Model Validation of PWM DC-DC Converters

***Abstract*—This paper presents hybrid automaton modeling, comparative model validation, and formal verification of stability through reachability analysis of PWM DC-DC converters. Conformance degree provides a measure of closeness between the proposed hybrid automata models and experimental data. Non-determinism due to variations in circuit parameters is modeled using interval matrices. In direct contrast to the unsound and computationally-intensive Monte Carlo simulation, reachability analysis is introduced to overapproximate the set of reachable states and ensure stable operation of PWM DC-DC converters. Using a 200 W experimental prototype of a buck converter, hybrid automata models of open-loop and hysteresis-controlled converters are first validated against experimental data using their conformance degrees. Next, converter stability is formally verified through reachability analysis and informally validated using Monte Carlo simulations and experimental results.**

***Index Terms*—DC-DC converter, formal verification, hybrid automaton, model validation, reachability analysis.**

I. Introduction

ABSTRACT models of PWM DC-DC converters should reasonably match the experimental data obtained from a hardware prototype despite parametric uncertainty. More-efficient stochastic simulation techniques are based on polynomial chaos, where parametric uncertainties are accounted for by a series of orthogonal polynomials that depends upon their probability distributions [1]. Series coefficients are computed using various intrusive (e.g., stochastic Galerkin [1]) or non-intrusive (e.g., stochastic collocation [2]) methods. Examples of such stochastic methods for electrical circuits and power systems include Galerkin-based generalized polynomial chaos [3], SPICE-compatible stochastic Galerkin [4], Galerkin-based generalized decoupled polynomial chaos [5], stochastic testing [6], and SPICE-compatible stochastic collocation approach [7]. In general, polynomial chaos methods suffer from the curse of dimensionality, slow convergence with discontinuous solutions, and substantial computational overhead [8]–[11].

Another conventional approach is the simulation-based Monte Carlo paradigm [12], [13], wherein considering all possible parameter variations and initial conditions is computationally-prohibitive. Moreover, for a higher level of confidence in results produced by the Monte Carlo analysis, greater number of simulation runs are required. Generally, the total number of Monte Carlo simulations, σ, has to be increased by 100-fold to achieve additional decimal place of precision, owing to the $O(\frac{1}{\sqrt{\sigma}})$ convergence rate [14]. Conceptually, to have a full confidence in Monte Carlo results, one would require infinite number of simulation runs [15], [16]. The level of required modeling fidelity depends on the critical nature of the application domain. For example, the root cause of the 2014 recall of around 700,000 Toyota Prius cars was attributed to an error in the interaction between a boost converter and its software controller [17]. Likewise, more than 100,000 Toyota Prius cars were recalled due to an inverter failure [18]. Therefore, this mission-critical domain would require significant confidence in the modeling accuracy. At the same time, the utilized model validation tool should be conservative enough to overapproximate all possible sets of states reachable by the model execution.

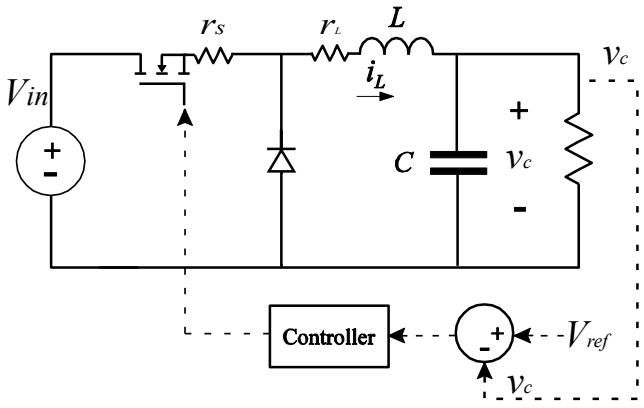

Fig. 1. Closed-loop DC-DC buck converter with main parasitic elements.

The formal verification community has been using reachability analysis-based model checking tools to have sufficient confidence in the model. Therefore, we first use rigorous model validation paradigms [19] to quantify the closeness between the abstract model waveforms and experimental data using the *conformance degree* [20]. Stable converter operation is then *formally verified* on the model using reachability analysis. The boundaries of state trajectories can be found from average-value models [21], [22]. Reachability analysis overapproximates the set of all possible reachable states (i.e., the reach sets) from a given set of initial states and parameter values. One can then confidently ascertain a stable converter operation if the reach sets remain within a desired region of the state space for a given time span. Without loss of generality, we have considered a DC-DC buck converter, with main parasitic elements, as shown in Fig. 1.

General reachability analysis tools include, but are not limited to, HyTech [23], PHAVer [24], UPPAAL [25], HSolver [26], d/dt [27], Flow* [28], and SpaceEx [29]. To effectively use such model checking tools, hybrid automata models of DC-DC converters are required [30]. Hybrid automaton modeling of DC-DC converters is presented in [31]–[36]. However, [33]–[35] do not consider component losses/variations and the discontinuous conduction mode (DCM), and do not perform the reachability analysis. PHAVer in [37] computes the reach sets for an open-loop boost converter but does not include DCM or component losses. MATLAB/Ellipsoidal Toolbox is used in [38] for the reachability analysis of DC-DC converters. However, Ellipsoidal-based set computations suffer

from the curse of dimensionality. SpaceEx (the successor of PHAVer) scales quite efficiently and is used as the reachability analysis tool in this paper.

We have formally defined a precise hybrid automaton model for PWM DC-DC converters, that accommodates main circuit parasitics, DCM, and the non-determinism due to parameter variations, that have not been considered altogether in any past work. We also use the notion of conformance degree to compare different model abstractions, using their output trajectories, that has not been used in any of the work cited above. Moreover, all the hybrid automata models are automatically generated. Herein, the proposed approach is shown to outperform the traditional Monte Carlo simulation in computation time. In summary, the main contributions of this paper are:

- Hybrid automata models for DC-DC converters are automatically generated, validated against Simulink/Stateflow, PLECS simulations, and hardware measurements, and verified using reachability analysis in SpaceEx. These models include component nonidealities and different operational modes.
- The conformance degree of the hybrid automata models validates these against the experimental data, by providing a proximity measure between executions/behaviors of these two in both time and space.
- Non-determinism due to parametric variations is modeled using interval matrices, which results in a set-valued additive input term in the system dynamics.
- The reachability analysis achieves a fixed point where there are no other reach sets (i.e., the model output will remain within reach sets as $t \rightarrow \infty$). It is impossible to get such success through Monte Carlo analysis.

The remainder of this paper is organized as follows: Hybrid automaton modeling is discussed in Section II. Application of conformance degree for model validation is discussed in Section III. Section IV uses interval analysis to model the non-determinism caused by the parameter variation. SpaceEx-based reachability analysis is discussed in Section V. Section VI validates the developed models against a 200 W buck converter prototype using the conformance degree, formally verifies the model properties using reachability analysis, and presents comparison with the Monte Carlo simulation. Section VII concludes the paper.

