set should be as small as possible. For instance, the optimal control inspired methods used in the Ellipsoidal Toolbox and other methods like [17] become impractical beyond tens of variables, at best. The tool we use, SpaceEx, is the successor of the successor of the tool PHAVer [12]. One reason we were interested using SpaceEx is for future work—the support function algorithm [18] used for reach set computation in SpaceEx scales quite efficiently in the number of variables, and examples with tens and hundreds of variables are feasible, and those with thousands of variables will be feasible soon. Another method that scales quite well in the number of variables is a method using zonotypes [16], [7], which has also reached hundreds to thousands of dimensions.

II. Reachability Analysis for DC-to-DC Switchmode Power Supplies

We will briefly review a switched-system model of a buck-converter, but the interested reader is referred to an in-depth derivation of the model in [19]. A buck-converter is a DC-to-DC switchmode step-down converter. The converter takes an input voltage of say 5V and "bucks" or drops the voltage to some lower DC voltage, say 2.5V. The circuit can be seen in Figure 1. The basic operation is that the voltage source is connected and disconnected from the load by toggling the

switch S at some switching frequency T_s and duty cycle. We will refer to the duty cycle with switch S open as $1 \geq \Delta_o \geq 0$ which is the percentage of time spent with S open. Likewise, the time spent with switch S closed is referred to as $\Delta_c = 1 - \Delta_o$. In an open-loop configuration, each of T_s, Δ_c, and Δ_o are fixed, while in closed-loop, they may vary based on different control strategies.

The position of the switch gives rise to different modes, of which there are a finite number in some index set M. In open-loop, this circuit can be modeled as a switched affine (linear with fixed input) system [20] of the following form,

$$\dot{x}_{\sigma(t)} = A_{\sigma(t)}x + B_{\sigma(t)},$$

where for each $i \in M$, $A_i \in \mathbb{R}^{n\times n}$, $B_i \in \mathbb{R}^n$, and $\sigma(t) : \mathbb{R} \to M$ is a function mapping time to a mode. For the buck-converter, the two state variables are the current through the inductor i_L and the voltage across the capacitor V_c, so we have

$$x = \begin{bmatrix} i_L \\ V_c \end{bmatrix}.$$

In continuous conduction (see, e.g., [19]), there are two modes based on whether the switch is open or closed, so $M = \{o, c\}$, and each mode is described by the following system matrix,

$$A_o = A_c = \begin{bmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{RC} \end{bmatrix}. \tag{1}$$

However, the affine input term differs based on the mode selection. If S is closed, then the voltage source is connected, so $B_c = [\frac{1}{L}; 0]V_s$, while if it is open, $B_o = [0; 0]V_s$. We chose not to model parasitics in favor of modeling nondeterministic parametric uncertainty in the components as described below in Subsection II-A, but one can make the extension by appropriately modifying Equation 1.

With such a system definition, one may analyze stability of this system using a variety of techniques. Aside from traditional tests like Bode analysis, there are other now classic techniques like averaging analysis [21], or one can use more recent switched-system techniques [20]. Since the system is linear, such techniques may account for small perturbations in the model, for instance, due to component variation of the circuit elements. However, such techniques as just described may be unable to account for variation *in the switching time itself*. For these reasons, and our interest in using a computational approach, we chose to analyze stability of this system using a recent hybrid systems reachability tool called SpaceEx [4].

A. Parameter Variation

Due to manufacturing processes and environmental factors (e.g., parameter variation due to temperature changes), any actual circuit implementation of the buck converter will not match the model from Equation 1. We now describe a method to capture such behaviors when performing a reachability computation. For our example, suppose we are given some tolerances for each component as shown in Table I. Due to component variation, one can no longer specify the system as a single matrix, but as a family of matrices. To over-approximate the behavior of such a system, one may use interval arithmetic and interval matrices [22], for which there exist necessary and sufficient conditions for stability [23], [24]. There are also more computational practical sufficient conditions [25]. There are some computational tools using interval arithmetic to compute reach sets [10], [16], [26]. SpaceEx does not support such interval matrices, so we describe a simple technique used to overapproximate the reach set.

