Converter Analysis Using Discrete Time State-Space Modeling

Jared A. Baxter and Daniel J. Costinett
The Bredesen Center
The University of Tennessee, Knoxville
Knoxville, Tennessee, USA
Email: jbaxte24@vols.utk.edu, daniel.costinett@utk.edu

***Abstract*—Modeling plays a vital role in the design of advanced power converters. Commonly, modeling is completed using either dedicated hand analysis, which must be completed individually for each topology, or time-stepping circuit simulations, which are insufficiently rapid for broad analysis considering a wide range of potential designs or operating points. Discrete time state-space modeling of switching converters has shown merits in rapid analysis and generality to arbitrary circuit topologies but is hampered by difficulty incorporating nonlinear elements. In this work, we investigate methods for the incorporation of nonlinear elements into a generalized discrete time state-space modeling framework and showcase the utility of the approach for use in the converter design process.**

I. Introduction

The design process for power electronics is a complex interplay of approximation, designer intuition, historical trends, and prototype/revision cycles. A simplified V-model diagram of the converter design process is shown in Fig. 1. On the left side of the "V", the design process is carried out, transitioning from an application-specific design specification to a complete design suitable for prototyping. On the right side, the design is tested and validated, often with multiple inner-cycles of prototyping and revision. As an example, [1] details a limited design process in which multiple candidate designs are assessed using general models, then gradually refined with more accurate, but complex, modeling efforts and empirical measurements to progress toward a final implementation.

In the design stage, analysis of the converter is predominately focused on periodic steady-state operation [2]. Ongoing developments in multi-domain simulation, combined with multi-objective optimization techniques, seek to automate much of the detailed design process [3]–[5]. These developments will eventually replace hardware prototype/revision cycles with "virtual prototyping" in simulation, accelerate the design process, and reduce the time-to-market for new technologies [6]. With increasing fidelity, developments in a wide variety of software simulation tools portend replacing a significant portion of the need for experimental validation in the long-term.

However, to achieve this level of fidelity, increasingly complex and computationally-intensive models are required. In applications requiring high performance power conversion, the scope of candidate-optimal design approaches is too broad to employ high-fidelity, model-based optimization. Even with continued advancement in the computational power available for the design process, a distinction remains between the broad-scale design, which translates a specification into a conceptual paper design, and the detailed design process, which translates this design into a physical implementation.

For certain applications, it may be desirable to consider topologies within pulse width modulation (PWM), soft-transition, resonant, quasi/multi-resonant [7], switched capacitor [8], resonant switched capacitor [9], hybrid [10], or composite [11] classifications. Considering multiple topologies in each classification, over a wide range of operating points and without constraint on electrical parameters or device implementation, the development of a schematic level candidate-optimal design for a single application becomes a substantial challenge even prior to detailed design stages.

This work focuses on the broad-scale design effort, including tasks such as circuit topology and component selection, component (electrical) sizing, and modulation pattern selection. Within this task, simplified models are used, either in simulation or derived from dedicated mathematical analysis. These simplified models allow a broader scope of designs to be considered within a feasible timeframe, narrowing to a single or small number of candidate designs suitable for detailed design and prototyping.

Individually-derived mathematical models of converters allow computationally efficient evaluation of a converter model;

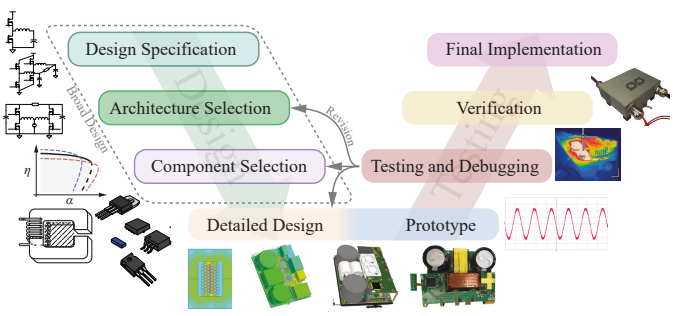

Fig. 1. Example model of the power converter design process.

978-1-7281-1842-0/19/$31.00 ©2019 IEEE

however, they are time-consuming to produce. The varying analysis methods and approximations employed in these individually-derived models, particularly for topologies of different classifications and operating modes, make the models unsuitable for broad comparative analysis. State-space models of switching circuits [12]–[16], as commonly employed in discrete time modeling, are suitably generic with respect to topology but lack closed-form solutions when nonlinear elements are included in the circuit.

Circuit simulation tools, e.g. LTspice or PLECS, use direct or modified [2] time-stepping and numerical integration to incorporate arbitrary nonlinear elements through local linearization at each timestep. This results in generic simulators which can accommodate arbitrary topology and device model complexity. However, direct time-domain numerical integration may result in slow convergence to steady-state when the converter exhibits dynamics well below the switching frequency, as is true in the vast majority of applications of power electronics.

In light of these drawbacks, this work examines the feasibility of employing state-space modeling techniques for switched, linear circuits to the broad design process of power electronics. The theory and modeling framework are reviewed in Section II, expansion to include select nonlinearities is discussed in Section III, and experimental results used to validate the modeling are given in Section IV. Section V concludes the paper.

