EFFICIENT AND ROBUST SIMULATION, MODELING AND CHARACTERIZATION OF IC POWER DELIVERY CIRCUITS

A Dissertation

by

YA WANG

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Peng Li
Committee Members,	Weiping Shi
	Prasad Enjeti
	Duncan M. Walker
Head of Department,	Miroslav M. Begovic

December 2017

Major Subject: Computer Engineering

ABSTRACT

As the Moore's Law continues to drive IC technology, power delivery has become one of the most difficult design challenges. Two of the major components in power delivery are DC-DC converters and power distribution networks, both of which are time-consuming to simulate and characterize using traditional approaches. In this dissertation, we propose a complete set of solutions to efficiently analyze DC-DC converters and power distribution networks by finding a perfect balance between efficiency and accuracy.

To tackle the problem, we first present a novel envelope following method based on a numerically robust time-delayed phase condition to track the envelopes of circuit states under a varying switching frequency. By adopting three fast simulation techniques, our proposed method achieves higher speedup without comprising the accuracy of the results. The robustness and efficiency of the proposed method are demonstrated using several DC-DC converter and oscillator circuits modeled using the industrial standard BSIM4 transistor models. A significant runtime speedup of up to 30X with respect to the conventional transient analysis is achieved for several DC-DC converters with strong nonlinear switching characteristics.

We then take another approach, average modeling, to enhance the efficiency of analyzing DC-DC converters. We proposed a multi-harmonic model that not only predicts the DC response but also captures the harmonics of arbitrary degrees. The proposed full-order model retains the inductor current as a state variable and accurately captures the circuit dynamics even in the transient state. Furthermore, by continuously monitoring state variables, our model seamlessly transitions between continuous conduction mode and discontinuous conduction mode. The proposed model, when tested with a system decoupling technique, obtains up to 10X runtime speedups over transistor-level simulations with a maximum output voltage error that never exceeds 4%.

Based on the multi-harmonic averaged model, we further developed the small-signal model that provides a complete characterization of both DC averages and higher-order harmonic responses. The proposed model captures important high-frequency overshoots and undershoots of the converter response, which are otherwise unaccounted for by the existing techniques. In two converter examples, the proposed model corrects the misleading results of the existing models by providing the truthful characterization of the overall converter AC response and offers important guidance for converter design and closed-loop control.

To address the problem of time-consuming simulation of power distribution networks, we present a partition-based iterative method by integrating block-Jacobi method with support graph method. The former enjoys the ease of parallelization, however, lacks a direct control of the numerical properties of the produced partitions. In contrast, the latter operates on the maximum spanning tree of the circuit graph, which is optimized for fast numerical convergence, but is bottlenecked by its difficulty of parallelization. In our proposed method, the circuit partitioning is guided by the maximum spanning tree of the underlying circuit graph, offering essential guidance for achieving fast convergence. The resulting block-Jacobi-like preconditioner maximizes the numerical benefit inherited from support graph theory while lending itself to straightforward parallelization as a partition-based method. The experimental results on IBM power grid suite and synthetic power grid benchmarks show that our proposed method speeds up the DC simulation by up to 11.5X over a state-of-the-art direct solver.

DEDICATION

To my mother, my father and my wife.

ACKNOWLEDGMENTS

First I would like to express my greatest gratitude for my advisor, Prof. Peng Li, for continuously providing support and help for the past five years. With his profound experience in circuit simulation, he introduced me to the frontier of electronic design automation research and offered many valuable insights to my research work. His diligence and perseverance have set the standard for me and keep driving me to reach for higher goals. I am truly thankful for having Prof. Peng Li as my Ph.D. advisor. This dissertation would not have been possible without him.

I would like to thank Dr. Weiping Shi, Dr. Prasad Enjeti and Dr. Hank Walker for serving on my Ph.D. committee and providing insightful comments and suggestions. I have greatly benefited from their feedbacks on my preliminary exam.

I would also like to thank my colleagues and friends in Computer Engineering & System Group. Di Gao and Wenrui Zhang have been truly reliable teammates while we are working on the same research projects. Dr. Yong Zhang, Dr. Qian Wang, Dr. Honghuang Lin, Jimmy Jin and Xin Zhan have all offered tremendous help by providing inspiring technical discussions.

Finally, I would like to thank my parents, who have always put their deepest trust in me. They encouraged me to study abroad for a Ph.D. degree and it turns out to be the best decision I ever made. Their selfless support and unconditional love make it possible for me to see beyond the horizons. I would also like to thank my wife, Sharon, for her continuous support. The past four years together at Texas A&M University will become my most cherished memories. Thanks for always being there for me and giving me a purpose to fight for.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a dissertation committee consisting of Professor Peng Li, Professor Weiping Shi and Professor Prasad Enjeti of the Department of Electrical and Computer Engineering and Professor Duncan Walker of the Department of Computer Science and Engineering.

All work conducted for the dissertation was completed by the student independently.

Funding Sources

This work was made possible in part by Semiconductor Research Corporation (SRC) through Texas Analog Center of Excellence at the University of Texas at Dallas under Grant Number 1836.115.

Its contents are solely the responsibility of the authors and do not necessarily represent the official views of the Semiconductor Research Corporation.

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

1. INTRODUCTION AND LITERATURE REVIEW

Power delivery has become one of the most important aspects of today's integrated circuit (IC) design. As shown in Fig. 1.1, there are two major components that process and deliver the power from an external power source to the individual functional circuit on the chip. The first component is the DC-DC converters, which step up/down the external power supply voltage to desired voltage levels for different functional circuits. The second component is the power distribution networks (PDN), which are large circuit networks that deliver power to each transistor. Both components are critical in delivering stable and robust voltages to functional circuits and may lead to catastrophic consequences when either component is designed incorrectly.

Nowadays, power delivery design circuit is becoming more and more challenging. As shown by International Technology Roadmap for Semiconductors (ITRS) in Fig. 1.2, two significant trends in power supply are observed. First of all, as the supply voltage keeps scaling down to reduce power consumption, the functionalities of the on-chip circuits are more vulnerable to supply voltage noise and IR drops on power distribution networks. Second, with more functional circuits being integrated into the chip, the load current continues to increase, which causes the DC-DC converter being more susceptive to output voltage drop in fast load transient and heavy load conditions. Overall, the situations demand better power delivery design methods that provide efficient and accurate analysis of DC-DC converters and power distribution networks.

While it is important to efficiently analyze, simulate and model DC-DC converters and power distribution networks, finding the proper methods is proved to be very challenging [2, 3]. Analyzing both circuits requires carefully designed algorithms to properly extract desired information from a large amount of data in an efficient manner. In this dissertation,

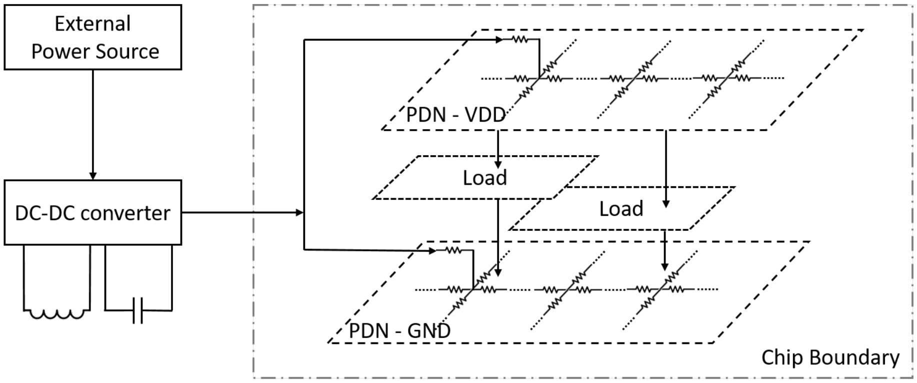


Figure 1.1: A power delivery system consists of an external power source, a DC-DC converter and power distribution networks (PDN).

we propose a set of solutions to efficiently analyze DC-DC converters and power distribution networks. At the core of the solutions, the philosophy of tackling the problem is the efficiency and accuracy trade-off. While it is time-consuming and usually not cost-efficient to process all simulation data and obtain exact solutions, it is desirable to neglect redundant information and only extract relevant results about the circuit behavior. For DC-DC converter, we propose to exploit envelope following and circuit averaging techniques to efficiently obtain circuit responses. The envelope following methods achieves efficiency by tracing the slowly varying envelope of the circuit and skipping many fast-changing switching cycles in between [4, 5]. The circuit averaging methods builds averaged behavior models of the periodic circuits and ignore the transient within one cycle. For power distribution networks, instead of using a direct solution like Cholesky factorization, we take a divide-and-conquer approach by partitioning the network and build an iterative solution using the result from each partition [6, 3].

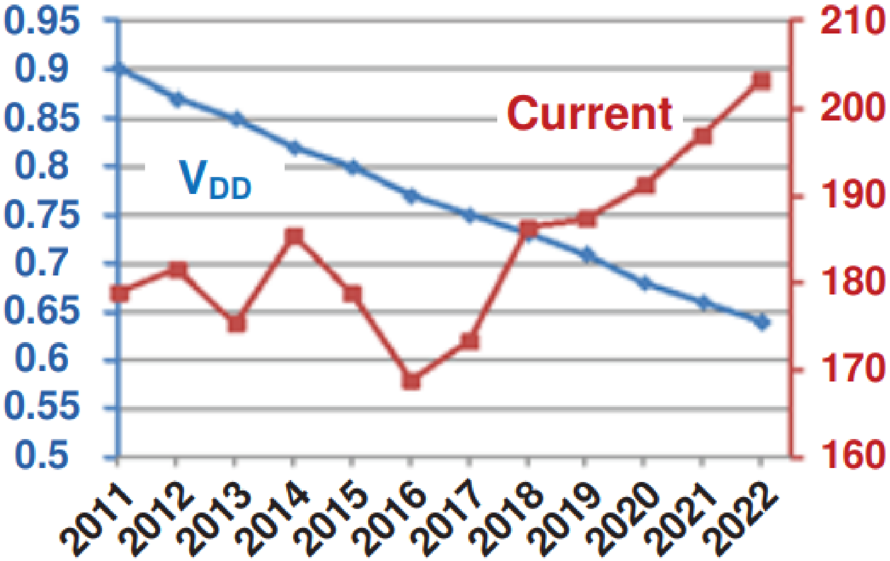


Figure 1.2: Power Supply trends. Reprinted from [1].

1.1 Efficient Simulation of DC-DC converters

DC-DC converters, due to their modulation mechanism, demonstrates multi-rate characteristics and leads traditional transient analysis method to severe over-sampling problem [4]. While the analysis has to take extremely small steps to capture the switching activities, it also needs to cover a long period of time due to slow load variations. The amount of data accumulated during the simulation is large and thus the efficiency of the traditional analysis methods degrades.

The time-domain envelope following (EF) method is well suited for the simulation of such circuits with a multi-rate characteristic [2, 4, 5]. The efficiency of envelope following stems from the fact that it efficiently traces the slowly varying envelope of the circuit by skipping many fast-changing switching cycles in between. An EF method is introduced in [2] to simulate open-loop switching power converters with a fixed clock frequency and the general difficulty in simulating closed-loop switching converters is discussed. [5] extends this method for closed-loop converters and the problem of quasi-algebraic variables is addressed. [7] introduce a method that exploits the parallelism in the envelope-following method and parallelize the Newton update solving part to boost the simulation performance. In [8] quadratic and exponential approximations of the envelope are used in EF simulation. However, only fixed-frequency PWM converters are targeted. To analyze variable-frequency converters, [9] approximates variations of the switching period by assuming that the envelope stepsize is an integer multiple of the last switching period of each EF step, which is not true in general. Another envelope following method for closed-loop converters is proposed in [10] for a specific type of converters with multiple switching intervals with a fixed clock (switching) period. The key limitation of [9] and [10] is that the entire converter is treated simplistically as a linear switched network with each switching interval modeled using a linear state transition function.

In this dissertation, we propose an enhanced envelope following method that efficiently simulates DC-DC converters operating with complex modulation controls, *e.g.*, pulse width modulation (PWM), pulse frequency modulation (PFM) or pulse skipping modulation (PSM). At the core of our new envelope following algorithm is a novel time-delayed phase condition that provides robust tracking of the circuit envelope in the transient phase, under a varying switching frequency. We further develop a mechanism that can smoothly track the transitions between the transient and steady-state phases, thereby providing a unifying solution to the EF simulation of both modes of operation. The proposed method is verified using several DC-DC converters and demonstrated improved efficiency compared with traditional transient analysis.

1.2 Multi-harmonic Modeling of DC-DC converter

While envelope following method reduces the simulation time by only capturing the envelope of the responses and skipping multiple switching cycles, the average modeling techniques achieve efficiency by replacing the circuit with a model that only capture averaged behaviors between cycles the ignores the details in between. In particular, state-space averaging has been a very popular simulation technique of pulse-width modulated (PWM) DC-DC converters. In [11, 12, 13], both large-signal and small-signal state-space average models for DC-DC converters operating in continuous conduction mode (CCM) and Discontinuous Conduction Mode (DCM) are presented. However, without considering non-ideal device characteristics, [12] and [13] are inaccurate in simulating low-power DC-DC converters. Furthermore, the modeling approach of [12] and [13] is circuit-specific, thereby limiting the ability to automate the modeling and simulation process on arbitrary DC-DC converter topologies.

Alternatively, the PWM switch model [14, 15], which is a linearization of the three-terminal switch cell, can be readily applied to a wide range of DC-DC converters. By tracking the circuit operating mode based on averaged circuit states, [16] enhances the basic PWM switch model to support simulation of DC-DC converters operating in both CCM and DCM. However, a major limitation of the PWM switch model is that it neglects the dynamic behavior of DC-DC converters within one cycle and provides no information about the waveform ripples. As suggested in [17], the PWM switch model has an underlying assumption of the small ripple condition, which prevents it from being applied to converters with large ripples. It has been shown that neglecting the ripples of state variables can lead to large discrepancies in the simulation results of converters operating at low frequencies [18].

To address this problem, a lot of work has been done on the generalized averaging techniques [19]. In [18], a multi-frequency averaged model is introduced which conducts frequency selective averaging on the switched state-space models of DC-DC converters. However, this method is based on boost converters and cannot be easily applied to other converter configurations. In [20], a flexible method of in-place averaging that replaces elements in DC-DC converters with the k^{th}-index average elements is presented, thereby allowing for the tracking of responses with harmonics up to k^{th} degree. However, the method in [20] assumes a continuous inductor current with ideal switches. As a result, such a model becomes neither suitable nor accurate enough for low-power DC-DC converter simulations.

To solve the problem, we propose a multi-harmonic averaged modeling method of DC-DC converters that combines the accuracy of enhanced state-space averaging, the flexibility of PWM switch models and the multi-frequency nature of the generalized models. The proposed model, which is based on the switch cell, can be readily applied to various types of DC-DC converters. It approximates the converter response using multiple harmonics up to an arbitrary degree. We demonstrate the efficiency and generality of the proposed method using different DC-DC converters.

1.3 Small Signal Modeling of DC-DC Converters

Both envelope following method and averaged modeling method provide paths for efficiently analyzing the large-signal behavior of DC-DC converters. As the complexity of the modern low-power integrated circuits grows exponentially, there is an increasing need for accurate control techniques in designing DC-DC converters [21]. Converter circuit behavior is typically highly nonlinear due to the strong switching activities and the presence of nonlinear devices. The small-signal model, which approximates the behavior of the DC-DC converter by linearizing the nonlinear devices and switches at a certain DC operating point, is widely used by designers to design the control blocks and closed-loop systems, as well as to analyze system stability.

Typically, a small-signal model is obtained by first deriving the averaged model of the DC-DC converter, then injecting perturbations to the averaged model through the control signal/supply voltage, followed by evaluating the sensitivity of the circuit states. For low-power DC-DC converters, it is critically important to capture the high-frequency circuit responses in stability analysis and closed-loop design. For example, [22] demonstrates significant accuracy improvement in small-signal models by capturing high-order harmonics in DC-DC series resonant converters.

Based on the multi-harmonic large-signal model, we further derive a small-signal model that captures the higher-order effects of the DC-DC converter behavior. The proposed model considers both the DC averages and the higher-order harmonic components as well as the interactions between them, thereby providing a complete small-signal characterization of the converter circuits. The proposed model forms a linear time-variant (LTV) system that can be empirically analyzed for stability. Using several closed-loop design examples, we demonstrate that in practice our model shows significant accuracy improvements in high-frequency responses over existing methods in the literature, thereby providing useful design insights for critical applications such as optimization and design centering.

1.4 Efficient Analysis of Power Distribution Networks

Power distribution networks is another important component in power delivery design. As shown in Fig. 1.1, the power distribution networks distribute the power drawn from DC-DC converters to individual transistor on each functional circuit. With the complexity of chip design continuing to increase to multi-million or even multi-billion gates, the size of the power distribution networks also increase dramatically and analyzing such circuits are becoming more and more challenging.

The bottleneck of analyzing the power distribution networks lies in efficiently finding the solution to the following linear system problem $Ax = b$, where A is a large sparse matrix representing the conductance of the circuit, b is a vector of the external current sources and x is a vector of node voltages. There are largely two families of methods to solve the sparse matrix system - the direct method and the iterative method. The direct solution methods [23, 24], which decompose the matrix into upper and lower triangular matrices and then solve the system using forward/backward substitution, have been widely adopted for their robustness and accuracy in circuit simulation. However, these methods have limited application in power grid analysis due to extremely large circuit size and the superlinear time and memory complexity of matrix factorization. Iterative solvers including conjugate gradient (CG) and generalized minimal residual method (GMRES), which update the solution step by step and solve the linear system iteratively, are considered more suitable for power grid analysis due to the favorable memory requirement.

To facilitate fast convergence, preconditioned iterative solvers are introduced and various preconditioning techniques have been studied. The support graph preconditioner, which is presented in [25, 26] as a general preconditioner for symmetric positive-definite (SPD) matrices, is particularly interesting. This technique identifies an underlying subgraph from the matrix, called the support graph, and uses it as the preconditioner of the matrix. For a particular type of support graphs, the maximum spanning tree (MST), it is proved that the preconditioned system has a bound of $O(n^2)$ on the condition number for any $n \times n$ SPD matrix. [27] extends the support graph technique to power grid analysis by building a hierarchical maximum spanning tree from the circuit, showing a significant improvement in runtime and memory usage over direct solution methods.