II. Hybrid Automaton Modeling

A. Preliminaries

DC-DC converters exhibit both continuous and discrete behaviors due to the presence of passive elements and switching components, respectively. Hybrid automaton modeling [39] integrates resulting differential equations and finite state machines in a single formalism. The state of a hybrid automaton model may change in two ways, i.e., through a continuous flow trajectory within a given topology (Definition 2.2), and through a discrete transition between two given topologies (Definition 2.3). A *topology* is defined as the circuit configuration in each switching sub-interval (Fig. 2). We define

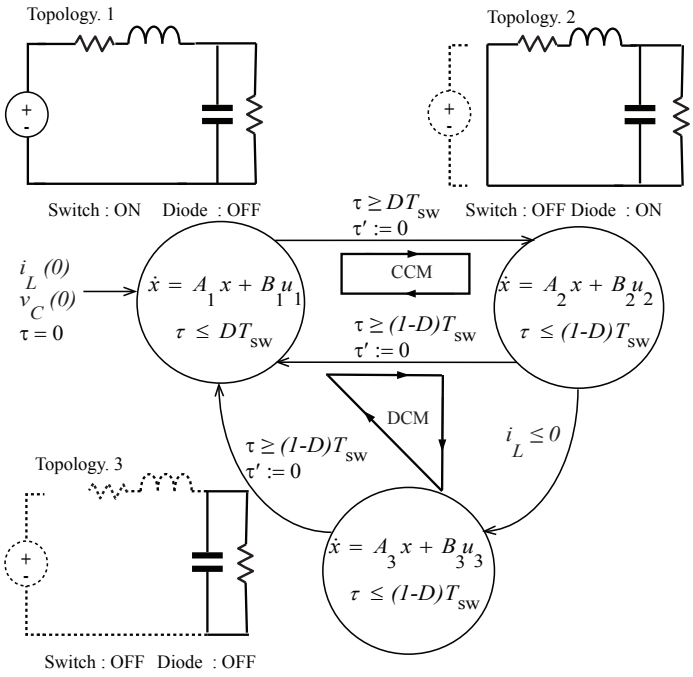

Fig. 2. Topologies, operational modes, and hybrid automaton modeling of a DC-DC buck converter.

$\mathbb{R}^n$ as the set of n-dimensional reals, and 2^X as the power set of a given set X, i.e., the set of all the subsets of X.

B. Hybrid Automaton Model Syntax and Semantics

We first formally define the model components in mathematical set representation, and then define the model execution as these components interact.

Definition 2.1: A *hybrid automaton model* is defined by a tuple $\mathcal{H} = \langle Q, X, init, U, E, g, G, inv, h, F \rangle$, which has the following components:

- *Topologies:* $Q = \{q_1, q_2, ..., q_N\}$ is a finite set of topologies.
- *State Variables*: $X \subseteq \mathbb{R}^n$ is set of continuous state variables. A *state* is defined by $(q, x) \in Q \times X$.
- *Initial Conditions*: $init \subseteq Q_0 \times X_0$ is a set of initial conditions, such that $Q_0 \subseteq Q$ and $X_0 \subseteq X$.
- *Inputs*: $U = \{u_1, u_2, ...u_N\}$ is the set of inputs for each topology.
- *Discrete Transitions*: $E \subset Q \times Q$ is a set of feasible discrete transitions allowed among the topologies, such that an element $e_{ij} = (q_i, q_j) \in E$ implies that a discrete transition from i^{th} topology to j^{th} topology is allowed. It might not be possible to visit the entire set of topologies from one particular topology (Definition 2.3).
- *Guard Function*: $g : E \rightarrow G$ is the guard function that maps each element $e_{ij} \in E$ to its corresponding guard $g(e_{ij}) \in G$.
- *Guards*: $G \subseteq 2^X$ is the guard set such that $\exists\ g(e_{ij}) \in G$ for each $e_{ij} \in E$. A *guard* is a property of the hybrid automaton model that must be satisfied by a state to take a discrete transition from a given topology to another pre-defined topology. A state $(q_k, x_k) \in Q \times X$ *satisfies* $g(e_{ij})$ (i.e, $(q_k, x_k) \vDash g(e_{ij})$) iff $q_k = q_i$ and $x_k \in g(e_{ij})$.

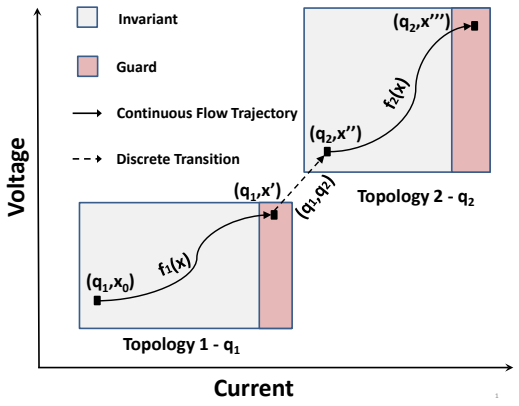

Fig. 3. Execution of the hybrid automaton model of DC-DC converters.

- *Invariants*: $inv : Q \rightarrow 2^X$ is a mapping that assigns an invariant $inv\,(q) \subseteq X$ for each topology $q \in Q$. An *invariant* is a property of the hybrid automaton model that must be satisfied by all the states for a given topology. A state $(q, x) \vDash inv(q)$ iff $x \in inv(q)$.
- *Reset of Continuous State*: $h : E \times X \rightarrow X$ resets the continuous state, i.e., if a discrete transition takes place from i^{th} topology to j^{th} topology as defined by $e_{ij} \in E$ with $x \in X$, the continuous state is reset to a new value $x' = h(e_{ij}, x) \in X$, such that $x' \in inv(q_j)$.
- *Set of ODEs*: F is the set of ordinary differential equations (ODEs) that are defined for each topology $q \in Q$ over the continuous variables $x \in X$. The continuous dynamics for each $q \in Q$ is defined by $F(q, x, u)$ over a given time horizon $t \in [\tau_1, \tau_2]$ that assigns a Lipschitz continuous vector space in $\mathbb{R}^n$.

Remarks: Here, $x' \in X$ symbolizes the new value of a continuous state $x \in X$ after a continuous flow or a discrete transition. If a state (q, x) does not satisfy an invariant $inv(q)$, then real time τ is stopped, forcing the continuous state x to stop evolving within a topology. The guard function ensures discrete transition to an appropriate topology, once the corresponding guard is satisfied. Here, invariants and guards are defined in the form of bounds over continuous state variables in Fig. 3.

Definition 2.2: The *continuous flow trajectory* $\mathcal{T}$ for a hybrid automaton model $\mathcal{H}$ is defined by the valuations of $x \in X$. For a given initial state $(q, x_0) \in Q \times X$ and $u \in U$, $\exists\, f(q, x, u) \in F$ that results in a final continuous state $x' \in X$, whereas q remains unchanged with given invariant $inv(q)$, iff $(q, x) \vDash inv(q)$. $\forall\ t \in [\tau_1\ \tau_2]$, $\mathcal{T}$ is given by

$$\mathcal{T}(q, x') = x_0 + \int_{\tau_1}^{\tau_2} f(q, x, u)dt. \tag{1}$$

and denoted by $(q, x_0) \xrightarrow{f} (q, x')$.

At each topology, converter dynamics can be modeled by ODEs; e.g., system matrices A_q and B_q describe the continuous flow trajectories in topology $q \in \{1, 2, 3\}$ of Fig. 2.

Definition 2.3: The *discrete transition* for a hybrid automaton model $\mathcal{H}$ is defined as: for a given state $(q_i, x) \in Q \times X$ and $u \in U$, there is a function $h(e_{ij}, x)$ that resets the continuous state to $x' \in X$, and the topology to q_j, iff $(q_i, x) \vDash inv(q_i)$ and $(q_i, x) \vDash g(e_{ij}) \in G$, and $\exists\ e_{ij} \in E$. The discrete transition is denoted by $(q_i, x) \xrightarrow{h} (q_j, x')$.

Definition 2.4: An *execution* of a hybrid automaton model $\mathcal{H}$ is an alternating sequence of continuous flow trajectories and discrete transitions.

The example of an execution is shown in Fig. 3.

The switching instance can be determined either externally (e.g., by a duty cycle command for the MOSFET) or internally (e.g., by meeting appropriate threshold conditions for the diode). The sequence of topologies, observed periodically in the steady state, defines an operational mode. Example of three topologies and two operational modes for a buck converter are shown in Fig. 2.