II. State-Space Modeling

Discrete time state-space modeling [13]–[16] has long been considered a suitable modeling framework for generalized analysis of switching circuits in both steady-state and dynamic conditions. The framework natively accommodates periodically switched circuits, given each switching configuration in the circuit can be represented as a linear equivalent circuit. Under this assumption, the framework builds upon the state-space representation of the linear equivalent circuit during each switching interval

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}_i\boldsymbol{x}(t) + \boldsymbol{B}_i\boldsymbol{u}(t) \tag{1}$$

$$\boldsymbol{y}(t) = \boldsymbol{C}_i\boldsymbol{x}(t) + \boldsymbol{D}_i\boldsymbol{u}(t) \tag{2}$$

where the state vector, $\boldsymbol{x}$, contains the n_s converter states and $\boldsymbol{u}$ contains the n_i independent inputs to the system. The matrices $\boldsymbol{A}_i \in \mathbb{R}^{n_s \times n_s}$ and $\boldsymbol{B}_i \in \mathbb{R}^{n_s \times n_i}$ describe the circuit topology. The index $i \in [1...m]$ iterates over different linear equivalent circuits that model the topological changes due to switching.

For any single time interval i, over which the linear equivalent circuit remains a valid approximation to the circuit behavior and approximating all independent inputs as constant during the interval, the solution to (1) is

$$\boldsymbol{x}(t) = \mathrm{e}^{\boldsymbol{A}_i t}\boldsymbol{x}(0) + \int_0^t \mathrm{e}^{\boldsymbol{A}_i(t-\tau)}\boldsymbol{B}_i\boldsymbol{u}(\tau)d\tau. \tag{3}$$

If $\boldsymbol{u}(t)$ is well-approximated as constant over the entirety of a single switching interval, (3) becomes

$$\boldsymbol{x}(t) = \mathrm{e}^{\boldsymbol{A}_i t}\boldsymbol{x}(0) + \boldsymbol{A}_i^{-1}[\mathrm{e}^{\boldsymbol{A}_i t} - \mathbf{I}]\boldsymbol{B}_i\boldsymbol{u}_i. \tag{4}$$

By repetitively applying (4) over the m linear equivalent switching subcircuits in one period of steady-state operation, the final states at the end of a switching period are

$$\boldsymbol{x}(T_s) = \left(\prod_{i=1}^{m} \mathrm{e}^{\boldsymbol{A}_i t_i}\right)\boldsymbol{x}(0)\times \sum_{i=1}^{m}\left[\left(\prod_{k=i+1}^{m} \mathrm{e}^{\boldsymbol{A}_k t_k}\right)\boldsymbol{A}_i^{-1}\left(\mathrm{e}^{\boldsymbol{A}_i t_i} - \mathbf{I}\right)\boldsymbol{B}_i\boldsymbol{u}_i\right]. \tag{5}$$

Setting $\boldsymbol{x}(T_s) = \boldsymbol{x}(0) = \boldsymbol{X}_{ss}$, the periodic steady-state solution to an arbitrary converter is

$$\boldsymbol{X}_{ss} = \left[\mathbf{I} - \prod_{i=1}^{m} \mathrm{e}^{\boldsymbol{A}_i t_i}\right]^{-1}\times \sum_{i=1}^{m}\left[\left(\prod_{k=i+1}^{m} \mathrm{e}^{\boldsymbol{A}_k t_k}\right)\boldsymbol{A}_i^{-1}[\mathrm{e}^{\boldsymbol{A}_i t_i} - \mathbf{I}]\boldsymbol{B}_i\boldsymbol{u}_i\right], \tag{6}$$

where $\boldsymbol{X}_{ss}$ describes the instantaneous states at the beginning of the period. The only approximation made in the derivation of (6) was that $\boldsymbol{u}(t) \approx \boldsymbol{u}_i$ within one switching interval. This approximation is valid for the vast majority of power electronics applications or can be enforced by expanding the scope of the modeled system.

Once the steady-state solution is solved, the value of the states $\boldsymbol{x}(t)$ at any instant within the period can be generated using (4), and the value of any other signal can be obtained using the result and (2), along with appropriate selection of $\boldsymbol{C}_i$ and $\boldsymbol{D}_i$. Coupled with methods to automatically parse a circuit description into state matrices [17], this approach can rapidly evaluate the steady-state electrical performance of an arbitrary switching circuit without restricting classification or requiring ideal assumptions.

Though relatively approximation-free, there are two key limitations to the application of (6) for generalized switching circuit analysis:

1) For direct computation, $\boldsymbol{A}_i$ must be invertible, and
2) Each switching interval must be well-approximated as a linear equivalent circuit over a known duration t_i.

Solutions and implications of these limitations are discussed in the following sections.

A. Singular $\boldsymbol{A}_i$

The first point is solved using the augmented state-vector method of [14]. Although (4) contains $\boldsymbol{A}_i^{-1}$, expansion of the power series of $(\mathrm{e}^{\boldsymbol{A}_i t} - \mathbf{I})$ eliminates it [18], though neither evaluation of the power series nor direct evaluation of the

convolution integral in (3) are computationally efficient [19]. Instead, [20] shows that for a square matrix

$$\hat{\boldsymbol{A}}_i = \left[\begin{array}{c|c} \boldsymbol{A}_i & \boldsymbol{B}_i \\ \hline \boldsymbol{0} & \boldsymbol{0} \end{array}\right], \tag{7}$$

the matrix exponential

$$\mathrm{e}^{\hat{\boldsymbol{A}}_i t} = \left[\begin{array}{c|c} \mathrm{e}^{\boldsymbol{A}_i t} & \int_0^t \mathrm{e}^{\boldsymbol{A}_i (t-\tau)} \boldsymbol{B}_i d\tau \\ \hline \boldsymbol{0} & \mathbf{I} \end{array}\right] \tag{8}$$