The limitation of the support graph technique is that as a flat preconditioning method, it can only be applied to the full power grid. Recent development in parallel processing has made partition-based preconditioners more favorable. [3] presents a parallel additive Schwarz preconditioner based on the algebraic partition on the circuit conductance matrix. [6] introduces an overlapping partition-based power grid analysis method using spatial locality to provide approximate boundary conditions. Multigrid method [28] is another family of preconditioners, which maps the original problem to a reduced system using certain geometric properties of the power distribution network (PDN). Both of these methods assume that the PDN is well-structured and the partition is already given. However, for large and complex IC designs with millions of nodes and beyond, the PDN is not always well-structured and finding a reasonable partition is not trivial. [29] presents a complete package including hypergraph partitioning and block Jacobi preconditioning techniques. But as a general-purpose circuit simulator, it lacks the efficiency for power grid analysis.

To address the problems mentioned above, we propose a hybrid method that combines the support graph preconditioner with the block Jacobi preconditioner. The proposed method efficiently finds a partition of the matrix based on the support graph extracted from the circuit. The result is an elegant yet powerful partition-based preconditioner that targets optimizing the numerical convergence property of the partitioned system. We verify the performance of our proposed method on several industrial power grid circuits. Compared with direct method and other partition-based iterative method, our proposed method demonstrates significant improvement in simulation efficiency.

1.5 Organization of Dissertation

The remaining part of this dissertation is organized as follows. In Chapter 2 we elaborate an efficient and robust envelope following algorithm for simulating DC-DC converters. In Chapter 3 a multi-harmonic large-signal averaging modeling technique that takes DC-DC converters' device non-idealities into account is discussed and verified. Chapter 4 extends the multi-harmonic average model to derive small-signal models for DC-DC converters which demonstrate significant accuracy improvements in high-frequency responses over existing methods. In Chapter 5 we present the convergence-boosted graph partitioning method that enables fast and accurate simulation of large power distribution networks. Finally in Chapter 6 we draw conclusions.

2. ROBUST AND EFFICIENT TRANSISTOR-LEVEL ENVELOPE-FOLLOWING ANALYSIS OF DC-DC CONVERTERS [*]

Highly efficient DC-DC converters are indispensable in today's low power microprocessors, embedded systems, and portable devices [30]. However, the simulation of these circuits is generally very challenging due to the existence of complex dynamics, widely spread time scales (*e.g.* fast switchings with slowly varying amplitudes) and feedback control. These difficulties render the use of the standard transient analysis very inefficient, for instance, by forcing the stepsize to be very small.

In this chapter, we develop a robust and efficient envelope following method to provide a unifying solution to the simulation of DC-DC converters with both transient and steady-state behaviors under a constant or varying switching frequency. Of our particular interest are real-life DC-DC power converters that are operated with complex modulation controls, *e.g.*, pulse width modulation (PWM), pulse frequency modulation (PFM) or pulse skipping modulation (PSM). Even with the standard transient analysis, these circuits are very challenging to simulate due to the co-existence of multi-rate nature, strong nonlinearities, hard switching activities, digital/memory and hysteretic effects and strong feedback control. These characteristics significantly stress the robustness and efficiency requirements of the applied envelope following method and prevent us from using techniques that have been shown to be successful for oscillators such as [4].

At the core of our new envelope following algorithm is a novel time-delayed phase condition that provides robust tracking of the circuit envelope in the transient phase, under a varying switching frequency. We further develop a mechanism that can smoothly track

[*] ©2016 IEEE. Reprinted, with permission, from Ya Wang, Peng Li and Suming Lai, "Robust and Efficient Transistor-Level Envelope-Following Analysis of PWM/PFM/PSM DC-DC Converters", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Volume: 35, Issue: 11, Nov. 2016

the transitions between the transient and steady-state phases, thereby providing a unifying solution to the EF simulation of both modes of operation. The implementation of three fast simulation techniques improves the efficiency of the algorithm without compromising the accuracy of the results. The proposed method can be transparently applied to PWM, PFM and PSM converters under a constant or varying switching frequency. We demonstrate the excellent robustness, generality, and efficiency of the proposed technique using several DC-DC converters and oscillator circuits.

2.1 Backgrounds

We review the basic backward-Euler based envelope following method for converters with a constant switching frequency (*e.g.* PWM converters). An electronic circuit can be described using a standard differential-algebraic equation (DAE)

$$\dot{\boldsymbol{q}}(\boldsymbol{x}) + \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{u}(t), \tag{2.1}$$

where $\boldsymbol{x} \in \mathbf{R}^N$ is a vector of state variables, $\boldsymbol{q}$ is a nonlinear charge function, $\boldsymbol{f}$ describes the resistive nonlinearities, and $\boldsymbol{u}(t)$ is the excitation to the circuit [31].

The output voltage of converters such as PWM converters demonstrates fast switching activities with a slowly varying amplitude as a result of load change, a characteristic that is well suited for EF analysis [32]. As shown in Fig. 2.1, denote the constant switching cycle of the circuit at t_0 by T_0 and switching cycle at t_n by T_n. The switching cycle is known as a constant, so we have $T_0 = T_n$. Denote the state variable at time t by $\boldsymbol{x}(t)$, accordingly the state at time t_0 by $\boldsymbol{x}(t_0)$ and that at t_n by $\boldsymbol{x}(t_n)$. Then define t_n as the time point that is n cycles after t_0, namely $t_n = t_0 + nT_n$, so that circuit at t_n has the same phase as at t_0. By the envelope following method, starting from a given $\boldsymbol{x}(t_n)$, we simulate the circuit for one cycle T_n to get another state vector of equal phase $\boldsymbol{x}(t_{n+1}) = \boldsymbol{x}(t_n + T_n)$. As shown in Fig. 2.1, if the amplitude of the circuit response changes slowly enough, a

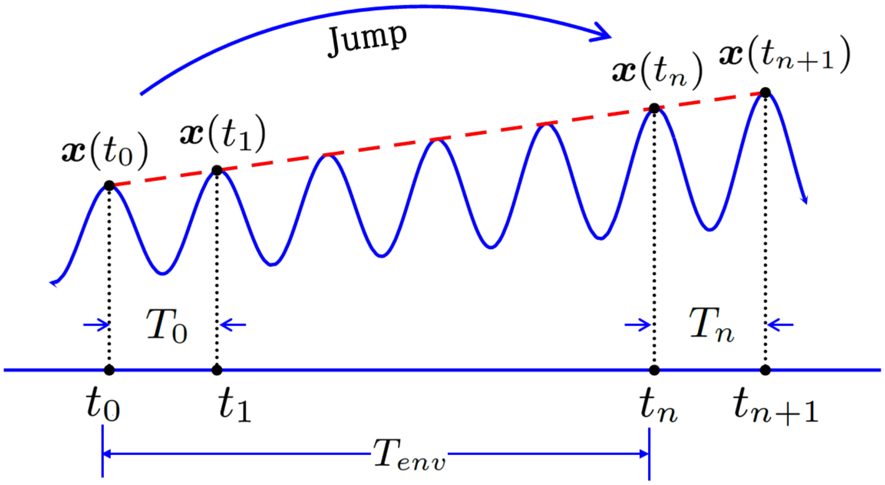


Figure 2.1: Basic envelope following method. t_0,t_1,t_n and t_{n+1} are four time points at equal phase and satisfy both $t_1 = t_0 + T_0$ and $t_{n+1} = t_n + T_n$.

line can be drawn to pass through these three equal-phase points, implying that

$$\frac{\boldsymbol{x}(t_n) - \boldsymbol{x}(t_0)}{nT_n} = \frac{\boldsymbol{x}(t_{n+1}) - \boldsymbol{x}(t_n)}{T_n}. \tag{2.2}$$

Note that $\boldsymbol{x}_{n+1}$ can be evaluated as $\boldsymbol{x}_{n+1} = \phi(\boldsymbol{x}_n, t_n, T_n)$, where ϕ is the state transition function of the circuit. Now (2) can be written as

$$\frac{\boldsymbol{x}(t_n) - \boldsymbol{x}(t_0)}{nT_n} = \frac{\phi(\boldsymbol{x}(t_n), t_n, T_n) - \boldsymbol{x}(t_n)}{T_n}. \tag{2.3}$$

The only unknown in this equation is $\boldsymbol{x}_n$, which can be solved by any nonlinear solution method such as Newton-Raphson Method.

Starting from a known initial state $\boldsymbol{x}(t_0)$, one may skip a large number of switching cycles which is $T_{env} = nT_n$ as in Fig. 2.1, to directly solve for the state $\boldsymbol{x}(t_n)$. To move one step forward, the same procedure is re-started by treating the solved $\boldsymbol{x}(t_n)$ as the new initial state $\boldsymbol{x}(t_0)$ and $\boldsymbol{x}(t_{n+1})$ as $\boldsymbol{x}(t_1)$, respectively. However, this basic EF method assumes that the switching cycle does not change, which prevents its application to circuits with dynamically changing switching frequencies like PFM DC-DC converters, which are the focuses of the next section.

2.2 Time-delayed Phasing Tracking for Circuits with Variable Switching Frequencies

For the types of circuits of interest here, the varying switching frequency is set by a specific modulation or tuning mechanism. As such, the switching period T_n is not known *a priori* and must be treated as an unknown variable. Another consequence of the varying frequency is that the skipped time interval between t_0 and t_n may not be an integer number of cycles, *i.e.*, it is generally true that $T_{env} \neq nT_n$. To see the issues involved, we rewrite (2) slightly as

$$\frac{\boldsymbol{x}(t_n) - \boldsymbol{x}(t_0)}{T_{env}} = \frac{\boldsymbol{x}(t_{n+1}) - \boldsymbol{x}(t_n)}{T_n}, \tag{2.4}$$

where the three unknown variables are $\boldsymbol{x}(t_n)$, T_n, and T_{env}. Since there are $N+2$ unknowns and only N equations (N being the dimensionality of (2.4)), this system is underdetermined. In fact, (4) alone does not guarantee $\boldsymbol{x}(t_0)$, $\boldsymbol{x}(t_n)$ and $\boldsymbol{x}(t_{n+1})$ being at the same phase. Varying switching frequencies introduce significant challenges to envelope following. We propose a novel and numerically robust technique for tracking the phase of circuits operating under a changing switching frequency.

2.2.1 Challenges and possible solutions

Several existing EF methods address varying switching or oscillation frequencies. In the oscillator simulation technique of [33], T_n is predetermined during the integration process by using the notion of Poincar$\acute{e}$ map. With T_n computed, T_{env} is set to be an integer multiple of T_n. However, the underlying assumption that T_n remains unchanged

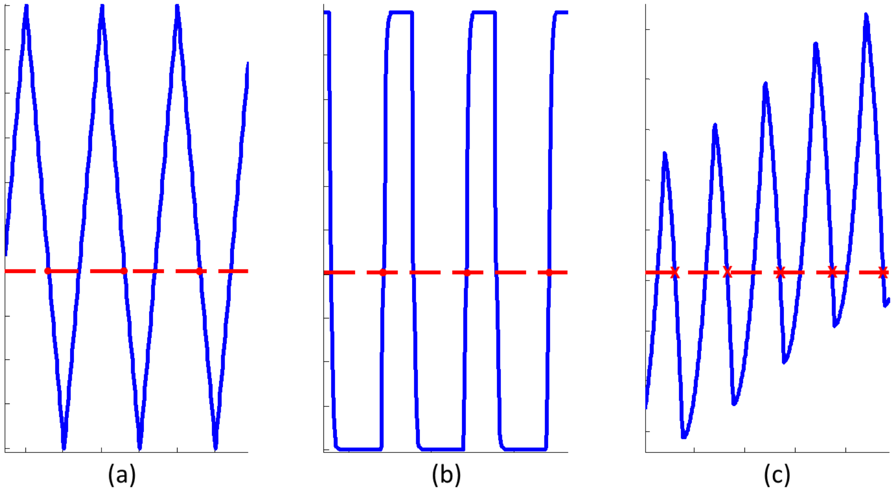


Figure 2.2: Three typical voltage waveforms in a DC-DC converter: (a) external triangular signal, (b) typical internal signal driving power switches, and (c) typical output response with both frequency and amplitude modulation. Specific level crossing times may be used to define equal phase conditions (red dashed lines).

within one envelope step is not true in general and can potentially prevent use of large envelope stepsizes for transient phases of the circuit. In addition, it not always possible to determine T_n by Poincar$\acute{e}$ map as suggested in [34]. This situation could be worse for DC-DC converters with digital behaviors and hard switching activities. [4] addresses this problem in oscillator simulation by adding two extra phase conditions at t_n and t_{n+1} by constraining the time derivatives of a nodal voltage at these two points. Though proved effective for oscillators, this technique may not be suitable for DC-DC converters due to the presence of digital characteristics and sharp signal transitions, which exacerbate the numerical noise inherent in the numerical evaluation of time derivatives.

Fig. 2.2 shows waveforms of three typical nodes in DC-DC converters. Fig. 2.2(a) shows the external triangular signal that is used in the PWM DC-DC converter of Fig. 2.12. Fig. 2.2(b) shows a typical internal voltage signal that drives the MOS switches in both PWM and PFM DC-DC converters. Fig. 2.2(c) depicts a typical output waveform with a varying amplitude during the transient phase of a PWM/PFW DC-DC converter. Common properties of these representative signals are that: they all have discontinuous first derivatives; and there are long periods of time in which the first-order derivative of a signal is either very large or approximately constant. These characteristics present practical challenges for the aforementioned EF techniques developed for oscillators.

On the other hand, sharp signal changes in a converter indicate certain controlled-switching events that are taking place in the circuit. These switching activities reliably reflect the onset or ending of a specific mode of operation and can be in principle leveraged to robustly identify equal-phase points that shall be sampled by an EF method. Clearly, the main challenge here is to achieve such in a numerically robust manner. For example, we may purposely choose to monitor one or multiple internal signals with large sharp swings and use the moments at which such signals cross given critical threshold levels to determine the sampling time instants for EF, as shown in Fig. 2.2 by the dashed lines. For instance, the output of the SR-latch of the PFM converter used in the experimental section (Fig. 2.16) can be a good choice as its switching activities reveal the on or off states of the power switches. However, using fixed signal crossing levels is problematic for signals that experience amplitude modulation as in Fig. 2.2(c).

2.2.2 Robust monitoring of phase change

Motivated by the above discussion, we introduce a new equal-phase condition that can capture both frequency and amplitude variations in a converter. To do that, we begin with a definition of slope function $slp(\cdot,\cdot)$. Slope function is defined by two points over a given period. As shown by points A and B in Fig. 2.3, the slope function of A and B is

$$slp(A,B)=\frac{x(B)-x(A)}{t(B)-t(A)}, \tag{2.5}$$

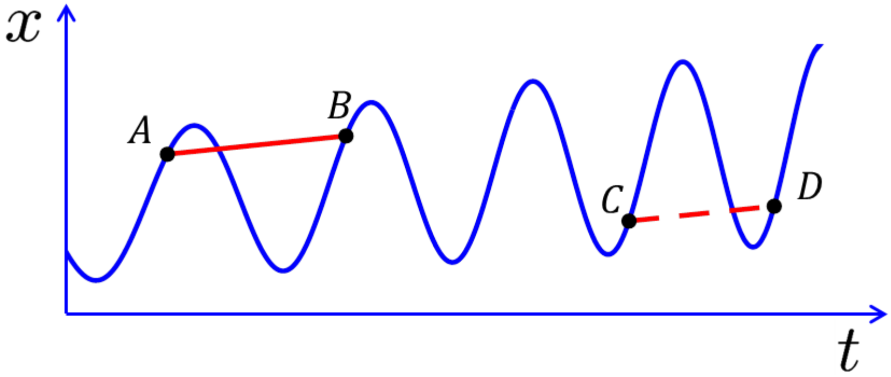


Figure 2.3: Slope function and equal-phase points.

where $x(A)$ and $x(B)$ are the amplitudes of points A and B, $t(A)$ and $t(B)$ the time instants of A and B. The slope function is useful in defining equal-phase condition because the slope function of two equal-phase points can be used to find other equal-phase points in the vicinity. For example in Fig. 2.3, assume that A and B are already known as equal-phase points, *i.e.* they mark the beginning and ending of one switching cycle. In this case, $slp(A, B)$ is in fact the cycle-slope of the signal during the corresponding switching period. Then, points C and D are also equal-phase points if and only if $slp(A, B) - slp(C, D) = 0$ is satisfied. Note that this new phase condition is a more general case of the scheme that is based on crossing times of fixed signal threshold levels discussed at the end of Section 2.2.1.

When computing slope function to find points with equal phase, we monitor one or multiple internal circuit nodes (branches), or phase monitoring nodes (or branches). To robustly specify equal-phase points, we assume that phase monitoring nodes are provided by the designer and exposed to the simulation algorithm. In general, these nodes can be chosen rather easily by leveraging a very minimum amount of design knowledge. For instance, for PFM controlled DC-DC converters, a natural choice is the regulated output node that drive internal comparators to alter the switching behavior. Using the idea of

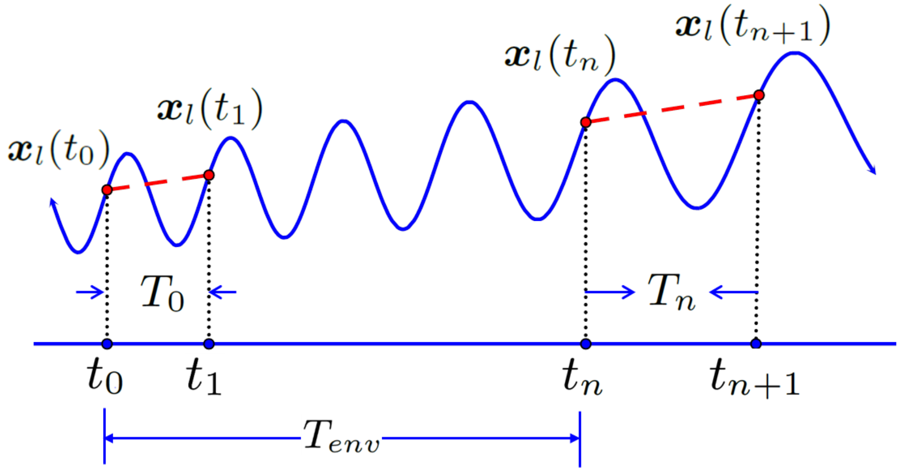


Figure 2.4: Equal-phase points defined by equal slope function value are shown in the dashed lines.

slope function, we now formally define the proposed equal-phase condition. As shown in Fig. 2.4, denote the voltage of a phase monitoring node by $\boldsymbol{x}_l$. Four nodes involving the envelope can then be denoted by $\boldsymbol{x}_l(t_0)$, $\boldsymbol{x}_l(t_1)$, $\boldsymbol{x}_l(t_n)$ and $\boldsymbol{x}_l(t_{n+1})$. Under the assumption that the variance of the envelope slope within one step is small, the equal-phase condition can be defined as

$$\frac{\boldsymbol{x}_l(t_{n+1}) - \boldsymbol{x}_l(t_n)}{T_n} - \frac{\boldsymbol{x}_l(t_1) - \boldsymbol{x}_l(t_0)}{T_0} = 0. \tag{2.6}$$

Note that results from previous envelope cycle $\boldsymbol{x}_l(t_0)$ and $\boldsymbol{x}_l(t_1)$ are already available and are at equal phase. When (6) is satisfied, the value of slope function of $\boldsymbol{x}_l(t_n)$ and $\boldsymbol{x}_l(t_{n+1})$ is equal to that of $\boldsymbol{x}_l(t_0)$ and $\boldsymbol{x}_l(t_1)$, meaning that $\boldsymbol{x}_l(t_n)$ is at the equal phase as $\boldsymbol{x}_l(t_{n+1})$.