More precisely, an $n \times n$ interval matrix is defined as a set of real matrices:

$$\mathcal{A} = \{A = [a_{ij}] : a_{ij} \in [b_{ij}, c_{ij}],\ i, j = 1, \ldots, n\},$$

that is, each entry a_{ij} in $\mathcal{A}$ varies between $b_{ij} \leq a_{ij} \leq c_{ij}$. Matrices of dimension $n \times m$ for $n \neq m$ can be defined similarly. The interval dynamic system described by $\dot{x} = A_i x$ for any $A_i \in \mathcal{A}$ was shown to be asymptotically stable in [23] if and only if the matrices A_i created from all possible combinations of interval endpoints b_{ij} and c_{ij} are stable. Similarly, to overapproximate the reach set, it is sufficient to consider every combination of interval extrema in $\mathcal{A}$. There may be an exponential number of such combinations, so the methods to overapproximate the interval matrix reach set in [26] may be better suited, but this tool was not available to us at the time of writing. For some systems, it is sufficient to compute a minimal and maximal matrix $\underline{A}$ and $\overline{A}$ respectively. These matrices satisfy the orders $\underline{A} \leq \mathcal{A} \leq \overline{A}$, where $\leq$ means $\underline{a}_{ij} \leq a_{ij} \leq \overline{a}_{ij}$ for $i, j = 1, 2, \ldots, n$ and $\underline{a}_{ij} \in \underline{A}$, $a_{ij} \in \mathcal{A}$, and $\overline{a}_{ij} \in \overline{A}$. Using this, we can overapproximate the reach sets of $\mathcal{A}$ by computing reachability using $\underline{A}$ and $\overline{A}$. Thus, for computing the reach set, we can look only at the "worst behaved" matrices, where worst behaved essentially means the matrices with the smallest stability margins, as also used in the more computationally efficient sufficient condition for stability from [25].

Concretely, suppose we allow the component variations for R, C, and L to take each take nondeterministically from a range like the $\pm 5\%$ variations indicated in Table I. We can also model nondeterministic uncer-

Component / Parameter Name	Symbol	Range
Input Voltage	V_s	$[11.95, 12.05]$ V
Load Resistance	R	$[0.95, 1.05]\Omega$
Capacitor	C	$[23.75, 26.25]$ uF
Inductor	L	$[47.5, 52.5]$ uH
Control Period	T_s	$[24.5, 25.5]$ us
Switch-open duty cycle	Δ_o	$[9.75, 10.25]$ us
Switch-closed duty cycle	Δ_c	$[14.75, 15.25]$ us

TABLE I
BUCK-CONVERTER PARAMETER VALUES AND VARIATIONS.

tainties in the switching times simply by allowing a mode switch to occur in a range of times instead of at a the particular time designated by the control period and mode duty-cycle. We then arrive at an interval system matrix for each mode of the buck-converter as

$$\mathcal{A}_o = \mathcal{A}_c = \begin{bmatrix} [0,0] & [-21053, -19048] \\ [38095, 42105] & [-44321, -36281] \end{bmatrix},$$

and the input matrices as

$$\mathcal{B}_o = \begin{bmatrix} [0,0] \\ [0,0] \end{bmatrix} \text{ and } \mathcal{B}_c = \begin{bmatrix} [19048, 21053] \\ [0,0] \end{bmatrix}.$$

To compute these interval matrices, we relied on the following interval extensions of multiplication and division. For two intervals $[a,b]$ and $[c,d]$, we have $[a,b] * [c,d] = [\min(ac,ad,bc,bd), \max(ac,ad,bc,bd)]$. Similarly, the quotient of two intervals (supposing the divisor interval $[c,d]$ does not contain zero) is defined as $[a,b]/[c,d] = [a,b] * (1/[c,d])$ for $1/[c,d] = [\frac{1}{d}, \frac{1}{c}]$.[1]

Figure 2 displays the reachable states from a startup condition, and proves that the system with the chosen parameters always has an output voltage within the range in the graph. We omit the reach set computation figure for steady-state due to space constraints, but it also satisfies a voltage ripple constraint. In Figure 3, we show the current-versus-voltage output from the hybrid systems reachability tool SpaceEx [4], which supports systems with affine dynamics (i.e., of the form $\dot{x} = Ax + B$ where B is a constant vector or more generally, takes values from a convex set of appropriate dimension).