inherently computes both terms in (3). Because the matrix exponential is well-defined for any square matrix and because modern numerical computation tools employ efficient methods for the computation of the matrix exponential, (7) presents an efficient method to calculate the necessary terms in the event of a singular matrix $\boldsymbol{A}_i^{-1}$. This approach is further leveraged in the augmented state-vector method [14], [16], which notes that

$$\frac{d}{dt}\hat{\boldsymbol{x}}(t) = \hat{\boldsymbol{A}}_i \hat{\boldsymbol{x}}(t), \tag{9}$$

has solution

$$\hat{\boldsymbol{x}}(t) = \left[\begin{array}{c} \boldsymbol{x}(t) \\ \hline \boldsymbol{u}_i \end{array}\right] = \mathrm{e}^{\hat{\boldsymbol{A}}_i} \hat{\boldsymbol{x}}(0). \tag{10}$$

Conceptually, this method can be understood as reforming the system description such that the (assumed constant) inputs in each switching interval are instead considered states. Creating a new state-space matrix $\hat{\boldsymbol{A}}$ enforces that the time-derivative of each of the states corresponding to $\boldsymbol{u}_i$ is zero.

Equation (10) can be applied in the same manner as in (6) to solve the steady-state solution directly using augmented matrices and finding the solution to [14]

$$\hat{\boldsymbol{X}}_{ss} \left(\mathbf{I} - \prod_{i=1}^{m} \mathrm{e}^{\hat{\boldsymbol{A}}_i t_i} \right) = 0, \tag{11}$$

or (8) can be used to compute solely $\int_0^t \mathrm{e}^{\boldsymbol{A}_i (t-\tau)} \boldsymbol{B}_i$, which is useful in cases where many evaluations of the same system with varying inputs $\boldsymbol{u}_i$ are needed.

A similar expansion of the state matrices, which sets the derivatives of the new states equal to the instantaneous value of the original states, can be used to directly compute the integral of each state over one period, thereby finding the average value of each state, as detailed in [14]. Because the integral of the states is generally not periodic, this should be applied using (5) after finding the steady-state solution.

B. Linear Circuit Requirement

Because this modeling approach is built upon the state-space description of (1), it is valid only when the system can be modeled within any arbitrarily small interval i, over which the circuit is adequately modeled as a linear equivalent. Though this may require a substantial increase in the number of switching intervals, m, there is no fundamental limit to the capability of the model to adapt to arbitrary nonlinear transfer characteristics of circuit elements or their loss models. This approach is analogous to the small-signal linearization employed in current commercial modified nodal analysis [21] and state-space time-stepping [22] simulators, which often require iteration and variable time-stepping to ensure that linearized models remain sufficiently accurate in numerical approximations of nonlinear circuit elements.

There is, however, a significant practical limit in the state-space direct steady-state solution approach discussed previously, as the time-duration of each equivalent circuit t_i must be known a priori. While this is well-suited to controlled switching actions such as the gate-controlled turn-on of a transistor, it is poorly suited to state-based nonlinear transfer characteristics. To retain the benefit of employing a direct steady-state solution as in (4), the number of intervals for which t_i is not directly controlled should be limited. Otherwise, a time-stepping approach is more suitable to track the nonlinear characteristics present in the circuit.

In any event, the incorporation of nonlinear circuit behaviors will increase the modeling complexity and, therefore, analysis time. As discussed in relation to Fig. 1, full modeling of all circuit nonlinearities using physics-based modeling techniques is often reserved for a small number of highly detailed designs due to the complexity of both creating and solving such models. In the initial broad design stage, particularly when dedicated mathematical analysis is employed, circuit operation is often solved using ideal models with any relevant loss mechanisms approximated from the ideal waveforms. Relative to this approach, even limiting the state-space model to only include the linear loss mechanisms when solving circuit operation represents an increase in model fidelity. Often, nonlinear loss mechanisms such as core loss or switching loss mechanisms (beyond C_{oss}) can be approximated from the solved waveforms retroactively using empirical data.

This approach employs an approximation of high efficiency in the design stage, i.e. it assumes that all nonlinear loss mechanisms are sufficiently small relative to the output power such that they have minimal impact on the converter waveforms. One primary exception to this assumption is the nonlinear state-controlled switching of diodes, including transistor body diodes. Even when modeled as a two-state switch, the switching times of a diode are not externally controlled and can substantially impact converter waveforms even in high efficiency implementations. As such, the following section examines approaches to incorporate this mechanism into the solution of (4) through iterative means, while attempting to maintain the benefits of solving steady-state operation directly.

III. Incorporation of Uncontrolled Switching

As proposed generally in [13], [15], this work treats the incorporation of nonlinear diode switching as a sub-problem pertaining to the selection of time t_i. With diodes modeled as