C. Model Instantiation for DC-DC Converters

We may now implement the syntax and semantics of the hybrid automaton model developed above for DC-DC converters. We define D as the duty cycle, T_{sw} as the switching period, and V_{in} as the DC input voltage. We can represent the continuous dynamics for a given topology as a standard set of state-space equations

$$\frac{dx}{dt} = A_q x + B_q u \tag{2}$$

where, $x \in \mathbb{R}^n$ is a vector of continuous states, Q is a finite set of topologies, $u \subseteq U$ such that $U \subseteq \mathbb{R}^m$ is a set of input vectors, and $A_q \in \mathbb{R}^{n \times n}$ and $B_q \in \mathbb{R}^{n \times m}$ are system matrices. Such formation can be readily created for the buck converter in Fig. 2. The instantiation of the hybrid automation model for an open-loop DC-DC converter, as per Definition 2.1, is:

- Three topologies are denoted by $Q = \{q_1, q_2, q_3\}$.
- The continuous state vector is $x = [i_L\ v_C\ \tau]'$, where τ represents real time such that $\frac{d\tau}{dt} = 1$.
- $U = \{[V_{in}, 0, 0]', [0, 0, 0]', [0, 0, 0]'\}$ forms the input vector set.
- $E = \{(q_1, q_2), (q_2, q_1), (q_2, q_3), (q_3, q_1)\}$ defines the feasible discrete transitions, e.g., (q_2, q_3) means a discrete transition from topology 2 to 3 is allowed.
- Guard set, for the corresponding elements of E, is defined by $G = \{(\tau \geq DT_{sw}), (\tau \geq (1 - D)\, T_{sw}), (i_L \leq 0), (\tau \geq (1 - D)\, T_{sw})\}$.
- The continuous flow trajectory is defined by (2), with the corresponding state matrices for each topology. For topology 1, this can be denoted by $(q_1, x_0) \xrightarrow{f} (q_1, x')$, as shown in Fig. 3. Here, (q_1, x_0) is the initial state and (q_1, x') is the final state as the automaton continuously evolves with the continuous flow dynamics $f_1(x)$.
- The reset function h defines a new continuous state x'' for the new topology. For example, if a transition is to take place from topology 1 to topology 2 with some final state $x' \in X' \subset X$ in topology 1, h assigns the new state $x'' \in X'' \subset X$ in topology 2. For topology 1 to topology 2, a discrete transition is denoted by $(q_1, x') \xrightarrow{h} (q_2, x'')$, as shown in Fig. 3.

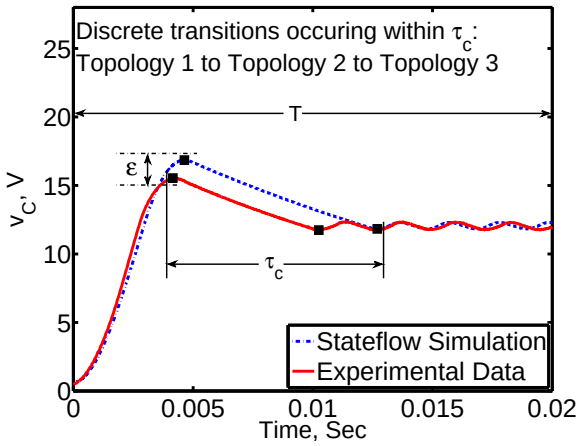

Fig. 4. Output trajectories of capacitor voltage for the closed-loop controlled buck converter - local mismatch for interval τ_c and ε.

The evolution of the hybrid automaton model starts with initial conditions from set $init$, e.g., $(q_1, x_0) \in init$ for a given input $u_1 = [V_{in}, 0, 0]'$ and, subsequently, the continuous state evolves according to the set of ODEs defined by (2) (i.e., F in Definition 2.1). The topology remains the same, i.e., $q(t) = q_1$, as x_0 evolves inside the invariant $inv(q_1)$, such that it attains a final value $x' \in inv(q_1)$. Once the continuous state x' satisfies the guard $g(e_{q_1 q_2})$ corresponding to the edge $e_{q_1 q_2} \in E$, the topology may transition from q_1 to q_2, and the continuous state is reset with a new value x'' in the new invariant set $inv(q_2) \subset X$ with a new input $u_2 = [0, 0, 0]'$.

This hybrid automaton model can be extended to closed-loop DC-DC converters, e.g., hysteresis-controlled converters. The tuple remains the same except that the guards are defined in terms of switching boundaries. The hysteresis band is formed by defining an upper switching boundary, $V_{ref} + \delta$, and a lower switching boundary, $V_{ref} - \delta$, where V_{ref} is the desired output voltage, and δ is the tolerance level. Thus, $G = \{(v_C \geq V_{ref} + \delta), (v_C \leq V_{ref} - \delta), (i_L \leq 0), (v_C \leq V_{ref} - \delta)\}$.

It should be noted that time τ does not appear in the guard expressions. Therefore, we have developed two hybrid automata models for the closed-loop buck DC-DC converter, i.e., one with variable τ (called the *time-dependent* hybrid automaton model), and another without variable τ (called the *time-independent* hybrid automaton model). For the time-independent hybrid automaton model, we perform the reachability analysis for an unbounded time, i.e., compute the reach sets as $t \to \infty$.

III. Validation through Conformance Degree

Model validation of DC-DC converters requires comparing *output trajectory* as defined by (1) for a given hybrid automaton model $\mathcal{H}$ with the measured data from an experimental prototype referred to as $\mathcal{I}$.

Our goal is to find an appropriate measure of distance for output trajectories of hybrid automata models. One can consider the output trajectories of the capacitor voltage (v_C) for a closed-loop buck converter shown in Fig. 4. The experimental data obtained from a prototype and output trajectory of the hybrid automaton model in Simulink/Stateflow are overlaid. Intuitively, the two output trajectories look similar; however, the sup norm would give a large value to the distance between them. This is, partly, because $\mathcal{I}$ and $\mathcal{H}$ might transition among various topologies at slightly different moments in time. Therefore, our distance measure should allow some *wiggle room* in time. Rather than comparing only the states that are exactly time-aligned, it should allow comparison of states that are within some $\tau_c > 0$ time units of each other.

Moreover, it is not appropriate to compare outputs when two systems have executed different numbers of discrete transitions. Thus, our distance measure must only compare states after an equal number of discrete transitions between topologies of the two systems. Note that within the time window τ_c in Fig. 4, both the hardware prototype as well as the Stateflow model exhibit two discrete transitions. To this end, we introduce the parameter $j \in \mathbb{N}$, that counts the number of discrete transitions each system makes, where $\mathbb{N}$ is the set of natural numbers. It is reasonable to require that the transition times of the two systems be close to consider that the systems themselves are close: the value τ_c will also bound the difference in transition times. The distance measure will account for the distance between output trajectories, captured by the value $\varepsilon > 0$. Thus, we have a 2-value distance measure, with values τ_c and ε capturing the time and space distance between the two output trajectories as illustrated in Fig. 4.

The output trajectories of hybrid automata models are parameterized with t and j. The time spent in a given converter topology is $t \in \mathbb{R}_{>0}$, and $j \in \mathbb{N}$ counts the number of discrete transitions between different topologies (where $\mathbb{R}_{>0}$ is the set of positive real numbers). We write $y(t, j)$ for the output trajectory at the hybrid time $(t, j) \in \mathbb{R}_{>0} \times \mathbb{N}$, i.e., at time t and after j transitions. Let $domy \subset \mathbb{R}_{>0} \times \mathbb{N}$ denote the domain of output trajectory y, i.e., the set of all (t, j), so that $(T, J, \tau_c, \varepsilon)$-closeness [20] can be formally defined.

Definition 3.1: Take an output trajectory for time $T \in \mathbb{R}_{>0}$, a maximum number of discrete transitions $J \in \mathbb{N}$, and parameters $\tau_c, \varepsilon > 0$. Two output trajectories y_1 and y_2 are $(T, J, \tau_c, \varepsilon)$-close, shown as $y_1 \approx_{(\tau_c, \varepsilon)} y_2$, if (a) for all $(t, j) \in \text{dom}y_1$ such that $t \leq T, j \leq J$, there exists $(s, j) \in \text{dom}y_2$ where $|t - s| \leq \tau_c$, and $\|y_1(t, j) - y_2(s, j)\| \leq \varepsilon$, and (b) for all $(s, j) \in \text{dom}y_2$ such that $s \leq T, j \leq J$, there exists $(t, j) \in \text{dom}y_1$ where $|t - s| \leq \tau_c$, and $\|y_2(s, j) - y_1(t, j)\| \leq \varepsilon$.