Since the system needs two phase conditions, we simply apply (6) to another phase monitoring node, denoted by $\boldsymbol{x}_k$. Now the new system of envelope following formulation is

$$
\begin{aligned}
\frac{\boldsymbol{x}(t_n) - \boldsymbol{x}(t_0)}{T_{env}} - \frac{\boldsymbol{x}(t_{n+1}) - \boldsymbol{x}(t_n)}{T_n} &= 0 \\
\frac{\boldsymbol{x}_l(t_{n+1}) - \boldsymbol{x}_l(t_n)}{T_n} - \frac{\boldsymbol{x}_l(t_1) - \boldsymbol{x}_l(t_0)}{T_0} &= 0 \\
\frac{\boldsymbol{x}_k(t_{n+1}) - \boldsymbol{x}_k(t_n)}{T_n} - \frac{\boldsymbol{x}_k(t_1) - \boldsymbol{x}_k(t_0)}{T_0} &= 0.
\end{aligned}
\tag{2.7}
$$

Here to distinguish between two different cycle periods, we denote the period at t_0 by T_0 (already known at this time) and one at t_n by T_n.

While (7) appears to be robust, a close examination reveals that (7) has a fundamental problem. Note that the l^{th} row of the first equation and the second equation in (7) have the shared term $(\boldsymbol{x}_l(t_{n+1}) - \boldsymbol{x}_l(t_n))/T_n$. Similarly, $(\boldsymbol{x}_k(t_{n+1}) - \boldsymbol{x}_k(t_n))/T_n$ is shared by the k^{th} row of the first equation and the third equation. Substituting the last two equal-phase equations into the l^{th} and k^{th} rows of the first equation leads to

$$
\begin{aligned}
\frac{\boldsymbol{x}_l(t_n) - \boldsymbol{x}_l(t_0)}{T_{env}} - \frac{\boldsymbol{x}_l(t_1) - \boldsymbol{x}_l(t_0)}{T_0} &= 0 \\
\frac{\boldsymbol{x}_k(t_n) - \boldsymbol{x}_k(t_0)}{T_{env}} - \frac{\boldsymbol{x}_k(t_1) - \boldsymbol{x}_k(t_0)}{T_0} &= 0,
\end{aligned}
\tag{2.8}
$$

which correspond to forward Euler integration of the envelope of $\boldsymbol{x}_l$ and $\boldsymbol{x}_k$, respectively. The explicit forward Euler method is not A-stable and has much degraded stability region. It is rarely used in practice. The formulation of (7) effectively applies the explicit forward Euler type integration to the two phase monitoring nodes, a problem that shall be remedied by an improved equal phase condition introduced next.

2.2.3 Time-delayed equal-phase condition

A deep investigation reveals that the above problem stems from the fact that the introduced two phase conditions do not provide fully independent new constraints of the circuit state. In (7), shared terms involving envelope states exist between the l^{th} row of the first equation and the second equation, and between the k^{th} row of the first equation and the third equation. Such sharing renders the phase conditions and the backward Euler based EF equation constrain a common set of state variables and may manifest itself in several different ways. If the phase conditions are constructed by forcing the two monitored nodal voltages (or branch currents in general) to cross a predetermined level at equal phase points, a special case of the more general phase conditions adopted in (7), it can be shown that the equal phase equations would be identical to the l^{th} and k^{th} rows of the first equation of (7), rendering the full system underdetermined. As discussed already, the more general formulation of (7) immediately reduces the integration of two monitoring nodes to forward Euler while the deeper cause of this phenomenon is due to the sharing.

The key observation behind our solution to the above problem is to note that an equal-phase condition needs not to be defined at the beginning nor end of each cycle; it can be forced anywhere within a cycle. This observation leads to a new equal-phase condition, termed *time-delayed phase condition*, resulting in a new EF formulation

$$\begin{aligned} \frac{\boldsymbol{x}(t_n) - \boldsymbol{x}(t_0)}{T_{env}} - \frac{\boldsymbol{x}(t_{n+1}) - \boldsymbol{x}(t_n)}{T} &= 0 \\ \frac{\boldsymbol{x}_l(t'_{n+1}) - \boldsymbol{x}_l(t'_n)}{T_n} - \frac{\boldsymbol{x}_l(t'_1) - \boldsymbol{x}_l(t'_0)}{T_0} &= 0 \\ \frac{\boldsymbol{x}_k(t'_{n+1}) - \boldsymbol{x}_k(t'_n)}{T_n} - \frac{\boldsymbol{x}_k(t'_1) - \boldsymbol{x}_k(t'_0)}{T_0} &= 0, \end{aligned} \tag{2.9}$$

where $t'_0 = t_0 + \alpha T_0$, $t'_1 = t_1 + \alpha T_0$, $t'_n = t_n + \alpha T_n$ and $t'_{n+1} = t_{n+1} + \alpha T_n$. $\alpha \in (0, 1)$ is a constant factor used to delay the sampling time of phase condition, as illustrated in Fig. 2.5. In (9), the first equation represents the same backward-Euler style equation involving the state variables at a set of four points t_0, t_1, t_n and t_{n+1}. In contrast, the last two equations specify the equal-phase condition at a different set of four points with each delayed by a fraction of the respective cycle time from the corresponding point in the first set. The two equal-phase condition in (9) constrain a different set of state variables from ones that are

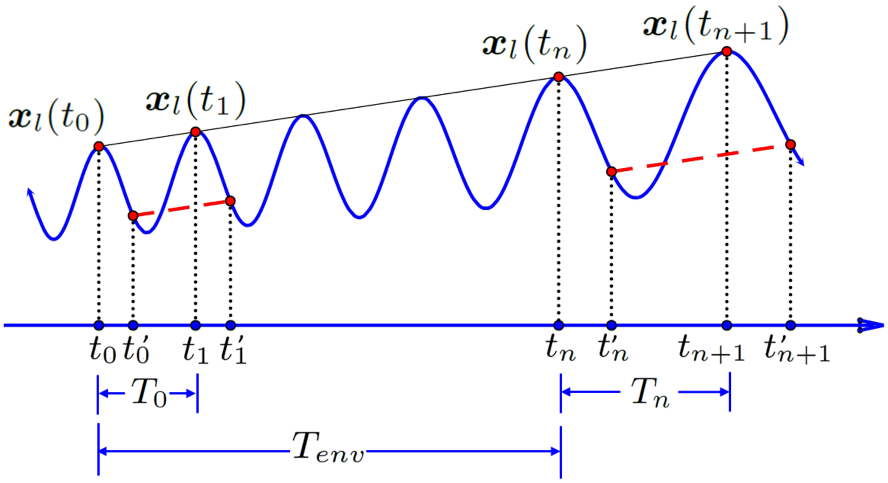


Figure 2.5: Time delayed phase tracking method.

in the first equation. The new state variables that are forced to be at an equal phase are the future states of the corresponding variables in the first equation and are related to the latter variables through the nonlinear state transition characteristics of the converter excited by the external input.

2.2.4 Robust numerical solution

Our proposed EF method is described in (9) and the unknown vector that needs to be solved is

$$\boldsymbol{X} = \left[\boldsymbol{x}^{\mathrm{T}}(t_n), T_n, T_{env}\right]^{\mathrm{T}}. \tag{2.10}$$

To solve the system in (9) by the Newton Raphson method, we need to evaluate the Jacobian matrix properly which involves computation of the sensitivities of each term in (9) with respect to any of the three unknown variables. The evaluation of sensitivities is done by computing the corresponding partial derivatives. Most of them are straightforward to evaluate except for those terms that involve the state transition function, explained as follows. Since in every Newton-Raphson iteration an inner-loop transient run from t_n to t'_{n+1}

is performed in order to get $\boldsymbol{x}(t_{n+1})$ and $\boldsymbol{x}(t'_{n+1})$, the desired sensitivities terms can be accumulated through every transient step [32]. For convenience of notation, denote $\boldsymbol{x}(t_n)$ by $\boldsymbol{x}_0$ (not to be confused with $\boldsymbol{x}(t_0)$ in Fig. 2.1) and the state at the k^{th} step of the inner-loop transient simulation by $\boldsymbol{x}_k$, (1) can be written as

$$\frac{\boldsymbol{q}(\boldsymbol{x}_k) - \boldsymbol{q}(\boldsymbol{x}_{k-1})}{h_k} + \boldsymbol{f}(\boldsymbol{x}_k) = \boldsymbol{u}(t_k), \tag{2.11}$$

at the k^{th} step of the transient simulation, with a stepsize of h_k. Differentiating (11) with respect to $\boldsymbol{x}_0$ gives

$$\frac{\partial \boldsymbol{x}_k}{\partial \boldsymbol{x}_0} = \left[\frac{1}{h_k}\frac{\partial \boldsymbol{q}(\boldsymbol{x}_k)}{\partial \boldsymbol{x}_k} + \frac{\partial \boldsymbol{f}(\boldsymbol{x}_k)}{\partial \boldsymbol{x}_k}\right]^{-1} \left[\frac{1}{h_k}\frac{\partial \boldsymbol{q}(\boldsymbol{x}_{k-1})}{\partial \boldsymbol{x}_{k-1}}\frac{\partial \boldsymbol{x}_{k-1}}{\partial \boldsymbol{x}_0}\right]. \tag{2.12}$$

Note that $\frac{1}{h_k}\frac{\partial \boldsymbol{q}(\boldsymbol{x}_k)}{\partial \boldsymbol{x}_k} + \frac{\partial \boldsymbol{f}(\boldsymbol{x}_k)}{\partial \boldsymbol{x}_k}$ is the Jacobian matrix of (1) and is available from the transient simulation. Starting from $\frac{\partial \boldsymbol{x}_0}{\partial \boldsymbol{x}_0} = I$, applying (12) repeatedly at every transient step and accumulating the results will give all the desired sensitivity terms with respect to $\boldsymbol{x_0}$ along the way including $\frac{\partial \boldsymbol{x}_{n+1}}{\partial \boldsymbol{x}_0}$ in the end. Other sensitivity terms can be found in a similar way. Differentiating (11) with respect to T_n and T_{env} gives

$$\begin{aligned}\frac{\partial \boldsymbol{x}_k}{\partial T_n} = &\left[\frac{1}{h_k}\frac{\partial \boldsymbol{q}(\boldsymbol{x}_k)}{\partial \boldsymbol{x}_k} + \frac{\partial \boldsymbol{f}(\boldsymbol{x}_k)}{\partial \boldsymbol{x}_k}\right]^{-1} \\ &\left[\frac{1}{h_k}\frac{\partial \boldsymbol{q}(\boldsymbol{x}_{k-1})}{\partial \boldsymbol{x}_{k-1}}\frac{\partial \boldsymbol{x}_{k-1}}{\partial T_n} + \frac{\partial \boldsymbol{u}(t_k)}{\partial T_n} + \frac{\boldsymbol{q}(\boldsymbol{x}_k) - \boldsymbol{q}(\boldsymbol{x}_{k-1})}{h_k T_n}\right],\end{aligned} \tag{2.13}$$

and

$$\frac{\partial \boldsymbol{x}_k}{\partial T_{env}} = \left[\frac{1}{h_k}\frac{\partial \boldsymbol{q}(\boldsymbol{x}_k)}{\partial \boldsymbol{x}_k} + \frac{\partial \boldsymbol{f}(\boldsymbol{x}_k)}{\partial \boldsymbol{x}_k}\right]^{-1} \left[\frac{1}{h_k}\frac{\partial \boldsymbol{q}(\boldsymbol{x}_{k-1})}{\partial \boldsymbol{x}_{k-1}}\frac{\partial \boldsymbol{x}_{k-1}}{\partial T_{env}} + \frac{\partial \boldsymbol{u}(t_k)}{\partial T_{env}}\right]. \tag{2.14}$$

Starting from $\frac{\partial \boldsymbol{x}_0}{\partial T_n} = 0$ and $\frac{\partial \boldsymbol{x}_0}{\partial T_{env}} = 0$, all sensitivity terms with respect to T_n and T_{env} can also be accumulated by applying (13)(14) repeatedly.

2.3 Unifying Phase Tracking

An EF method transparently applicable to a variety of converters with PWM, PFM modulation or a combination of thereof, is highly desirable. In fact, a PFM converter may effectively operate under a constant switching frequency in steady state. We consider steady-states of PFM converters to motive the need for a new unifying phase condition.

2.3.1 Problems with steady state EF analysis

To see why the formulation of (9) may not be applied to steady state, we examine the following partial derivatives of the two phase conditions. Denote the first phase condition in (9) by g_1 and the second one by g_2. It can be shown that

$$\frac{\partial g_1}{\partial T_{env}} = \frac{1}{T_n}\left(\frac{\partial \boldsymbol{x}_l(t'_{n+1})}{\partial T_{env}} - \frac{\partial \boldsymbol{x}_l(t'_n)}{\partial T_{env}}\right) \tag{2.15}$$

$$\frac{\partial g_2}{\partial T_{env}} = \frac{1}{T_n}\left(\frac{\partial \boldsymbol{x}_k(t'_{n+1})}{\partial T_{env}} - \frac{\partial \boldsymbol{x}_k(t'_n)}{\partial T_{env}}\right). \tag{2.16}$$

When the circuit gets settled in steady state under constant input excitations, $\frac{\partial \boldsymbol{u}(t_k)}{\partial T_{env}} = 0$. According to (14), the sensitivity terms of (15) and (16) are both zero, implying that both phase conditions do not constrain unknown T_{env} and (9) becomes under-determined as a result. The root cause of this pathological situation is that the circuit appears to be autonomous in steady sate with its current/future states only depend on its past states, but not on time.

2.3.2 Phase condition for steady state

As a first step to addressing the above problem, we adopt a new steady state phase condition that involves T_{env} by noting that cycle T becomes a constant in steady state

$$T_n - T_0 = 0, \tag{2.17}$$

where T_0 is the cycle time at time t_0. Also in steady state, T_{env} is an integer multiple of T

$$T_{env} - nT_n = 0, \tag{2.18}$$

where n is the number of cycles skipped in one envelope following step. Replacing two phase conditions in (9) by (17) and (18) yields

$$\begin{aligned} \frac{\boldsymbol{x}(t_n) - \boldsymbol{x}(t_0)}{T_{env}} - \frac{\boldsymbol{x}(t_{n+1}) - \boldsymbol{x}(t_n)}{T_n} &= 0 \\ T_n - T_0 &= 0 \\ T_{env} - nT_n &= 0. \end{aligned} \tag{2.19}$$

The system described in (19) is equivalent to the classic envelope following algorithm for circuits with fixed switching frequencies described in (2.2), which can be used to reliably solve any steady state solutions.

2.3.3 Smooth circuit state tracking and unifying phase condition

Till this point, we have developed two separate EF problem formulations in (9) and (19), respectively for transient and steady states. However, a circuit may transition between the two modes of operation back and forth and such transitions may be smooth and are not known *a priori*. Clearly, a unifying formulation is desirable. Denote the first phase condition in (9) by g_{11} and the phase condition described in (17) by g_{12}, namely

$$\begin{aligned} g_{11} &= \frac{\boldsymbol{x}_l(t'_{n+1}) - \boldsymbol{x}_l(t'_n)}{T_n} - \frac{\boldsymbol{x}_l(t'_1) - \boldsymbol{x}_l(t'_0)}{T_0} && (2.20) \\ g_{12} &= T_n - T_0. && (2.21) \end{aligned}$$

Similarly, denote the second phase condition in (9) by g_{21} and the phase condition described in (18) by g_{22}, namely

$$\begin{aligned} g_{21} &= \frac{\boldsymbol{x}_k(t'_{n+1}) - \boldsymbol{x}_k(t'_n)}{T_n} - \frac{\boldsymbol{x}_k(t'_1) - \boldsymbol{x}_k(t'_0)}{T_0} && (2.22) \\ g_{22} &= T_{env} - nT_n. && (2.23) \end{aligned}$$

The key idea in developing a unifying solution is to define a set of weighted new phase conditions,

$$\begin{aligned} g_1 &= \beta g_{11} + (1-\beta) g_{12} \\ g_2 &= \beta g_{21} + (1-\beta) g_{22} \end{aligned} \qquad (2.24)$$

where $\beta \in [0, 1]$ is a *continuous* internal parameter that continuously tracks the current circuit mode of operation, as illustrated in Fig. 2.6. Using this new phase conditions, the unifying formulation of the envelope following problem is

$$\begin{aligned} T_{env}[\boldsymbol{x}(t_{n+1}) - \boldsymbol{x}(t_n)] - T_n[\boldsymbol{x}(t_n) - \boldsymbol{x}(t_0)] &= 0 \\ \beta g_{11} + (1-\beta) g_{12} &= 0 \\ \beta g_{21} + (1-\beta) g_{22} &= 0. \end{aligned} \qquad (2.25)$$

We now discuss the significance and the implementation of the internal parameter β. Since the transitions between transient and steady states may not be abrupt, the goal be-

