Computer-Aided Formal Verification of Power Electronics Circuits

Omar Ali Beg*, Luan Nguyen†, Ali Davoudi* and Taylor T. Johnson‡
*Department of Electrical Engineering
University of Texas at Arlington, Arlington, Texas 76019, USA
Email: omar.beg@mavs.uta.edu and davoudi@uta.edu
†Department of Computer Science and Engineering
University of Texas at Arlington, Arlington, Texas 76019, USA
Email: luanvnguyen@mavs.uta.edu
‡Department of Electrical Engineering and Computer Science
Vanderbilt University, Nashville, Tennessee 37235, USA
Email: taylor.johnson@vanderbilt.edu

***Abstract*—Formal verification requires extensive analysis of a given mathematical model with respect to some correctness requirements using various tools and techniques. Manually constructing models of a given device in various formats requires considerable time and efforts. Thus we automatically generate the hybrid automaton models in SpaceEx format using HyST (Hybrid Source Transformer) tool, which is a source-to-source transformation and translation tool. We then automatically translate these SpaceEx models into Mathworks Simulink Stateflow (SLSF) for analysis thus saving significant amount of time and efforts. We present various power electronics circuits benchmarks to demonstrate the efficiency and effectiveness of HyST in model-based design process. Safe and reliable operation of these circuits in safety-critical applications necessitates a rigorous modeling and verification process. In this work, we use SpaceEx reachability analysis tool for formal verification of such circuits. We have used this computer-aided modeling technique to automatically generate and translate the models and verify that the output of a given model remains within a defined stable region in steady state.**

I. Introduction

Formal verification involves constructing a mathematical model $\mathcal{M}$ with precise semantics, extensive analysis with respect to some correctness requirement $\mathcal{P}$, and verifying that $\mathcal{M} \models \mathcal{P}$ [1]. Reachability analysis has been used for formal verification of pre-defined correctness requirements for analog mixed signal circuits [2]. In this work, we use SpaceEx [3], a reachability analysis tool, for formal verification of power electronics circuits[1]. Since one needs to build the model of a given device in various formats so as to perform extenive analysis using various tools for formal verification. Manually building the models in various formats requires significant time and efforts. Therefore, we have used a new tool HyST (Hybrid Source Transformer) [4] to automatically generate the hybrid automaton models in SpaceEx compatible format. We also use HyST to automatically convert the hybrid automaton models developed in SpaceEx to MathWorks Simulink/Stateflow (SLSF) models [2]. It is a source-to-source transformation and translation tool that takes input in the SpaceEx model format, and translates it to various other formats such as HyCreate, Flow*, dReach, C2E2, Passel 2.0, and HyComp. HyST tool is being updated over the time to add support for other analysis tools. The verification and validation research community is encouraged to use HyST as this computer-automated analysis saves significant time and efforts in model-based design process.

Power electronics form the energy middle-ware and used in automobiles, industrial automation, aerospace, and defense. Power electronics devices, such as DC-DC power converters contain switching components which lead to discrete behaviors, and have passive components that exhibit continuous dynamics within each discrete event. Such devices can be modeled as hybrid automata to perform reachability analysis. A signifcant rise in the safety recalls of cars manufactured by automotive industry due to malfunction of power electronics devices has been reported. As an exmaple, about 700,000 Toyota Prius cars were recalled in year 2014 due to an error in interaction between a boost converter and its software controller [5]. Later in year 2015, more than 100,000 Toyota Prius cars were recalled due to an inverter malfunction [6]. Therefore, such mission-critical devices would require formal verification prior implementation.

In this paper, we demonstrate effectiveness of HyST tool in automatic model-based design and formal verification process using four case studies of power electronics circuits. First two being special types of DC-DC power converters called center-tapped Buck and boost converters. In the last two case studies, we use two improved models of the transformer-isolated DC-DC power converters that were earlier presented in [7], namely, flyback converter (that acts as a Buck-boost converter) and forward converter (that acts as a Buck converter). This work is continuation of a series of benchmarks for power

[1]The tool is available online from the SpaceEx website at: http://spaceex.imag.fr/.

[2]The executable models are available online from the HyST website at: http://verivital.com/hyst/.