$(T, J, \tau_c, \varepsilon)$-closeness gives a proximity measure between the two output trajectories in both time and space. It shows that for every point $y_1(t, j)$, y_2 has a point $y_2(s, j)$ which is ε-close to it, and may occur anywhere in the window $[t - \tau_c, t + \tau_c]$ (and vice versa). Allowing this wiggle room in time is important when comparing the output trajectories, because the discrete transitions could occur at different times. The two values T and J limit our testing horizon. $(T, J, \tau_c, \varepsilon)$-closeness can be lifted from output trajectories to systems. One can validate the model through the conformance degree between its output trajectory and measured data.

Definition 3.2: Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be two hybrid automata models. The *conformance degree* of $\mathcal{H}_1$ to $\mathcal{H}_2$, given τ_c, is defined as the smallest ε such that for every trajectory y_1 of $\mathcal{H}_1$, there exists a trajectory y_2 of $\mathcal{H}_2$, where $y_1 \approx_{(\tau_c, \varepsilon)} y_2$. We denote this conformance degree by $\mathbf{CD}_\tau(\mathcal{H}_1, \mathcal{H}_2)$.

We will use this definition intuitively for model validation of DC-DC converters. We compute the conformance degree $\mathbf{CD}_\tau(\mathcal{H}_1, \mathcal{H}_2)$ for some $\tau_c > 0$ in different case studies of Section VI, and effectively say that some local mismatch is permissible within a window τ_c for the output trajectories of the models and the hardware prototype.

IV. Modeling Non-Determinism using Interval Analysis

The system matrices in the hybrid automata models of DC-DC converters depend on component values. The variations due to manufacturing tolerance, aging, and temperature result in non-determinism of component values. Analysis of electrical circuits with such variations has been reported in literature using interval arithmetic-based genetic optimization [40] and affine arithmetic [41]. We use the interval arithmetic [42] to incorporate the parameter variations within the reachability analysis framework. The range of component values are represented in terms of intervals. A real interval v is a set of real numbers given by

$$[\underline{v}, \overline{v}] = \{v \in \mathbb{R} : \underline{v} \leq v \leq \overline{v}\}, \tag{3}$$

where $\underline{v}$ is the infimum and $\overline{v}$ is the supremum. Given two intervals, $[\underline{u}, \overline{u}]$ and $[\underline{v}, \overline{v}]$, their product is another interval given by

$$[\underline{u}, \overline{u}] * [\underline{v}, \overline{v}] = [min(\underline{uv}, \underline{u}\overline{v}, \overline{u}\underline{v}, \overline{uv}), max(\underline{uv}, \underline{u}\overline{v}, \overline{u}\underline{v}, \overline{uv})]. \tag{4}$$

The quotient of two intervals, with a non-zero divisor, is

$$\frac{[\underline{u}, \overline{u}]}{[\underline{v}, \overline{v}]} = [\underline{u}, \overline{u}] * \left(\frac{1}{[\underline{v}, \overline{v}]}\right), \tag{5}$$

where

$$\left(\frac{1}{[\underline{v}, \overline{v}]}\right) = \left[\frac{1}{\overline{v}}, \frac{1}{\underline{v}}\right]. \tag{6}$$

If $[\underline{v}, \overline{v}]$ has both bounds negative, then

$$\left(\frac{1}{[\underline{v}, \overline{v}]}\right) = \left[\frac{1}{\underline{v}}, \frac{1}{\overline{v}}\right]. \tag{7}$$

These intervals may also be defined by the midpoint-radius representation

$$mid(v) = \frac{1}{2}(\underline{v} + \overline{v}), \tag{8}$$

$$rad(v) = \frac{1}{2}(\overline{v} - \underline{v}). \tag{9}$$

The interval matrix for the system matrix is $\mathcal{A} = [\underline{A}, \overline{A}]$. System stability can be deferred by examining matrix extrema, i.e., $\underline{A}$ and $\overline{A}$ [43]. Therefore, it is sufficient to consider every combination of matrix extrema to overapproximate the reach set. The overapproximation of an interval matrix $\mathcal{A}$ is given by splitting it into two parts, i.e., a nominal part and a symmetric part [44]. Consider a linear dynamic system with n state variables having single deterministic input V_{in}, with the following state-space representation

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} V_{in}. \tag{10}$$

We use SpaceEx reachability analysis tool (discussed in Section V) to compute the reach sets for non-deterministic hybrid automaton model, with linear dynamics defined by (10) for a given topology. It may be mentioned that SpaceEx, in its present version, does not fully handle the matrix algebra operations. Hence, we need to define the state dynamics as scalar combination of other state variables. For example, for the i^{th} state variable in (10), one has

$$\dot{x}_i = a_{i1}x_1 + a_{i2}x_2 + ... + a_{ij}x_j + ... + a_{in}x_n + b_iV_{in}. \tag{11}$$

To incorporate parameter variation, one can replace the above coefficients with intervals, and write the expression as a differential inclusion

$$\dot{x}_i \in [\underline{a_{i1}}, \overline{a_{i1}}]x_1 + ...[\underline{a_{ij}}, \overline{a_{ij}}]x_i ... + [\underline{a_{in}}, \overline{a_{in}}]x_n + [\underline{b_i}, \overline{b_i}]V_{in}. \tag{12}$$

Since SpaceEx, in its present version, does not support the interval arithmetic, the intervals $[\underline{a_{ij}}, \overline{a_{ij}}]$ of (12) are computed outside the SpaceEx environment using (4), (5), (6), and (7). Subsequently, these intervals are transformed into the midpoint-radius representation to include the state and parametric intervals before implementing in the SpaceEx environment. Using (8) and (9), one can write (12) in a midpoint-radius representation as

$$\begin{aligned} \dot{x}_i \in\ & \{mid(a_{i1}) \pm rad(a_{i1})\}x_1 + ... + \{mid(a_{ij}) \pm \\ & rad(a_{ij})\}x_j + ... + \{mid(a_{in}) \pm rad(a_{in})\}x_n \\ & + \{mid(b_i) \pm rad(b_i)\}V_{in}. \end{aligned} \tag{13}$$

The mid-points correspond to the nominal parameter values that are constant terms, which can be separated as

$$\begin{aligned} \dot{x}_i \in\ & (a_{i1}x_1 + r_{i1}) + ... + (a_{ij}x_j + r_{ij}) + ... + \\ & (a_{in}x_n + r_{in}) + (b_iV_{in} + r_{bi}). \end{aligned} \tag{14}$$

This defines the continuous dynamics in the hybrid automaton model for the state variable x_i. The radii $r_{i1}, r_{i2}, ..., r_{ij}, ..., r_{in}$, and r_{bi} are expressed as product of the state and parametric intervals, such that r_{ij} is given by

$$r_{ij} \in [-rad(a_{ij}), rad(a_{ij})] * [\underline{x_j}, \overline{x_j}], \tag{15}$$

where, x_j varies between $\underline{x_j}$ and $\overline{x_j}$. For example, for the hysteresis-controlled DC-DC buck converter considered here, $\underline{v_C} = 0\ V$ and $\overline{v_C} = 20\ V$ in (15). Thus the coupling between the state variables is accommodated in the amended SpaceEx model by formulating r_{ij} in terms of $[\underline{x_j}, \overline{x_j}]$, and incorporating it in the dynamics in (14). The product of the two intervals in (15) is yet another interval, obtained using (4). The intervals thus computed are used in the model to define the lower and upper bounds for respective radii. Since this treatment of the state variables as intervals is not catered in Monte Carlo simulations, SpaceEx provides more reliable results.