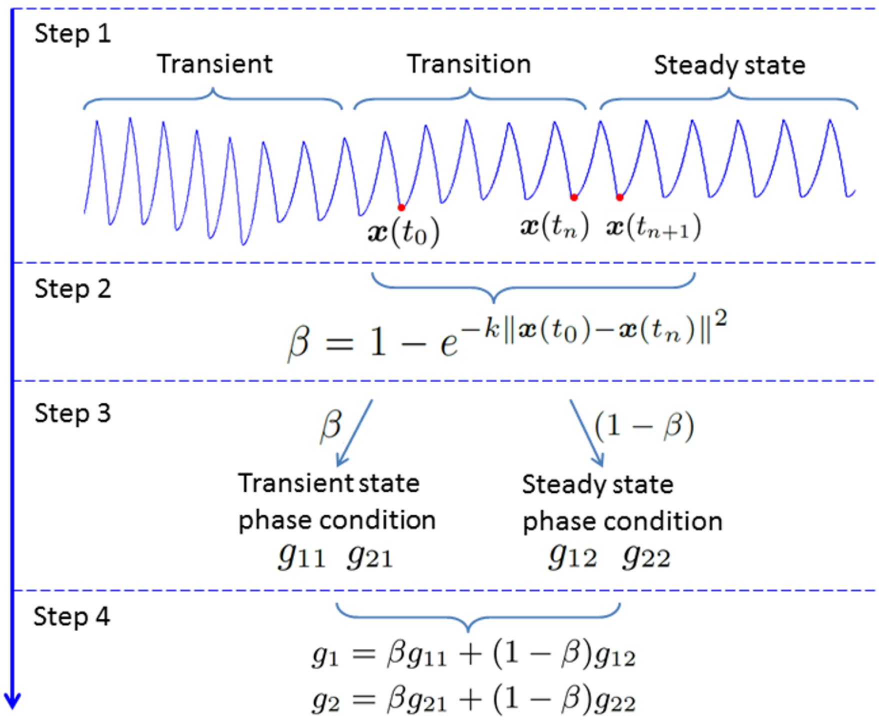

Step 1
Transient
Transition
Steady state
$\boldsymbol{x}(t_0)$
$\boldsymbol{x}(t_n)$
$\boldsymbol{x}(t_{n+1})$
Step 2
$\beta = 1 - e^{-k\|\boldsymbol{x}(t_0)-\boldsymbol{x}(t_n)\|^2}$
Step 3
β
$(1-\beta)$
Transient state phase condition
g_{11} g_{21}
Steady state phase condition
g_{12} g_{22}
Step 4
$g_1 = \beta g_{11} + (1-\beta) g_{12}$
$g_2 = \beta g_{21} + (1-\beta) g_{22}$

Figure 2.6: The unifying phase condition. Steps 1 and 2 illustrate the dependency of the continuous mode tracking parameter β on current circuit state. Steps 3 and 4 weight phase conditions according to β to define the unifying phase conditions.

hind the introduction of β is to provide a formal mathematical mechanism to track such transitions smoothly. As such, β is continuous valued in $[0, 1]$. Furthermore, such state transitions are not known *a prior*, hence, β shall not be set externally and must be an internal parameter that is dependent of the current circuit state. For this, while there exist multiple implementation choices for β, the following choice is found to be effective

$$\beta = 1 - e^{-k\gamma}, \tag{2.26}$$

where $\gamma = \|\boldsymbol{x}(t_0) - \boldsymbol{x}(t_n)\|^2$ and k is a scalar to balance between the values of g_{11} and g_{12} as well as g_{21} and g_{22}. When the circuit is at steady state, $\beta = 1 - e^{-k\|\boldsymbol{x}(t_0)-\boldsymbol{x}(t_n)\|^2}$=0. Therefore, g_1 and g_2 are the same as steady state phase conditions g_{12} and g_{22}, respectively. As more transient behavior is excited in the circuit, the norm γ will gradually increase and β will increase accordingly, putting more weight on g_{11} and g_{21}. This mechanism has the desired properties to track the operation mode transition faithfully. For example, with $\frac{\partial u(t)}{\partial t} = 0$ and as the circuit transitions into steady state, β will not vanish immediately until it gets completely settled. It is also worth noticing that β is an unknown dependent on the unknown current state $\boldsymbol{x}(t_n)$. The value of β can only become known after (25) is solved as a whole, providing generality and robustness for mode tracking.

2.4 Fast Simulation Techniques

In this section three techniques are introduced to further improve the efficiency of the proposed EF algorithm. At the beginning of each envelope following step, a dynamic prediction scheme generates the best initial guess for $x(t_n)$ from three different initial solution predictors. During each envelope following step, the simulator excludes certain digital nodes from convergence check, which reduces the number of iterations to reach convergence without degrading the accuracy of simulation results. At the end of each EF step, local truncation errors are estimated and the envelope following stepsize is changed

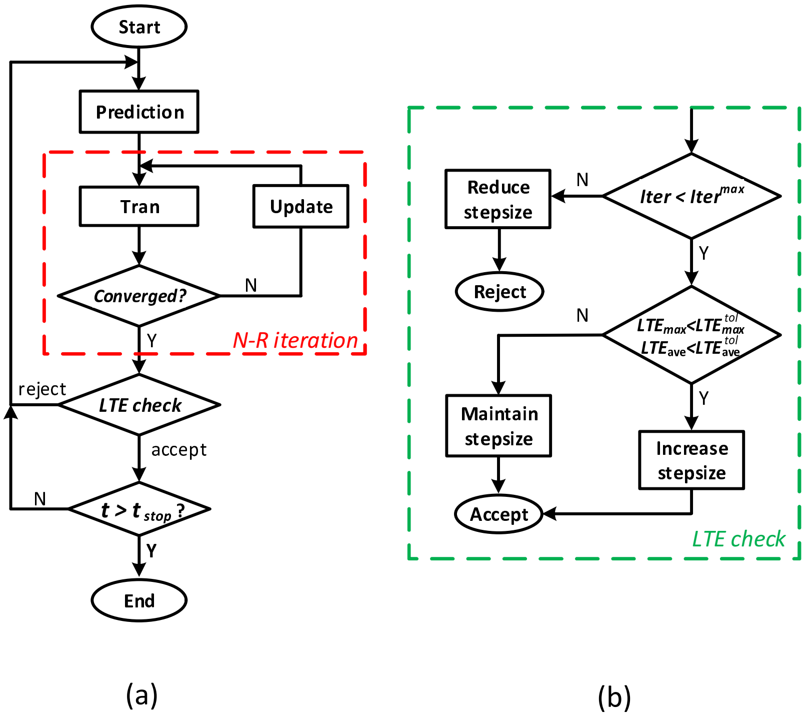


Figure 2.7: The simulation flow of proposed envelop following algorithm (a) and the flow of adaptive envelope step selection (b).

using adaptive envelope step selection.

2.4.1 Dynamic prediction scheme

As shown in Fig. 2.7(a), a full cycle of transient simulation is performed for each Newton-Raphson iteration in one EF step. Since simulation of DC-DC converters is generally time-consuming, the full cycle transient analysis dominates the simulation time of envelope following in each step. To reduce the total simulation time and achieve higher speedups, the number of iterations in each step needs to be minimized. A good prediction on $x(t_n)$ can provide the simulator with a starting point that is close to the actual solution and need fewer iterations to reach convergence. However, no prediction methods can always generate the best initial guess due to the fact that DC-DC converters have different dynamic behaviours in different modes. Therefore, we proposed a prediction-selection scheme to provide the best initial guess for solution vector $x(t_n)$ from three different initial solution predictors (ISPs). The selections are made by comparing three perdition results and selecting the one with the least error.

The first predictor ISP$_1$ directly uses the solution from the last step as an initial guess for the current step, namely

$$x(t_n) = x(t_0). \tag{2.27}$$

The second predictor ISP$_2$ leverages the two solution vectors of the previous steps $x(t_0)$ and $x(t_1)$ and uses a linear extrapolation of the two as the initial guess for $x(t_0)$. Recall that in Fig. 4 we denote T_0 as the time period between t_0 and t_1, we have

$$x(t_n) = x(t_0) + \frac{T_{env}}{T_0}\left[x(t_1) - x(t_0)\right]. \tag{2.28}$$

The third predictor ISP$_3$ is first introduced in [34], which uses a linear extrapolation of the circuit transfer function ϕ to predict $x(t_0)$. To achieve this, we first observe that $\phi(x(t_0), t_0, T_0)$, which is the circuit transfer function from $x(t_0)$ to $x(t_1)$, is already available from the transient simulation of the last step. With this, we can approximate the circuit transfer function from $x(t_n)$ to $x(t_{n+1})$ using the first-order Tyler expansion:

$$\phi(x(t_n), t_n, T_n) = \phi(x(t_0), t_0, T_0) + \left[x(t_n) - x(t_0)\right]\phi_x(x(t_0)), \tag{2.29}$$

where $\phi_x(x(t_0)) = \frac{\partial \phi}{\partial x}|_{x=x(t_0)}$. The left-hand-side of (2.29) is essentially equal to the solution vector $x(t_{n+1})$. Substituting (2.29) into the envelope following equation (2) results in

$$x(t_n) = \left[(1 + \frac{T_n}{T_{env}})I - \phi_x(x(t_0))\right]^{-1} \left[\phi(x(t_0), t_0, T_0) - \phi_x(x(t_0))x(t_0) + \frac{T_n}{T_{env}}x(t_0)\right]. \quad (2.30)$$

Now we have three different predictors and each of them generates accurate initial guesses for DC-DC converters operating in different modes. The first predictor (2.27) is essentially a zero-th order method, which generates accurate prediction when converters are near steady state. The second predictor method (2.28) is a first-order method and thus is effective for DC-DC converters with linearly changing load conditions. The third predictor (2.30) makes extrapolation of solution vectors based on circuit transfer functions. So it is most accurate when the circuit transfer function has almost linear changes between cycles and can be precisely calculated from the transient simulation.

To make the best uses of all three predictors, we design a selector that evaluates all three prediction results and chooses the one with the least error, which is measured by calculating distance from the given predicted solution to a valid circuit operating state. This can be done by setting the prediction result as the solution to the circuit and evaluate (1) with $u(t)$ moved to the left-hand-side. A non-zero residue vector on the right-hand-side can be acquired and the l_2 norm of such residue vector indicates how close the prediction is to the real circuit operating state. The predicted solution with the least l_2 error will be selected as the initial solution of the current step.

2.4.2 Automatic node selection for convergence check

DC-DC converters are mixed-signal circuits that include digital components with highly nonlinear behaviours. The outputs of those digital components have large swings and sharp transitions. As a result, the simulation is very challenging even for standard transient anal-

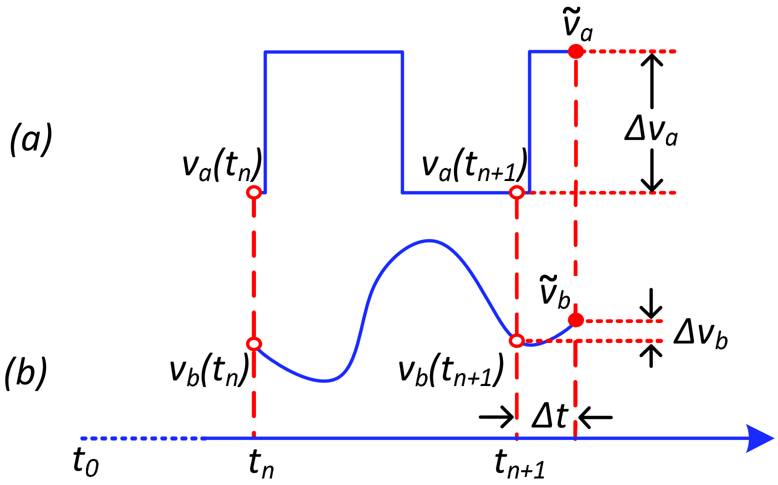


Figure 2.8: The transient analysis waveforms of one iteration of an envelope following step.

ysis with very small stepsizes. It is even more difficult for envelope following algorithm to efficiently reach convergence on these signals.

To illustrate this, the envelope following waveforms of two nodes in a PWM DC-DC converter are shown in Fig. 2.8, where the first node a is an internal node that drives the power switches and the second node b is the output node of the DC-DC converter. One Newton-Raphson iteration is shown where one cycle of transient simulation from t_n to t_{n+1} is performed for both nodes. At this iteration, the envelope following algorithm has found an accurate circuit state $v_a(t_n)$ and $v_b(t_n)$. Due to a small error Δt in cycle time T, the cycle starts from t_n and ends at $t_{n+1} + \Delta t$ instead of t_{n+1}. The corresponding end-of-cycle solution is $\tilde{v}_a$ and $\tilde{v}_b$, whereas the actual end-of-cycle solution is $v_a(t_{n+1})$ and $v_b(t_{n+1})$. For node b, if the error Δv_b and time difference Δt are small enough, this solution will be accepted by the EF convergence check. However, the same conclusion cannot be drawn for node a, which shows a huge error in Δv_a due to the sudden change of voltage level within a short period Δt. As a result, this solution is not accepted by convergence check and the simulator has to keep iterating. However, even with more iterations, the Newton-Raphson method cannot efficiently converge to the actual solution, due to the fact that the waveform between t_{n+1} and $t_{n+1} + \Delta t$ of node a has discontinuous first order derivatives [35].

A simple way to fix this problem is to exclude node a and other nodes with sharp transitions from the convergence check. It is important to note that this does not affect the accuracy of the results since the solution of one envelope following step is $x(t_n)$, which is the circuit state at t_n rather than t_{n+1}. By avoiding choosing nodes excluded from convergence check as the phase monitoring nodes to calculate (20) and (22), we also make sure that the nodal voltage error of $x(t_{n+1})$ does not propagate to the evaluation of the phase conditions. To select the proper nodes for convergence check, a few cycles of transient simulation is conducted before the envelope following simulation is started. During the transient simulation, a node selector continuously monitors all node voltages and exclude any node that satisfies both of the following two criteria, (1) nodes with large swings that the nodal voltage reaches both 1% and 99% of the supply voltage V_{dd} within one cycle and (2) nodes with sharp slopes by checking if the maximum first order derivative of the nodal voltage exceeds certain threshold. In practise nodes that satisfy these two criterion are output nodes of the digital components of the internal control circuitry. After removing nodes that satisfy these two criteria, the EF simulation takes fewer iterations to reach convergence and the accuracy of the results are not compromised.

2.4.3 Adaptive envelope step selection by LTE

To achieve good speedup factor while controlling accuracy, a backward-Euler based local truncation error (LTE) is utilized to predict the next envelope stepsize. The value of

LTE is estimated during each step of EF [33]

$$LTE = \frac{T_{env}}{2}\left(\frac{\boldsymbol{x}(t_{n+1}) - \boldsymbol{x}(t_n)}{T_n} - \frac{\boldsymbol{x}(t_1) - \boldsymbol{x}(t_0)}{T_0}\right). \tag{2.31}$$

Note that the above LTE is a vector. We compute its l_2-norm LTE_{ave} and the l_∞-norm LTE_{max}. After solving each envelope step, LTE_{ave}, LTE_{max} and the number of Newton-Raphson iterations $Iter$ are checked. Based on the LTE information, the current solution will be categorized into three different types, each leading to a specific action. If the number of iterations exceeds the maximum threshold $Iter^{max}$ then the solution is rejected and a smaller stepsize is used to redo this EF step. If the iteration number is accepted, LTE_{ave} and LTE_{max} are compared with two prescribed tolerance values LTE^{tol}_{ave} and LTE^{tol}_{max}. If both values are smaller than the tolerances, the stepsize is increased for the next EF step. Otherwise the same stepsize is maintained. Fig. 2.7(b) shows the detailed process of the adaptive envelope step selection.

2.5 Simulation Results

The proposed envelope following technique with the unifying phase condition of (21) and fast simulation techneques has been implemented in a comprehensive in-house C++ based SPICE simulation environment with interfaces to industry standard BSIM4 transistor-level models. We first test the generality of the proposed unifying solution by applying the formulation in (22) to an oscillator. Then we move on to the circuits of our interest, PWM, PFM and PSM DC-DC converters, and evaluate the efficiency and robustness of the proposed unifying method. We compare our method with the standard transient analysis in terms of speedups and accuracy and also comment on improvements over existing EF methods.

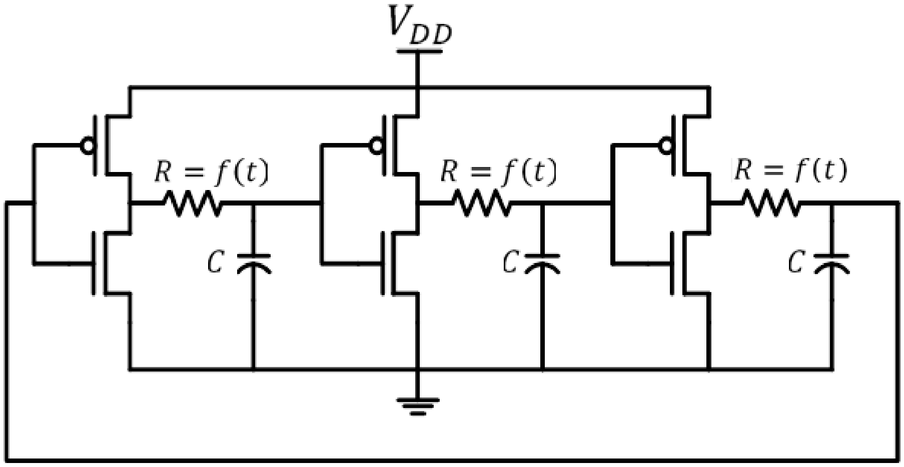


Figure 2.9: A three-stage ring oscillator with the value of R controlled by a function of time $f(t)$.

2.5.1 A three-stage ring oscillator

The oscillator of Fig. 2.9 has three identical stages and the resistor of each stage is controlled by the same time varying function $f(t)$. To evaluate the performance of the proposed unifying EF method, $f(t)$ is designed in such a way that the oscillator transits between transient state and steady state. As shown in Fig. 2.10, the frequency of the ring oscillator first linearly decreases from $2.025MHz$ to $2.01MHz$ in the first $2ms$ and then remains at about $2.01MHz$ for another $2ms$. Fig. 2.10 also shows in parallel the corresponding waveform of the second stage output nodal voltage simulated by the proposed EF method. From an initial value of $T_{env} = 32T_n$, the envelope step increases to $T_{env} = 128T_n$ during the transient phase. As circuit enters to steady state and the estimated LTE decreases, the envelope step further increases to $T_{env} = 512T_n$ until the simulation ends. Fig. 2.11 compares the proposed EF method with the conventional transient analysis. The EF results match with those of the transient analysis with a high accuracy. An average speedup of $60\times$ is achieved with respect to the transient analysis. Among the existing methods, [4] uses similar circuits and obtains speedup factors of about $35\times$. [33, 36]

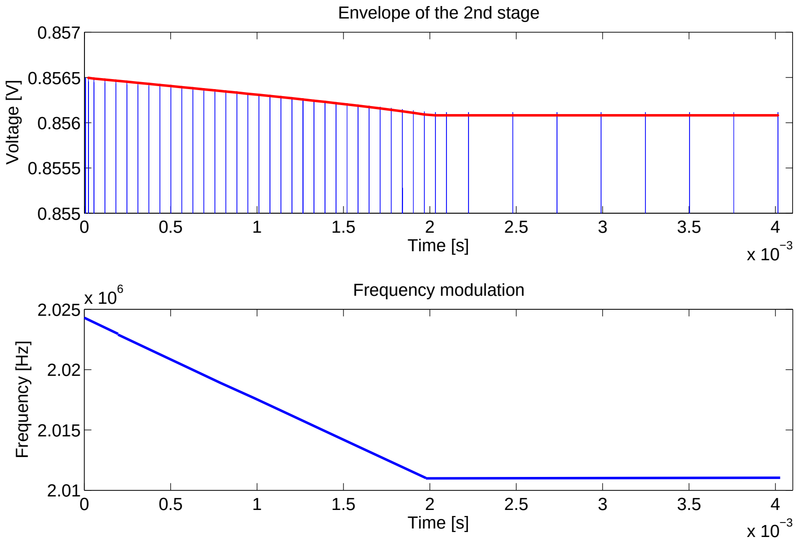


Figure 2.10: The envelope of the second stage output voltage of the ring oscillator (top) and its frequency variation (bottom) simulated by the proposed EF method. In the top figure, the red line is the envelope while the blue lines are the transient responses obtained in each cycle of envelope following.

