2023 IEEE 24th Workshop on Control and Modeling for Power Electronics (COMPEL) | 979-8-3503-1618-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/COMPEL52896.2023.10221030

Power Converter and Discrete Device Optimization Utilizing 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*—Broad-scale modeling and optimization play a vital role in the design of advanced power converters. Optimization is normally implemented via brute force iterations of design variables or utilizing metaheuristic techniques which are time consuming for a wide range of potential topologies, device implementations, and operating points. Recently, discrete time state-space modeling has shown merits in rapid analysis and generality to arbitrary circuit topologies but has not yet been utilized under rapid optimization techniques across multiple converter parameters. In this work, we investigate methods to incorporate rapid gradient-based optimization techniques to leverage discrete time state-space modeling and showcase the approach in the power converter design process. The method is validated on a 48-to-1V converter designed using the proposed techniques.**

I. Introduction

In the design of power electronics, steady-state modeling forms the basis of many metrics for the assessment of converter design [1]. As such, techniques for general, rapid, and highly-accurate steady-state analysis of arbitrary power converters are of broad appeal [2]–[6]. Simultaneously, the increasing availability of significant computational power has expanded the role of design automation and optimization in the converter design process for both late-stage, hardware implementation and early-stage topology selection and schematic-level design [7], [8].

Fig. 1 shows an example of the latter. Given a set of constraint and objectives for a power converter, the goal of the early-stage process is to generate a schematic-level implementation suitable for physical prototyping or detailed multi-physics modeling. This process includes selection of converter topology, components, modulation, and some basic considerations of form factor that can be modeled prior to a circuit implementation. In most cases, this design process is iterative, and requires assessing many devices, components, modulations, and topologies to arrive at a suitable candidate with acceptable performance.

This work gives a framework to optimize converters utilizing the rapid modeling method of discrete time state-space modeling [3], [4], [6] and adapting gradient-based optimization techniques for discrete component selection as well as continuous parameters such as device area and frequency. The method is used to design a 48-to-1 V converter through a multi-topology optimization.

Section II briefly reviews state space modeling and the nature of externally-controlled and state-dependent switching actions. Section III develops gradient-based techniques for incorporating state-dependent switching into the steady-state solution. Section IV extends the approach to design optimization, including component and switching frequency selection. Section V gives details of an experimental prototype designed using the developed techniques. Finally, Section VI concludes the paper.

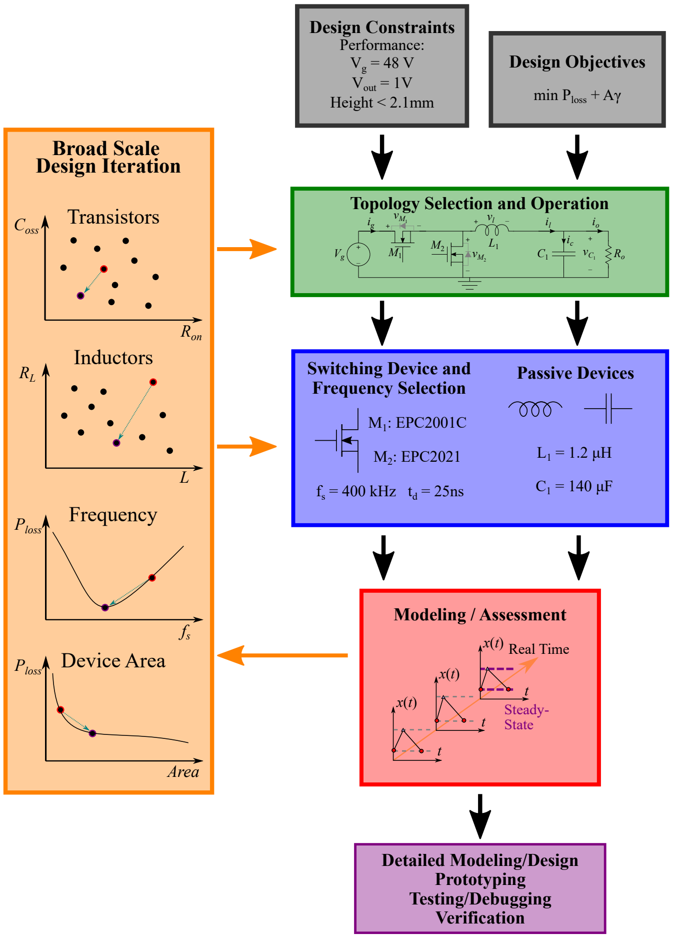


Fig. 1. Early stage broad-scale optimization flowchart for the design of power converters.

II. Discrete Time State-Space Modeling

Discrete time state-space modeling is a proven steady-state and dynamic analysis method for switched circuits [2]–[4], [9]–[11]. By approximating a switched mode power supply as a linear equivalent circuit within each switching interval, the entire period of the converter can be modeled using discrete time state-space equations,

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

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

where $\boldsymbol{A}_i$, $\boldsymbol{B}_i$, and $\boldsymbol{u}$ represent the state matrix, input matrix, and constant inputs respectively. During any switching interval i the state vector can be solved for any time t within the interval if an initial condition $\boldsymbol{x}(0)$ is known, $\boldsymbol{A}_i$ is invertable and all input sources are constant throughout the interval,

$$\boldsymbol{x}(t) = \mathrm{e}^{\boldsymbol{A}_i t}\boldsymbol{x}(0) + \boldsymbol{A}_i^{-1}[\mathrm{e}^{\boldsymbol{A}_i t} - \mathbf{I}]\boldsymbol{B}_i\boldsymbol{u}. \tag{3}$$