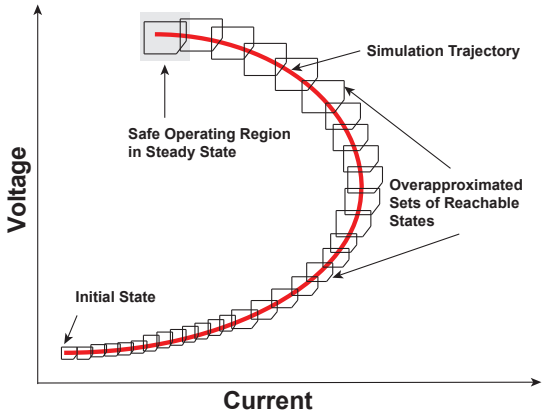

Fig. 5. Reachability analysis using reach sets for formal verification of a hybrid automaton model.

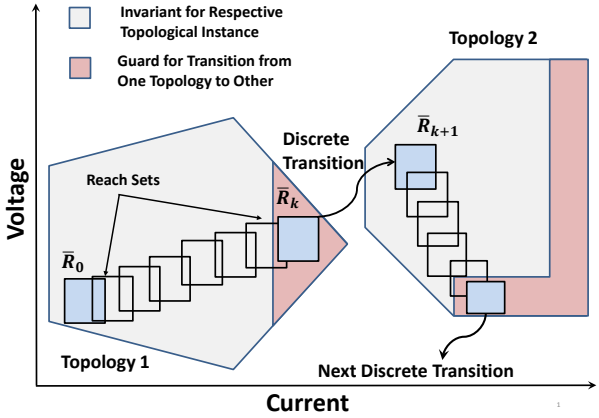

Fig. 6. Reach sets in different topologies with transitions imposed by guards.

V. Reachability Analysis for Hybrid Automata

Reachability analysis has been used by the formal verification community, and we have implemented it in its entirety for PWM DC-DC converters modeled as hybrid automata. In general, reachability analysis has been documented to produce more reliable results than Monte Carlo simulations:

1. The reachability analysis is more efficient. Monte Carlo analysis becomes computationally less tractable with increased size and complexity of a given system [38].
2. Reachability analysis is conclusive. In contrast, infinitely many Monte Carlo simulations are required to span the entire design parameter space and operational conditions and have full confidence in the final results [15], [16].
3. SpaceEx-based reachability analysis considers the entire state space [45], while Monte Carlo simulations only sample the parameter space. Generally, reachability analysis is theoretically superior and more sound [46].

We formally verify the stability properties of non-deterministic hybrid automata models of PWM DC-DC converters through the reachability analysis. We define the stability in the sense of Lyapunov, i.e., $\dot{x} = f(x(t))$ is stable if $\forall\ \theta > 0\ , \exists\ \beta > 0$ such that if $\|x(0)\| \leq \beta \Rightarrow \|x(t)\| \leq \theta\ \forall\ t \geq 0$. We may define a bounded region and verify that the output of the hybrid automaton model eventually reaches, and always remains, in this stable region, as seen in Fig. 5. We define the stability specification such that from the settling time t_s, the output voltage $V_C(t)$ should remain bounded within a tolerance γ of the reference voltage $V_{ref}(t)$, i.e., for $t \geq t_s \Rightarrow V_C(t) = V_{ref}(t) \pm \gamma$.

Definition 5.1: State x is *reachable* iff $\exists$ an execution α such that $x \in \alpha$.

Definition 5.2: The *set of reachable states* contains all the states that are reachable from a given set of initial conditions for a given time.

Consider an example of an autonomous system $\dot{x} = Ax$. The set of reachable states from initial time t_0 to final time t_f, from a given initial set X_0, is

$$\mathcal{R}_{t_0}^{t_f}(X_0) = \bigcup_{t \in [t_0, t_f]} e^{At} X_0. \tag{16}$$

However, (16) does not cater to the discrete transitions associated with the hybrid dynamical systems. Additionally, the exact set of all reachable states is undecidable [29].

In practice, overapproximations of the reachable states are computed using geometrical data structures (e.g, boxes, polytopes, ellipsoids, or zonotopes [47]), and denoted by $\overline{\mathcal{R}}$. For simplicity, we call these overapproximations as the *reach sets* in this paper. This framework can be extended to hybrid dynamical systems by including invariants and guard sets (Fig. 6), and implemented in various reachability analysis tools by software research community as mentioned in Section I. The reach sets for continuous dynamics can be computed using continuous post-operators so long as the continuous dynamics of DC-DC converter are contained within the invariant set defined for the corresponding topology or do not enter the guard set. Once the guard condition is satisfied within an invariant, a transition takes place from topology 1 to topology 2 such that the next reach set is computed using discrete post-operator. This process goes on until either the final time in a local time horizon, or a fixed point, is reached. A *fixed point* signifies that the reachability algorithm cannot find any new reach set during the current iteration other than those computed in the previous iteration. SpaceEx reachability tool computes the reach sets of a hybrid dynamical system. It is a classical fixed point algorithm based on computation of symbolic states [29].

Definition 5.3: A *symbolic state* is defined as a pair (q, Θ), where q is a topological instance, and Θ is the corresponding convex continuous set.

The reach set $\overline{\mathcal{R}}$ is obtained by computing the set of symbolic states. This reach set is the fixed point of the sequence $\overline{\mathcal{R}}_o = post_c\,(Init)$, and the successors are

$$\overline{\mathcal{R}}_{k+1} := \overline{\mathcal{R}}_k \bigcup post_c\left(post_d\left(\overline{\mathcal{R}}_k\right)\right) \tag{17}$$

where, $post_d$ is the *discrete post-operator* that defines the reach sets after a discrete transition from $\overline{\mathcal{R}}_k$. This corresponds to the h function defined in Definition 2.1. The *continuous post-operator*, $post_c$, defines the reach sets for the continuous states from $\overline{\mathcal{R}}_k$ after an arbitrary amount of time is elapsed. This corresponds to F in Definition 2.1.

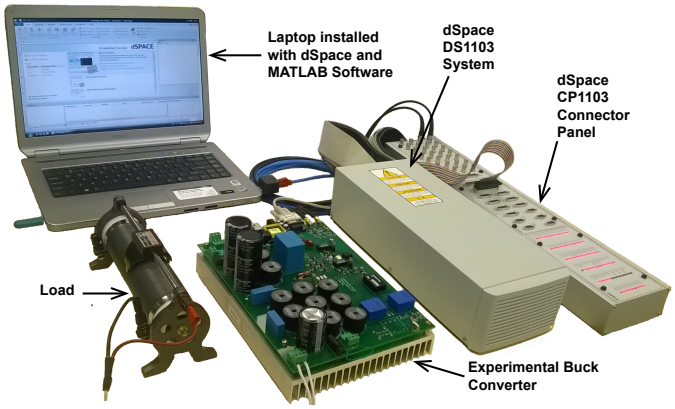

Fig. 7. Buck converter prototype controlled with dSPACE DS1103.

An approximated computation of Θ_k is given in [29] for the k^{th} time step. Hence, a sequence of convex continuous sets $\Theta_0, \Theta_1, ...\Theta_{N-1}$ is computed to form a *flowpipe* that covers the reach sets up to a pre-defined time such that N represents the number of time steps. This flowpipe is then used to compute the transition successors. Only those states can take the transition that satisfy the guard associated with the present topology and the invariant of the target topology. This process is continued until a fixed point is reached, i.e., if all the reach sets that are computed in the present iteration, are contained in reach sets computed in the previous iteration, i.e., $\overline{\mathcal{R}}_{k+1} \subseteq \overline{\mathcal{R}}_k$. This signifies that no new reach sets could be found and the computation process may be terminated.