[1]We can also symbolically define this matrix, letting $\overline{R}$ be the upper value of this range for R, $\underline{R}$ be the lower value, so $\underline{R} \leq \overline{R}$, and analogously for the other components, but do not do so because writing the entry $-\frac{1}{RC}$ is tedious with the numerous min and max operators.

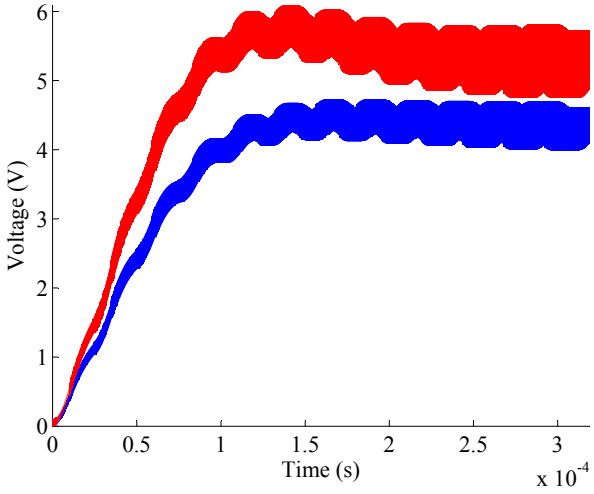

Fig. 2. Startup verification of buck-converter output voltage using parameters from Table I. The upper red set shows the reachable states due to $\overline{A}$, and the lower blue set shows the reachable states due to $\underline{A}$. The initial set of states were $i_L \in [0, 0.1]$A and $V_c \in [0, 0.1]$V.

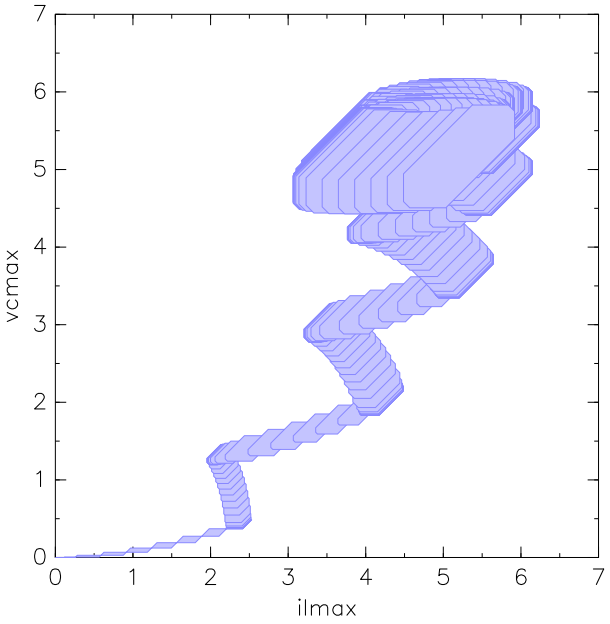

Fig. 3. Current-versus-voltage plot of the reachable states from startup for the buck-converter. Units are in amps and volts.