The dynamic switching behavior of the converter is captured by iterating through the linear equivalent switching intervals. Assuming the converter is operating in steady-state over the period,

$$\begin{aligned}\boldsymbol{X}_{ss} = {} & \left[\mathbf{I} - \prod_{i=m}^{1} \mathrm{e}^{\boldsymbol{A}_i t_i}\right]^{-1} \\ & \times \sum_{i=1}^{m}\left[\left(\prod_{k=m}^{i+1} \mathrm{e}^{\boldsymbol{A}_k t_k}\right)\boldsymbol{A}_i^{-1}[\mathrm{e}^{\boldsymbol{A}_i t_i} - \mathbf{I}]\boldsymbol{B}_i\boldsymbol{u}\right],\end{aligned} \tag{4}$$

solves the set of linear equivalent circuits over the entire period to calculate the steady-state solution.

Utilizing (4) to solve for the steady-state of a switch mode power supply does have limitations discussed in prior work [6], [8]. Some conditions such as requiring a $\boldsymbol{A}_i$ be invertible and $\boldsymbol{u}$ constant throughout the period can be quickly alleviated. However, in order to solve (4) the switch timing $\boldsymbol{t}_i$ between each linear equivalent circuit must be known a priori and are discussed in the following sections.

Some limitations to discrete time state-space modeling have been addressed in literature, such as dependent switching due to time interval $\boldsymbol{t}_i$ being set before solving for steady-state of the converter [8]. The common issue resulting from this limitation is inaccurate diode conduction timing. The framework presented in [8] to solve this issue iterate individually through each diode violation and time interval found in the converter.

Discrete time state-space modeling allows for rapid analysis of switch mode power supplies without the need for additional approximation or dedicated analysis of each topology. With an accelerated modeling method, brute force and metaheuristic methods execute faster; however, there are inherent techniques used in discrete time state-space modeling that can be applied to optimization [5]. A general approach to this is shown in Fig. 3. Due to the speed of discrete time state-space modeling, it is feasible to run a multi-loop optimization without sacrificing breadth of analysis. In the inner loop, the solution is iterated to correctly account for all nonlinear state-dependencies; in the outer loop, the hardware design or modulation is altered to optimize the performance.

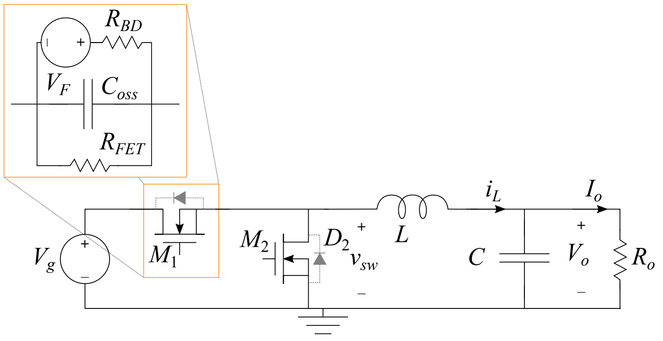


Fig. 2. Synchronous buck converter schematic, including switching transistor piecewise-linear circuit models. R_{BD} and R_{FET} are varied depending on the conduction state of the transistor.

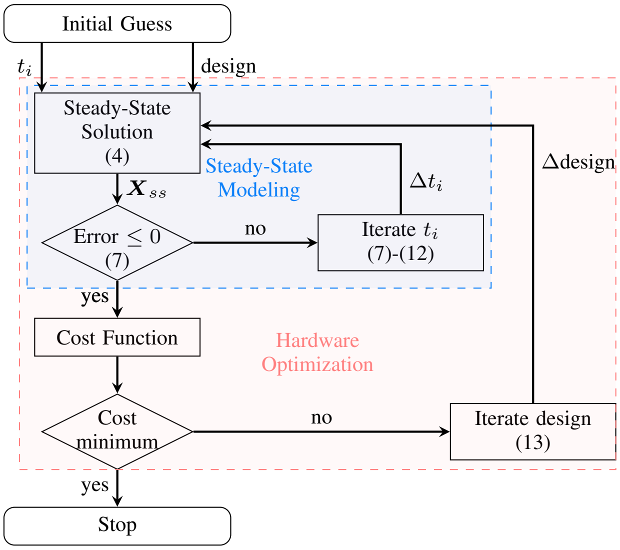


Fig. 3. Flowchart of schematic-level design optimization. In an inner loop, the discrete time steady-state model is iterated to find a valid converter solution. In an outer loop, the design (including e.g. component selection, switching frequency) is iterated to optimize an objective function.

A. State-Dependent Switching

Fig. 4 shows an example steady-state solution of the synchronous Buck converter of Fig. 2. The steady-state is solved using (4), then expanded to full waveforms for visualization. Corresponding state models and conducting devices are given in Table I for each interval as labeled at the bottom of Fig. 4.

In the initial solution, the deadtimes, t_1 and t_3, are set substantially longer than the switching transients for the circuit design and operation. This may occur from an inaccurate initial guess to the circuit solution, or may model the real, programmed deadtime from the transistor modulation signals in a design. In any event, the initial solution defines the externally-controlled switching actions, i.e. the gate-controlled transistor turn-on and turn-off, but does not define the state-dependent switching actions, i.e. the turn-on and turn-off of the body diodes.