ISBN 978-3-8007-4442-8

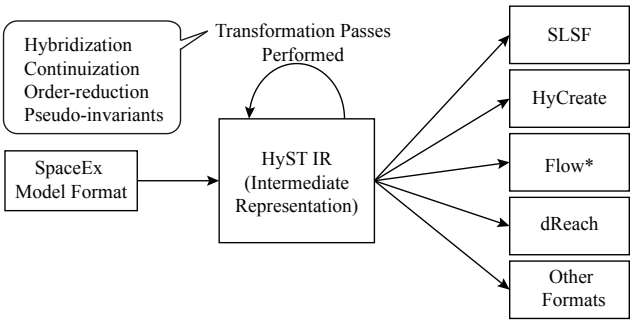

Fig. 1. Overview of the HyST conversion process.

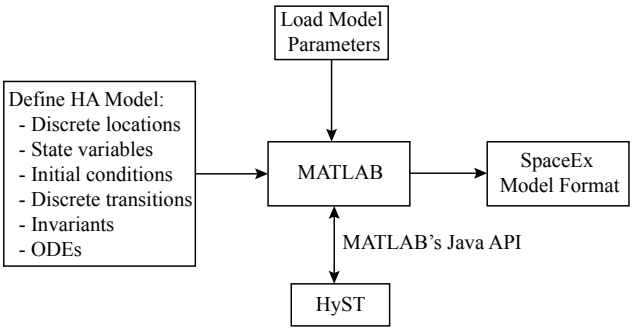

Fig. 2. Overview of automatic model generation in SpaceEx format.

electronics circuits [8]–[10] that are being developed to benefit from formal verification prior to field implementation and deployment.

II. Automatic Model Generation Using HyST

HyST is an automatic source-to-source model translation and transformation tool that takes input in SpaceEx format and generates models in SLSF, HyCreate, Flow*, dReach, C2E2, Passel, and HyComp formats [4]. The support for other reachability analysis tools will be added from time to time. HyST can be beneficial to the hybrid systems verification community in following ways:

1. The user may automatically generate a model file for numerous other tools, carry out the analysis, and choose the best suitable tool for the system under consideration.
2. The researcher involved in development of hybrid systems model checkers may quickly compare the performance of the newly developed tool with other tools.

HyST takes input in SpaceEx source format, parses it into an intermediate representation (IR), and finally prints the output source in a format specified by the user. This conversion architecture is shown in Fig. 1. IR is implemented as Java data structures to encode the hybrid automaton model components, whereas, transformation passes may be regarded as the model-to-model conversions. More details regarding HyST can be found in [4].

In this paper, we use HyST as a benchmark generator for automatic generation of hybrid automata models in SpaceEx format. Thus the user needs not to manually create the hybrid automata models through SpaceEx model editor saving considerable time and effort. We use MATLAB's API (application program interface) for Java that enables MATLAB to interact with Java programs synchronously or asynchronously. In this automatic model generation process, we need to instantiate the model components per Definition 2.1.

Definition 2.1: We define a *hybrid automaton model* by a tuple $\mathcal{M} = \langle L, X, Init, \mathcal{T}, Inv, F \rangle$, where:

- $L = \{l_1, l_2, ..., l_N\}$ is a finite set of discrete locations.
- X is a finite set of continuous state variables, such that $\forall\ x \in X\ \exists\ val(x) \in \mathbb{R}$, where $val(x)$ is a vluation of x resulted due to function mapping.
- $Init \subseteq L_0 \times X_0$ is a set of initial conditions, such that $L_0 \subseteq L$ and $X_0 \subseteq X$.
- $\mathcal{T} = \langle l_s, l_e, g, r \rangle$ is a set of feasible discrete transitions allowed among the discrete locations, where the corresponding elements of the tuple are the start location, end location, relevant guard, and the subsequent reset, respectively.
- Inv is a finite set of invariants for each discrete location.
- F is a set of ordinary differential equations (ODEs) that are defined for each location $l \in L$ over the continuous variables $x \in X$.

We implement following steps (Fig. 2) to automatically generate the hybrid automaton model using MATLAB:

1. Instantiate the matrix/string to define various components of the hybrid automaton model as per Definition 2.1.
2. Load parameter values and initialize the state variables.
3. Call the parser in HyST to represent these components into SpaceEx data structures.
4. Print into the SpaceEx model format, i.e., '.cfg', and '.xml' files.
5. Translate and print the model into the SLSF format.

III. Hybrid Automaton Model Formulation

The power electronics devices can be modeled as hybrid automata as these exhibit both the continuous and discrete behaviors due to the inherent passive elements and switches, respectivley [11]. In this section, we discuss the modeling of such circuits for use in automatic SpaceEx model generation process and translation to SLSF format. We demonstrate the effectiveness of HyST tool in model-based design process using four different types of power electronics circuits.