III. MODEL CHECKING FOR MULTILEVEL CONVERTERS

In this section, we present a multilevel converter model and model checking procedure used to verify properties of it. A multilevel converter takes several DC voltage sources and creates an AC output as shown

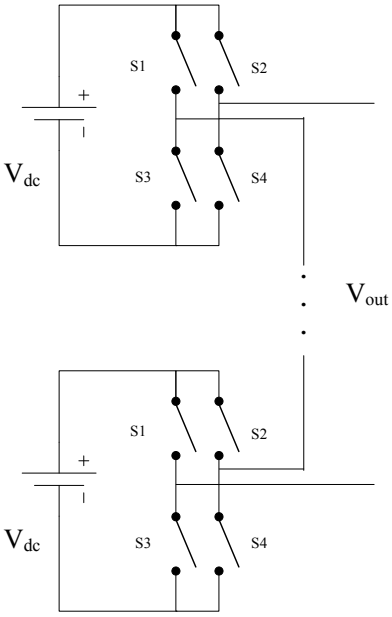

Fig. 4. Multilevel converter topology.

in Figure 4. Such converters have been investigated recently as they are useful in allowing grid connections from photovoltaic cells [27]. In an actual circuit, the switches are usually realized using H-bridges and may suffer from some non-idealities, but in our model, we assume the switches are ideal.

While the circuit relies on H-bridges, we can simply abstract the circuit elements away, and arrive at a basic timed automata model shown in Figure 5. In essence, the multilevel converters sums different numbers of the DC voltage sources at different points in time to generate the time-varying output. These switching times are usually chosen to minimize total harmonic distortion of the desired output wave, for instance, a sinusoid if going to the grid [28].

We implemented a simple model of this in the timed automata model checker Uppaal [6]. Uppaal works on a class of systems with simpler dynamics than the SpaceEx tool used for verifying the buck-converter, and for this class of systems, one can exactly compute the set of reachable states, so no overapproximation is necessary. The example multilevel converter had 5 DC voltage sources of 2V each, when connected by an H-bridge can easily reverse polarity to provide $\pm$2V sources. A single execution (simulation) of the system is shown in Figure 6.

We also verified several properties for all possible runs using Uppaal. We verified a simple property that the output voltage always lies in the range -10V to 10V.

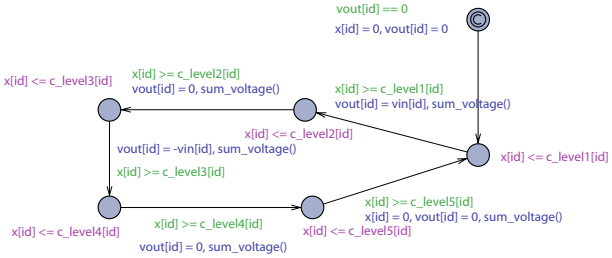

Fig. 5. Timed automata model of a multilevel converter. The model is parameterized by the variable *id*, which ranges in the number of levels of the converter, so here, *id* ranges from 1 to 5 for the 5 levels. The states are circles and the initial state has a second concentric circle in it. The states have guards in green between them on transitions (edges) which specify the values the timer $x[id]$ must be in. The transitions also allow resets, which we use to model updating the output voltage $vout[id]$ to appropriate combinations of the input voltages $vin[id]$.

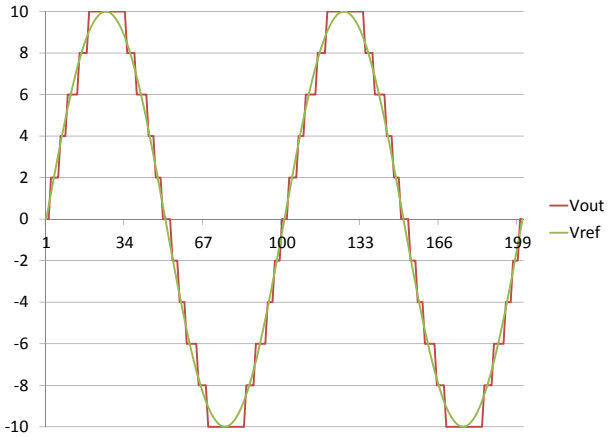

Fig. 6. Multilevel converter simulation from Uppaal using 5 DC voltage sources of 2V each. The horizontal axis is in discrete time steps and the vertical is in volts.

