Steady-State Convergence of Discrete Time State-Space Modeling with State-Dependent Switching

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*—Steady-state modeling plays an important role in the design of advanced power converters. Typically, steady-state modeling is completed by time-stepping simulators, which may be slow to converge to steady-state, or by dedicated analysis, which is time-consuming to develop across multiple topologies. Discrete time state-space modeling is a uniform approach to rapidly simulate arbitrary power converter designs. However, the approach requires modification to capture state-dependent switching, such as diode switching or current programmed modulation. This work provides a framework to identify and correct state-dependent switching within discrete time state-space modeling and shows the utility of the proposed method within the power converter design process.**

I. Introduction

The power electronics design process often utilizes several iterations of modeling and prototyping to achieve the goal performance. The general process from design specification to verification typically involves several iterations of steady-state converter modeling or simulation across varying topologies, followed by dedicated converter-specific simulation to reach a prototype-ready design [1]. The broad-scale initial design phase is the focus of this paper, where the steady-state modeling of many converter topologies, discrete components, magnetics, and frequencies results in a quasi-ordered optimization problem.

Several techniques can be utilized to complete the initial design phase. Mathematical approximations such as averaged, lossless, or fundamental harmonic analysis can capture the dominant dynamics of specific converter topologies. However, these models take time to produce and do not readily translate to other topologies without experience-based assessment of the validity of the approximations. More detailed models that can capture a wider array of converters become too large to employ rapidly, even with advanced computational power [2], [3].

Commercial time-stepping simulators such as LTspice or Simulink are built to handle arbitrary circuit topologies and provide a singular platform to draw comparisons. These simulators typically use nodal or state-space analysis to linearize the converter waveforms within a small time step [4]. However,

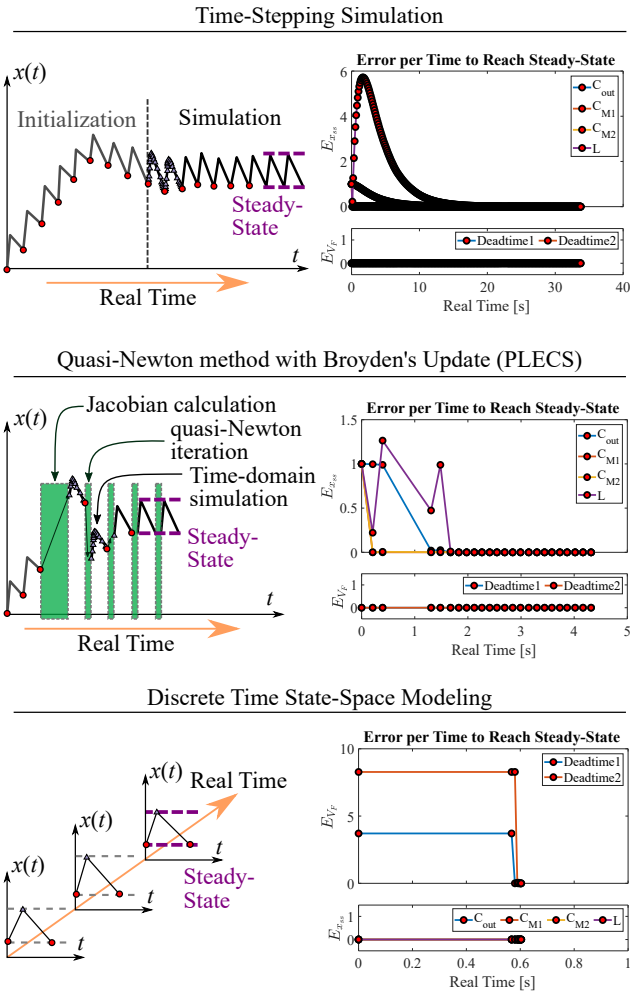


Fig. 1. Detailed flow diagram of broad electrical domain design of a synchronous buck converter. $E_{x_{ss}}$ is the error between the initial and final state values, and E_{V_F} is the error of the final voltage value at the end of each deadtime to the forward voltage of the transistors. The red dots on the right figure indicate the timestamp for one period of simulation. The blue triangles represent intermediate values solved for within the period.

to produce an optimal design, thousands or millions of configurations are iteratively modeled and assessed, which can be extremely time consuming for time-stepping simulators. The top row of Fig. 1 shows an example of the convergence of time-stepping simulators. Unless accurate initial conditions are provided, an initialization stage is used to reach the correct operating dynamics of the converter. During this time, waveforms and values within the converter can widely vary and vastly deviate from the steady-state operating point. Convergence to steady-state may require simulation over many periods, as large, low-loss filter elements will exhibit underdamped low-frequency transient resonance. Thus, the time to reach steady-state can take multiple seconds for a single converter design.