In the steady-state solution, the long deadtimes allow the switched-node voltage v_{sw} to resonate below 0 V during subintervals 1 and 3, where the circuit model assumes all semiconductors are in their off states. This initial solution is therefore errant, as the body diodes (or reverse conduction mechanism for GaN HEMTs) would turn on and disallow this voltage from resonating negative. In order to correct this error and achieve a valid steady-state solution, the model must alter both the circuit model and time intervals as shown in the final solution in both Fig. 4 and Table I. The correct steady-state solution inserts circuit model $\boldsymbol{A}_{1,2} = \boldsymbol{A}_{3,2}$ for the circuit during the deadtime with both transistors turned off, but the low-side body diode forward-biased. Additionally, intervals t_1 and t_3 are split into two subintervals, $t_i = t_{i,1} + t_{i,2}$, with the same total duration, as shown in greater detail for t_1 in Fig. 4(b). More generally, the intervals may be split into more than two subintervals based on any state-dependent switching action,

$$t_i = t_{i,1} + t_{i,2} + \ldots + t_{i,N} \ . \tag{5}$$

The total duration of the subintervals must remain constant, as the end of the deadtime is controlled by external signals and not, in general, affected by diode conduction dynamics.

Note from Fig. 4 that the solution for $t_{1,1}$ is not the zero-crossing time of v_{sw} in the initial solution. Because both the initial and final solutions are given by (4), both are in steady-state, even if state-dependent switching is accounted for incorrectly. Thus, the additional negative volt-seconds applied to the inductor by the errant negative v_{sw} in the initial solution significantly alters the operating point, as shown by i_L and v_{out}.

Though the example presented in Fig. 4 ignores how the final solution is solved, it is clear that two steps must be completed, iteratively

1) Insert subintervals for state-dependent switching actions at required interval interfaces
2) Adjust subinterval durations to drive towards zero-error solution

In this work, a gradient descent method is developed which uses a numerical gradient between the subinterval duration and the steady-state error in state-dependent switching to address the second point.

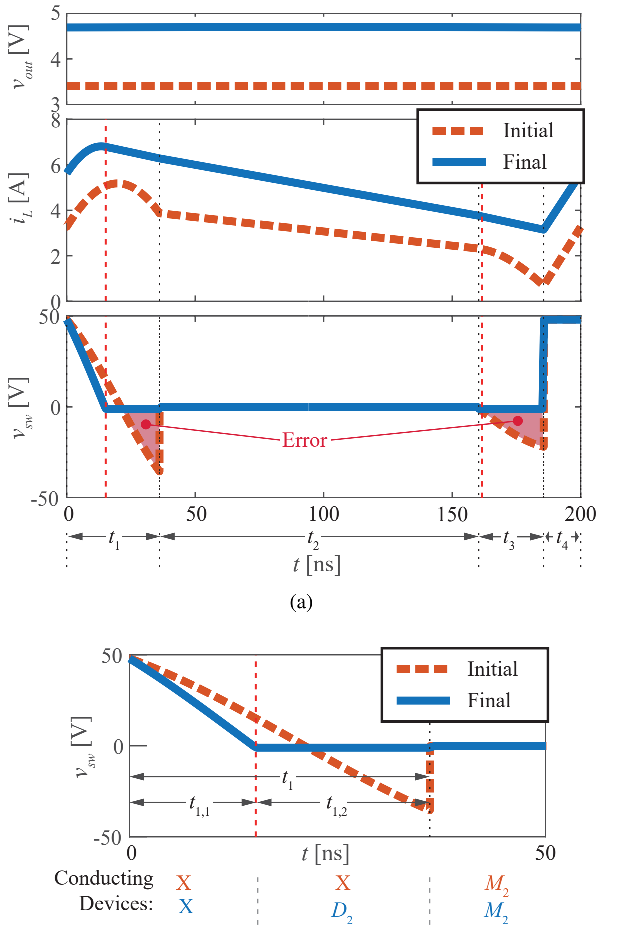
III. Steady-State Solution

For the example of diode state-dependent switching, error in the steady-state solution is identified by violation of one of the pair of inequalities,

$$\begin{cases} v_d(t) & \leq V_F, \quad\quad i \text{ s.t. diode is reverse biased} \\ i_d(t) & \geq 0, \quad\quad\ \ i \text{ s.t. diode is forward biased} \end{cases} \tag{6}$$

These constraints are both of the general form

$$\boldsymbol{E}_i = \boldsymbol{C}_{\boldsymbol{bnd}_i} \boldsymbol{X}_i + \boldsymbol{D}_{\boldsymbol{bnd}_i} \boldsymbol{u} \leq \boldsymbol{0} \tag{7}$$



(a)

(b)

Fig. 4. Example steady-state waveforms of synchronous buck converter (a) before and after state-dependent switching correction and (b) zoomed-in waveforms of initial dead time

TABLE I
Buck Converter Deadtime Solution

Inital Solution						
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_2	—		M_1
Final Solution						
Time Interval	t_1		t_2	t_3		t_4
Circuit Model	$\boldsymbol{A}_1$	$\boldsymbol{A}_{1,2}$	$\boldsymbol{A}_2$	$\boldsymbol{A}_3$	$\boldsymbol{A}_{3,2}$	$\boldsymbol{A}_4$
Conducting Devices	—	D_2	M_2	—	D_2	M_1

where $\boldsymbol{C}_{\boldsymbol{bnd}} \in \mathbb{R}^{p \times n_s}$ and $\boldsymbol{D}_{\boldsymbol{bnd}} \in \mathbb{R}^{p \times 1}$ with p the number of constraint equations in the circuit and n_s the number of states in the circuit. This approach formulates the error signals due to state-dependent switching as though they were outputs of the system.