VI. Case Studies

An experimental setup of a buck converter, controlled with a dSpace DS1103 unit, has been prototyped, as shown in Fig. 7. The experimental results are used for benchmarking purposes against MATLAB/PLECS [48], Simulink/Stateflow [49], Monte Carlo simulations, and SpaceEx reachability analysis. Circuit parameters $L = 2.65$ mH, $C = 2.2$ mF, and $R = 10$ Ω are used throughout this study. The non-determinism due to the parameter variations is modeled using the interval matrices in SpaceEx model. For a coherent comparison in terms of parameter variations in R, L, and C, we have used 15% tolerance for Monte Carlo simulations and SpaceEx reachability analysis. We have used the Hybrid Source Transformer (HyST) which is a source-to-source conversion tool for hybrid automata models [50]. The hybrid automaton model is developed using the java interface in MATLAB, and transformed into a SpaceEx compatible model using HyST data structures. We use the conformance degree to validate the hybrid automaton model against the experimental data. Then, the reachability analysis results are provided for formal verification of an open-loop and a hysteresis-controlled buck converter.

A. Model Validation Using Conformance Degree Testing

We use notations $\mathcal{I}_O$ and $\mathcal{I}_C$ for hardware prototypes in open-loop and closed-loop configurations, respectively. PLECS and Stateflow models are denoted by $\mathcal{H}_{OP}$, $\mathcal{H}_{CP}$ and $\mathcal{H}_{OS}$, $\mathcal{H}_{CS}$, respectively, where subscript O denotes an open-loop and C denotes a closed-loop configuration. The computed ε values against τ_c (as defined in Section III) are tabulated in Table I for the corresponding output trajectories. It is evident from Table I that the ε values of $\mathcal{H}_{OP}$ and $\mathcal{H}_{OS}$ as well as $\mathcal{H}_{CP}$ and $\mathcal{H}_{CS}$ are close enough (also, as seen in Fig. 8). We have computed conformance degrees for the prototype buck converters, i.e., $\mathcal{I}_O$ and $\mathcal{I}_C$, in comparison with other models, i.e., $\mathcal{H}_{OP}$, $\mathcal{H}_{OS}$ and $\mathcal{H}_{CP}$, $\mathcal{H}_{CS}$. We also define the absolute value of the maximum difference measured between the two given output trajectories as Δ for a given time duration τ_c. The measured Δ values are tabulated in Table I. The ε values depicted in Table I provide enough wiggle room in comparison with the corresponding Δ to validate that $\mathcal{H}_{OP}$ and $\mathcal{H}_{OS}$ are reasonable abstractions for $\mathcal{I}_O$, whereas $\mathcal{H}_{CP}$ and $\mathcal{H}_{CS}$ are reasonable abstractions for $\mathcal{I}_C$. Consider, for example, the case of a closed-loop buck converter. The 1^{st} row under closed-loop configuration in Table I provides the ε value (i.e., $\varepsilon = 3.6667$ A) and Δ value (i.e., $\Delta = 3.0590$ A), as we compare the inductor current (i_L) output trajectories for PLECS and experimental prototype. Δ of the two output trajectories remain within ε (as also depicted in Fig. 9 (a)). This is also true for the corresponding output trajectories of capacitor voltage (v_C). Accordingly, the hybrid automata models are validated in conformance with both the open-loop and the closed-loop converter prototypes.

TABLE I
Conformance Degree Analysis

Config.	Type of Output Trajectories	τ_c Value (s)	ε Value	Δ Value
Open Loop	i_L - PLECS vs Experiment	3×10^{-4}	5.1515 A	4.5570 A
	i_L - Stateflow vs Experiment	3×10^{-4}	5.0008 A	4.5570 A
	i_L - Stateflow vs PLECS	3×10^{-4}	0.1785 A	0 A
	v_C - PLECS vs Experiment	3×10^{-4}	1.8945 V	1.7202 V
	v_C - Stateflow vs Experiment	3×10^{-4}	2.3201 V	1.7202 V
	v_C - Stateflow vs PLECS	3×10^{-4}	0.6666 V	0 V
Closed Loop	i_L - PLECS vs Experiment	8×10^{-4}	3.6667 A	3.0590 A
	i_L - Stateflow vs Experiment	8×10^{-4}	3.6643 A	3.0590 A
	i_L - Stateflow vs PLECS	8×10^{-4}	0.0878 A	0 A
	v_C - PLECS vs Experiment	8×10^{-4}	2.8014 V	1.5905 V
	v_C - Stateflow vs Experiment	8×10^{-4}	2.7677 V	1.5905 V
	v_C - Stateflow vs PLECS	8×10^{-4}	0.0580 V	0 V

B. Formal Verification of the Open-loop Buck Converter

We consider the voltage stability specification to perform formal verification. For example, for $t_s = 0.025$ s, and $V_{ref} = 48$ V, we define $\gamma = 6$ V. This results in an upper voltage bound of 54 V, and lower voltage bound of 42 V, as shown in Fig. 8(b) by dotted lines. The input parameters are $V_{in} = 100$ V, and $f_s = 60$ kHz. The output trajectories and phase-plane responses are considered for the startup transients of the open-loop buck converter. The parameters' variations have been modeled using interval analysis in SpaceEx model, and also included in the Monte Carlo simulation. The reachability analysis results, obtained

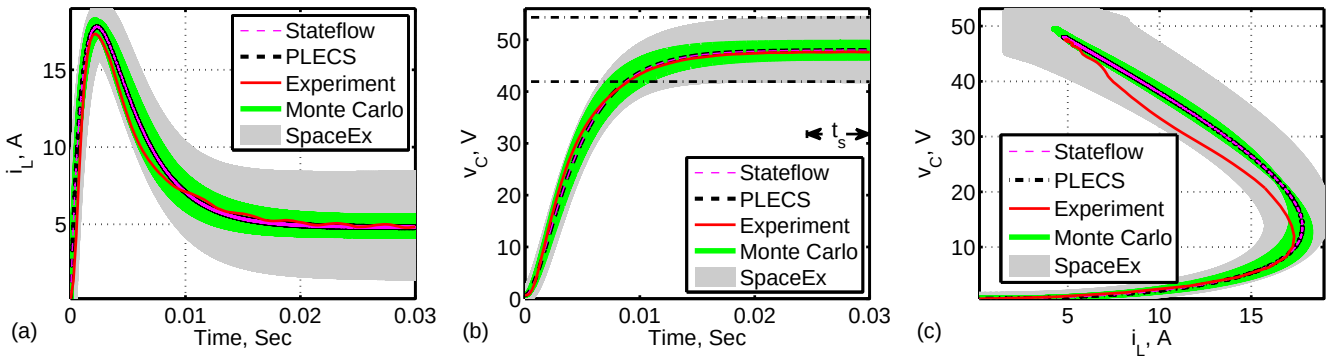

Fig. 8. Startup transients for an open-loop buck converter using interval matrices including Stateflow, PLECS, experiment, Monte Carlo, and SpaceEx; (a) current vs. time, (b) voltage vs. time, and (c) phase portrait.

using SpaceEx, are plotted in Fig. 8. It can be seen that the steady-state inductor current and capacitor voltage waveforms lie within the reachability analysis results, i.e., the simulations and measurement data are contained within the reach sets. Moreover, we verify that $v_C(t) \in [42, 54]$ for $t \geq t_s$ for Stateflow, PLECS, measurement data, Monte Carlo analysis, and SpaceEx analysis results.

C. Verification of the Hysteresis-controlled Converter

We define the voltage stability specification for the closed-loop buck converter to perform formal verification. For $t_s = 0.012$ s, and $V_{ref} = 12$ V, we define $\gamma = 1$ V. This leads to upper and lower voltage bounds of 13 and 11 V, respectively, as shown by dotted lines in Fig. 9(b). In this case study, the time-dependent and the time-independent models (as mentioned in Section II) are considered. First, SpaceEx reachability analysis is performed for the time-dependent model. The new parameters are $V_{in} = 24$ V, $V_{ref} = 12$ V, and $f_s = 50$ kHz. The trajectories are shown in Fig. 9 for Stateflow, PLECS, and experimental data along with reach sets computed using SpaceEx. The Stateflow, PLECS, and SpaceEx results match right from the start until the steady state is reached. Experimental results match that of Stateflow, PLECS, and SpaceEx in the steady state. It can be observed in Fig. 9 that Stateflow, PLECS, and measured results remain within the reach sets computed using SpaceEx, verifying $v_C(t) \in [11, 13]$ for $t \geq t_s$.