For the model formulation, we assume the transformer losses to be negligible. The winding at the input is called primary, whereas that towards the output is called secondary. The dynamics of such circuits depends on the operation of the MOSFET switch, i.e., being ON and OFF. We consider open loop DC-DC power converters such that the MOSFET switch is operated by a pulse generator of constant duty cycle D, over the switching time period T. The state variables are defined by the voltage across the capacitor v_C, and current through the inductor i_L.

A. Center-Tapped Buck Converter Model

It is a special type of DC-DC Buck converter, wherein, the inductor is center-tapped, i.e., a contact is made to a point

ISBN 978-3-8007-4442-8

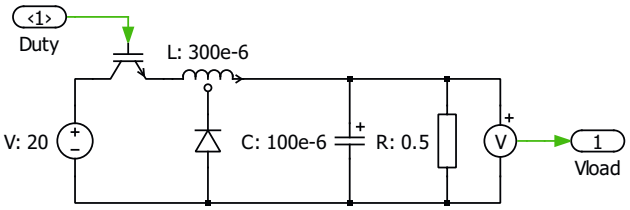

Fig. 3. Schematic diagram of center-tapped Buck converter.

halfway along the winding of an inductor. The schematic of the converter is shown in Fig. 3. Let n be the turns ratio of primary to secondary windings, n_1 be the number of turns before the center-tap, and n_2 after the center-tap. For a tapped inductor, let v_L be the overall voltage across the entire number of turns, then

$$n = \frac{v_L}{v_2} = \frac{n_1 + n_2}{n_2} = 1 + \frac{n_1}{n_2}. \tag{1}$$

The state of the MOSFET switch, i.e., being ON and OFF, results into two modes. The third mode results, when the MOSFET switch is OFF and $i_L \leq 0$.

1. Mode 1: During the switching cycle $0 < t \leq DT$, MOSFET switch is ON and diode is OFF. The input DC voltage source V_{in} supplies the primary of the inductor. In this mode, the entire inductor is charged and diodes acts as an open switch to charge the capacitor and supply the load resistance. The ODEs for i_L and v_C may be formulated using conventional Kirchoff' voltage law (KVL) and Kirchoff's current law (KCL). We use KVL on the outer loop containing L, R, and C that results in

$$\frac{di_L}{dt} = -\frac{1}{L}v_C + \frac{V_{in}}{L}, \tag{2}$$

whereas, applying KCL on the node joining L, R, and C results

$$\frac{dv_C}{dt} = \frac{1}{C}i_L - \frac{1}{RC}v_C. \tag{3}$$

The state space matrices, during the switching cycle $0 < t \leq DT$, are thus given by

$$A_1 = \begin{bmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{RC} \end{bmatrix}, B_1 = \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix}, X = \begin{bmatrix} i_L \\ v_C \end{bmatrix}, u = V_{in}. \tag{4}$$

2. Mode 2: In this mode, the MOSFET switch is OFF during the switching cycle $DT < t \leq T$, thus V_{in} is disconnected from the primary of the transformer. However, the current in the secondary (equivalent to ni_L as derived from (1)) still flows hence the diode is forward biased (in ON state). We first consider the secondary winding loop, apply KVL and use (1) to form ODE as

$$\frac{di_L}{dt} = -\frac{n}{L}v_C. \tag{5}$$

Applying KCL on the node joining L, R, and C, we obtain following ODE.

$$\frac{dv_C}{dt} = -\frac{n}{C}i_L - \frac{1}{RC}v_C. \tag{6}$$

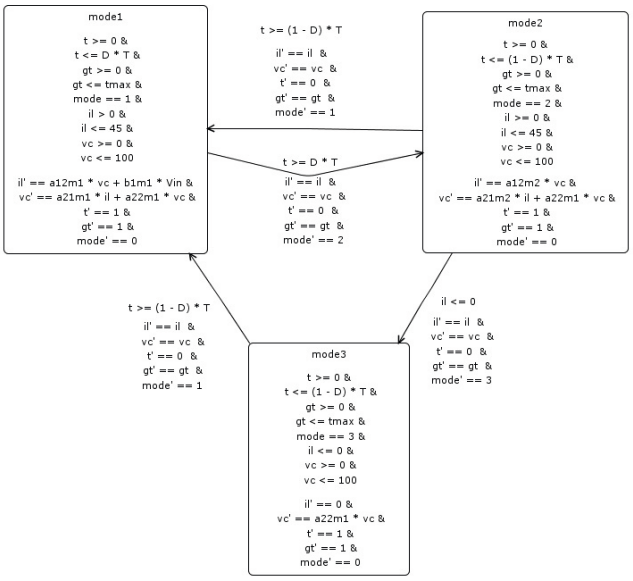

Fig. 4. Hybrid automaton model in SpaceEx format is automatically generated using HyST for center-tapped Buck converter.