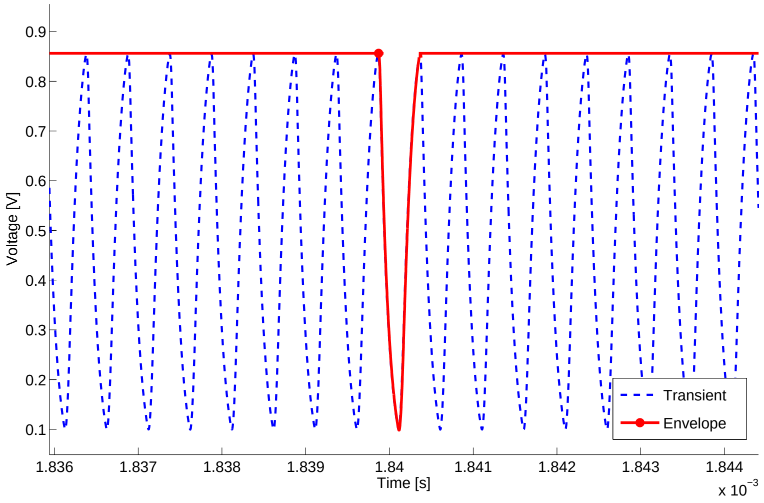


Figure 2.11: Comparison between the proposed EF analysis (red solid line) and transient analysis (blue dashed line) of the ring oscillator.

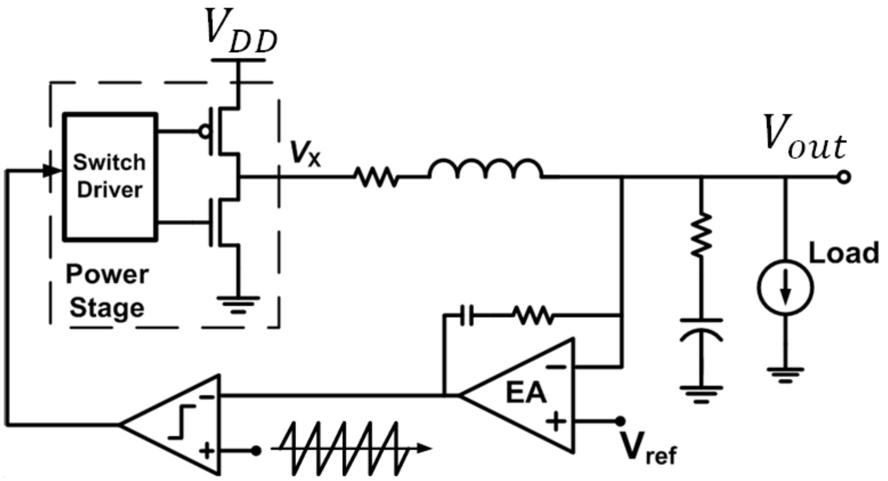


Figure 2.12: A PWM controlled DC-DC converter.

report speedup factors higher than $60\times$ for oscillators. Though proven to be effective for ring oscillators, the proposed method may not be optimized for high-Q oscillators like crystal oscillators, due to the fact that the Backward Euler integration method, which we use for inner-loop transient simulation, is a low-order method and has damping effects. In the next three examples we will simulate DC-DC converters with different modulation schemes, which are the main targeted circuits of our proposed methods.

2.5.2 A PWM controlled DC-DC converter

Next, we focus on our targeted class of circuits, DC-DC converters that posses strong nonlinearities, hard switching activities, digital/memory and hysteretic effects and strong feedback control. We first consider the startup transient of a standard DC-DC converter with PWM control that has all the essential real-life characteristics as shown in Fig. 2.12, where the switching frequency is determined by an external $10MHz$ triangular signal applied to one of the comparator's inputs. The transistor-level schematics of the two key blocks, the error amplifier (EA) and comparator, are shown in Fig. 2.13. While operating with PWM control under a fixed switching frequency, testing our unifying EF technique

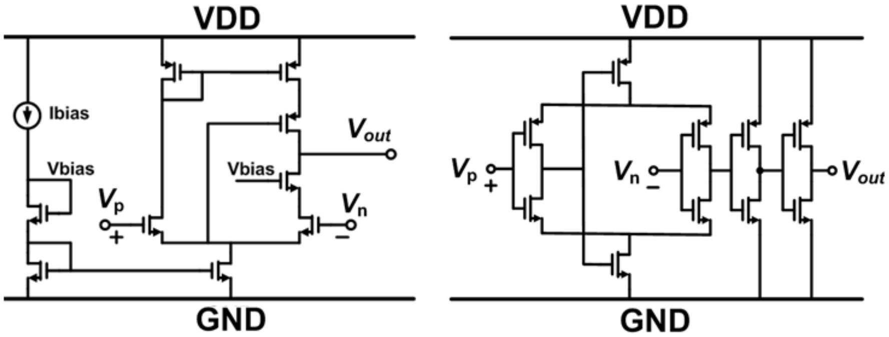


Figure 2.13: The error amplifier (left) and comparator (right) of the PWM controlled DC-DC converter.

without any algorithmic-level change on this circuit would demonstrate its generality.

Fig. 2.14 shows the envelope following results of the PWM converter output response. The envelope step is gradually increased until steady state is reached. Fig. 2.15 compares our EF method with the transient analysis for the early startup phase, where the output node has the most dynamic response, showing the ability of our method in closely tracking the changing amplitude of output voltage.

Overall a speedup factor of $20\times$ is achieved in this example. In comparison, speedup factors in the range of $6\times$ are reported for PWM converters in [2]. The simulation speedup improvement of our proposed method over [2] is mainly due to the use of the fast simulation techniques. First of all, the dynamic prediction scheme selects a mix of ISP_2 and ISP_3 at the beginning of the startup simulation. Then it switches to ISP_1 to predict the initial solution using the solution from previous step as the output voltage of the converter reaches the peak and the dynamic behaviors of the converter gradually settle down. The second technique that bring the performance improvement is the adaptive envelope step selection scheme. As shown in Fig. 2.14 the envelope step selection scheme dynamically adjust the envelope step size T_{env} of the algorithm based on the LTE of each step, which make best use of the multi-rate characteristics of DC-DC converters. Finally, the auto-

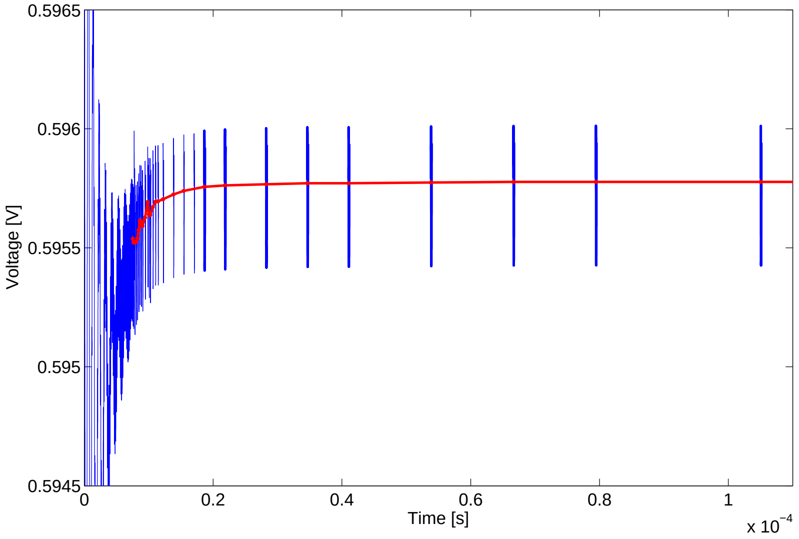


Figure 2.14: Simulation of the PWM DC-DC converter using the proposed EF method. The red line is the envelope of the output voltage while the blue lines are the transient responses obtained in each cycle of envelope following.

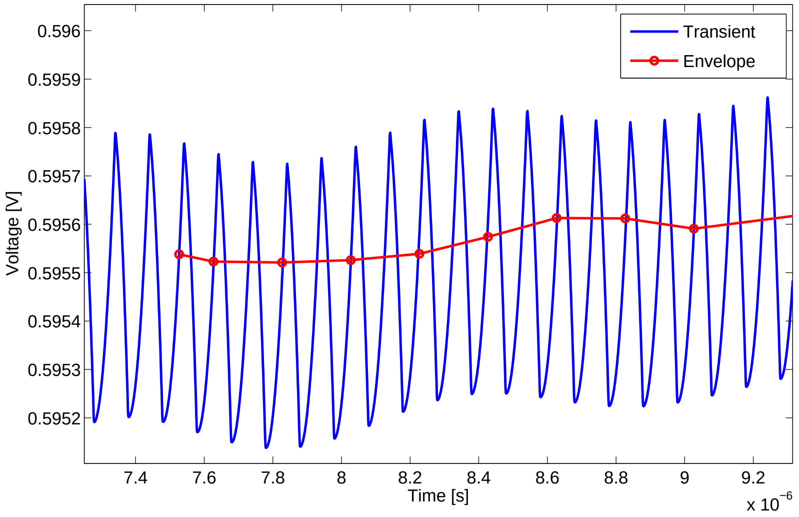


Figure 2.15: Detailed comparison of envelope following results (red) and transient simulation results (blue) of the PWM DC-DC converter.

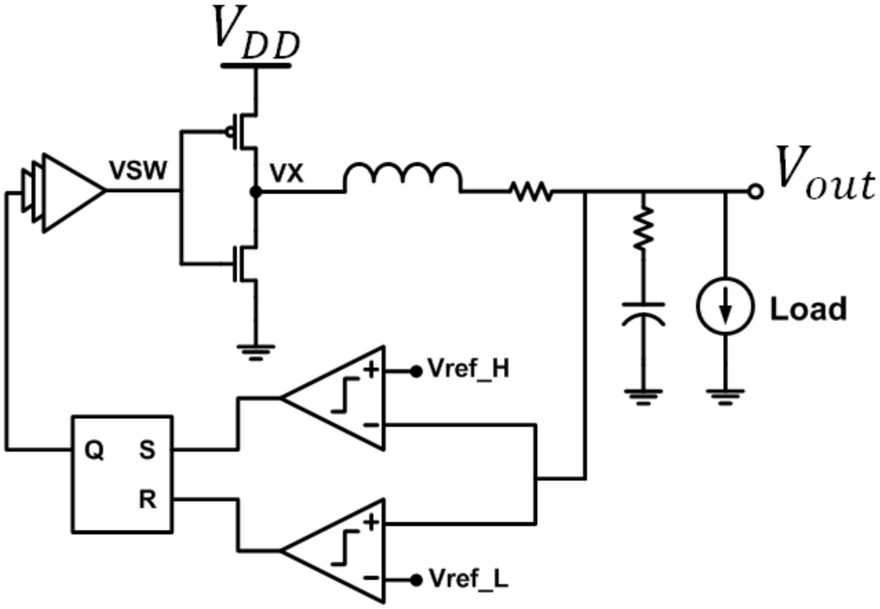


Figure 2.16: A hysteretic/PFM DC-DC converter.

matic node selection excludes nodes with digital behavior and sharp transitions from the convergence check, resulting in a fast convergence without compromising the accuracy.

2.5.3 A PFM DC-DC converter

The next circuit we consider is a hysteretic/PFM DC-DC converter in Fig. 2.16. This circuit, along with other autonomous DC-DC converters, are the primary target of our proposed method. As shown in Fig. 2.16, a hysteretic comparator that consists of two separate high-gain comparators constantly compare the output voltage with a high (V_{ref_H}) and low (V_{ref_L}) threshold voltages. The outputs of the hysteretic comparator feed an SR latch and then drive the two power switches. As such, the converter forces the output voltage to stay between V_{ref_L} and V_{ref_H} by dynamically adjusting the turn-on time of the PMOS power switch.

This is a challenging circuit for testing the robustness of the proposed EF technique due to its strong nonlinearity, high-gain blocks, hysteretic/memory effects and feedback

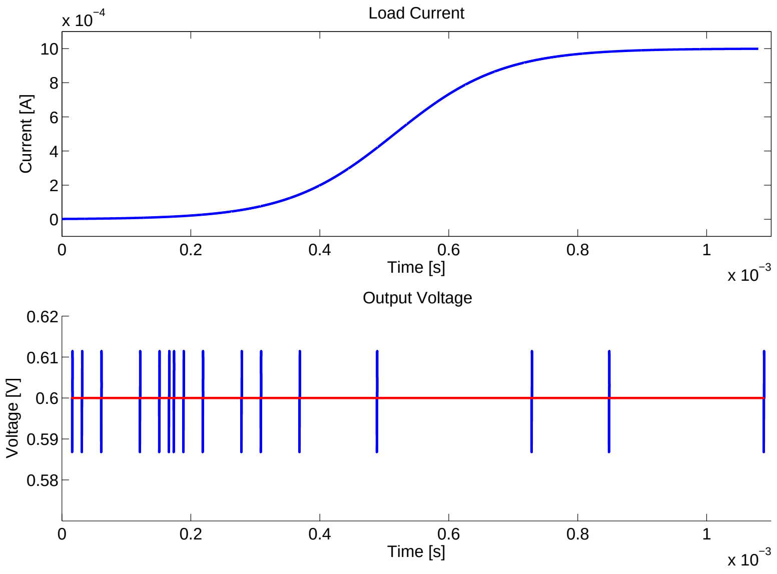


Figure 2.17: Load current waveform (top) and the envelope of the hysteretic converter output node response simulated by our EF method (bottom). In the bottom figure the red line is the envelope while the blue lines are the transient responses obtained in each cycle of envelope following.

control. Even for the standard transient analysis, a source ramping scheme and/or tight stepsize control need to be in place to guarantee convergence and accuracy.

The load current of the converter increases from $0mA$ to $1mA$ following a sigmoid function, resulting in a switching frequency change from 1.06MHz to 1.11MHz. The corresponding envelope of the output response is shown in Fig. 2.17. From a conservatively chosen envelope stepsize of $T_{env} = 16T_n$, the simulator gradually scales up the envelope stepsize according to the estimated LTE until the circuit enters into steady state. After that, the load current approximately linearly increases from $0.4ms$ to $0.7ms$. As shown in Fig. 2.17, during this period of time, the envelope stepsize actually increases. This boost of performance comes from an internal scheme for guessing the new $\boldsymbol{x}_n$ value at the beginning of each envelope step that is based upon a linear extrapolation of $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$, which are available before the Newton iteration starts. This prediction mechanism happens to

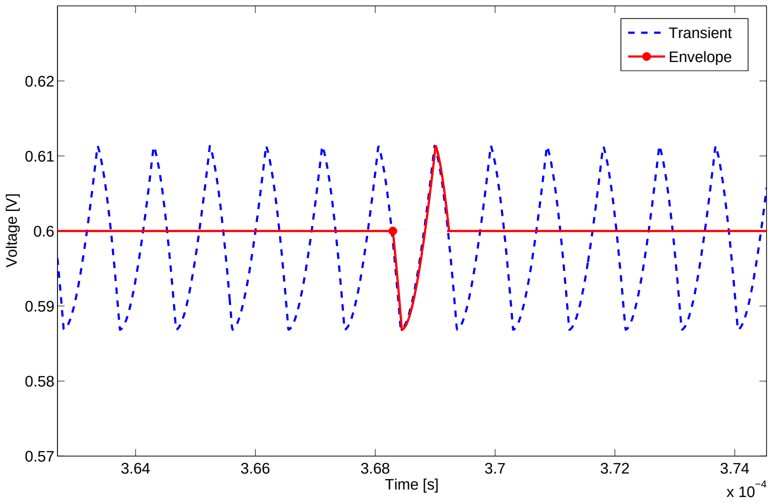


Figure 2.18: Detailed comparison of the proposed EF method (red solid line) with the transient analysis (blue dashed line) for the PFM/hysteretic converter.

provide fairly accurate initial guesses while the load current approximately rises up linearly. In conjunction with the implemented LTE stepsize control, the scheme boosts the envelope stepsize allowed by the LTE tolerance around time 6ms as shown in the bottom plot of Fig. 2.17. Fig. 2.18 compares the detailed EF results with the transient simulation results. The computed envelope accurately matches the transient simulation, demonstrating ability of our proposed method to closely track the envelope of highly dynamic and nonlinear converters with a varying switching frequency.

Our EF method achieves a speedup of $30\times$ with respect to the transient analysis for this example. Very few reported works target autonomous DC-DC converters under varying load conditions. [9] demonstrates analysis of two autonomous converters using constant envelope steps of 3 and 5 without reporting any speedup factors. As discussed in Section **??**, a simplistic switched linear system model is adopted to model converters with an additional assumption on variation of cycle time that is not generally true in [9].

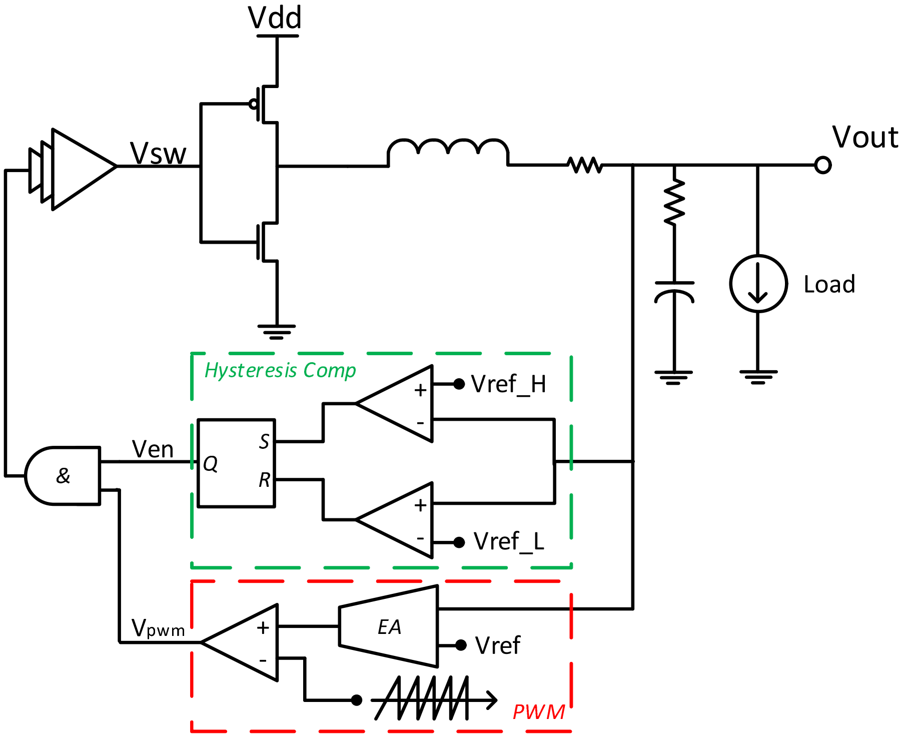


Figure 2.19: A PSM DC-DC converter.