In each switching interval i, only one of the constraints in

(6) will apply for each diode in the circuit, determined by whether the matrices $\boldsymbol{A}_i$ and $\boldsymbol{B}_i$ are derived with the diode forward or reverse-biased. For other types of state-dependent switching, there may be conditions which appear only in certain intervals without dual counterparts in the remaining intervals.

To reduce computational burden, the method used in this work initially examines the error only at the discrete points at the interface between intervals. The steady-state solution is solved using (4) for $\boldsymbol{X}_{ss} = \boldsymbol{X}_0$, then the states at the end of each subsequent interval are reconstructed using recursive application of (3),

$$\boldsymbol{X}_{i+1} = \mathrm{e}^{\boldsymbol{A}_i t_i}\boldsymbol{X}_i + \boldsymbol{A}_i^{-1}[\mathrm{e}^{\boldsymbol{A}_i t_i} - \mathbf{I}]\boldsymbol{B}_i\boldsymbol{u}. \tag{8}$$

or appropriate methods when $\boldsymbol{A}_i$ is singular [6]. If interval dynamics are sufficiently fast [8], it may be necessary to increase the resolution of discrete points to identify all errors, but this does not alter the approach and can be done as a final check once a valid solution has been solved based on the discrete points at the end of each interval.

A. Inserting Additional Subintervals

The error calculation (7) is applied to both the states at the beginning and end of each interval (8), resulting in two error signals for the i^{th} interval, $\boldsymbol{E}_i^-$ for the initial states and $\boldsymbol{E}_i^+$ for the final states, both with the same conducting devices.

In the initial solution of Fig. 4, both interval 1 and 3 have $\boldsymbol{E}_i^- \leq 0$ but $\boldsymbol{E}_i^+ \nleq 0$. This implies that the error occurs somewhere within the interval, and a valid solution may be found by inserting a new state at the end of the interval. The necessary diode which must change commutation state is found by examining which row of (7) is in violation, and which equation from (6) generated the constraint.

In this example, both require the insertion of a new subinterval with D_2 turned on. These subintervals, $\boldsymbol{A}_{1,2} = \boldsymbol{A}_{3,2}$, are initially inserted with minimal time duration $t_{1,2}$ and $t_{3,2}$ so that they only nominally affect the steady-state waveforms. After inserting these intervals, the following all hold simultaneously, at subinterval (i, j), which is defined by the state-dependent turn-on of D_2

$$\begin{cases} \boldsymbol{E}_{i,j}^- \leq 0 \\ \boldsymbol{E}_{i,j}^+ \nleq 0 \\ \boldsymbol{E}_{i,j+1}^- \leq 0 \end{cases} \tag{9}$$

This situation implies that by increasing $t_{i,j+1}$, subject to (5), the error can be eliminated. From the previous example, this corresponds to the dead times with body diode turned on for an instant at the end of the dead time. The remaining task is to increase the duration of body diode conduction, from the end of the interval back, until the diode properly turns on as the diode voltage increases to $v_d(t) = V_F$.

In more complex topologies, additional situations may arise, including persistent error across multiple intervals. In any event, the appropriate candidate states to be added are selected by examining $\boldsymbol{E}_{i,j}^{\pm}$

B. Adjusting Subinterval Durations

Once the correct subintervals have been inserted (or a best guess at the appropriate subintervals based on the current steady-state solutions), the subinterval time durations are adjusted using a gradient-based minimization of the signals in violation of (7).

A Jacobian is calculated for all time intervals of the converter to determine each time intervals impact on the steady-state as well as each steady-state violation.

$$\boldsymbol{J} = \begin{bmatrix} \frac{\partial X_1(t_1)}{\partial t_1} & \frac{\partial X_1(t_2)}{\partial t_1} & \cdots & \frac{\partial X_1(t_n)}{\partial t_1} \\ \frac{\partial X_2(t_1)}{\partial t_1} & \frac{\partial X_2(t_2)}{\partial t_1} & \cdots & \frac{\partial X_2(t_n)}{\partial t_1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial X_m(t_1)}{\partial t_1} & \frac{\partial X_m(t_2)}{\partial t_1} & \cdots & \frac{\partial X_m(t_n)}{\partial t_1} \end{bmatrix}$$

where $\boldsymbol{J} \in \mathbb{R}^{n_s \times m \times m}$ for a system with n_s states and $m = m_i + m_j$ subintervals. Entry j_{ijk} is the response of the i^{th} state, at the j^{th} subinterval, to a perturbation in the duration of the k^{th} subinterval time.

$\boldsymbol{J}$ is evaluated numerically for small perturbations to each time interval. To construct the complete matrix, the steady-state solution (4) and discrete time reconstruction (8) need to be repeated m times. Because these equations are all closed-form, doing so requires relatively little computation burden.