Additionally, we verified properties that show that the output is always sinusoidal with a desired frequency. Showing this essentially amounted to establishing that the output voltage is at the desired sum of levels at the appropriate times. Using Uppaal, we could extend the model to allow for nondeterminism in the switching times. However, also allowing for variation in the DC voltage sources would be more interesting, and could be accomplished using the hybrid systems reachability tools described earlier.

IV. Conclusion

In this paper, we presented our results on using verification techniques from the hybrid systems literature applied to the problem of formal verification of two types of circuits. The first class of circuits we addressed use traditional circuit elements with parameter variation

in switched-mode power supplies. The second class of circuits could be modeled as simple timed automata, such as multilevel converters. For future work, we are interested in applying these reachability methods to larger circuit designs with more non-idealities.

ACKNOWLEDGMENTS

We acknowledge the support of the Promoting Undergraduate Research in Engineering (PURE) program at the University of Illinois at Urbana-Champaign, which helped begin this work. We also acknowledge the support Rockwell Collins has provided the PURE program. Lastly, we acknowledge the help former PURE participants Soonwoo (Daniel) Chang, Haeran Lee, Youngho (Ryan) Park, and Yosub Shin who contributed to this work in fall 2008.

REFERENCES

[1] T. Dang, A. Donz, and O. Maler, "Verification of analog and mixed-signal circuits using hybrid system techniques," in *Formal Methods in Computer-Aided Design*, ser. Lecture Notes in Computer Science, A. Hu and A. Martin, Eds. Springer Berlin / Heidelberg, 2004, vol. 3312, pp. 21–36.

[2] C. Yan and M. Greenstreet, "Faster projection based methods for circuit level verification," in *Design Automation Conference, 2008. ASPDAC 2008. Asia and South Pacific*, Mar. 2008, pp. 410–415.

[3] E. M. Hope, X. Jiang, and A. D. Dominguez-Garcia, "A reachability-based method for large-signal behavior verification of dc-dc converters," *Circuits and Systems I, IEEE Transactions on*, vol. 58, no. 12, pp. 2944–2955, Dec. 2011.

[4] G. Frehse, C. Le Guernic, A. Donzé, S. Cotton, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, and O. Maler, "Spaceex: Scalable verification of hybrid systems," in *Proc. 23rd International Conference on Computer Aided Verification (CAV)*, ser. LNCS. Springer, 2011.

[5] R. Alur and D. L. Dill, "A theory of timed automata," *Theoretical Computer Science*, vol. 126, pp. 183–235, 1994.

[6] J. Bengtsson, K. Larsen, F. Larsson, P. Pettersson, and W. Yi, "Uppaal: a tool suite for automatic verification of real-time systems," in *Hybrid Systems III*, ser. Lecture Notes in Computer Science, R. Alur, T. Henzinger, and E. Sontag, Eds. Springer Berlin / Heidelberg, 1996, vol. 1066, pp. 232–243.

[7] M. Althoff, A. Rajhans, B. Krogh, S. Yaldiz, X. Li, and L. Pileggi, "Formal verification of phase- locked loops using reachability analysis and continuization," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2011.

[8] T. A. Henzinger, P.-H. Ho, and H. Wong-Toi, "Hytech: amodel checker for hybrid systems," *International Journal on Software Tools for Technology Transfer (STTT)*, vol. 1, pp. 110–122, 1997.

[9] E. Asarin, O. Bournez, T. Dang, and O. Maler, "Approximate reachability analysis of piecewise-linear dynamical systems," in *Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Science, N. Lynch and B. Krogh, Eds. Springer Berlin / Heidelberg, 2000, vol. 1790, pp. 20–31.

[10] T. Henzinger, B. Horowitz, R. Majumdar, and H. Wong-Toi, "Beyond hytech: Hybrid systems analysis using interval numerical methods," in *Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Science, N. Lynch and B. Krogh, Eds. Springer Berlin / Heidelberg, 2000, vol. 1790, pp. 130–144.