Improvements have been made to expedite the steady-state convergence of time-stepping simulation using optimization techniques. The Quasi-Newton method with Broyden's Update [4], used in PLECS, can provide a much faster solution, as shown in the middle row of Fig. 1. This technique uses Newtonian iteration to solve the error minimization problem

$$\min_{\boldsymbol{x}(0)} \Big[f(\boldsymbol{x}(0)) = ||\boldsymbol{x}(T_s) - \boldsymbol{x}(0)|| \Big] \tag{1}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^{n_s}$ is the vector of the n_s circuit states and $\boldsymbol{x}(T_s)$ is evaluated through time-stepping simulation over one switching period. While this approach works well for simple topologies, Jacobian calculation becomes expensive and convergence is not guaranteed for more-complicated topologies with many energy storage elements or simple topologies with many parasitic elements, due to iteration in an n-dimensional space.

The iterative optimization techniques in broad-scale design require a method that is general, yet fast to model the steady-state of converters. Discrete time state-space modeling is a uniform approach to rapidly simulate arbitrary power converter designs. Instead of reducing the error of the initial and final state values, discrete time state-space modeling directly solves for the steady-state vector $\boldsymbol{x}(0)$ in closed-form. However, the solution requires a predefined switching pattern, which may not be known a priori in the presence of state-dependent switching, such as diode switching or current programmed modulation. In this event, the technique again requires an iterative solution, this time altering the switching times (as opposed to the initial states), as shown in the bottom row of Fig. 1.

This work provides a framework to identify and correct state-dependent switching within discrete time state-space modeling and shows the utility of the proposed method within the power converter design process. Section II reviews discrete time state-space modeling and the difference between state-independent and state-dependent switching. Section III provides an overview of the different categories of state-dependent switching, presents the proposed algorithm to find the steady-state of a converter with state-dependent switching, and shows a comparison to other simulation methods. Experimental results in Section IV illustrate the accuracy of the proposed modeling method, and the paper is concluded in Section V.

II. Discrete Time State-Space Modeling

Discrete time state-space modeling is a generalized analysis framework for both steady-state and dynamic modeling of switched circuits [2], [5]–[9]. As long as the converter can be approximated as a linear-equivalent circuit within each switching interval, a switched mode power supply can be modeled within each switching interval i using the state-space equations

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}_i \boldsymbol{x}(t) + \boldsymbol{B}_i \boldsymbol{u}(t) \tag{2}$$

$$\boldsymbol{y}(t) = \boldsymbol{C}_i \boldsymbol{x}(t) + \boldsymbol{D}_i \boldsymbol{u}(t). \tag{3}$$

The state vector, $\boldsymbol{x}$, contains the capacitor voltages and inductor currents from the circuit, and the output vector, $\boldsymbol{y}$, is any combination of node voltages or component currents contained within the converter. The solution to the state equation during any single switching interval, assuming all independent inputs are constant, is

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

By solving (4) over the set of n linear-equivalent circuits within one switching period, the periodic steady-state solution ($\boldsymbol{x}(0) = \boldsymbol{x}(T_s)$) is

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

After solving (5), the value of $\mathbf{x}(t)$ at any time within the period can be solved for using (4).

The initial limitations of discrete time state-space modeling via (5) include assuming all inputs to be constant during each switching interval $\boldsymbol{u}(t) \approx \boldsymbol{u}$, ensuring the state matrix $\boldsymbol{A}_i$ is invertible during each switching interval, approximating each switching interval as a linear-equivalent circuit, and knowing the duration of each switching interval t_i before solving the steady-state of the converter using (5). Implications and mitigation methods for the first three of these limitations are discussed in [3]; this work focuses on the last.

A. State-Independent and State-Dependent Switching

Power converters are assumed to be well-modeled as piecewise linear to employ (5). Each switching action, either by a transistor or diode, alters the linear-equivalent model of the converter by changing the conduction resistance or forward voltage lumped elements in the linear-equivalent model. The time intervals and their associated linear-equivalent circuits must be known before solving (5).

State-independent or active switching actions caused by a controller driving a transistor are set independently by an external signal. Thus, the timing of each of these switching actions are known a priori for a given converter design and steady-state operating point. However, state-dependent or passive switching, such as a diode conducting due to its forward threshold voltage being reached, occurs at a time dictated by

the internal waveforms of the converter. The presence of state-dependent devices results in a nonlinear state-space description within a single switching interval,

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}_i(\boldsymbol{x})\boldsymbol{x}(t) + \boldsymbol{B}_i(\boldsymbol{x})\boldsymbol{u}(t). \tag{6}$$

Alternatively, if the state-dependence can be modeled as piecewise linear, the system can continue to be modeled as linear as in (2). In this case, matrices within an interval where a state-dependent switching action occurs will be split into two (or more) subintervals, e.g. for a diode with different forward biased (on) and reverse biased (off) equivalent circuits,

$$\boldsymbol{A}_i = \begin{cases} \boldsymbol{A}_{i,D(on)}, & \text{if } -V_d \geq V_{FW} \\ \boldsymbol{A}_{i,D(off)}, & \text{otherwise} \end{cases} \tag{7}$$

and similar for $\boldsymbol{B}_i$. The time-duration of each individual subinterval impacted by state-dependent switching is not known without additional analysis. Because (5) always solves for the steady-state of the converter, other states throughout the converter are altered whenever the timing of these switching actions is altered. To maintain the merits of the direct steady-state solution (5), the time-duration of each subinterval must be solved such that (7) remains valid at every instant in the steady-state solution waveforms.