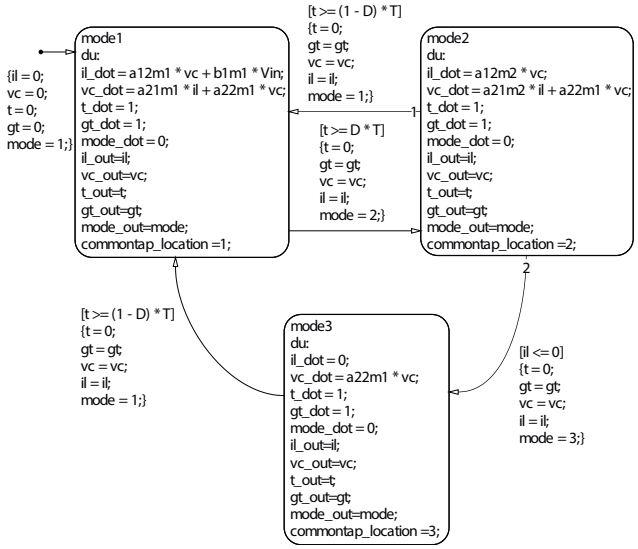

Fig. 5. SLSF model is automatically generated using HyST for center-tapped Buck converter.

The corresponding state space matrices, during the switching cycle $DT < t \leq T$, are thus given by

$$A_2 = \begin{bmatrix} 0 & -\frac{n}{L} \\ \frac{n}{C} & -\frac{1}{RC} \end{bmatrix}, B_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{7}$$

We skip the ODEs for the third mode being quite straightforward. Using HyST, we have automatically generated the models of Buck converter based on above ODEs in SpaceEx and SLSF formats as shown in Fig. 4 and Fig. 5, respectively. The component values used in the model are mentioned in Fig. 3, and adopted from [12].

B. Center-Tapped Boost Converter Model

It is a special type of DC-DC boost converter with a center-tapped inductor as shown in Fig. 6. As in the above

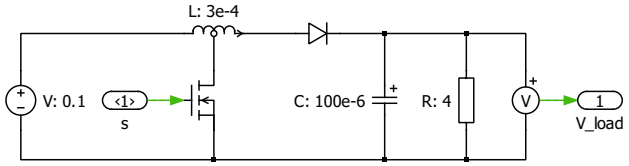

Fig. 6. Schematic diagram of the center-tapped boost converter.

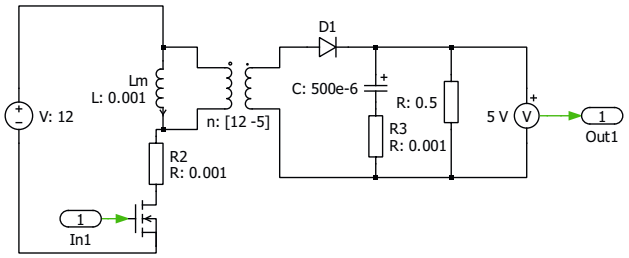

Fig. 7. Schematic diagram of flyback converter.

case, the dynamics of the circuit depends on the operation of the MOSFET switch resulting in two modes. We have automatically generated the models of center-tapped boost converter in SpaceEx and SLSF formats using HyST. Due to space limitation, we skip the formulation of ODEs and corresponding model figures. The component values used in the model are mentioned in Fig. 6.

C. Improved Model of Flyback Converter

For flyback and forward transformer-isolated DC-DC power converters, we model the transformer by L_m, a parallel magnetizing inductance, at the input side. The magnetizing current through L_m is denoted by i_{Lm}. In case of the flyback converter there are two state variables (i.e., i_{Lm} and v_C) and two modes. A simple model was presented in [7] for this type of transformer-isolated converter. This model may be improved by adding an ESR (equivalent series resistor) for the capacitor [13] as shown in Fig. 7. For space limitation, we skip the detailed model formulation. We have automatically generated SpaceEx and SLSF models of flyback converter as shown in Fig. 8 and Fig. 9, respectively. The component values used in the model are mentioned in Fig. 7, and adopted from [12].