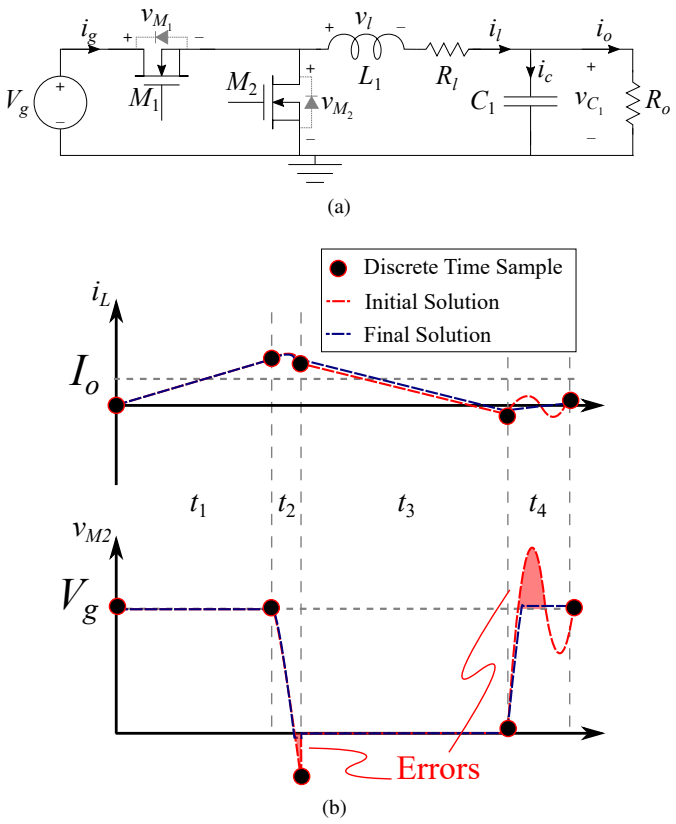

Fig. 2. Example circuit (a) and waveforms (b) showing violations of diode operation. Body diodes, which are eliminated from the state matrix description, are shown grayed out.

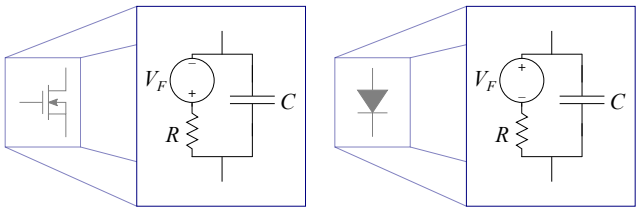

Fig. 3. FET and diode equivalent circuits in state-space representation. The transition between linear equivalent circuits is accomplished by adjusting R and V_F based on the current conduction state of the device.

linear equivalent circuits in both the on- and off-state, (4) is solved subject to constraints given by

$$\boldsymbol{y_D}(t) = \boldsymbol{C_D}\boldsymbol{x}(t) + \boldsymbol{D_D}\boldsymbol{u}(t) \leq \boldsymbol{V_F}, \quad \forall t \in [0, \Sigma t_i], \quad (12)$$

where, for the i^{th} time interval, the vector $\boldsymbol{y}_D(t)$ contains all diode voltages within the circuit, and $\boldsymbol{V_F}$ is a vector containing the diode forward voltages.

Two possible violations of (12) are shown by an example converter and waveforms of its solution to (6) in Fig. 2. Two transistors in a synchronous buck converter are modeled as ideal switches, thus eliminating their body diodes from the state equations. For an initial set of times t_i, (6) gives a steady-state solution which allows v_{M_2} to resonate both above V_g and below zero volts during the two deadtimes, resulting in two time intervals where violations of (12) are present. Further,

TABLE I
INITIAL SOLUTION CONDUCTING DEVICES

Time Interval	t_1	t_2	t_3	t_4
Circuit Model	$\boldsymbol{A}_1$	$\boldsymbol{A}_2$	$\boldsymbol{A}_3$	$\boldsymbol{A}_4$
Conducting Devices	M_1	—	M_2	—

TABLE II
FINAL SOLUTION CONDUCTING DEVICES

Time Interval	t_1	t_2		t_3	t_4	
Circuit Model	$\boldsymbol{A}_1$	$\boldsymbol{A}_2$	$\boldsymbol{A}_2^*$	$\boldsymbol{A}_3$	$\boldsymbol{A}_4$	$\boldsymbol{A}_4^*$
Conducting Devices	M_1	—	D_2	M_2	—	D_1

the violation in t_4 is not detectable solely from examining the converter at the discrete switching times. The initial solution assumes that the converter periodic behavior is as shown in Table I, i.e. the switching deadtimes in the synchronous buck converter are set ideally such that no body diode conduction occurs. The actual steady-state behavior at the operating point shown in Fig. 2 is described in Table II where $\boldsymbol{A}_2^*$ and $\boldsymbol{A}_4^*$ are similar to $\boldsymbol{A}_2$ and $\boldsymbol{A}_4$ but exhibit loss mechanisms of body diode conduction rather than MOSFET conduction. Both t_2 and t_4 remain the same total duration, defined by the controlled switching actions, but are now split across two different intervals.

To obtain the final solution of Fig. 2, the times t_i are updated to values such that the body diodes conduct during the deadtimes once their forward voltage is exceeded. Depending on the modeling of the diodes, the resulting diode conduction can be modeled as ideal or with linear loss mechanisms different from transistor channel conduction. To find the correct t_i, the rapid solution to (6) is leveraged to take a numerical partial derivative of the steady-state solution under perturbed timing t_i and evaluate the constraint on the perturbed solution, which informs the adjustment of t_i to reduce the violation. For violations with interval dynamics faster than the time duration, such as in t_4 in Fig. 2, small perturbations will not yield accurate insight into the necessary adjustments to timing intervals.

A. Algorithm

A circuit parser is implemented based on the graph theory presented in [17]. The parser algorithm automatically produces one set of state-space matrices, with each switch parameterized by R and V_F as shown in Fig. 3. These values can then be adjusted for each switch to yield the appropriate $\boldsymbol{A}_i$, $\boldsymbol{B}_i$, $\boldsymbol{C}_i$, and $\boldsymbol{D}_i$ matrices for any switching state. The $\boldsymbol{C_D}$ and $\boldsymbol{D_D}$ matrices, representing diode conduction constraints, are generated in the same manner. A SPICE netlist is used as the input circuit description to the parser.