An example of passive switching due to the body diodes in the synchronous buck converter of Fig. 2 is shown in Fig. 3. The piecewise linear model of each MOSFET is implemented as a variable voltage source and resistance,

$$\boldsymbol{V}_{F,M_i} = \begin{cases} V_{FW}, & \text{if } V_{ds,i} \leq -V_{FW} \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

$$\boldsymbol{R}_{on-off,M_i} = \begin{cases} R_{ds(on),i}, & \text{if } M_i \text{ on} \\ R_{d,i}, & \text{if } V_{ds,i} \leq -V_{FW} \\ \infty, & \text{otherwise} \end{cases}. \tag{9}$$

The diode conduction behavior is unknown prior to solving the steady-state of the converter via (5). An initial guess for the switching interval time-durations supposes that the body diodes do not turn on during the deadtime. The steady-state solution under this assumption, given by the red dashed line in Fig. 3, contains a violation of (7). Namely, the switched-node voltage v_{sw} resonates below $-V_{FW}$, where the diode should turn on. Only observing the discrete time samples, this error can be seen because the voltage v_{sw} at the end of the period is well below the forward voltage.

To correct this violation, a new time interval must be introduced to the converter where the body diode of M_2 conducts. This splits the original deadtime interval into two separate subintervals with their own linear-equivalent circuit as shown in Table I. Assuming the correct time-duration of each subinterval is found, this change results in a valid steady-state solution as shown by the solid blue line in Fig. 3. Note that, by introducing or changing the duration of switching intervals in the converter, the initial states at the beginning of the deadtime have changed due to the new steady-state solution; the correct

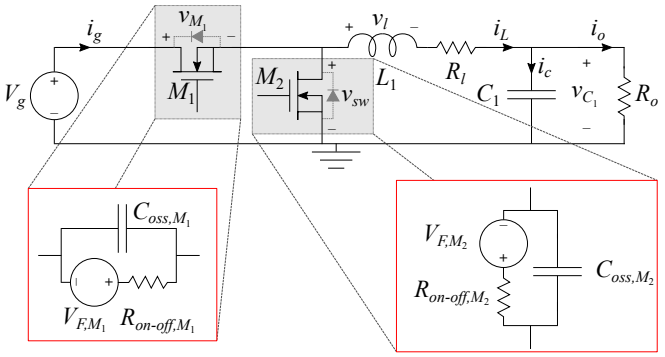


Fig. 2. Synchronous buck converter and equivalent transistor representation used in Fig. 1 and Fig. 3.

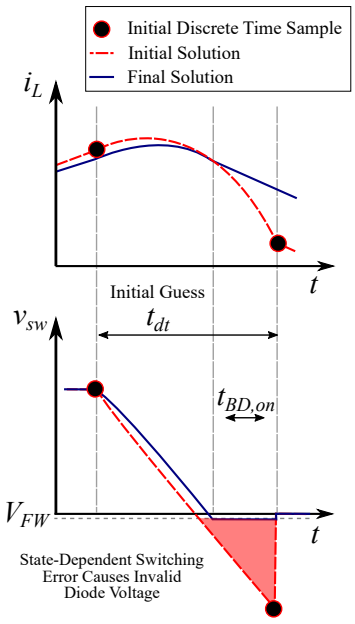


Fig. 3. Waveform of a synchronous buck converter after the high side device turns off. The initial guessed deadtime length is too long between active switching actions.

$t_{BD(on)}$ is not equal to the amount of time that $v_{sw} < -V_{FW}$ in the intial solution.

Both the initial and final solutions in Fig. 3 are in steady-state, i.e. $\boldsymbol{x}(0) = \boldsymbol{x}(T_s)$, as enforced by the solution of (5). However, only the final solution exhibits a steady-state waveform with no violations of (8)-(9).

B. Steady-State Solution with State-Dependent Switching

To model the presence of state-dependent switching actions, the following notation is used. The index i consistently refers to the switching interval order as determined by only the state-independent (i.e. active, or externally-controlled) switching actions, $i \in \{1, 2, ..., n\}$ for n active switching intervals. The vector $\boldsymbol{p} \in \mathbb{N}^n$ contains the number of distinct subintervals in each active switching interval due to state-dependent switching. The index j refers to switching subintervals within the

TABLE I
BUCK CONVERTER DEADTIME SOLUTION

Initial Solution		
Time Interval	t_{dt}	
Circuit Model	$\boldsymbol{A}_{i,BD(off)}$	
Conducting Devices	—	
Final Solution		
Time Interval	t_{dt}	
	$t_{BD(off)}$	$t_{BD(on)}$
Circuit Model	$\boldsymbol{A}_{i,BD(off)}$	$\boldsymbol{A}_{i,BD(on)}$
Conducting Devices	—	$M_{2,BD}$

i^{th} interval that occur due to the presence of state-dependent switching; in interval i, $j \in \{1, ..., p_i\}$. State matrices $\mathbf{A}_{(i,j)}$ and $\mathbf{B}_{(i,j)}$ model the linear-equivalent circuit within the j^{th} passive subinterval of the i^{th} active interval. In all cases $\mathbf{A}_{(i,1)} = \mathbf{A}_i$, as defined by (2).