2.5.4 A PSM DC-DC converter

The last circuit we consider is a pulse skipping modulated (PSM) DC-DC converter in Fig. 2.19, which represents one of the most popular DC-DC converter topologies in low-power design. This circuit is essentially an enhanced PWM DC-DC converter with the ability to skip pulses when the output voltage is above certain threshold voltage. The hysteretic comparator compares V_{out} with two threshold voltages V_{ref_L} and V_{ref_H} and generates a binary enabling signal V_{en}. The PWM module compares the voltage difference between reference voltage V_{ref} and generates a series of pulses V_{pwm} to drive the power switches. When the output voltage is below V_{ref_L}, the enabling signal V_{en} is high and the power switches are directly driven by the PWM pulse signals, resulting in an increase of output voltage. By the time V_{out} reaches the upper threshold voltage V_{ref_H}, the enabling

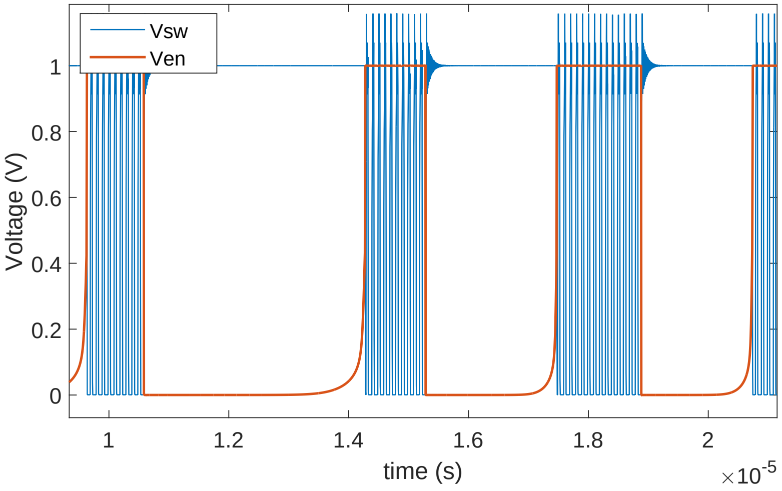


Figure 2.20: The switching signal V_{sw} and the enabling signal V_{en}.

signal V_{en} becomes zero and the pulse signal generated by the PWM module is blocked by the AND gate. Thus the converter skips all PWM pulse signals until V_{out} drops below the lower threshold again.

Fig. 2.20 shows the waveforms of V_{en} and V_{sw}. When enabling signal $V_{en} = 1$, the PWM module generates multiple pulses to drive the power switches. The occurrence and number of PWM pulses depend on the load condition and are not known *a priori*. Thus envelope following simulation of this type of circuits is extremely challenging. In order to robustly apply the proposed envelope following algorithm to this type of circuits, we pick V_{out} and V_{en} as the phase monitoring nodes and define the cycle of the circuit by the periodic behaviour of V_{en} rather than that of V_{sw}.

The results of envelope following simulation are shown in Fig. 2.21. The top waveform shows a linearly changing load current increased from $50mA$ to $51mA$. The bottom waveform shows the voltage responses of the output voltage V_{out}. Starting from $T_{env} = 8T$, our proposed algorithm automatically adjusts stepsize based on the estimated LTE values. With the help of prediction, our proposed algorithm maintains a rather large stepsize

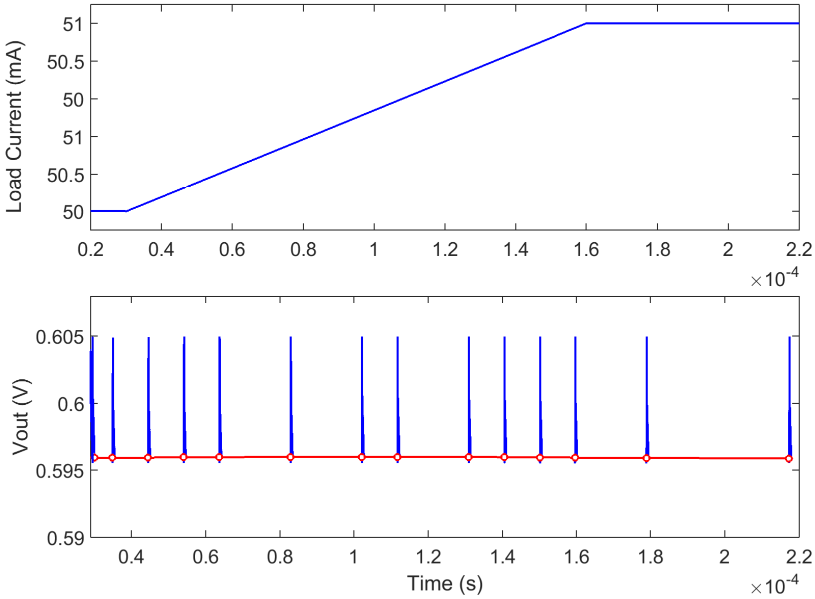


Figure 2.21: Linearly increasing load current (top) and the output node voltage response simulated by the proposed EF method (bottom). In the bottom figure the red line is the envelope while the blue lines are the transient responses obtained in each step of envelope following.

during the period when the load current linearly increases. The circuit finally reaches the steady state at $t = 0.18ms$ and after that the stepsize quickly increases to $T_{env} = 64T$. Overall, a speedup of $10X$ is achieved with respect to transient analysis.

Table 2.1 summarizes the simulation results of PWM, PFM and PSM DC-DC converters. The speedup of the proposed envelope following method with respect to the standard transient simulation are calculated based on the simulation time of both methods.

2.6 Summary

A robust and efficient envelope following method is presented for circuits with constant or variable switching frequencies. At the core of our new EF algorithm are a novel time-delayed equal-phase condition and a mechanism that smoothly tracks the transitions of the circuit state. The implementation of fast simulation technique improves the efficiency

Table 2.1: Simulation Results for DC-DC converters

		PWM	PFM	PSM
Number of nodes		22	31	41
Number of transient steps	TR	192,473	10,881,234	1,100,001
	EF	9,780	322,307	115,115
Total number of transient N-R iteration	TR	415,737	1,121,5154	1,374,336
	EF	20,737	365,974	132,590
Simulation time (sec)	TR	906	112,321	24,707
	EF	54	3,691	2,203
Speedup		20X	30X	11X

of the algorithm while maintaining the same accuracy level. We verify the robustness, generality and efficiency of the proposed technique using several test circuits for which our technique offers excellent simulation speedups and robustness.

3. MULTI-HARMONIC LARGE-SIGNAL NONLINEAR MODELING OF LOW-POWER PWM DC-DC CONVERTERS*

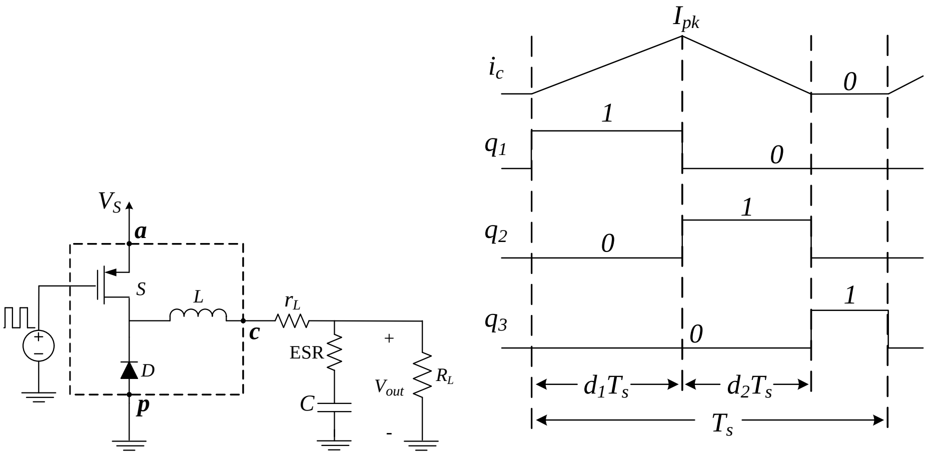
Beside envelope following introduced in Chapter 2, circuit averaging is another technique that is well suited for the efficient simulation of DC-DC converters.

In this chapter, we present a multi-harmonic averaged model for low-power PWM DC-DC converter. The proposed model combines the accuracy of enhanced state-space averaging, the flexibility of PWM switch models and the multi-frequency nature of the generalized models. The multi-harmonic model has four main features. First, the model approximates the actual converter responses by multiple harmonics of ripples up to an arbitrary degree. Second, the model accounts for the effects of device non-idealities like diode forward voltage drop in an efficient manner. The third feature lies in its generality. Our proposed model is based on the switch cell and can be applied to many DC-DC converters including buck, boost and buck-boost converters without any modifications. It is also general in the sense that both CCM and DCM operations are supported. Finally, a system decoupling technique is introduced to simulate converters with improved efficiency. As a result, when applied to several open-loop DC-DC converters, the proposed model generates almost identical responses as the transistor-level simulations with a runtime speedup of one order of magnitude.

3.1 Generalized Switch Model with Device Non-idealities

In this section, we present a generalized model of a three-terminal switch cell based on a standard buck converter shown in Fig. 3.1a. The generalized switch model can also be applied to other PWM DC-DC converters with no additional changes. To accurately

*©2016 IEEE. Reprinted, with permission, from Wang, Ya, et al. "Multi-harmonic nonlinear modeling of low-power PWM DC-DC converters operating in CCM and DCM." *Proceedings of the 2016 Conference on Design, Automation & Test in Europe*. EDA Consortium, 2016.



(a) Standard Buck converter

(b) The inductor current and the three switching functions.

Figure 3.1: Standard Buck PWM DC-DC converter

capture the circuit behavior, the dynamic non-idealities are accounted for in the switch cell. Furthermore, by continuously monitoring the circuit state, our model seamlessly transitions between CCM and DCM.

3.1.1 Equivalent switch model based on switching functions

Consider the buck converter shown in Fig. 3.1a operating in DCM with a discontinuous inductor current i_c shown in Fig. 3.1b. The switching cycle T_s is divided into three operation intervals by three binary switching functions $q_1(t)$, $q_2(t)$ and $q_3(t)$ using the parameters d_1 and d_2 as shown in Fig. 3.1b. Fig. 3.2 (a) shows the extracted switch cell while Figs. 3.2 (b),(c) and (d) show the three equivalent circuits corresponding to the three intervals.

Next, we analyze two circuit variables, v_{ag} and i_p, of particular interest in each operation interval and construct a complete nonlinear DCM switch model. In the first interval, the MOS transistor operates in the triode region with a nonlinear resistance of $R_{ds(on)}$ resulting in a voltage drop of $V_{ds(on)}$ while the diode is reverse biased. As shown in Fig.

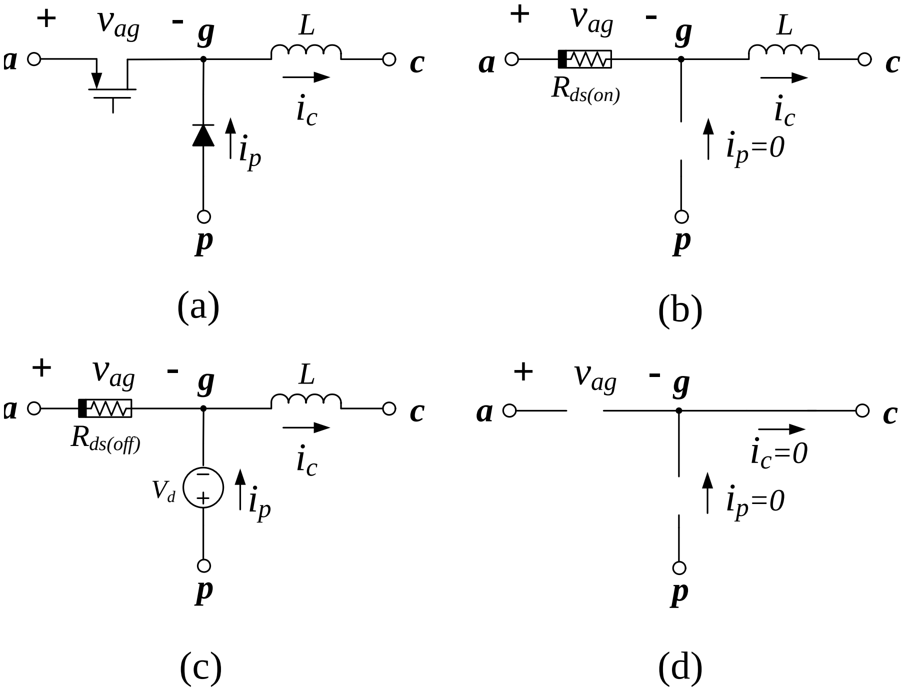


Figure 3.2: The three-terminal switch cell(a) and its equivalent circuits in three operation intervals (b)(c)(d).

3.2(b), in the first interval, we have $v_{ag} = q_1 V_{ds(on)}$ and $i_p = 0$. In the second interval, the diode is forward biased and holds a forward voltage drop V_d while the MOS transistor is operating in sub-threshold conduction with a leakage current I_{leak}. An equivalent resistor $R_{ds(off)}$ is used to model the effect of sub-threshold conduction. As shown in Fig. 3.2(c), in the second interval we have $v_{ag} = q_2 \left(v_{ap} + V_d\right)$ and $i_p = q_2 \left(i_c - I_{leak}\right)$. In the third interval, the inductor is completely discharged and the inductor current is zero. Both the transistor and the diode are off. According to the constitutive equation of inductors $v_L(t) = L\frac{di_L(t)}{dt}$, the inductor is shortened such that $v_{gc} = 0$. In the third interval, we have $V_{ag} = q_3 V_{ac}$ and $i_p = 0$. Combining the expressions of V_{ag} and i_p over the three intervals

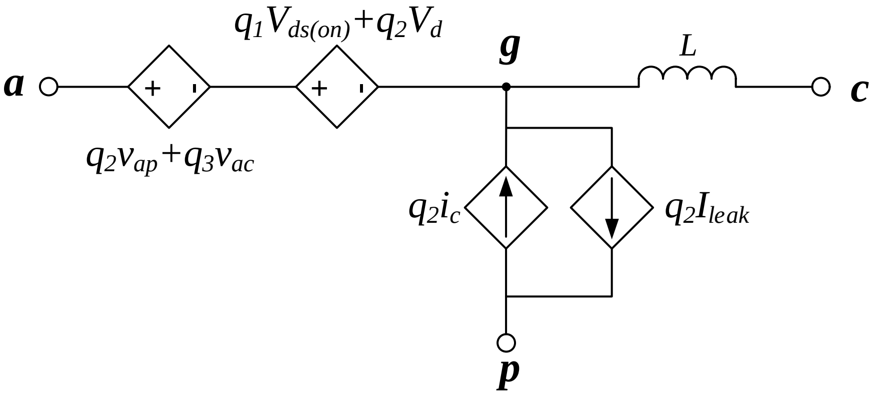


Figure 3.3: The equivalent model of the switch cell with non-ideal device characteristics.

results in:

$$v_{ag} = q_1(t)V_{ds(on)} + q_2(t)(v_{ap} + V_d) + q_3(t)v_{ac} \tag{3.1}$$

$$i_p = q_2(t)(i_c - I_{leak}). \tag{3.2}$$

We can now replace the transistor and the diode in the original circuit by controlled sources using the relations in (3.1) and in (3.2). This leads to the model shown in Fig. 3.3, which is essentially an equivalent switch model that accounts for the non-ideal device characteristics. Inserting the model of Fig. 3.3 in place of the switch cell in Fig. 3.1a results in a circuit model that accurately captures the behavior of the converter. The switch model can also be applied to converters operating in CCM with one additional constraint of $d_1+d_2 = 1$. The process of determining the operation mode is automated by calculating d_2 concurrently as the model is being simulated using

$$d_2 = \frac{2L}{d_1T_s}\frac{\bar{i}_c}{\bar{v}_{ac} - V_{ds(on)}} - d_1, \tag{3.3}$$

where $\bar{i}_c$ and $\bar{v}_{ac}$ are the average inductor current and the average voltage across nodes a

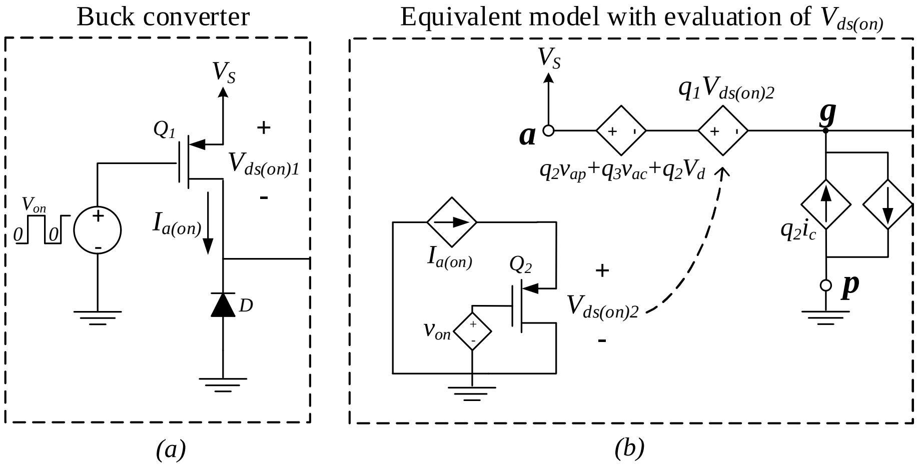


Figure 3.4: A buck converter operating at the first interval(a) and its equivalent model with an additional circuit to evaluate the voltage drop $V_{ds(on)}$ using an accurate device model(b).

and c in one cycle, both of which are immediately available in the multi-harmonic model as will be shown later. If $d_1 + d_2 < 1$, the converter operates in DCM. Otherwise, it operates in CCM with $d_2 = 1 - d_1$[15]. Note that(3.3) accounts for the non-idealities of the transistors and therefore results into a more accurate evaluation of d_2 than [13].