By multiplying every j^{th} column by $\boldsymbol{C}_{bnd,j}$ the resulting matrix $\boldsymbol{J}_{err} \in \mathbb{R}^{p \times m \times m}$ has elements $j_{err,ijk}$ that represent the linearized response of the p^{th} constraint equation (7) to the same perturbation and time.

In order to eliminate the errors present in any iteration of the steady-state solution, $\boldsymbol{J}_{err}$ is evaluated only for the states and times where error with state-dependent switching is present in the steady-state solution. This is combined with equations of the form of (5) for every subdivided interval from the controlled switching pattern to give

$$\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{J}_{err,ij}^{\pm} \\ \boldsymbol{1_T} \end{bmatrix}, \tag{10}$$

$$\boldsymbol{\Gamma} = \begin{bmatrix} \boldsymbol{D}_{bnd,ij}^{\pm}\boldsymbol{u} \\ \boldsymbol{t_i} \end{bmatrix} \tag{11}$$

where $\boldsymbol{1_T}$ is a binary matrix indicating which timing subintervals are part of each controlled interval in $\boldsymbol{t_i}$. The required change to each time subinterval $\Delta\boldsymbol{t_i}$, based on the linearized steady-state Jacobian, is then given by the solution to

$$\boldsymbol{\Phi}(\Delta\boldsymbol{t_i}) + \boldsymbol{\Gamma} = 0 \tag{12}$$

where the equation is an equality, rather than inequality, because only the errant signals are include which must be driven back to zero from currently violating (7).

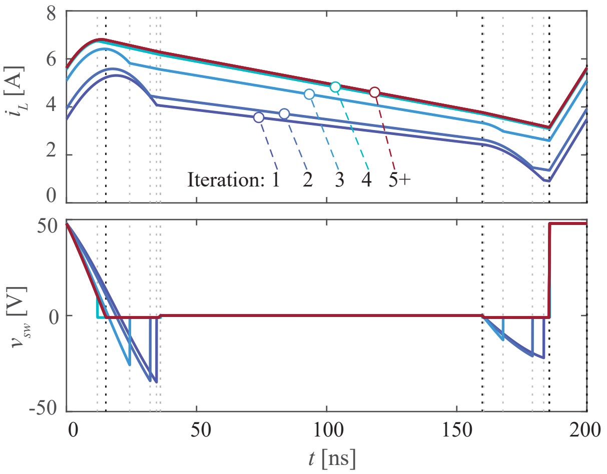


Fig. 5. Waveforms showing convergence of state-dependent switching due to deadtime body diode conduction in the synchronous buck converter

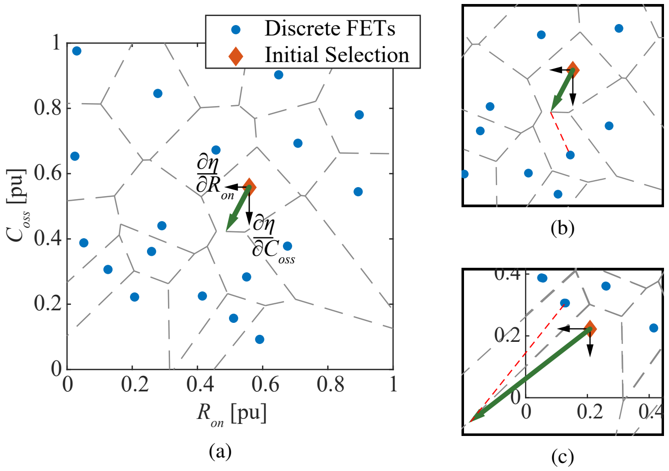


Fig. 6. Example iterative transistor selection from discrete devices. Individual FETs are shown with a voronoi diagram (dashed gray lines) showing the regions closest to each discrete device. Cases of selecting subsequent iteration are shown in (b) and (c)

In practice, $\boldsymbol{\Delta t_i}$ must then be checked to ensure no time intervals are driven below zero. If the solution attempts to do so, subintervals inserted in the prior section may need to be removed, and the algorithm should iterate back and forth between the subinterval insertion and time adjustment steps.

Fig. 5 plots i_L and v_{sw} over the seven iterations required to reach a valid steady-state solution. On the fourth iteration, the diode conduction time during t_1 overshoots the final solution slightly. Iterations 5-7 are nearly indistinguishable from the final solution at the plotted resolution. The process takes less than 100 ms[†] when complete waveforms are not plotted at each iteration.

IV. Optimization Method

Following a similar approach to the gradient-based error-minimization used to solve steady-state waveforms, additional numerical gradient computations are used to optimize a schematic-level converter implementation. The objective function for this optimization depends on the intended application, but in addition to parameters such as component size/weight and cost, will usually include some accounting for converter efficiency or power loss. Using the method detailed in [3], average input and output power are simple to solve using the steady-state solution $\boldsymbol{X}_ss$ and applying (8) to the augmented system to compute average values of input and output waveforms.

For continuous parameters including switching frequency (or period), the application is straightforward. After solving the objective function at an initial design, the switching frequency is perturbed and the objective function resolved to find a numerical gradient that is descended to optimize the circuit operation.

Hardware optimization, in which we may be required to select between discrete components, requires additional consideration. Fig. 6 gives an example design space for transistor selection for the synchronous buck converter examined previously. Based on the model of Fig. 2, transistors are characterized by their on-resistance R_{on} and output capacitance C_{oss} with only discrete points available as valid selections.