Subinterval time-durations are indexed accordingly, such that

$$t_i = \sum_{j=1}^{p_i} t_{(i,j)}. \tag{10}$$

To find the valid steady-state of a power converter using discrete time state-space modeling, each passive switching action must be found within each active switching action. The active switching time intervals are modeled as

$$\boldsymbol{X}_{ss} = \left[\mathbf{I} - \prod_{i=1}^{n} \boldsymbol{\Phi}_i\right]^{-1} \sum_{i=1}^{n} \left[\boldsymbol{\Psi}_i\right] \tag{11}$$

where

$$\boldsymbol{\Phi}_i = \prod_{j=1}^{p_i} \mathrm{e}^{\boldsymbol{A}_{(i,j)} t_{(i,j)}} \tag{12}$$

and

$$\boldsymbol{\Psi}_i = \sum_{j=1}^{p_i} \left[\left(\prod_{k=i+1}^{n} \boldsymbol{\Phi}_k \right) \left(\prod_{k=j+1}^{p_i} \mathrm{e}^{\boldsymbol{A}_{(i,k)} t_{(i,k)}} \right) \times \boldsymbol{A}_{(i,j)}{}^{-1} \left[\mathrm{e}^{\boldsymbol{A}_{(i,j)} t_{(i,j)}} - \mathbf{I} \right] \boldsymbol{B}_{(i,j)} \boldsymbol{u} \right]. \tag{13}$$

For each active switching time interval i, there is a set of p_i passive switching time intervals which solves the steady-state of the converter such that all state-dependent switching constraints (e.g. (7)) are satisfied. In general, this sequence is solved through numerical iteration.

For the specific case of state-dependence arising from diode conduction, this iteration is employed to ensure the inequality

$$\boldsymbol{E}_{(i,j)} \big(\boldsymbol{y}_D + \boldsymbol{V}_F \big) < 0 \tag{14}$$

where $\boldsymbol{E}_{(i,j)}$ is a vector of ± 1 values to indicate whether a diode is conducting or blocking, $\boldsymbol{V}_F$ is a vector of each diode's forward voltage, and $\boldsymbol{y}_D$ is a vector of instantaneous diode voltages during the entirety of the steady-state waveforms,

$$\boldsymbol{y}_D(t) = \boldsymbol{C}_{\boldsymbol{D}(i,j)} \boldsymbol{x}(t) + \boldsymbol{D}_{\boldsymbol{D}(i,j)} \boldsymbol{u}(t) \tag{15}$$

for appropriately-derived $\boldsymbol{C_D}$ and $\boldsymbol{D_D}$.

The goal of the framework is to solve a steady-state solution consisting of interval durations $t_{i,j}$ and the associated conduction ordering of all switching devices constrained by (10) and solve the steady-state initial condition for each iteration from (11) while ensuring that (14) holds for any time $t \in [0, T_s]$ within the switching period.

C. Example Convergence of Synchronous Buck

To examine the constraints on an iterative solution accounting for passive switching, Fig. 4 shows an example of the solution space for the synchronous buck converter of Fig. 2. The converter operates in CRM with ZVS and has parameters as listed in Table II.

TABLE II
EXAMPLE SYNCHRONOUS BUCK PARAMETERS

L_1	500 nH	$C_{oss,M1}$	190 pF
C_1	100 µF	$C_{oss,M2}$	880 pF
R_l	1 mF	R_{ds}	2.5 mΩ
V_g	48 V	f_s	200 kHz
I_o	1 A	D	$\frac{1}{48}$

Fig. 4 shows the error in state-dependent conditions at the discrete time sampling points, as the two deadtimes of the buck converter are independently varied from 0.1% to 3.3% of the switching period T_s. In all cases, both transistors are treated as switches as modeled in Fig. 2; no attempt is made in this example to account for body diode conduction in order to examine the solution space. The error shown in Fig. 4 is

$$\text{Err} = \begin{cases} v_{sw} - (V_g + V_{FW}), & \text{if } v_{sw} > (V_g + V_{FW}) \\ v_{sw} + V_{FW}, & \text{if } v_{sw} < -V_{FW} \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

and models the magnitude of the spurious over-voltage or under-voltage at the switch node, which would be clamped by body diode conduction in the converter.

When only the discrete sampling points are examined, there are multiple zero-error regions, where only the points near the minimum values of t_{d1} and t_{d2} are valid. Fig. 5 shows the waveforms during each deadtime interval of two operating points from Fig. 4. Operating point 1 is in the valid zero-error region, with $t_{d1}/Ts = 0.17\%$ and $t_{d2}/Ts = 0.23\%$. Operating point 2 is in an invalid zero-error region, with $t_{d1}/Ts = 1.8\%$ and $t_{d2}/Ts = 1.5\%$. Note that, examining only the discrete time points (green diamonds), both operating points have sampled v_{sw} within the boundaries of diode conduction at the beginning and end of the subinterval. However, when the full waveforms of both points are examined, only operating point 1 is valid at every time instant within the subinterval.