[11] A. Chutinan and B. Krogh, "Computational techniques for hybrid system verification," *Automatic Control, IEEE Transactions on*, vol. 48, no. 1, pp. 64–75, Jan. 2003.

[12] G. Frehse, "Phaver: Algorithmic verification of hybrid systems past hytech," in *HSCC*, 2005, pp. 258–273.

[13] A. Girard, C. Le Guernic, and O. Maler, "Efficient computation of reachable sets of linear time-invariant systems with inputs," in *Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Science, J. Hespanha and A. Tiwari, Eds. Springer Berlin / Heidelberg, 2006, vol. 3927, pp. 257–271.

[14] A. Kurzhanskiy and P. Varaiya, "Ellipsoidal toolbox (et)," in *Decision and Control, 2006 45th IEEE Conference on*, Dec. 2006, pp. 1498–1503.

[15] E. Asarin, T. Dang, and A. Girard, "Hybridization methods for the analysis of nonlinear systems," *Acta Informatica*, vol. 43, pp. 451–476, 2007.

[16] M. Althoff, C. Le Guernic, and B. H. Krogh, "Reachable set computation for uncertain time-varying linear systems," in *Proceedings of the 14th international conference on Hybrid systems: computation and control*, ser. HSCC '11. New York, NY, USA: ACM, 2011, pp. 93–102.

[17] C. Tomlin, I. Mitchell, A. Bayen, and M. Oishi, "Computational techniques for the verification of hybrid systems," *Proceedings of the IEEE*, vol. 91, no. 7, pp. 986–1001, 2003.

[18] C. L. Guernic and A. Girard, "Reachability analysis of linear systems using support functions," *Nonlinear Analysis: Hybrid Systems*, vol. 4, no. 2, pp. 250–262, 2010.

[19] R. P. Severns and G. Bloom, *Modern DC-to-DC Switchmode Power Converter Circuits*. New York, New York: Van Nostrand Reinhold Company, 1985.

[20] D. Liberzon, *Switching in Systems and Control*. Boston, MA, USA: Birkhäuser, 2003.

[21] P. Krein, J. Bentsman, R. Bass, and B. Lesieutre, "On the use of averaging for the analysis of power electronic systems," *Power Electronics, IEEE Transactions on*, vol. 5, no. 2, pp. 182–190, Apr. 1990.

[22] E. Hansen, "Interval arithmetic in matrix computations, part i," *Journal of the Society for Industrial and Applied Mathematics: Series B, Numerical Analysis*, vol. 2, no. 2, pp. 308–320, 1965.

[23] S. Bialas, "A necessary and sufficient condition for the stability of interval matrices," *International Journal of Control*, vol. 37, no. 4, pp. 717–722, 1983.

[24] Y.-T. Juang and C.-S. Shao, "Stability analysis of dynamic interval systems," *International Journal of Control*, vol. 49, no. 4, pp. 1401–1408, 1989.

[25] J. A. Heinen, "Sufficient conditions for stability of interval matrices," *International Journal of Control*, vol. 39, no. 6, pp. 1323–1328, 1984.

[26] M. Althoff, B. H. Krogh, and O. Stursberg, "Analyzing reachability of linear dynamic systems with parametric uncertainties," in *Modeling, Design, and Simulation of Systems with Uncertainties*, ser. Mathematical Engineering, A. Rauh and E. Auer, Eds. Springer Berlin Heidelberg, 2011, vol. 3, pp. 69–94.

[27] L. Tolbert and F. Peng, "Multilevel converters as a utility interface for renewable energy systems," in *Power Engineering Society Summer Meeting, 2000. IEEE*, vol. 2, 2000, pp. 1271–1274.

[28] L. Tolbert, F. Z. Peng, and T. Habetler, "Multilevel converters for large electric drives," *Industry Applications, IEEE Transactions on*, vol. 35, no. 1, pp. 36–44, Jan. 1999.