We can formally verify the time-independent SpaceEx model for an unbounded time, i.e., $t \to \infty$, by excluding τ. This would not be possible through Monte Carlo analysis as, even for a limited time span, one has to take into account infinite number of possible combinations. We have successfully achieved a fixed point using SpaceEx, with unbounded time, and with all possible parameter variations. The phase-plane plots are given for the start-up transients in Fig. 10. As seen, all results remain within the computed reach sets as $t \to \infty$, verifying $v_C(t) \in [11, 13]$ as $t \to \infty$.

A comparison of Monte Carlo analysis and SpaceEx reachability analysis, in term of computation times, is shown in Table II. Both are run on a Windows 7 SP1 (64 bit) platform, with Intel (R) core i7-2600 CPU with 3.40 GHz, 16.0 GB RAM, MATLAB version 8.5.0.197613 (R2015a), PLECS version 3.7.3, and SpaceEx version 0.9.8d. While infinite iterations are required to have full confidence in model validation through Monte Carlo analysis, we have only used finite (i.e., 2000) iterations as would be done in practice. Even then, it is evident that the SpaceEx reachability outperforms the Monte Carlo analysis in computation time, as seen in Table II.

VII. Conclusion

A hybrid automaton modeling approach for PWM DC-DC converters is developed. We have used the conformance testing for model validation when compared with a hardware prototype of DC-DC converters. The interval matrices analysis accommodates the model non-determinism caused by variations in component values. Reachability analysis frameworks are developed for formal verification of the resulting hybrid automata models. It is shown that the proposed reachability analysis outperforms the brute force Monte Carlo analysis in computation time and confidence level.

Acknowledgment

The authors would like to thank the anonymous reviewers whose valuable suggestions/comments have added value to this work and enabled us to effectively present our contributions.

References

[1] D. Xiu and G. E. Karniadakis, "The wiener–askey polynomial chaos for stochastic differential equations," *SIAM J. Sci. Comput.*, vol. 24, no. 2, pp. 619–644, Oct 2002.

[2] D. Xiu and J. S. Hesthaven, "High-order collocation methods for differential equations with random inputs," *SIAM J. Sci. Comput.*, vol. 27, no. 3, pp. 1118–1139, Dec 2005.

[3] A. Monti *et al.*, "A polynomial chaos theory approach to the control design of a power converter," in *Proc. 35th Power Electron. Specialists Conf.*, Aachen, Germany, 2004, pp. 4809–4813.

[4] P. Manfredi *et al.*, "Stochastic analysis of switching power converters via deterministic spice equivalents," *IEEE Trans. Power Electron.*, vol. 29, no. 9, pp. 4475–4478, Sept 2014.

[5] ——, "Generalized decoupled polynomial chaos for nonlinear circuits with many random parameters," *IEEE Microw. Wireless Compon. Lett.*, vol. 25, no. 8, pp. 505–507, Aug 2015.

[6] Z. Zhang *et al.*, "Stochastic testing method for transistor-level uncertainty quantification based on generalized polynomial chaos," *IEEE Trans. Comput.-Aided Des. Integr. Syst.*, vol. 32, no. 10, pp. 1533–1545, Oct 2013.

[7] A. Prasad and S. Roy, "Multidimensional variability analysis of complex power distribution networks via scalable stochastic collocation approach," *IEEE Trans. Compon. Packag. Technol.*, vol. 5, no. 11, pp. 1656–1668, Nov 2015.

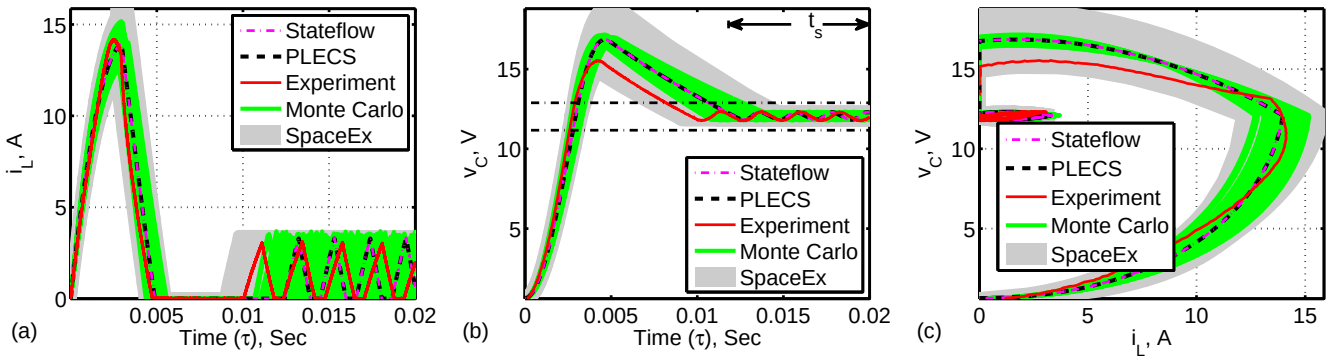

Fig. 9. Time-dependent hysteresis-controlled converter analysis using interval matrices including Stateflow, PLECS, experiment, Monte Carlo, and SpaceEx; (a) current vs. time, (b) voltage vs. time, and (c) phase portrait.

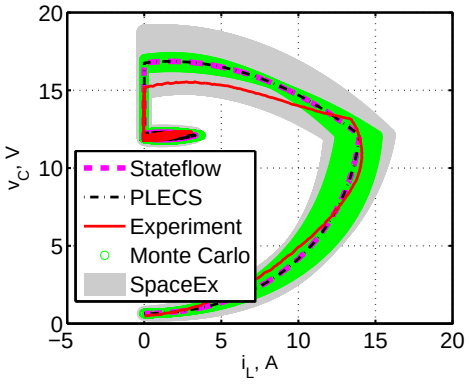

Fig. 10. Time-independent hysteresis-controlled converter analysis using interval matrices: Stateflow, PLECS, experiment, Monte Carlo, and SpaceEx.

TABLE II
COMPARISON OF MONTE CARLO AND SPACEEX ANALYSIS

System Configuration	Monte Carlo Iterations	Monte Carlo Time (s)	SpaceEx Time (s)	Times SpaceEx is Faster
Open Loop	2000	1.0151×10^4	872.9	11.63
Hysteresis control (time-dependent)	1000	315.43	1.5	210.29
Hysteresis control (time-independent)	1000	315.43	1.3	242.64
Hysteresis control, steady state (time-independent)	2000	1327	1.3	1.0208×10^3

[8] Z. Zhang *et al.*, "Efficient uncertainty quantification for the periodic steady state of forced and autonomous circuits," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 60, no. 10, pp. 687–691, Oct 2013.

[9] F. Lu *et al.*, "Limitations of polynomial chaos expansions in the bayesian solution of inverse problems," *J. Computational Physics*, vol. 282, pp. 138–147, Feb 2015.

[10] K. Konakli and B. Sudret, "Polynomial meta-models with canonical low-rank approximations: Numerical insights and comparison to sparse polynomial chaos expansions," *J. Computational Physics*, vol. 321, pp. 1144 – 1169, Jun 2016.

[11] M. Gerritsma *et al.*, "Time-dependent generalized polynomial chaos," *J. Computational Physics*, vol. 229, no. 22, pp. 8333–8363, Nov 2010.

[12] W. Huang *et al.*, "System accuracy analysis of the multiphase voltage regulator module," *IEEE Trans. Power Electron.*, vol. 22, no. 3, pp. 1019–1026, May 2007.

[13] M. del Casale *et al.*, "Selection of optimal closed-loop controllers for dc-dc voltage regulators based on nominal and tolerance design," *IEEE Trans. Ind. Electron.*, vol. 51, no. 4, pp. 840–849, Aug 2004.