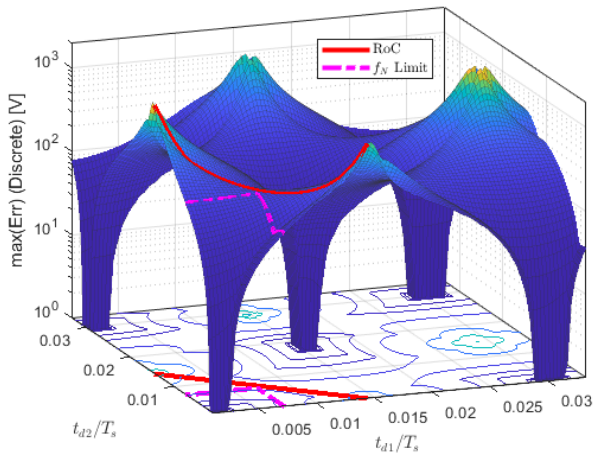


Fig. 4. Passive switching error for the synchronous buck converter considering only the discrete sampling points.

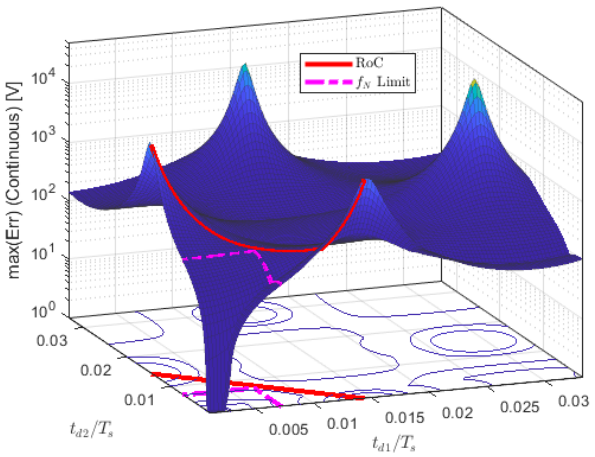


Fig. 6. Passive switching error for the synchronous buck converter considering full continuous-time waveforms.

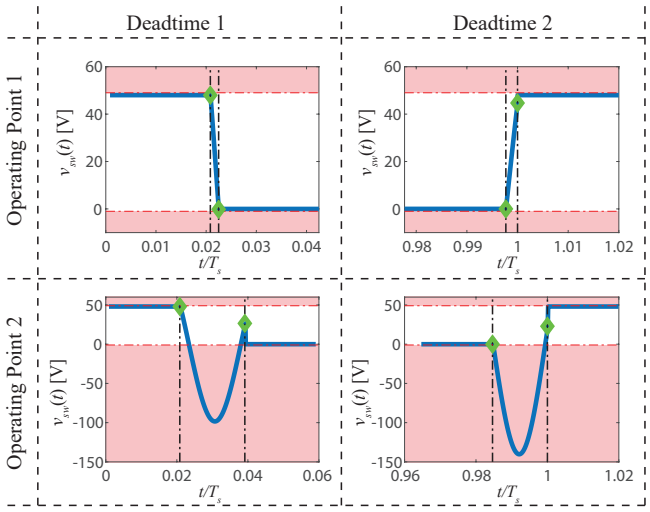


Fig. 5. Example complete waveforms (solid blue lines) and discrete sampling points (green diamonds) for both deadtimes, and two operating points, of the synchronous buck converter.

Fig. 6 shows the error function over every point in the waveform, rather than solely the discrete time samples. In this case, only valid waveforms show zero error; however, local minima still exist at the invalid operating points from Fig. 4.

Data in Fig. 6 is obtained by time-stepping through a single period of the steady-state solution from (5) with the prescribed deadtimes. Though errors outside of the discrete sampling points can be identified, this process is computationally intensive and greatly increases the simulation time. Thus, the process should be avoided unless necessary.

The solid red line in both Fig. 4 and Fig. 6 indicates the region of convergence (RoC) for a theoretical gradient-based error minimization algorithm. As long as the initial guess for the converter deadtimes is within this region, a gradient-descent method will arrive at a valid operating point. The dashed magenta line in both figures represents a boundary based on the natural frequencies of the converter, as discussed in the following section. In this case, limiting both deadtimes according to the natural frequencies of the state matrix guarantees timing durations within the RoC. The synchronous buck converter presents a relatively simple case for examining the solution space for a passive switching algorithm. Other topologies, which may exhibit multiple passive switching actions within a single interval or where error resulting from passive switching violations has a stronger impact on the complete steady-state waveform, exhibit a higher degree of non-convexity in the solution space.

The following section develops an algorithm to systematically check and address passive switching constraints to eventually converge to a valid steady-state solution.

III. Passive Switching Correction

Due to the nonlinearity of the problem, an iterative numerical approach is used to obtain the correct steady-state solution in the presence of state-dependent switching actions. In each iteration, the duration of passive switching intervals may be altered, or new switching states may be inserted or removed from the switching pattern. This section discusses the implementation of a method to determine the iterative changes to the switching pattern and subinterval timing such that the steady-state solution converges to a violation-free result in the presence of state-dependent switching actions.