Based on an initial guess of the switching state order and interval times, (6) is used to solve the steady-state initial condition, and (4) is used repetitively to find the state vector

at each switching action, corresponding to the discrete time sample points shown in Fig. 2.

The states at each discrete time sample are then tested against diode conduction constraints using (12) through the algorithm presented in Fig. 4. If a violation occurs, either the appropriate matrices must be inserted into the switching pattern, or the time intervals must be adjusted to account for this error. To make this determination, t_{i-1} or t_{i+1} , depending on whether the beginning or ending discrete time sample was found to be in violation, is evaluated. If the next or previous time interval has the correct devices conducting to alleviate the error, the correct state matrix order is already in place and only the duration of the intervals needs to be adjusted. A linear projection adjustment can be made to adjust t_i,

$$\Delta t_i = \frac{\big(V_F - y_D(t_i)\big)\Delta t}{y_D(t_i + \Delta t) - y_D(t_i)}. \tag{13}$$

This requires solving for the steady-state with a new perturbed time interval to calculate $y_D(t_i + \Delta t)$. If constant switching period modulation is employed, $-\Delta t_i$ must be applied to the next or previous time interval.

If the correct state matrices are not already present in the switching pattern, then a new interval must be inserted. The new state matrices and time intervals are set to be added to the switching pattern once the current iteration completes, and the algorithm proceeds to the next step without re-solving the steady-state solution, thereby preventing oscillations due to multiple, simultaneous diode switching events, which may be present, for example, in symmetric full bridge circuits.

B. Simulation Comparison

Three test converters are used to compare the aforementioned algorithm and parser to PLECS and LTSpice in terms of solving time and solution accuracy. The dual active bridge (DAB) converter of Fig. 5(a), Hybrid Dickson Switched Capacitor Converter (HDSC) of Fig. 5(b), and a synchronous buck converter shown in Fig. 2 are used as test cases. In each case, the transistors are modeled including only constant values for r_{on}, C_{oss}, and a body diode.

Table III, shows the time needed for each simulation method to reach a steady-state solution along with the state variables and time intervals of each converter. The proposed method shows both the total time to reach a steady-state solution and the solver time. The solver time excludes the time needed for the parser to produce the state matrices of the converter from a SPICE netlist file. Parsing is only done once per topology to achieve a symbolic state-space representation of the circuit. The steady-state solution and the incorporation of uncontrolled switching are achieved during the solve time. Thus, the repetitive time that is consumed for a topology optimization is better approximated by the solve time presented.

The relative differences in simulation time are explained graphically in Fig. 6. In all figures, t is simulated time while "Time" represents real-world duration of the simulation. In Fig. 6(a), direct time-stepping simulation is used, wherein a variable timestep is used to ensure fidelity of the results at each

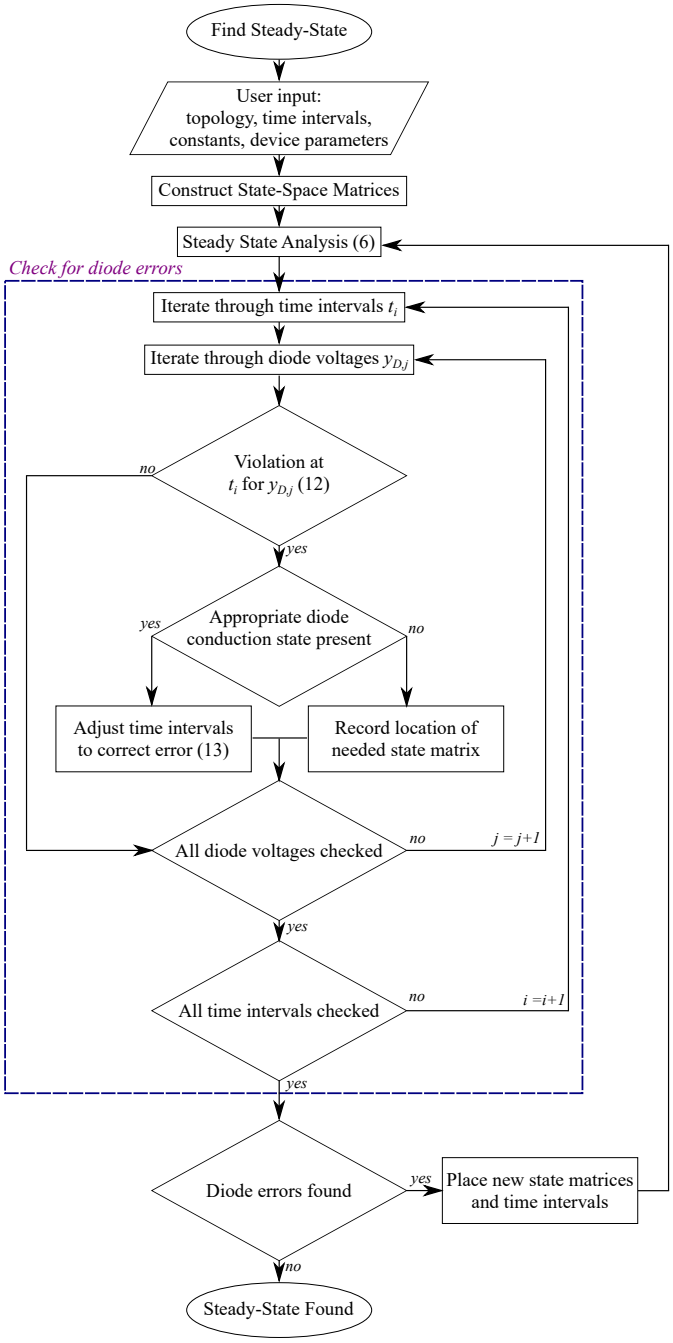

Fig. 4. Flow diagram of the algorithm used to find steady-state of a converter while accounting for uncontrolled diode conduction.