Though only the discrete points are feasible designs, the optimization can nonetheless treat the design space as continuous in the R_{on}-C_{oss} plane, compute numerical gradients, then revise the next selected point to conform to the feasible design points. Fig. 6(a) shows this process for an assumed objective to maximize efficiency, η. Using small perturbations to each parameter and re-solving steady-state operation, the optimization finds numerical approximations to $\partial\eta/\partial R_{on}$ and $\partial\eta/\partial C_{oss}$ and combines them for a net vector of steepest descent. In Fig. 6(b), the steepest descent vector is extended until it crosses from the current region into a new region closer to an alternate discrete point. Fig. 6(c) gives a second example on the boundary of the points.

To select a discrete point for a subsequent iteration, the optimization computes the minimum distance d that must be traveled along the vector of steepest descent before each point will be closer than the original. Generally, for an initial point $p_0 = (p_{0,x}, p_{0,y})$ and any other point p_i, the distance is

$$d_i = \frac{1}{2}\frac{(p_{0,x}-p_{i,x})^2 + (p_{0,y}-p_{i,y})^2}{(p_{0,x}-p_{i,x})\frac{\partial\eta}{\partial x} + (p_{0,y}-p_{i,y})\frac{\partial\eta}{\partial y}} . \qquad (13)$$

Then, the best selection for the next iteration of the optimization is the point with minimum nonnegative d_i.

This approach is again applied to a simplified synchronous buck converter for demonstration. The buck converts 48 V input to 5 V output, has an ideal 250 nH inductor, and switching frequency is optimized per-design to maximize efficiency. Fig. 7 shows the precomputed efficiency and optimal switching frequency for a range of candidate discrete transistors. For this conceptual example, the transistors are randomly generated near-to, but bounded by, a figure of merit $R_{on}C_{oss} \leq 2$ mΩnF.

[†] on a Intel Core i7-8700 running Matlab R2022b

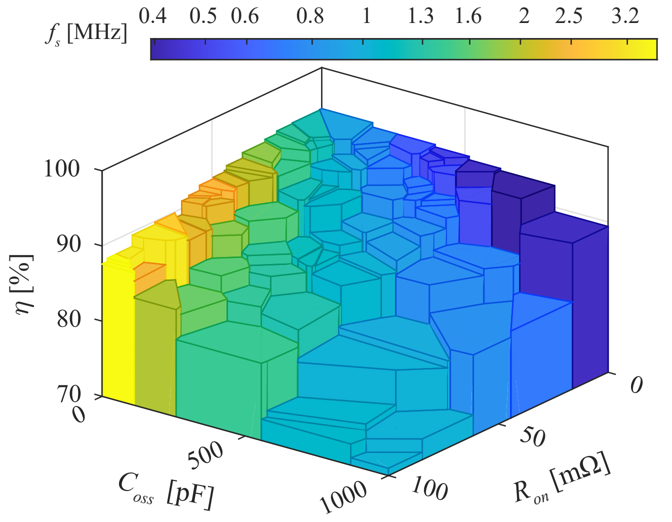


Fig. 7. Precomputed maximum efficiency over f_s and transistor selection for a synchronous buck converter with zero dead time

Initially, dead times are set to zero, forcing non-ZVS operation of each switching action and making it feasible to run the optimization without an inner iterative loop solving steady state; no state-dependent switching actions are present. Under this condition, the precomputed maximum efficiency is 94.54% at a frequency of 917 kHz.

From an initial point, steady-state efficiency under perturbations to R_{on} and C_{oss} are solved and (13) is used to compute the discrete FET selection for the following iteration. The gradient of efficiency with respect to switching frequency is also calculated, and on the updated switching frequency is scaled proportional to this gradient and the discrete step size, d_i.

Fig. 8 diagrams the iterative convergence to near-optimal designs from three initial points, overlaid on the precomputed values of Fig. 7. Though the plot overlays the convergence path on the optimal efficiencies for clarity, the iterations align with these efficiencies only near the final convergence. Due to non-optimal switching frequencies given at the initial operating points, the initial efficiencies are 21%, 33%, and 73% for the three starting points. Nonetheless, all three converge to designs with over 94.45% efficiency, with minor variations between them.

Next, both the steady-state algorithm of Section III and the optimization formulation in this section are combined by setting nonzero deadtime for the buck converter. For each design, deadtimes are limited to one-quarter of the resonant frequency between L and C_{oss} which allows critical ZVS but prevents nonmonotonic convergence [8].

Rather than the proportional step in switching frequency used previously, the switching frequency is optimized to convergence at each iteration. Though this requires additional computation, it was found to be necessary. With variable deadtime and switching frequency, the optimization will converge towards designs with critical ZVS of both transistors.

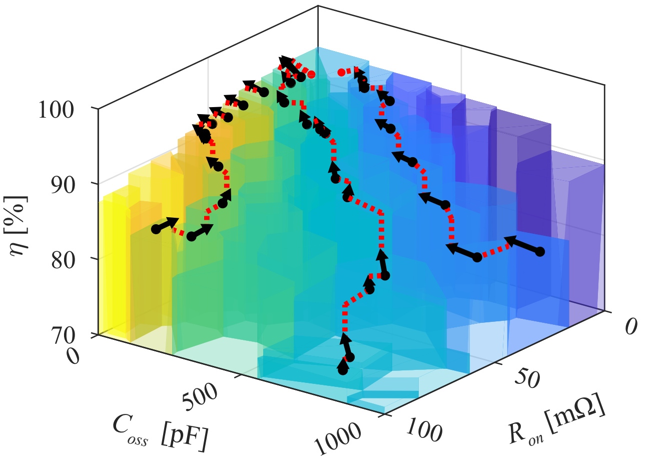


Fig. 8. Paths of iterative convergence to maximum efficiency designs for three starting point overlaid on Fig. 7. Formatting of path vectors follows that shown in Fig. 6(b)-(c).