For subintervals where the slope of the waveform exhibiting a violation is nearly constant, such as v_{sw} in Fig. 3, the identification of an invalid steady-state solution due to passive switching is easily solved by evaluating the discrete time points and their derivatives. The length of the body diode time interval in the synchronous buck converter example can be solved by taking the partial derivative of the state variables at the discrete switching points with respect to the time

TABLE III
CATEGORIES OF FREQUENCY DYNAMICS IN THE PROPOSED METHOD

Tier	Condition
Constant slope	$\text{sgn}(\dot{x}(t_b)) = \text{sgn}(\dot{x}(t_e))$ & $\frac{\pi}{\text{Im}(\text{eig}(A))} > t_e - t_b$
Single-Frequency	One violation of: $\frac{\pi}{\text{Im}(\text{eig}(A))} > t_e - t_b$ ‖ $\text{sgn}(\dot{x}(t_b)) \neq \text{sgn}(\dot{x}(t_e))$
Multi-Frequency	Multiple violations of: $\frac{\pi}{\text{Im}(\text{eig}(A))} > t_e - t_b$

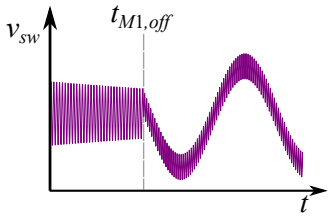


Fig. 7. Non-synchronous buck converter switch node waveform with a small stray loop inductance added in series with transistor.

interval [3]. Nearly-constant slope subintervals occur when the duration of the subinterval is significantly shorter than the dominant dynamic frequencies of the circuit. When the inverse is true, highly resonant or multi-resonant dynamics are present, and the violations are more difficult to identify and correct.

Eigenvalues of $\boldsymbol{A}_i$ are used to observe the natural frequencies of each sub-interval. The natural frequencies of each state matrix, along with the discrete time sample derivatives, separate each time interval of a converter into three distinct categories, each having their own identification and feasible state-dependent switching correction method.

The first tier, the 'constant slope' category, is similar to the example in Fig. 3. The natural frequencies of the circuit are sufficiently small such that, within the subinterval, the derivatives of the state variables associated with the violation are nearly constant. The second tier, the 'single-resonant' category, arises when a single natural frequency has a period which is less than, or nearly equal to, the subinterval time-duration. The third tier, the 'multi-resonant' category, occurs when the eigenvalues indicate multiple natural frequencies are faster than the time interval.

The conditions of these tiers are outlined in Table III, with t_b and t_e representing the beginning and ending times of the time interval, respectively, and an example of the single- and multi-resonant categories is shown for a non-synchronous buck converter with a small power loop inductance in Fig. 7. The loop inductance causes a single-resonance while the transistor is turned on and multi-resonance while the transistor is off during the deadtime interval.

When dynamics in the second or third tier occur, the simple slope-projection method of [3] is insufficient to dictate a correction for the ensuing iteration.

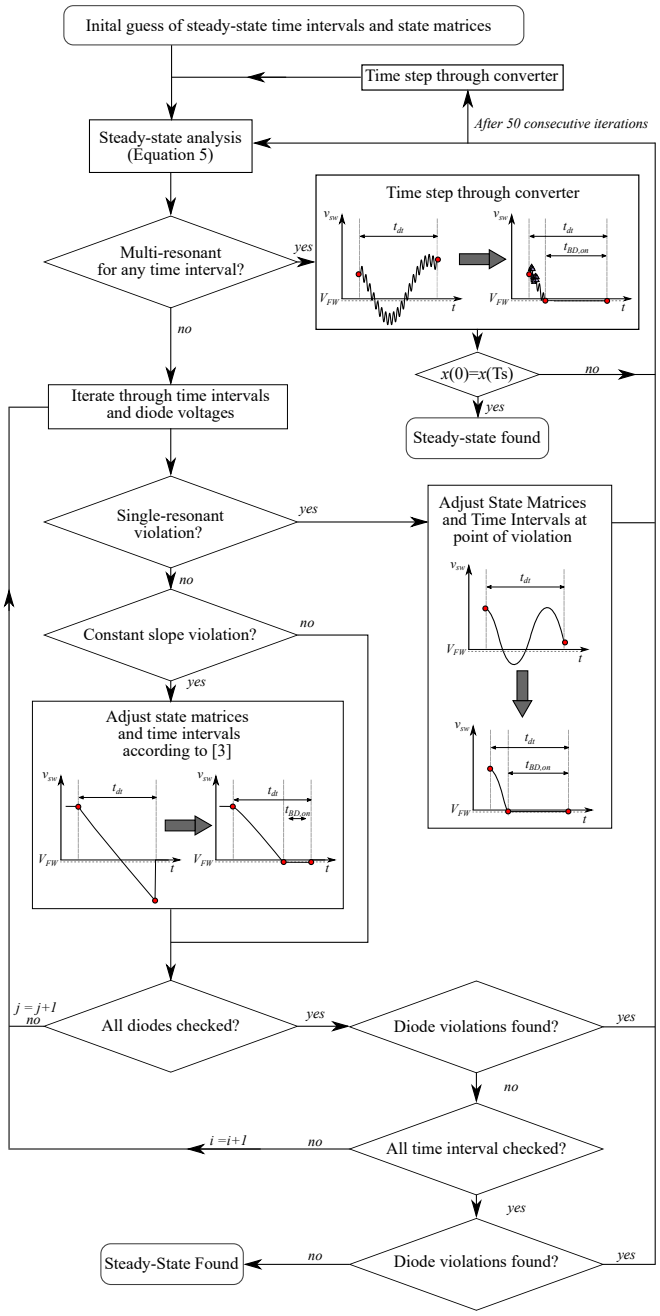


Fig. 8. Flow diagram of how multi-resonant and single-resonant circuits are addressed in the proposed method.