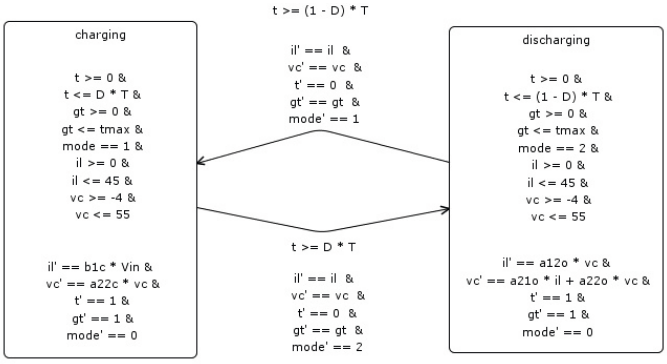

Fig. 8. Hybrid automaton model in SpaceEx format is automatically generated using HyST for flyback converter.

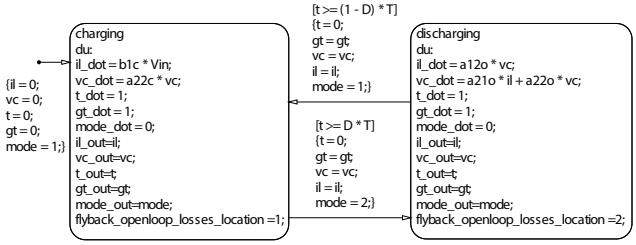

Fig. 9. SLSF model is automatically generated using HyST for flyback converter.

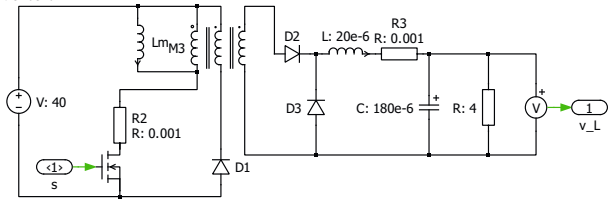

Fig. 10. Schematic diagram of forward converter.

D. Improved Model of Forward Converter

We present an improved model of the forward converter that was earlier presented in [7] to include the MOSFET switching loss (modeled by a series resistance r_{sw}) and ESR (r_L) for the inductor, as illustrated in Fig. 10. There are three state variables, i.e., i_{Lm}, i_L, and v_C. The switching modes depend on the state of the MOSFET switch as well as the fact that whether $i_L \leq 0$ and $i_{Lm} \leq 0$. This results in five different modes as shown in Fig. 11 and Fig. 12 for SpaceEx and SLSF models, respectively. Due to space limitation, we skip the ODE formulation. The component values used in the model are mentioned in Fig. 10.

E. Formal Requirements for Verification of Power Electronics Circuits

Formal verification requires that a given model of a power electronics device does not violate a predefined stability specification. We use the Lyapunov stability to define this

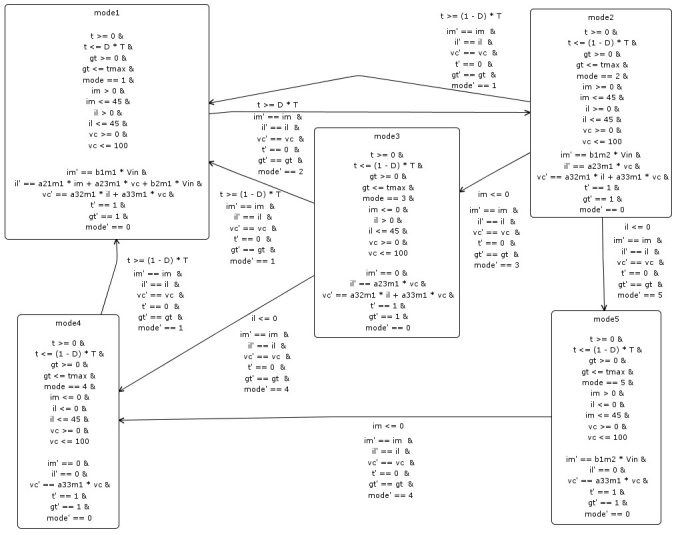

Fig. 11. Hybrid automaton model in SpaceEx format is automatically generated using HyST for forward converter.