step. This results in a variable number of time-steps per period, $N_{s,p}$. Because a full transient simulation is used, the total simulation must cover a number of switching periods great enough to ride through the low-frequency dynamics of the circuit being simulated, which are typically much slower than the switching dynamics. Table III shows direct time-stepping simulation performed best under fewer time intervals and fewer state variables in LTspice. The low frequency dynamics of the HDSC, however, severely slowed the PLECS transient

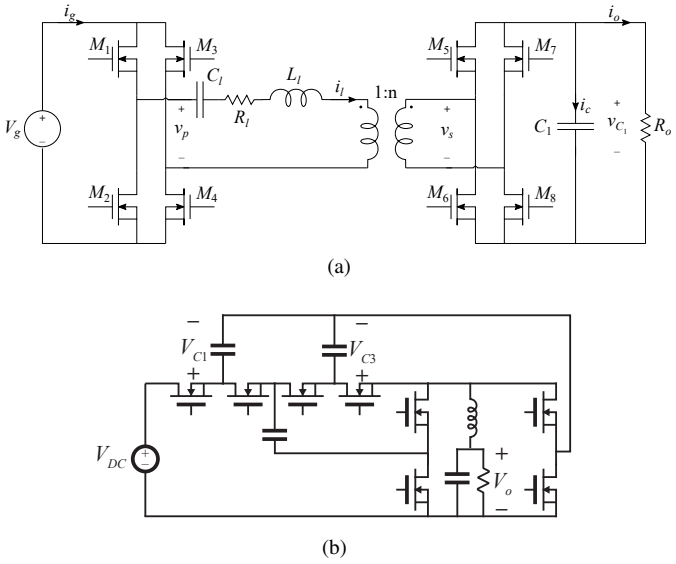

Fig. 5. DAB (a) and HDSC (b) converter circuits used for validation.

TABLE III
STEADY-STATE SOLVE TIME OF SIMULATION METHODS

	Convergence Time (sec)[1]		
Simulation Method	DAB	HDSC	Buck
PLECS (Transient)	4.39	173.61	1.98
PLECS (Steady-State)	-	-	2.48
LTspice	5.50	4.60	0.27
Proposed Method[2]	5.54 (1.67)	3.14 (0.62)	0.66 (0.15)
		Amount	
Parameter	DAB	HDSC	Buck
State Variables	11	13	4
Time Intervals	8	4	4

[1] All simulations performed on a desktop computer with an Intel(R) Core(TM) i7-4790 CPU @ 3.60 GHz processor, 8 GB of RAM, and a 64 bit operating system.
[2] Listed as "Total (Solver Only)"

solver.

In Fig. 6(b), the steady-state solver in PLECS [22], based on [2], uses a similar approach over a single switching period but uses quasi-Newton iteration between adjacent periods to attempt to accelerate convergence to steady-state and divorce the simulation convergence time from the low-frequency circuit dynamics. The approach solves the error x_e between the initial and final states in one period, then uses the quasi-Newton iteration to provide a correction $x_{0,c}$ to the initial states used in the next period. From the simulation results, a large number of states and time intervals proved to be limiting factors of this method. The DAB converter would cause the program to stall during its calculations, and the solver would iterate hundreds of times while attempting to solve the HDSC.

In this study, shown in Fig. 6(c), each iteration of the algorithm produces a steady-state solution. However, when additional nonlinearities such as uncontrolled switching are included, multiple iterations may be required to produce a valid solution, subject to constraints outlined in (12). As a result, the computation time is decoupled from the time-domain circuit dynamics completely and instead relates to the rate of convergence in the unconstrained solution space. The proposed method needed seven iterations of the diode error algorithm in Fig. 4 to reach steady-state; however, the HDSC had no diode conduction errors and thus a much faster solve time.

Conceptually, and as demonstrated in Table III, the approach in Fig. 6(c) can achieve accelerated convergence in situations where sufficient modeling fidelity can be obtained with a relatively small number of nonlinearities modeled. Additional nonlinear effects that do not have a significant impact on converter waveforms can be calculated afterward using the complete steady-state waveforms generated by the model.

IV. EXPERIMENTAL VERIFICATION

The modeling method was compared to experimental results using prototype DAB and HDSC converters. The DAB converter is used as an example converter in which multiple resonant time intervals are present, but each has a duration shorter than one quarter of the resonant period. The HDSC of Fig. 8 is an example of a switched capacitor converter, emphasizing the generality of the proposed method. Both converters are modeled through the same analytical framework without any dedicated analysis for either. The two converters use EPC GaNFETs; the DAB has a 48V-to-1.2V conversion ratio, employing a 36:1 transformer while the HDSC converts 24V to 5V with three flying capacitors. The values of r_{on}, C_{oss}, and reverse conduction voltage of the GaNFETs, as reported on their datasheets, are used in the switch models. Additional resistances of the magnetic components are included.