A. Algorithm

The proposed implementation of the eigenvalue and derivative checks for discrete time state-space modeling of power converters is shown in Fig. 8. After finding the steady-state of the active switching intervals of a converter, the eigenvalues are first checked and recorded for each subinterval.

If there is a violation during a multi-resonant interval, the converter is time-stepped for one period to find the approximate location of any state-dependent switching actions. Because the time-stepping begins from an initial state $\boldsymbol{X_{ss}}$,

TABLE IV
STEADY-STATE SOLVE TIME OF SIMULATION METHODS

	Convergence Time (sec)[1]				
Simulation	DAB	HDSC	Buck	Flyback	Buck-boost
LTspice	3.78	7.39	0.79	8.33	2.95
PLECS					
-Transient	7.82	3.20	2.20	9.16	4.34
-Steady-State	-	-	21.11	-	-
Proposed Method					
-Total	5.00	2.96	0.52	6.98	3.52
-Solver only	0.87	0.01	0.03	5.85	0.71
-Time-stepping iterations	0	0	0	7	0

[1] All simulations performed on a laptop computer with an Intel(R) Core(TM) i7-8750H CPU @ 2.20 GHz processor, 16 GB of RAM, and a 64 bit operating system.

TABLE V
STEADY-STATE SOLVE CONVERTER PROPERTIES

	Amount				
Parameter	DAB	HDSC	Buck	Flyback	Buck-boost
State Variables	11	13	4	7	9
Time Intervals	8	4	4	2	12

which exhibited passive switching violations, the time-stepped period will, in general, not result in steady-state. Steady-state is then re-solved using (5), with the new switching pattern and time intervals found from time-stepping.

If no multi-resonance is detected, diode voltages at each discrete time interval are first examined for violations. If a single-resonance is detected, additional discrete time samples are solved for every quarter period of the fastest natural frequency of the converter. Using these additional discrete time points, a binary search is completed to quickly determine if there is a state-dependent switching violation within the time interval. Intervals where states have semi-constant slopes are identified and corrected using the method from [3].

There are certain cases in which the proposed algorithm still exhibits slow convergence to steady-state. For example, if an initial multi-frequency violation is detected, additional diode conducting time intervals may be placed throughout the steady-state waveform of the converter. If the steady-state solution is not sufficiently close to the final steady-state value, small adjustments in state-dependent switching times can greatly alter the behavior of the converter. Such conditions are likely to occur if multiple extended diode conduction times are required to achieve steady-state or resonant combinations throughout the period. Evaluating and eliminating each violation takes multiple iterations of the algorithm; thus, after 50 iterations, the converter is solved via time-stepping through one period. This resets the diode conduction time intervals at an initial condition closer to the steady-state value.

B. Simulation Verification

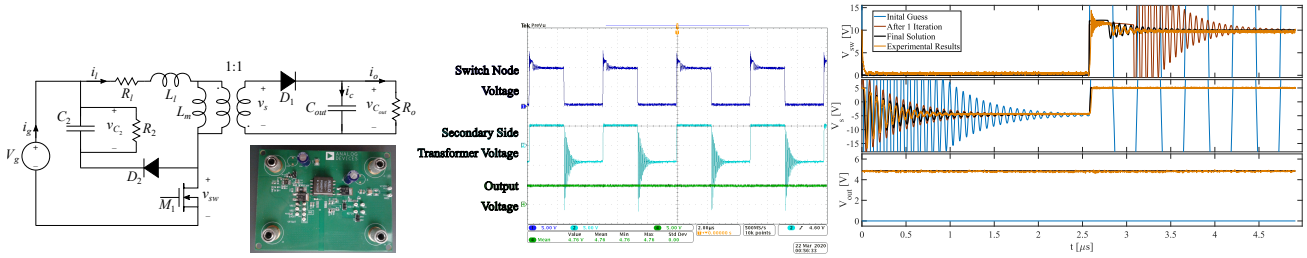
Several different types of converters are used to verify the proposed algorithm against other simulation tools in terms of real-time convergence speed. The DAB, HDSC, and buck converters are the same as detailed in [3]. The flyback and buck-boost are modeled after the experimental prototypes in Fig. 9 and Fig. 10, respectively. In each case, the transistors are modeled as shown in Fig. 2. Additionally, for LTspice and the PLECS transient method, waveform data is not saved until the steady-state of the converter is reached. Although this method requires the simulation time to reach steady-state be known before simulation, it provided a fairer comparison across all methods since only steady-state waveforms are being considered. The results of the simulation verification are shown in Table IV, with Table V displaying the number of state variables and time intervals for each converter. The proposed method does require some initialization time, as indicated by the large initial gap in error point in Fig. 1. This initialization consists of parsing a circuit netlist to obtain the $\boldsymbol{A}_i$ and $\boldsymbol{B}_i$ matrices of the topology. The solver time for the proposed method shows the time to reach steady-state after initialization. Circuit parsing is completed symbolically, such that the process would not be repeated in an optimization routine if there are no changes to the base topology.