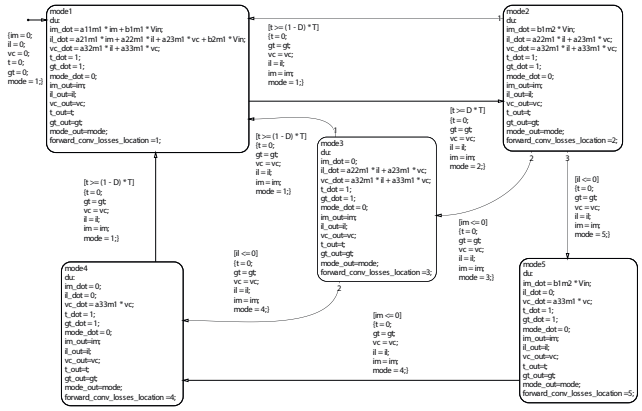

Fig. 12. SLSF model is automatically generated using HyST for forward converter.

specification, i.e., $\dot{x} = f(x(t))$ is stable if $\forall\ \epsilon > 0\ \exists\ \delta > 0$ such that if $\|x(0)\| \leq \delta \Rightarrow \|x(t)\| \leq \epsilon\ \forall\ t \geq 0$. Therefore, we may define a bounded region and verify that the output of the power electronics device eventually reaches and always remains in this stable region. This is hypothetically equivalent to requiring that both the state variables of interest, i.e., i_L and v_C attain a stable limit cycle in finite time. Accordingly, we define the stability specification for DC-DC power converters in steady state, such that i_L and v_C should attain a stable limit cycle within a finite settling time t_S.

IV. SLSF Simulations and Reachability Analysis

We have automatically generated SpaceEx models using HyST tool and analyze these in SpaceEx environment. We have also automatically translated the same SpaceEx models into SLSF format using HyST. For the flyback converter, we require that v_C and i_{Lm} should exhibit a stable limit within settling time t_S. For the center-tapped Buck, boost, and forward converters, we require that v_C and i_L should exhibit a stable limit within settling time t_S.

For center-tapped Buck, center-tapped boost, flyback, and forward converters, the SpaceEx and SLSF results for the capacitor voltage and inductor current are shown in Fig. 13, Fig. 14, Fig. 15, and Fig. 16, respectively. SLSF simulation traces are contained within the over-approximated sets of reachable states computed using SpaceEx. We also conclude that these results exhibit stable limit cycle, and that stable voltage is attained within $1.5\ ms$, $5\ ms$, $3\ ms$, and $2\ ms$ for the respective power converters.

V. Conclusion

HyST significantly reduces the time and efforts in model-based design process and formal verification. Verification and validation research community may use HyST to automatically transform the hybrid automaton models in SpaceEx format to other formats and perform reachability analysis using aforesaid model checking tools. The hybrid automaton models of power electronics circuits that we provide in this paper form part of a benchmark library. It is being developed to evaluate various reachability analysis and verification methods. This benchmark library is open to the continuous and hybrid systems verification community for testing and evaluation of their methods and tools.

Acknowledgments

The material presented in this paper is based upon work supported by the National Science Foundation (NSF) under grant numbers CNS 1464311, ECCN 1405173, and SHF 1527398, the Air Force Research Laboratory (AFRL) through contract numbers FA8750-15-1-0105 and FA8650-12-3-7255 via subcontract number WBSC 7255 SOI VU 0001, and the Air Force Office of Scientific Research (AFOSR) under contract numbers FA9550-15-1-0258 and FA9550-16-1-0246. The U.S. government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of AFRL, AFOSR, or NSF.

References

[1] R. Alur, "Formal verification of hybrid systems," in *Embedded Software (EMSOFT), 2011 Proceedings of the International Conference on*. IEEE, 2011, pp. 273–278.

[2] M. Althoff, A. Rajhans, B. H. Krogh, S. Yaldiz, X. Li, and L. Pileggi, "Formal verification of phase-locked loops using reachability analysis and continuization," *Commun. ACM*, vol. 56, no. 10, pp. 97–104, Oct. 2013. [Online]. Available: http://doi.acm.org/10.1145/2507771.2507783

[3] G. Frehse *et al.*, "Spaceex: Scalable verification of hybrid systems," in *Proc. 23rd International Conference on Computer Aided Verification (CAV)*, ser. LNCS, S. Q. Ganesh Gopalakrishnan, Ed. Springer, 2011.

[4] S. Bak, S. Bogomolov, and T. T. Johnson, "Hyst: a source transformation and translation tool for hybrid automaton models," in *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*. ACM, 2015, pp. 128–133.

[5] 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

[6] ——. (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

[7] O. A. Beg, A. Davoudi, and T. T. Johnson, "Reachability analysis of transformer-isolated dc-dc converters," in *Proceedings of Applied Verification for Continuous and Hybrid Systems Workshop (ARCH)*, Pittsburgh,PA, Apr. 2017, pp. 1–13.