TABLE IV
EFFICIENCY COMPARISON

Method	DAB Efficiency
Experimental	91.78%
Proposed Method	95.62%
LTspice	95.92%
PLECS	95.01%

Fig. 7(c) compares the measured converter waveforms to the model results of the proposed method. The full waveforms are reconstructed from the discrete time solution for clarity. Table IV shows an efficiency comparison of the proposed method and experimental results. As illustrated in the table, all simulations achieved nearly the same efficiency, within 1%. The experimental results show a reduced efficiency due to the absence of nonlinear core loss present in the converter. The additional nonlinear core loss was calculated afterword to be approximately 350 mW. For the 8.5 W experimental converter, the core loss leads to the approximate 4% discrepancy between simulation and experimental results.

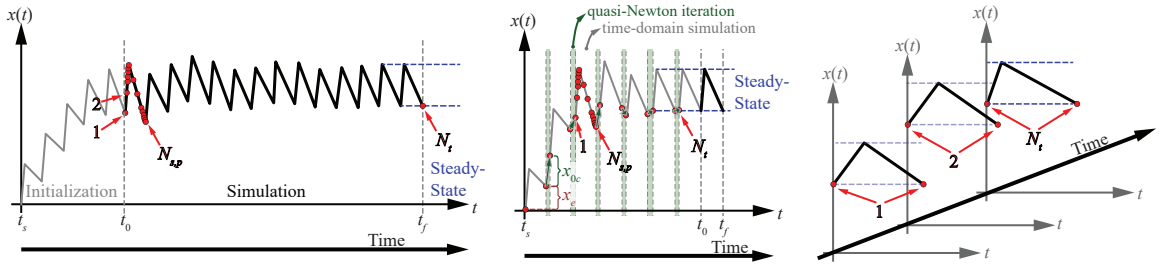

Fig. 6. Graphical comparison of simulation approaches in SPICE (a), PLECS (b), and the proposed direct steady-state method (c).

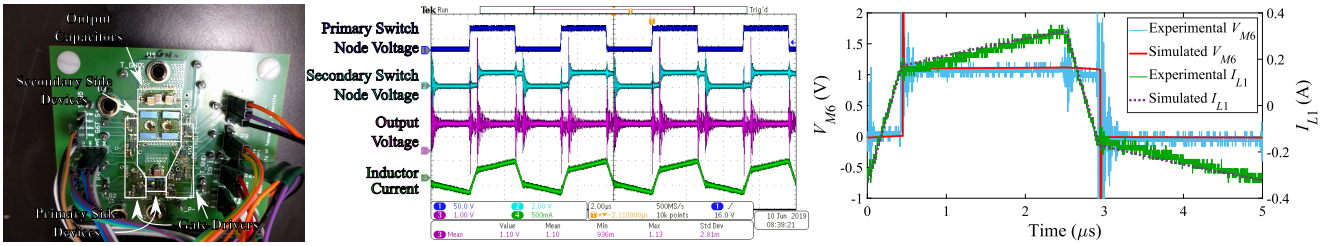

Fig. 7. DAB converter prototype (a), experimental waveforms (b), and modeled waveforms (c).

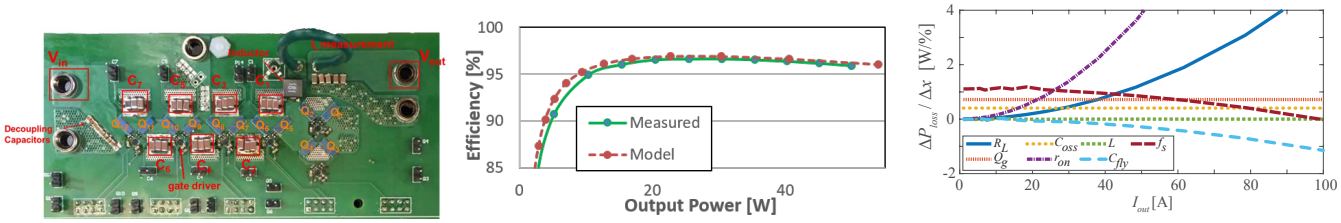

Fig. 8. HDSC converter prototype (a), measured and modeled efficiency (b), and loss sensitivity to parameter variation (c).

Fig. 8(b) shows a comparison between measured and modeled efficiency for the HDSC. Efficiency is evaluated solely by computing the average input and output voltage/current from the steady-state solution, containing only the modeled linear loss mechanisms.

In order to assess the HDSC design for purposes of optimization, the numerical partial derivative of the converter power loss with respect to various parameters is computed using the method detailed in this work. Fig. 8(c) was generated by solving the steady-state solution of the converter over 280 operating points and parameter variations, which took 14.8 seconds on a standard desktop computer. The resulting plots show the relative importance of altering each parameter to increase efficiency. At light load, f_s should be decreased (from 500 kHz) to reduce switching losses, but at heavy load, f_s should increase to reduce charge sharing losses. At intermediate and heavy loads, reducing r_{on} is the top priority in this prototype.

The results of Fig. 8(c) are intuitive, however, they highlight the merit of the proposed modeling technique in the broad design stage of converter development. The speed and generality of the approach allow rapid evaluation for the purpose of design analysis, insight, or optimization.

V. Conclusion

This work investigates discrete time state-space modeling for power electronics design. The assessment framework is developed to rapidly compute the steady-state operation of a converter without the need for dedicated analysis. The approach is general to different classifications of switching converters and scalable to varying levels of model fidelity. Techniques for the inclusion of nonlinearities, such as passive diode commutation, are developed. A simulation comparison is included and shows comparably faster steady-state convergence of test circuits. Experimental verification is performed to confirm model accuracy and simulation results.

Acknowledgment

The authors would like to thank Mr. Andrew Foote and Ms. Taylor Short for their assistance developing the DAB converter used for experimental verification.