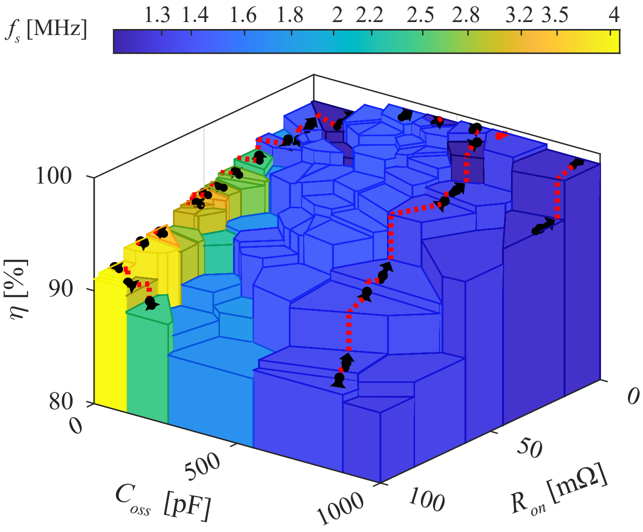


Fig. 9. Paths of iterative convergence to maximum efficiency designs for three starting point for the synchronous buck converter with dead time considered.

Perturbing only one of f_s or C_{oss} away from these design points will predict rapid reduction in efficiency, even if a simultaneous perturbation can increase it.

Due to the lack of other losses included in this example (e.g. inductor loss), the optimal designs exhibit very high efficiency under ZVS operation, with the maximum precomputed design have $\eta = 99.5\%$ efficiency at $f_s = 1.3$ MHz. As shown in Fig. 9, the presence of ZVS shifts optimal devices to the lowest R_{on} transistors.

The same three starting points are considered, and again all three rapidly converge to designs above 99% efficiency with the optimal device.

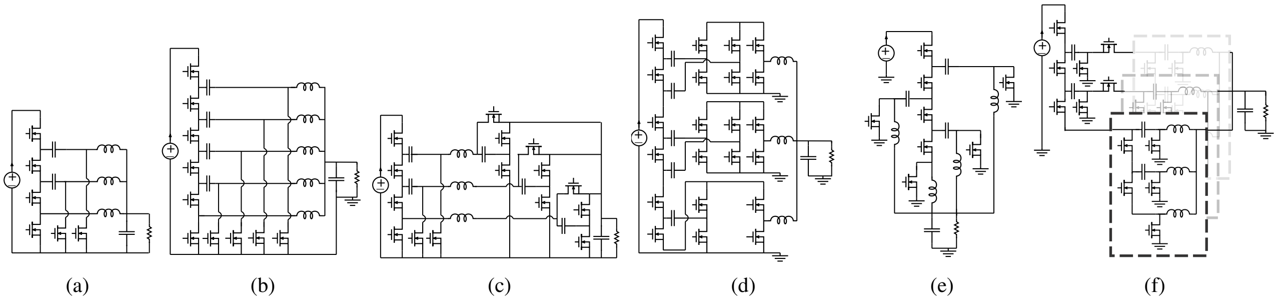


Fig. 10. Topologies included in example 48-to-1 V optimization. 3-phase (a) and 5-phase (b) series capacitor buck [12], series capacitor buck with switched-capacitor step-down [13] (c) LEGO-POL [14] (d) Multi-Inductor Hybrid [15](e) and Dickson^2 [16]

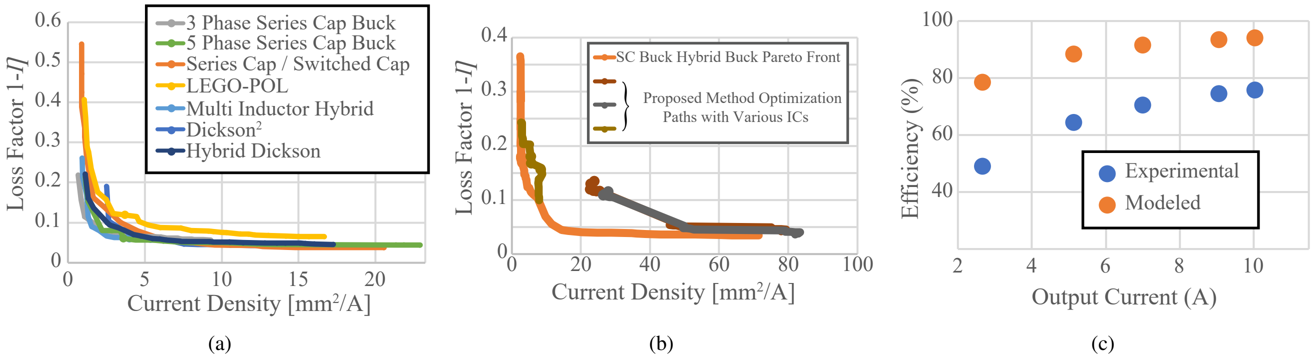


Fig. 11. GA-based pareto front for all topologies (a), convergence from multiple initial conditions of the proposed method when optimizing for efficiency only (b) and measured-vs-modeled efficiency of the prototype up to $I_{out} = 10$ A (c)