3.1.2 Non-ideal device modeling and the complete equivalent converter model

The accuracy of the model in Fig. 3.3 depends on the evaluation of $V_{ds(on)}$, V_d and I_{leak}, which are nonlinear functions of the circuit state and vary dynamically during the circuit simulation. These three values can be evaluated without simulating the actual buck converter, but by exploiting the information available in the equivalent model in Fig. 3.3. The process is illustrated by the example of computing $V_{ds(on)}$ shown in Fig. 3.4, where a buck converter is operating in the first interval with transistor Q_1 in the triode region with a limited conductance between the drain and the source. Due to the presence of a nonlinear resistance $R_{ds(on)}$, a voltage drop of $V_{ds(on)}$ is found between the drain and the source.

In the equivalent model, the evaluation of the voltage drop $V_{ds(on)}$ is achieved by adding the additional circuit shown in Fig. 3.4(b), which is constructed such that transistor Q_2 has the same bias (i.e. gate-to-source voltage and drain-to-source current) as transistor Q_1. To achieve this, we observe that in the first interval the drain-to-source current of Q_1 corresponds to $I_{a(on)}$ of the equivalent model and the gate-to-source voltage of Q_1 is V_{on}. Thus by forcing the drain-to-source current of Q_2 to be $I_{a(on)}$ and connecting the gate of Q_2 to the voltage source V_{on}, we ensure that the voltage drop produced by Q_2 is the same as that of Q_1 in the buck converter. Returning the voltage drop of Q_2 to the equivalent model allows the model to account for this type of non-ideality. It is important to note that the transistor Q_2 uses the same device model as Q_1(e.g. BSIM3/4). Thus the behaviour captured by transistor Q_2 fully accounts for the device non-idealities of Q_1.

Similar procedures can be used to obtain I_{leak} and V_d. With three additional circuits included, the complete equivalent model automatically accounts for all non-ideal device characteristics. By including simple additional transistor circuits as shown in Fig. 3.4(b), we are able to construct a coupled DC-DC converter model in which essential device non-idealities from the MOS switch and the diode can be accurately evaluated as part of the overall converter model.

3.2 Efficient Multi-harmonic Modeling of the Switch Cell

Fig. 3.3 presents a switch model with non-ideal device characteristics. However, it is essentially a switching circuit due to the presence of $q(t)$ and is computationally expensive for performing transient simulations. In this section, we propose a multi-harmonic averaged model that is both accurate and efficient. The proposed model approximates the true response of a DC-DC converter using a Fourier series of any order of choice. Since the switching functions $q(t)$ are decomposed into several Fourier series and are no longer present in the model, the proposed model is highly efficient.

3.2.1 Multi-harmonic model of switch cell

Multi-harmonic modeling decomposes each state variable into a Fourier series. Recall that a time-domain periodic signal $x(\tau)$ can be expanded using the Fourier series as

$$x(\tau) = \sum_{k=-\infty}^{\infty} \langle x \rangle_k(t) e^{jk\omega_s \tau}, \tag{3.4}$$

where $\omega_s = 2\pi f_s$ and f_s is the switching frequency. $\langle x \rangle_k(t)$ is the k^{th} complex Fourier coefficient[17], which is given by

$$\langle x \rangle_k(t) = \frac{1}{T_s} \int_{t-T_s}^{t} x(\tau) e^{-jk\omega_s \tau} d\tau. \tag{3.5}$$

As shown in (3.5), $\langle x \rangle_k(t)$ is defined by taking the average of $x(\tau)$ at a frequency kf_s. For instance, $\langle x \rangle_0(t)$ represents the dc component and $\langle x \rangle_1(t)$ represents the harmonic component at the fundamental frequency. Thus, we denote $\langle x \rangle_k(t)$ by the index-k average. By including higher orders of harmonics, the multi-harmonic model is able to approximate the non-linear behaviour of the switch cell with higher accuracy. However, as a result, the multi-harmonic model becomes more complex in circuit topology and more time-consuming to solve. Essentially a trade off needs to be found between the accuracy and efficiency of the multi-harmonic model. Going forward, we will develop the proposed model using only the index-0 and index-1 terms in the derivation as we believe they will present sufficient accuracy for most practical applications while still having the important ability to obtain design insights from the model that are critically important for low-power converter design, such as performing a stability analysis. It is important to note here that the developed models can be augmented by including higher orders of index-k averages when necessary. This would result in a multi-harmonic model that approximates the non-linear behavior of the switch model more accurately, but at the cost of increased complex-

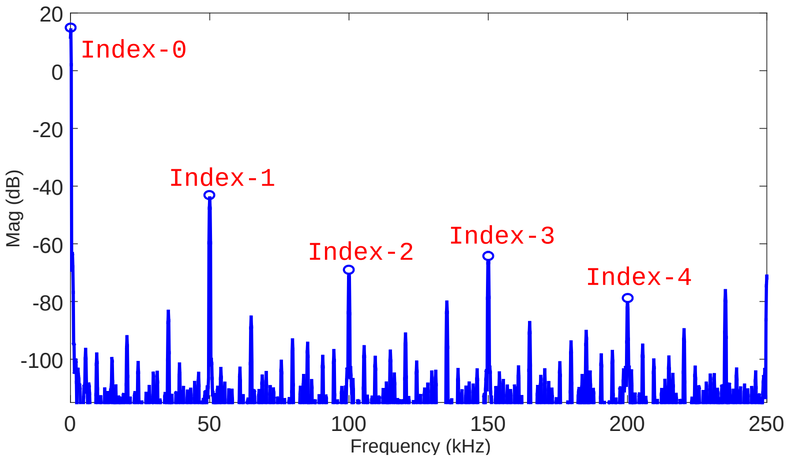


Figure 3.5: The FFT spectrum of the output voltage of a boost converter operating in the steady state.

ity. However, the fact that index-1 averages correspond to the most significant fundamental harmonics implies it is sufficient to only have index-0 and index-1 components for most practical cases of DC-DC converters which are not typically used as conventional analog circuits. This is more clearly illustrated in Fig. 3.5, which shows the FFT spectrum of the output voltage of a standard boost converter operating in the steady-state. As can be seen, the magnitudes of the index-0 and the index-1 components are measured to be $18dB$ and $-40dB$, respectively, which dominate the overall circuit response. In comparison, the magnitudes of all the other higher order harmonic components are less than $-60dB$, which is no more than 4% of the magnitude of the index-1 component, thus confirming our original assumption that by including the index-0 and the index-1 components, the multi-harmonic model is able to capture the circuit behavior with sufficient accuracy. This is also verified by the FFT analysis of a typical DC-DC converter in [37].

According to [18], two properties can be immediately identified from (3.4) and (3.5) that allow us to calculate the index-k averages of the switch model. The first property is discrete convolution. Using this property, each index-0 average term in (3.1) and (3.2) can be written as

$$\langle qx\rangle_k = \langle q\rangle_0\langle x\rangle_0 + \langle q\rangle_1\langle x\rangle_{-1} + \langle q\rangle_{-1}\langle x\rangle_1. \tag{3.6}$$

According to (3.5), every complex Fourier coefficient includes a real and an imaginary part. We also know that $\langle x\rangle_k$ and $\langle x\rangle_{-k}$ are conjugates of each other, which means $\langle x\rangle_k^R = \langle x\rangle_{-k}^R$ and $\langle x\rangle_k^I = -\langle x\rangle_{-k}^I$. Exploiting the conjugation property provides the following expression for the index-0 average of a product

$$\langle qx\rangle_0 = \langle q\rangle_0\langle x\rangle_0 + 2\left(\langle q\rangle_1^R\langle x\rangle_1^R + \langle q\rangle_1^I\langle x\rangle_1^I\right). \tag{3.7}$$

Substituting (3.7) into (3.1) and (3.2) gives the index-0 average model of the switch cell as follows

$$\begin{aligned}
\langle v_{ag}\rangle_0 =& \langle q_1\rangle_0 V_{ds(on)} + \langle q_2\rangle_0 V_d \\
&+\langle q_2\rangle_0\langle v_{ap}\rangle_0 + 2\left(\langle q_2\rangle_1^R\langle v_{ap}\rangle_1^R + \langle q_2\rangle_1^I\langle v_{ap}\rangle_1^I\right) \\
&+\langle q_3\rangle_0\langle v_{ac}\rangle_0 + 2\left(\langle q_3\rangle_1^R\langle v_{ac}\rangle_1^R + \langle q_3\rangle_1^I\langle v_{ac}\rangle_1^I\right) \\
\langle i_p\rangle_0 =& \langle q_2\rangle_0 \cdot I_{leak} \\
&+\langle q_2\rangle_0\langle i_c\rangle_0 + 2\left(\langle q_2\rangle_1^R\langle i_c\rangle_1^R + \langle q_2\rangle_1^I\langle i_c\rangle_1^I\right).
\end{aligned} \tag{3.8}$$

Similarly, the index-1 average of a product is given by

$$\langle qx\rangle_1^R = \langle q\rangle_0\langle x\rangle_1^R + \langle q\rangle_1^R\langle x\rangle_0 \tag{3.9}$$

$$\langle qx\rangle_1^I = \langle q\rangle_0\langle x\rangle_1^I + \langle q\rangle_1^I\langle x\rangle_0. \tag{3.10}$$

Substituting (3.9) and (3.10) into (3.1) and (3.2) gives the real part of the index-1 average

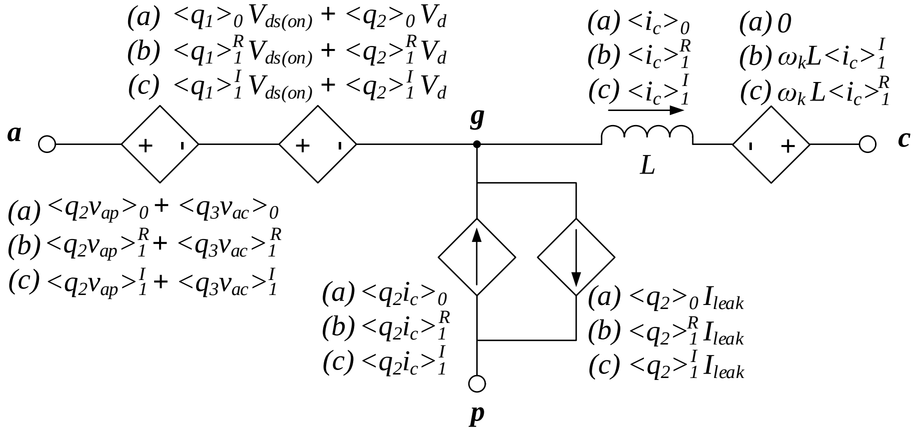

(a) $<q_1>_0 V_{ds(on)} + <q_2>_0 V_d$
(b) $<q_1>_1^R V_{ds(on)} + <q_2>_1^R V_d$
(c) $<q_1>_1^I V_{ds(on)} + <q_2>_1^I V_d$
(a) $<i_c>_0$
(b) $<i_c>_1^R$
(c) $<i_c>_1^I$
(a) 0
(b) $\omega_k L<i_c>_1^I$
(c) $\omega_k L<i_c>_1^R$
a
g
L
c
(a) $<q_2 v_{ap}>_0 + <q_3 v_{ac}>_0$
(b) $<q_2 v_{ap}>_1^R + <q_3 v_{ac}>_1^R$
(c) $<q_2 v_{ap}>_1^I + <q_3 v_{ac}>_1^I$
(a) $<q_2 i_c>_0$
(b) $<q_2 i_c>_1^R$
(c) $<q_2 i_c>_1^I$
(a) $<q_2>_0 I_{leak}$
(b) $<q_2>_1^R I_{leak}$
(c) $<q_2>_1^I I_{leak}$
p

Figure 3.6: The multi-harmonic model of the three-terminal the switch cell including (a) index-0 average model, (b) the real part of index-1 average model and (c) the imaginary part of index-1 average model.

model of the switch cell as

$$\begin{aligned}\langle v_{ag}\rangle_1^R =&\langle q_1\rangle_1^R V_{ds(on)} + \langle q_2\rangle_0 \langle v_{ap}\rangle_1^R + \langle q_2\rangle_1^R \langle v_{ap}\rangle_0\\ &+\langle q_2\rangle_1^R \cdot V_d + \langle q_3\rangle_0 \langle v_{ac}\rangle_1^R + \langle q_3\rangle_1^R \langle v_{ac}\rangle_0\\ \langle i_p\rangle_1^R =&\langle q_2\rangle_0 \cdot I_{leak} + \langle q_2\rangle_0 \langle i_c\rangle_1^R + \langle q_2\rangle_1^R \langle i_c\rangle_0,\end{aligned} \tag{3.11}$$

and the imaginary part of the index-1 average model as

$$\begin{aligned}\langle v_{ag}\rangle_1^I =&\langle q_1\rangle_1^I V_{ds(on)} + \langle q_2\rangle_0 \langle v_{ap}\rangle_1^I + \langle q_2\rangle_1^I \langle v_{ap}\rangle_0\\ &+\langle q_2\rangle_1^I \cdot V_d + \langle q_3\rangle_0 \langle v_{ac}\rangle_1^I + \langle q_3\rangle_1^I \langle v_{ac}\rangle_0\\ \langle i_p\rangle_1^I =&\langle q_2\rangle_0 \cdot I_{leak} + \langle q_2\rangle_0 \langle i_c\rangle_1^I + \langle q_2\rangle_1^I \langle i_c\rangle_0.\end{aligned} \tag{3.12}$$

The second property we exploit is the average of differentiation

$$\langle \frac{dx}{dt}\rangle_k(t) = \frac{d\langle x\rangle_k(t)}{dt} + jk\omega_s\langle x\rangle_k(t). \tag{3.13}$$

Applying (3.13) to the constitutive equation of inductors $v_L = L\frac{di_L}{dt}$ reveals that an index-k average of an inductor can be modeled by an inductor model with the same inductance in series with an impedance of $jk\omega_s L$.

Combining (3.8), (3.11), (3.12) and (3.13) produces the final model of the switch including index-0 and index-1 averages, as shown in Fig. 3.6. It should also be noted that a multi-harmonic model is generally highly coupled due to the interactions between averages of different orders.

3.2.2 Multi-harmonic model of the DC-DC converter

Apart from the switch cell, we need to find the index-k model for all other components of the DC-DC converter, e.g. the load capacitor and parasitic resistors. By the definition of index-k average shown in (3.5), it is easy to verify that $\langle v_R \rangle_k = R\langle i_R \rangle_k$, which implies the index-k average of a resistor has the same branch constitutive relationship of the resistor in terms of the branch voltage and current. Like the index-k average of an inductor, a physical capacitor can be modeled as an average capacitor model with the same capacitance in parallel with an admittance of $jkw_s C$.

Using the multi-harmonic average models for the switch cell in addition to the capacitors and the resistors derived above, we construct an index-0 and index-1 average model of a DC-DC converter. This results in three coupled circuits, which can be solved for the dc and the first-order harmonic responses.

3.3 System Decoupling

As pointed out earlier, solving mutually dependent averages of different orders is a computationally intensive task. In this section, we provide two rules for system decoupling that can remove the dependency of 0-index average models on all higher order models, which also improves the model accuracy.

Consider the standard buck converter shown in Fig. 3.1a and its 0-index average model

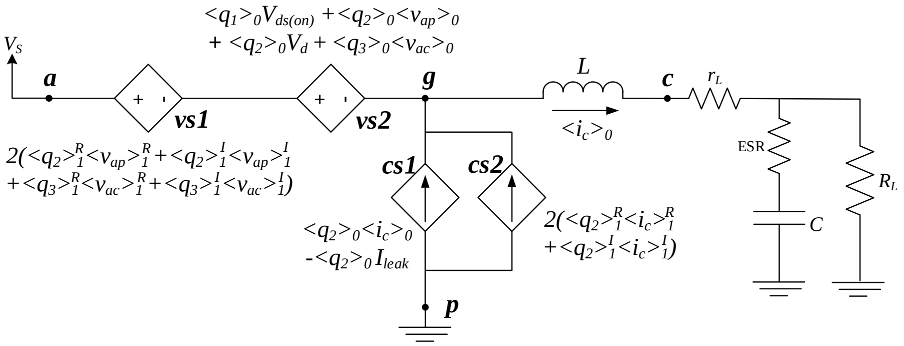

$<q_1>_0 V_{ds(on)} + <q_2>_0 <v_{ap}>_0$
$+ <q_2>_0 V_d + <q_3>_0 <v_{ac}>_0$
V_S
a
g
L
c
r_L
vs1
vs2
$<i_c>_0$
$2(<q_2>_1^R <v_{ap}>_1^R + <q_2>_1^I <v_{ap}>_1^I$
$+ <q_3>_1^R <v_{ac}>_1^R + <q_3>_1^I <v_{ac}>_1^I)$
cs1
cs2
ESR
R_L
$<q_2>_0 <i_c>_0$
$-<q_2>_0 I_{leak}$
$2(<q_2>_1^R <i_c>_1^R$
$+ <q_2>_1^I <i_c>_1^I)$
C
p

Figure 3.7: The index-0 average model of the buck converter. Controlled sources $vs1$ and $cs2$ depend on the index-1 average model and will be completely removed in the decoupled index-0 average model.

shown in Fig. 3.7. Our objective is to remove the VCVS $vs1$ and the CCCS $cs2$, both of which depend on components from the index-1 average model. Recall that the index-0 average of a product can be evaluated from the discrete convolution relation by (3.6). One useful observation is that if x has a very small variance within one cycle and hence can be viewed as a constant, then for any $k \neq 0$, $\langle x \rangle_k = 0$. Thus, we suggest the first rule of system decoupling:

Rule 1. *If x has small variance within one cycle, then $\langle qx \rangle_0 \approx \langle q \rangle_0 \langle x \rangle_0$. All higher order terms can be removed from the index-0 average calculation while only introducing negligible errors.*

In Fig. 3.6(a), there are three averages of product terms $\langle q_2 v_{ap} \rangle_0$, $\langle q_3 v_{ac} \rangle_0$ and $\langle q_2 i_c \rangle_0$. In the buck converter, we have $v_{ap} = V_s$ and $v_{ac} = V_s - V_{out}$, where V_s denotes the supply voltage and V_{out} dc component of the output voltage. Thus the voltage swings in v_{ap} and v_{ac} are small and can be neglected. Therefore, we can apply rule 1 to $\langle q_2 v_{ap} \rangle_0$ and $\langle q_3 v_{ac} \rangle_0$ for the case of buck converters. This results in the complete removal of $vs1$ in Fig. 3.7.