[14] X. Luo, P. V. Shevchenko, and J. B. Donnelly, "Addressing the impact of data truncation and parameter uncertainty on operational risk estimates," *J. Operational Risk*, vol. 2, no. 4, pp. 3–27, Mar 2007.

[15] M. Althoff *et al.*, "Formal verification of phase-locked loops using reachability analysis and continuization," *ACM Commun.*, vol. 56, no. 10, pp. 97–104, Oct 2013.

[16] M. Althoff, "Formal and compositional analysis of power systems using reachable sets," *IEEE Trans. Power Syst.*, vol. 29, no. 5, pp. 2270–2280, Sep 2014.

[17] Toyota. (2014, Feb. 12) Defect information report (nhtsa recall 14v-053). [Online]. Available: http://www-odi.nhtsa.dot.gov/acms/cs/jaxrs/download/doc/UCM450071/RCDNN-14V053-0945.pdf

[18] ——. (2015, Jul. 15) Defect information report (nhtsa recall 15v-449). [Online]. Available: http://www-odi.nhtsa.dot.gov/acms/cs/jaxrs/download/doc/UCM482439/RCORRD-15V449-4622.pdf

[19] L. Ljung, *System Identification: Theory for the User*, 2nd ed. New Jersey, USA: Prentice-Hall, Inc., 1999.

[20] H. Abbas *et al.*, "Formal property verification in a conformance testing framework," in *Proc. ACM-IEEE 12th Int. Conf. Formal Methods and Models for Syst. Design*, Lausanne, 2014, pp. 155–164.

[21] H. Behjati *et al.*, "Alternative time-invariant multi-frequency modeling of pwm dc-dc converters," *IEEE Trans. Circuits Syst. I: Regular Papers*, vol. 60, no. 11, pp. 3069–3079, Nov 2013.

[22] J. Kimball and P. Krein, "Singular perturbation theory for dc-dc converters and application to pfc converters," *IEEE Trans. Power Electron.*, vol. 23, no. 6, pp. 2970–2981, Nov 2008.

[23] T. Henzinger *et al.*, "HyTech: A model checker for hybrid systems," in *Computer Aided Verification*, O. Grumberg, Ed. Berlin Heidelberg: Springer, Mar. 1997, pp. 460–463.

[24] G. Frehse, "PHAVer: Algorithmic verification of hybrid systems past HyTech," vol. 10, no. 3, Jun. 2008, pp. 263–279.

[25] J. Bengtsson *et al.*, "UPPAAL a tool suite for automatic verification of

real-time systems," in *Hybrid Systems III*. Berlin Heidelberg: Springer, Jun. 2005, pp. 232–243.
[26] S. Ratschan and Z. She, "Safety verification of hybrid systems by constraint propagation based abstraction refinement," in *Hybrid Systems: Computation and Control*, M. Morari and L. Thiele, Eds. Berlin Heidelberg: Springer, Mar. 2005, pp. 573–589.
[27] E. Asarin *et al.*, "The d/dt tool for verification of hybrid systems," in *Computer Aided Verification*, E. Brinksma and K. G. Larsen, Eds. Berlin Heidelberg: Springer, Sep. 2002, pp. 365–370.
[28] X. Chen *et al.*, "Flow*: An analyzer for non-linear hybrid systems," in *Computer Aided Verification*, ser. Lecture Notes in Computer Science, N. Sharygina and H. Veith, Eds. Springer Berlin Heidelberg, 2013, vol. 8044, pp. 258–263.
[29] G. Frehse *et al.*, "SpaceEx: Scalable verification of hybrid systems," in *Proc. 23rd Int. Conf. on Comput. Aided Verification*, Snowbird, UT, 2011, pp. 379–395.
[30] M. Miranda and A. Lima, "Formal verification and controller redesign of power electronic converters," in *Proc. IEEE Int. Symp. Indust. Electron.*, Ajaccio, France, 2004, pp. 907–912.
[31] T. Johnson *et al.*, "Design verification methods for switching power converters," in *Proc. 3rd Power and Energy Conf. at Illinois*, Urbana, IL, 2012, pp. 1–6.
[32] S. Hossain *et al.*, "Reachability analysis of closed-loop switching power converters," in *Proc. 4th Power and Energy Conf. at Illinois*, Urbana, IL, 2013, pp. 130–134.
[33] M. Hongbo and F. Quanyuan, "Hybrid modeling and control for buck-boost switching converters," in *Proc. Int. Conf. Commun., Circuits and Syst.*, Milpitas, CA, 2009, pp. 678–682.
[34] M. Senesky *et al.*, "Hybrid modelling and control of power electronics," in *Proc. 6th Int. Workshop on Hybrid Systems: Computation and Control*, Prague, Czech Republic, 2003, pp. 450–465.
[35] C. Sreekumar and V. Agarwal, "A hybrid control algorithm for voltage regulation in dcdc boost converter," *IEEE Trans. Ind. Electron.*, vol. 55, no. 6, pp. 2530 – 2538, Jun. 2008.
[36] Y. Quan *et al.*, "Simultaneous ccm and dcm operations of boost converter by a pwm hybrid control strategy," in *Proc. IEEE 39th Annual Conf. Ind. Electron. Society*, Vienna, Austria, 2013, pp. 1260–1265.
[37] U. Kuhne, "Analysis of a boost converter circuit using linear hybrid automata," ENS Cachan, Cedex, France, Tech. Rep., 2010.
[38] E. Hope *et al.*, "A reachability-based method for large-signal behavior verification of dc-dc converters," *IEEE Trans. Circuits Syst. I*, vol. 58, no. 12, pp. 2944–2955, Dec. 2011.
[39] T. Henzinger, "The theory of hybrid automata," in *Proc. IEEE Symp. on Logic in Comput. Science*, New Brunswick, NJ, 1996, pp. 278–292.
[40] N. Femia and G. Spagnuolo, "Genetic optimization of interval arithmetic-based worst case circuit tolerance analysis," *IEEE Trans. Circuits Syst. I: Fundam. Theory Appl.*, vol. 46, no. 12, pp. 1441–1456, Dec 1999.
[41] T. Ding *et al.*, "How affine arithmetic helps beat uncertainties in electrical systems," *IEEE Circuits Syst. Mag.*, vol. 15, no. 4, pp. 70–79, Fourthquarter 2015.
[42] R. Moore *et al.*, *Introduction To Interval Analysis*. Philadelphia, PA, USA: Soc. Ind. Appl. Math., 2009.
[43] J. Rohn, "Stability of interval matrices: the real eigenvalue case," *IEEE Trans. Autom. Control*, vol. 37, no. 10, pp. 1604–1605, Oct. 1992.
[44] M. Althoff *et al.*, "Analyzing reachability of linear dynamic systems with parametric uncertainties," in *Modeling, Design, and Simulation of Systems with Uncertainties*, A. Rauh and E. Auer, Eds. Berlin Heidelberg: Springer, May 2011, pp. 69–94.
[45] A. Donz and G. Frehse, "Modular, hierarchical models of control systems in spaceex," in *European Control Conf.*, 2013, pp. 4244–4251.
[46] A. Aziz *et al.*, "Efficient control state-space search," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 20, no. 2, pp. 332–336, Feb 2001.
[47] P. Hnsch *et al.*, "Reachability analysis of linear systems with stepwise constant inputs," *Electronic Notes in Theoretical Computer Science*, vol. 297, no. 0, pp. 61–74, Dec. 2013.
[48] *PLECS Manual Version 3.7*, Plexim Inc., Cambridge, MA, USA, 2015.
[49] *MATLAB Stateflow User's Guide*, Mathworks, MA, USA, 2015.
[50] S. Bak *et al.*, "HyST: A source transformation and translation tool for hybrid automaton models," in *Proc. ACM 18th Int. Conf. on Hybrid Syst.: Computation and Control*, Seattle, WA, 2015, pp. 128–133.