This material is based upon work supported by the National Science Foundation under Grant Number 1751878.

This work made use of Engineering Research Center shared facilities supported by the Engineering Research Center Program of the National Science Foundation and the Department of Energy under NSF Award Number EEC-1041877 and the CURENT Industry Partnership Program.

REFERENCES

[1] C. Zhao, B. Trento, L. Jiang, E. A. Jones, B. Liu, Z. Zhang, D. Costinett, F. F. Wang, L. M. Tolbert, J. F. Jansen, R. Kress, and R. Langley, "Design and implementation of a GaN-based, 100-kHz, 102-W/in^3 single-phase inverter," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 4, no. 3, pp. 824–840, Sep. 2016.

[2] D. Maksimovic, "Automated steady-state analysis of switching power converters using a general-purpose simulation tool," in *Record 28th Annual IEEE Power Electronics Specialists Conference*, vol. 2, 1997, pp. 1352–1358.

[3] T. M. Evans, Q. Le, S. Mukherjee, I. A. Razi, T. Vrotsos, Y. Peng, and A. H. Mantooth, "Powersynth: A power module layout generation tool," *IEEE Transactions on Power Electronics*, pp. 1–17, 2018.

[4] T. Wu, B. Ozpineci, M. Chinthavali, W. Zhiqiang, S. Debnath, and S. Campbell, "Design and optimization of 3d printed air-cooled heat sinks based on genetic algorithms," in *2017 IEEE Transportation Electrification Conference and Expo (ITEC)*, 2017, pp. 650–655.

[5] A. Bindra and A. Mantooth, "Modern tool limitations in design automation: Advancing automation in design tools is gathering momentum," *IEEE Power Electronics Magazine*, vol. 6, no. 1, pp. 28–33, March 2019.

[6] J. W. Kolar and F. Krismer, "Power electronics design 4.0," in *IEEE Design Automation for Power Electronics Workshop (DAPE)*, September 2018.

[7] F. C. Lee, "High-frequency quasi-resonant and multi-resonant converter technologies," in *Proceedings.14 Annual Conference of Industrial Electronics Society*, vol. 3, Oct 1988, pp. 509–521.

[8] M. D. Seeman and S. R. Sanders, "Analysis and optimization of switched-capacitor dc–dc converters," *IEEE Transactions on Power Electronics*, vol. 23, no. 2, pp. 841–851, March 2008.

[9] K. W. E. Cheng, "New generation of switched capacitor converters," in *PESC 98 Record. 29th Annual IEEE Power Electronics Specialists Conference (Cat. No.98CH36196)*, vol. 2, May 1998, pp. 1529–1535 vol.2.

[10] Y. Lei, R. May, and R. Pilawa-Podgurski, "Split-phase control: Achieving complete soft-charging operation of a dickson switched-capacitor converter," *IEEE Transactions on Power Electronics*, vol. 31, no. 1, pp. 770–782, Jan 2016.

[11] H. Chen, H. Kim, R. Erickson, and D. Maksimović, "Electrified automotive powertrain architecture using composite dc–dc converters," *IEEE Transactions on Power Electronics*, vol. 32, no. 1, pp. 98–116, Jan 2017.

[12] R. D. Middlebrook and S. Cuk, "A general unified approach to modelling switching-converter power stages," in *1976 IEEE Power Electronics Specialists Conference*, 1976, pp. 18–34.

[13] G. C. Verghese, M. E. Elbuluk, and J. G. Kassakian, "A general approach to sampled-data modeling for power electronic circuits," *IEEE Transactions on Power Electronics*, vol. 1, no. 2, pp. 76–89, 1986.

[14] H. R. Visser and P. P. J. van den Bosch, "Modelling of periodically switching networks," in *PESC '91 Record 22nd Annual IEEE Power Electronics Specialists Conference*, June 1991, pp. 67–73.

[15] F. C. Y. Lee, R. P. Iwens, Y. Yu, and J. E. Triner, "Generalized computer-aided discrete time-domain modeling and analysis of dc-dc converters," *IEEE Transactions on Industrial Electronics and Control Instrumentation*, vol. IECI-26, no. 2, pp. 58–69, May 1979.

[16] A. Kumar, J. Lu, and K. K. Afridi, "Enhanced-accuracy augmented state-space approach to steady-state modeling of resonant converters," in *2015 IEEE 16th Workshop on Control and Modeling for Power Electronics (COMPEL)*, July 2015, pp. 1–6.

[17] L. O. Chua and P.-M. Lin, *Computer-Aided Analysis of Electrionic Circuits*. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1975.

[18] D. Costinett, R. Zane, and D. Maksimovic, "Discrete time modeling of output disturbances in the dual active bridge converter," in *2014 IEEE Applied Power Electronics Conference and Exposition - APEC 2014*, March 2014, pp. 1171–1177.

[19] C. Moler and C. Van Loan, "Nineteen dubious ways to compute the exponential of a matrix," *SIAM Review*, vol. 20, no. 4, pp. 801–836, 1978.

[20] C. Van Loan, "Computing integrals involving the matrix exponential," *IEEE Transactions on Automatic Control*, vol. 23, no. 3, pp. 395–404, June 1978.

[21] L. W. Nagel, "SPICE2: A computer program to simulate semiconductor circuits," Ph.D. dissertation, EECS Department, University of California, Berkeley, 1975.

[22] "Plecs user manual," Plexim, Tech. Rep., 2016. [Online]. Available: https://www.plexim.com/files/plecsmanual.pdf