TABLE II
SERIES CAPACITOR BUCK HYBRID BUCK CONVERTER PARAMETERS

L	0.251 µH	$M_{1,2,3}$	EPC2044
R_L	0.31 mΩ	$M_{4,5,6}$	EPC2088
$C_{1,2}$	2x10 µF	M_{7-15}	EPC2055
$C_{3,4,5}$	2x47 µF	f_s	500 MHz
V_g	48 V		

V. EXAMPLE APPLICATION

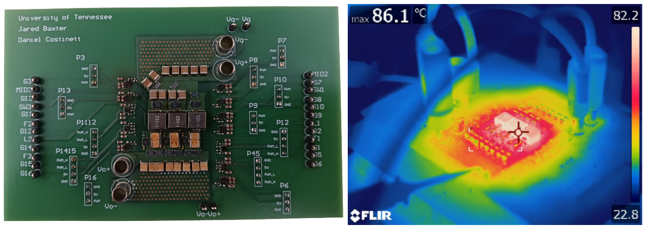
A 48-1 V converter was designed to verify the proposed method. The optimization database included a set of converters from literature [12], [14]–[16] shown in Fig. 10. A database of 20 EPC GaN devices modeled approximately by R_{on} and C_{oss}, 9 Coilcraft and 7 ICE Components discrete inductors modeled approximately by $R_{L,dc}$ and L was used to select implementation of each component. Additionally, the number of components in parallel, switching frequency, and output current were taken as variables for the optimization.

Fig. 11 shows the results of the optimization. In Fig. 11(a), a genetic algorithm is used to sweep a pareto front comprised by the efficiency and current density (assessed using the total area of power stage transistors and passives) for each topology. In Fig. 11(b), the proposed gradient-based optimization is used to separately select each transistor, inductor, and the switching frequency. In this case, only efficiency is used as the design objective, resulting in high-efficiency but large-footprint designs.

A prototype using the series capacitor buck with switched capacitor step-down topology is fabricated to test the modeling. The prototype uses the components listed in Table II which was selected from the Pareto front with a selection bias towards high efficiency designs without paralleled transistors. The prototype is designed for $I_{out} = 40$ A. Testing results up to $I_{out} = 10$ A are shown in Table II(c) and an image of the PCB along with thermal performance at 10 A are given in Fig. 12. From the thermal images and the comparison between modeled and measured efficiency, the prototype appears to exhibit about 2 W of additional ac loss in the inductor, which was not modeled. Nonetheless, the trends in efficiency match fairly well neglecting the constant offset in power loss.

VI. CONCLUSION

This work proposes a method to optimize power converters by leveraging discrete time state-space modeling. Techniques to accelerate both the convergence to a valid steady-state solution in the presence of state-dependent switching and hardware optimization via component selection and modulation design are developed. The techniques allow the accurate and rapid discrete time modeling to be used for broad early-stage converter design optimization.



(a) (b)

Fig. 12. Prototype PCB photograph (a) and thermal image at $I_{out} = 10$ A (b)

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] 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.

[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] 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.

[4] 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.

[5] J. A. Baxter and D. J. Costinett, "Broad-scale converter optimization utilizing discrete time state-space modeling," in *2022 IEEE Design Methodologies Conference (DMC)*, 2022, pp. 1–6.

[6] 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.

[7] 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.

[8] J. A. Baxter and D. J. Costinett, "Steady-state convergence of discrete time state-space modeling with state-dependent switching," in *2020 IEEE 21st Workshop on Control and Modeling for Power Electronics (COMPEL)*, Nov. 2020, pp. 1–8.

[9] 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.

[10] 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.

[11] 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.

[12] P. S. Shenoy, M. Amaro, J. Morroni, and D. Freeman, "Comparison of a buck converter and a series capacitor buck converter for high-frequency, high-conversion-ratio voltage regulators," *IEEE Transactions on Power Electronics*, vol. 31, no. 10, pp. 7006–7015, 2016.

[13] G. Gabian, J. Gamble, B. Blalock, and D. Costinett, "Hybrid buck converter optimization and comparison for smart phone integrated battery chargers," in *2018 IEEE Applied Power Electronics Conference and Exposition (APEC)*, 2018, pp. 2148–2154.

[14] J. Baek, Y. Elasser, K. Radhakrishnan, H. Gan, J. P. Douglas, H. K. Krishnamurthy, X. Li, S. Jiang, C. R. Sullivan, and M. Chen, "Vertical stacked lego-pol cpu voltage regulator," *IEEE Transactions on Power Electronics*, vol. 37, no. 6, pp. 6305–6322, 2022.

[15] R. Das and H.-P. Le, "A regulated 48v-to-1v/100a 90.9converter for pol applications in data centers and telecommunication systems," in *2019 IEEE Applied Power Electronics Conference and Exposition (APEC)*, 2019, pp. 1997–2001.

[16] Y. Zhu, T. Ge, Z. Ye, and R. C. Pilawa-Podgurski, "A dickson-squared hybrid switched-capacitor converter for direct 48 v to point-of-load conversion," in *2022 IEEE Applied Power Electronics Conference and Exposition (APEC)*, 2022, pp. 1272–1278.