The simulation results show the dual-active bridge converter and the hybrid Dickson switch capacitor converter take longer to initialize in the proposed method because of their high number of states. Only the flyback converter takes an extended period of time to solve due to the extended ringing caused by the leakage inductance of the transformer. Regardless, the solver time of the proposed method is well below that of any commercial tool in Table IV.

IV. EXPERIMENTAL VERIFICATION

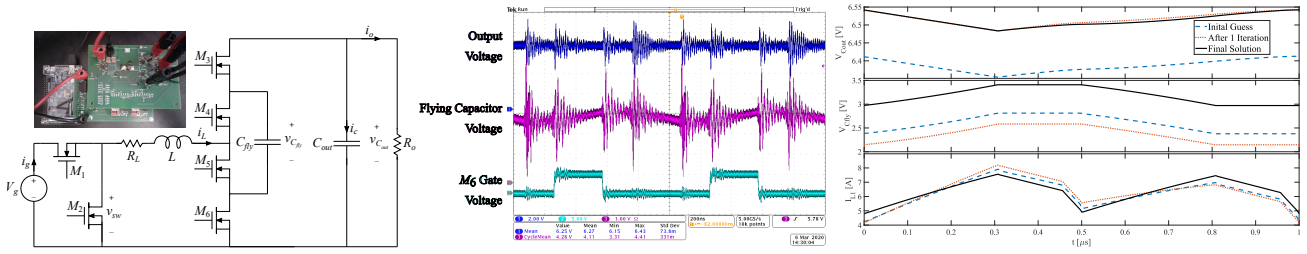
The flyback converter shown in Fig. 9 shows the progression of the proposed method through all three possible conditions. The converter is an evaluation board (EVAL-CN0342-EB1Z) with a 1:1 transformer and a RCD snubber network. Initially, only two switching intervals are known, since there is only one active switch in the converter, M_1. During the extended time when D_2 should be on, there is a multi-resonance that requires time-stepping to correct. The steady-state of the converter is then solved using the linear projection method. The simulation timing results are shown in Table IV.

The buck-boost converter shown in Fig. 10 is an example of the versatility of the proposed method for more than just diode violation correction, but also for other types of state-dependent switching. The converter uses EPC GaN devices with a 4.4 μF flying capacitor and 200 nH inductor. The initial guess of time intervals causes the converter to behave non-ideally, since the inductor current is unbalanced across $\frac{T_s}{2}$ and the flying capacitor is less than $\frac{V_{out}}{2}$. Other simulators would require a control loop to account for this imbalance [10], [11]. However, in the proposed method, a constraint is placed on the inductor current, similar to the diode forward voltage constraint, to force $i_L = 4.9$ A at $\frac{T_s}{2}$ and T_s, emulating



(a) (b) (c)

Fig. 9. Flyback converter (EVAL-CN0342-EB1Z) schematic (a), experimental waveforms (b), modeled waveforms (c).



(a) (b) (c)

Fig. 10. Buck-boost converter prototype and schematic (a), experimental waveforms (b), modeled waveforms (c).

a valley-current programmed mode controller. The proposed method finds the final solution consistently in under 4 seconds.

V. Conclusion

This work proposes a method to quickly solve the steady-state of power converters with state-dependent switching using discrete time state-space modeling. The state matrix eigenvalues are examined to determine the method needed to identify and correct passive switching. The proposed method gives an expanded capability to rapidly model state-dependent switching in power converters. Simulation results show the rapid convergence of the method compared to other simulation tools.

Acknowledgment

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] K. Hermanns, Y. Peng, and A. Mantooth, "The increasing role of design automation in power electronics: Gathering what is needed," *IEEE Power Electronics Magazine*, vol. 7, no. 1, pp. 46–50, March 2020.

[2] 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.

[3] J. A. Baxter and D. J. Costinett, "Converter analysis using discrete time state-space modeling," in *2019 20th Workshop on Control and Modeling for Power Electronics (COMPEL)*, June 2019, pp. 1–8.

[4] 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.

[5] 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.

[6] D. J. Costinett, "Analysis and design of high efficiency, high conversion ratio, dc-dc power converters," Ph.D. dissertation, University of Colorado at Boulder, 2013.

[7] 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.

[8] 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.

[9] 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.

[10] Z. Xia, B. L. Dobbins, and J. T. Stauth, "Natural balancing of flying capacitor multilevel converters at nominal conversion ratios," in *2019 20th Workshop on Control and Modeling for Power Electronics (COMPEL)*, 2019, pp. 1–8.

[11] J. Celikovic, R. Das, H. Le, and D. Maksimovic, "Modeling of capacitor voltage imbalance in flying capacitor multilevel dc-dc converters," in *2019 20th Workshop on Control and Modeling for Power Electronics (COMPEL)*, 2019, pp. 1–8.