[8] T. T. Johnson, Z. Hong, and A. Kapoor, "Design verification methods for switching power converters," in *Proceedings of the 3rd IEEE Power and Energy Conference at Illinois (PECI)*, Urbana, Illinois, USA, Feb. 2012, pp. 1–6.

[9] S. Hossain, S. Dhople, and T. T. Johnson, "Reachability analysis of closed-loop switching power converters," in *Proceedings of the 4th IEEE Power and Energy Conference at Illinois (PECI)*, Urbana, Illinois, USA, Feb. 2013.

[10] L. V. Nguyen and T. T. Johnson, "Benchmark: Dc-to-dc switched-mode power converters (buck converters, boost converters, and buck-boost converters)," in *Applied Verification for Continuous and Hybrid Systems Workshop (ARCH)*, Berlin, Germany, Apr. 2014.

[11] O. A. Beg, H. Abbas, T. T. Johnson, and A. Davoudi, "Model validation of pwm dc-dc converters," *IEEE Transactions on Industrial Electronics*, 2017, dOI: 10.1109/TIE.2017.2688961.

[12] *PLECS Manual Version 4.0.4*, Plexim Inc., Cambridge, MA, USA, 2016.

[13] S. K. Pandey, S. Patil, and V. S. Rajguru, "Isolated flyback converter designing modeling and suitable control strategies," in *Proceedings of the International Conference on Advances in Power Electronics and Instrumentation Engineering*, 2014.

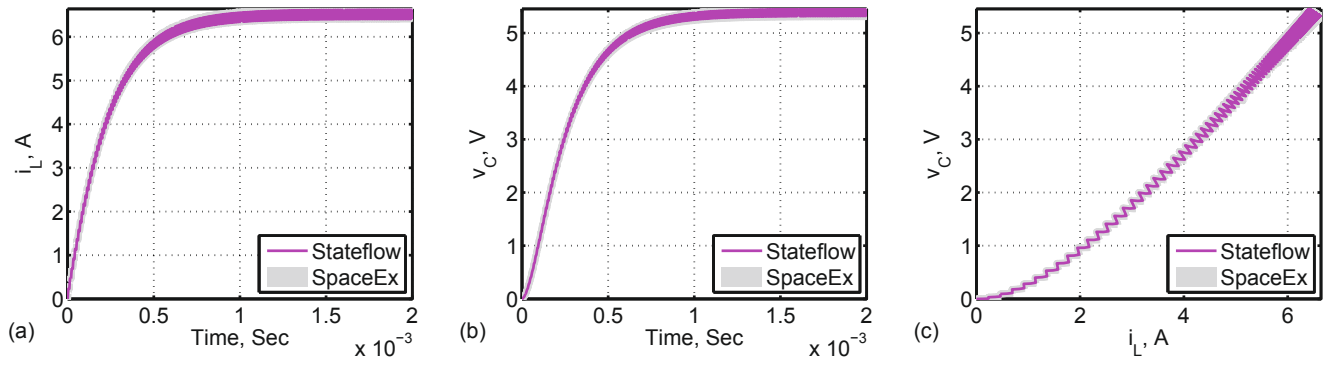

Fig. 13. Comparison of SpaceEx reach sets and SLSF trajectories for the center-tapped Buck converter showing the simulation trace containment within overapproximated sets of reachable states: (a) Inductor current vs time (b) Capacitor voltage vs time (c) Phase-plane plot of capacitor voltage and inductor current.

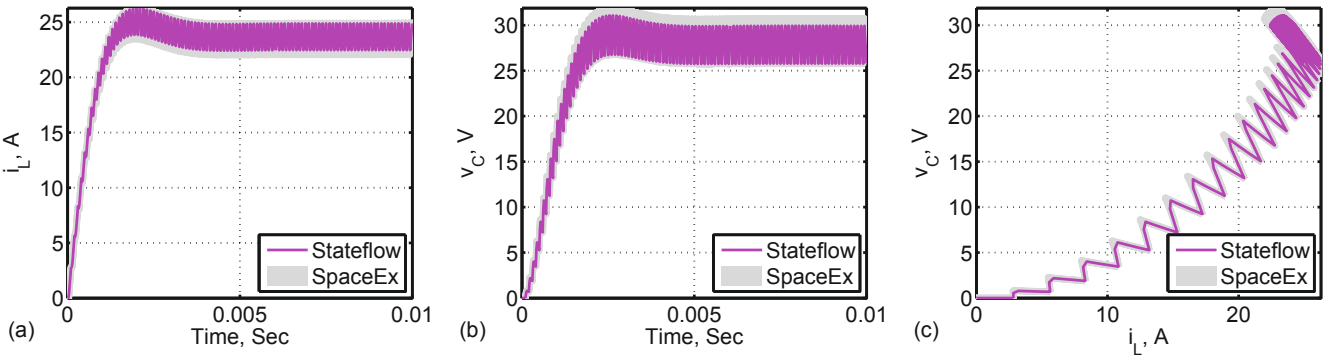

Fig. 14. Comparison of SpaceEx reach sets and SLSF trajectories for the center-tapped boost converter showing the simulation trace containment within overapproximated sets of reachable states: (a) Inductor current vs time (b) Capacitor voltage vs time (c) Phase-plane plot of capacitor voltage and inductor current.

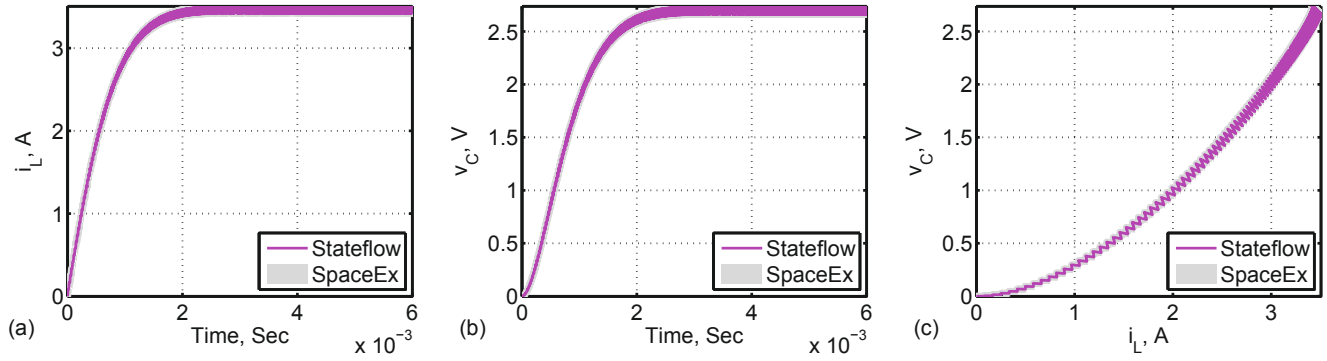

Fig. 15. Comparison of SpaceEx reach sets and SLSF trajectories for the flyback converter showing the simulation trace containment within overapproximated sets of reachable states: (a) Inductor current vs time (b) Capacitor voltage vs time (c) Phase-plane plot of capacitor voltage and inductor current.

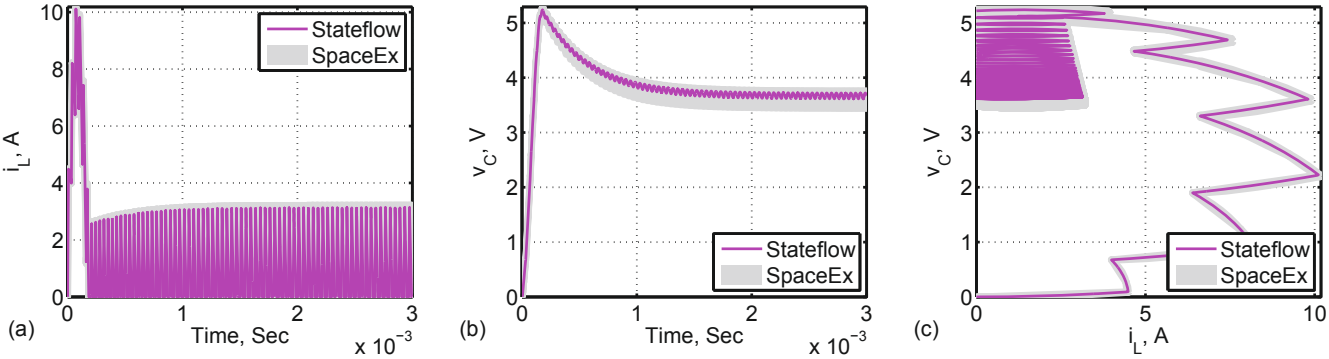

Fig. 16. Comparison of SpaceEx overapproximations and SLSF trajectories for the forward converter, showing the simulation trace containment within overapproximated sets of reachable states: (a) Inductor current vs time (b) Capacitor voltage vs time (c) Phase-plane plot of capacitor voltage and inductor current.

ISBN 978-3-8007-4442-8