On the other hand, $\langle q_2 i_c \rangle_0$ cannot be simplified by rule 1 due to the large variance of i_c shown in Fig. 3.8(a). (3.7) computeds the discrete convolution of $\langle q_2 i_c \rangle_0$, which includes

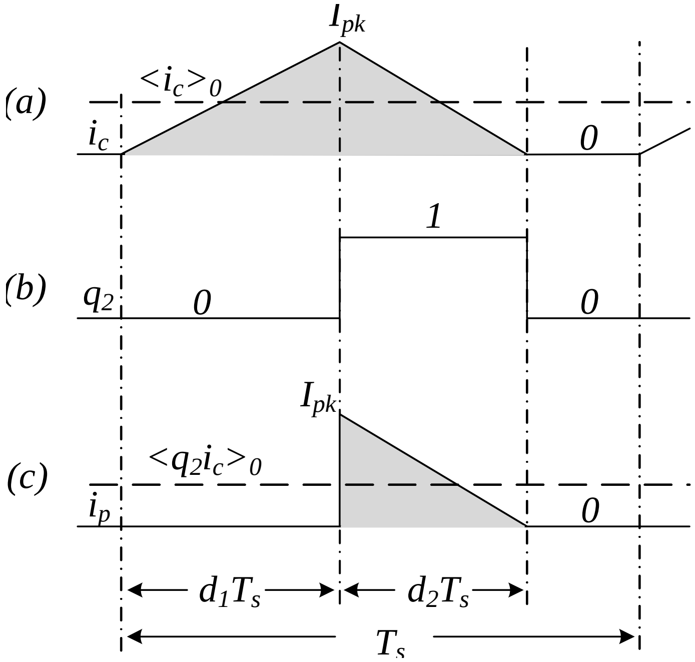


Figure 3.8: Illustration of $\langle q_2 i_c \rangle_0$ calculation: (a) inductor current i_c, (b) switch function q_2, and (c) product term $q_2 i_c$. The 0-index average can be calculated by dividing the shaded area by the length of interval.

the index-1 terms. To remove these terms, we need to consider the shape of i_c and q_2. In Fig. 3.8(a), the index-0 average value of i_c is essentially the dc value of the inductor current, which can be calculated by dividing the shaded area by the total time interval

$$\langle i_c \rangle_0 = \bar{i}_c = \frac{1}{2}(d_1 + d_2) I_{pk}, \tag{3.14}$$

where $d_1 = \langle q_1 \rangle_0$, $d_2 = \langle q_2 \rangle_0$ and I_{pk} is the peak value of i_c. Similarly, the index-0 average value of $q_2 i_c$ is

$$\langle q_2 i_c \rangle_0 = \bar{i}_p = \frac{1}{2} d_2 I_{pk}. \tag{3.15}$$

Substituting (3.14) into (3.15) leads to an equation that expresses $\langle q_2 i_c \rangle_0$ as

$$\langle q_2 i_c \rangle_0 = \frac{d_2}{d_1 + d_2} \langle i_c \rangle_0, \tag{3.16}$$

which is a function of only the index-0 average $\langle i_c \rangle_0$. Since no approximation is made in the derivation process, (3.16) is precise and equivalent to computing $\langle q_2 i_c \rangle_0$ using discrete convolution with infinite terms, which is not practical to compute directly. Thus replacing (3.7) by (3.16) not only decouples the system and improves efficiency but also results in a model with enhanced accuracy. As we will later show in a buck converter simulation example, simulation using the multi-frequency model presented in [18], which approximates $\langle q_2 i_c \rangle_0$ by (3.7), shows significant errors when the converter operates in DCM. While our proposed model using (3.16) accurately and efficiently captures converter behaviour in both CCM and DCM.

Now we suggest the second rule of system decoupling which also enhances the accuracy:

Rule 2. *For DC-DC converters in any condition,* $\langle q_2 i_c \rangle_0 = \frac{d_2}{d_1+d_2} \langle i_c \rangle_0$*. All higher order terms can be removed from the index-0 average calculation without introducing any errors.*

Applying rule 2 to $\langle q_2 i_c \rangle_0$ results in the removal of $cs2$ in Fig. 3.7 as well as changing the current source of $\langle q_2 \rangle_0 \langle i_c \rangle_0$ into $\frac{d_2}{d_1+d_2} \langle i_c \rangle_0$. Fig. 3.9 summarizes the decoupled buck converter index-0 average model where all of the components can be evaluated independently of high-order averages. Combining this model with the index-1 average model results in a more efficient and accurate multi-harmonic model.

3.4 Experimental Results

We test the accuracy and efficiency of our new proposed model by performing simulations of different DC-DC converters and compare the results with two existing averaged

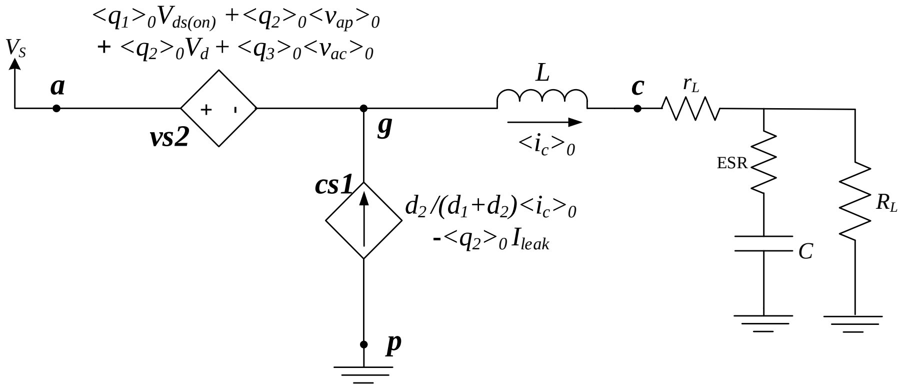


Figure 3.9: The decoupled index-0 average model of the buck converter.

models. The first example of the boost converter simulation demonstrates the improved accuracy of our model by considering device non-idealities. The second example of the buck converter simulation shows improved accuracy and efficiency by calculating $\langle q_2 i_c \rangle_0$ using (3.16). All models and circuits are implemented in Cadence Virtuoso and simulated by Spectre [38]. Our proposed model is described using Verilog-AMS and the converter circuits are described in transistor-level netlists with industry-standard BSIM4 models.

The accuracy of our proposed model is quantified using the following error metric defined with respect to transistor-level simulation:

$$\sigma = \frac{\sqrt{\sum_{k=1}^{n} \left(V_{model}(T_s \cdot k) - V(T_s \cdot k)\right)^2}}{\sqrt{\sum_{k=1}^{n} V(T_s \cdot k)^2}} \tag{3.17}$$

In (3.17), V is a targeted output voltage obtained by transistor-level simulation while V_{model} is the corresponding output voltage in our proposed model. T_s is the sampling stepsize which is set to one tenth of the switching period of a given converter circuit.

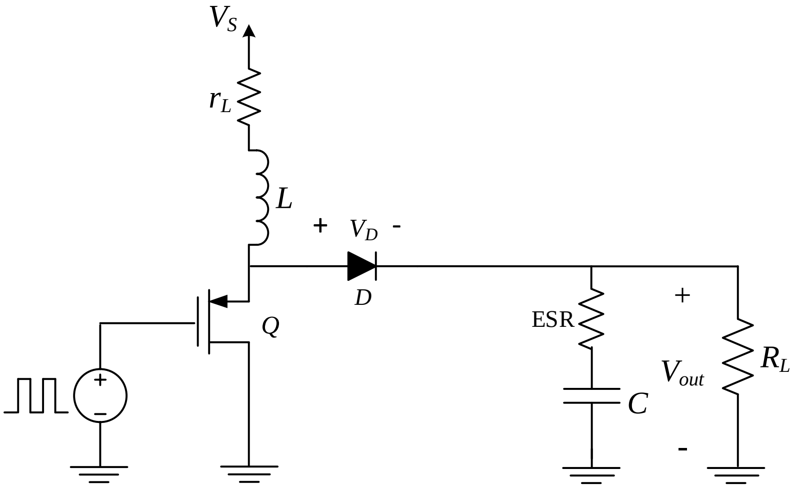


Figure 3.10: A standard boost converter with $V_s = 2V$, $L = 300\mu H$, $C = 1\mu F$ and $R_L = 50\Omega$.

3.4.1 Boost converter open-loop simulation

Consider the standard PWM boost converter shown in Fig. 3.10. A small capacitance value is selected for better demonstration of the ripple. The boost converter operates in CCM with a switching frequency of $f = 50$kHz. For this example, the converter simulation starts with a duty ratio of $d = 0.4$. This ratio is maintained until $t = 0.4$ms at which the duty ratio ramps up to 0.5 and remains at this level for the remainder of the simulation. Fig. 3.11 shows a comparison of the transistor-level simulation, our proposed model, and the equivalent circuit derived from the in-place circuit averaging with index-0 and index-1 averages considered [20]. The results from the in-place averaging show visible errors in the steady state and the transient state, because it fails to consider the effect of device non-idealities. On the other hand, the output voltage and the inductor current waveforms predicted by our proposed method match very well with the transistor level simulation. The percentage errors for the output voltage are found to be $\sigma_v = 3.46\%$ using (3.17). Furthermore, for this example, our proposed model achieved $6X$ runtime speedup with

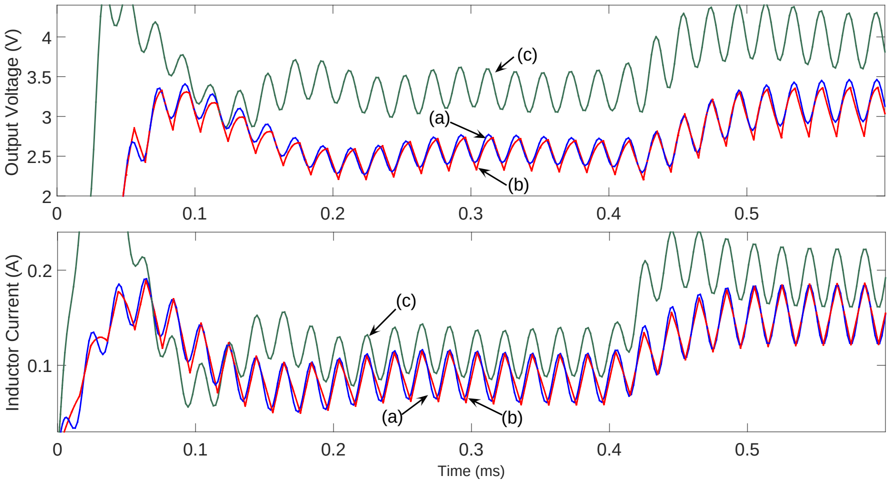


Figure 3.11: Boost converter open-loop simulation with varying duty ratio using our proposed model(a), transistor-level circuit(b), and the in-place circuit averaging technique(c).

respect to the transistor-level simulation.

3.4.2 Buck converter open-loop simulation

Next we test our proposed model on a buck converter operating in CCM and DCM. The stability and robustness of the proposed model is investigated by varying the duty ratio over a wide range. The standard buck converter circuit modeled is shown in Fig. 3.1a with $L = 100\mu H$, $C = 500nF$, $R_L = 10\Omega$ and $f_s = 50kHz$. As shown in Fig. 3.12(a), the duty ratio is varied in such a way that the converter transitions from CCM to DCM at $t = 0.2ms$.

The simulation results for the output voltage and inductor current of the transistor-level circuit, our proposed model and a modified version of the multi-frequency average model [18] are shown in Fig. 3.12(b)(c). The multi-frequency average model introduced in [18] does not consider device non-idealities and thus is inaccurate in low-power DC-DC converter simulations. To differ from the previous comparison with in-place averaging, we modified the multi-frequency average model by adding controlled sources that account

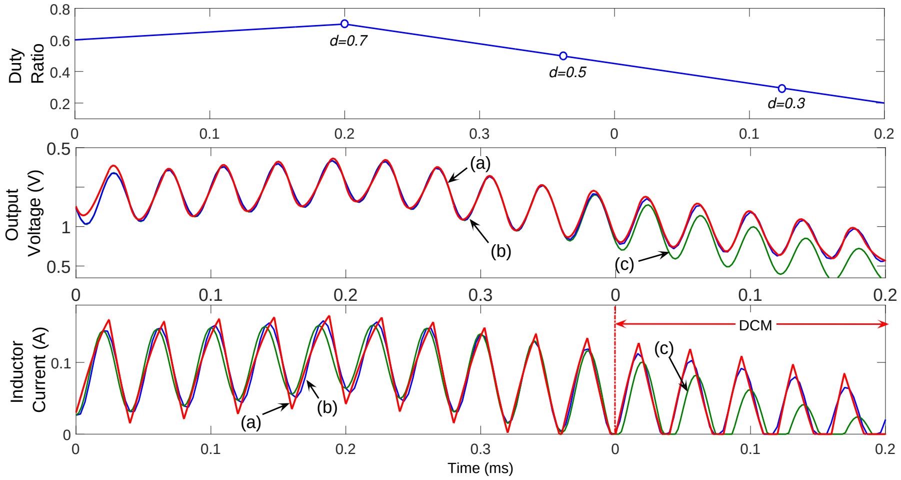


Figure 3.12: Buck converter open loop simulation of the transistor-level circuit(a), our proposed model(b), and the modified multi-frequency average model(c).

for device non-idealities including $R_{ds(on)}$, I_{leak} and V_d. As a result, both our model and the multi-frequency model make good predictions of the voltage and current responses in the first $0.2ms$ of simulation when the converter is operating in CCM. However, as the duty ratio continues to drop and the converter enters into DCM, the multi-frequency average model significantly deviates from the true response. The reason for the large errors is that the multi-frequency average model approximates $\langle q_2 i_c \rangle_0$ by (3.7), which only considers the 0-index and 1-index average and is not accurate when the inductor current i_c is discontinuous. Our proposed model, on the other hand, successfully detects the presence of the DCM and precisely calculates $\langle q_2 i_c \rangle_0$ using (3.16) and therefore closely tracks the actual circuit response.

In this example, our proposed model replaces the discontinuous inductor current with continuous averaged currents, which results in a $10X$ speedup with respect to the simulation of the transistor-level circuit. The output voltage error in this case is found to be $\sigma_v = 3.97\%$, which once again demonstrates the high accuracy of our proposed model. In comparison, the multi-frequency average model shows an error of $\sigma_v = 18.13\%$. Table 3.1

Table 3.1: Errors and speedups of the proposed model

	boost converter	buck converter
Transistor-level ckt size	11 nodes	8 nodes
Multi-harmonic ckt size	95 nodes	90 nodes
Error of multi-harmonic ckt	3.46%	3.97%
Speedup over transistor-level ckt	6X	10X

summarizes the speedup and error of our proposed model with respect to the transistor-level simulations.

3.5 Summary

In this chapter we present a multi-harmonic model for PWM DC-DC converters in various topologies. Our proposed model can capture the dc response as well as higher-order harmonics. As a full order model, it retains the inductor current as a state variable and is accurate even when the converter is in the transient state. Our model can seamlessly transition between CCM and DCM during the simulation. Moreover, we suggest two rules for system decoupling in order to achieve better efficiency without compromising accuracy. Our model was tested on two different DC-DC converters and speedups of one order of magnitude were achieved with respect to transistor-level simulations.

4. MULTI-HARMONIC SMALL-SIGNAL MODELING OF LOW POWER PWM DC-DC CONVERTERS [*]

As the complexity of the modern low-power integrated circuits grows exponentially, there is an increasing need for accurate control techniques in designing DC-DC converters [21]. Converter circuit behavior is typically highly nonlinear due to the strong switching activities and the presence of nonlinear devices. The small-signal model, which approximates the behavior of the DC-DC converter by linearizing the nonlinear devices and switches at a certain DC operating point, is widely used by designers to design the control blocks and closed-loop systems, as well as to analyze system stability.

The focus of this chapter is on presenting an efficient and accurate model for the analysis and design of low-power PWM converters. We first derive a multi-harmonic averaged model, which is a six-state variable system that considers both the DC response and the first-order harmonics as well as the interactions between them. We then develop a small-signal model to accurately capture the small-signal dependencies of each harmonic component on the targeted input perturbation. The proposed model is presented in both the state-space form and in the frequency domain. We perform an empirical analysis on the linear time-variant (LTV) of the proposed model and derive several key parameters to characterized the high-frequency behaviors. Then we compare the proposed small-signal model with the conventional small-signal model, which is based on the average model that only considers the DC component of the Fourier series of the signals [39], in two converter examples to demonstrate the significant accuracy enhancement that is offered by using the proposed model in high-frequency circuit responses. We will show that the proposed

[*] Wang, Ya, et al. "Multiharmonic Small-Signal Modeling of Low-Power PWM DC-DC Converters." *ACM Transactions on Design Automation of Electronic Systems (TODAES)* 22.4 (2017): 68.APA ©2017 ACM, Inc. Reprinted by permission. http://doi.acm.org/10.1145/3057274

model has a time-varying aspect and has the ability to identify misleading results that can result from using conventional small-signal models. Therefore, the proposed model provides the actual behavior of the converters that will, in turn, lead to stable closed-loop designs.

4.1 Multi-Harmonic Average Model

In this section, we revisit the large-signal multi-harmonic model introduced in Chapter 3 by deriving the averaged model for a boot converter shown in Fig. **??**. The multi-harmonic average model takes both the DC average and the multiple harmonic components into account [18]. Note that while the derivation will be based on the standard boost converter shown in Fig. 3.10, fundamentally similar derivations can be applied to other PWM DC-DC converter topologies, including buck and buck-boost type converters [40, 41].

We begin the derivation with the equations that describe the switched model of the boost converter operating in CCM and controlled by the switching function $q(t)$:

$$\frac{di(t)}{dt} = \frac{1}{L}(V_S - (1 - q(t))v(t)) \tag{4.1}$$

$$\frac{dv(t)}{dt} = \frac{1}{C}((1 - q(t))i(t) - \frac{1}{R}v(t)), \tag{4.2}$$

The switching function is binary and can be expressed as

$$q(t) = \begin{cases} 1 & t \in (0, dT_s) \quad \text{switch is ON} \\ 0 & t \in (dT_s, T_s) \quad \text{switch is OFF,} \end{cases} \tag{4.3}$$

where d is the duty ratio of the switching function and T_s is the switching period. Using the same derivation shown in 3.2, we obtain the index-0 average of (4.1) as

$$\begin{aligned}\frac{d\langle i\rangle_0}{dt} &= \frac{1}{L}\left(-\langle q'\rangle_0\langle v\rangle_0 + 2\left(\langle q\rangle_1^R\langle v\rangle_1^R + \langle q\rangle_1^I\langle v\rangle_1^I\right)\right) && (4.4)\\ \frac{d\langle v\rangle_0}{dt} &= \frac{1}{C}\left(\langle q'\rangle_0\langle i\rangle_0 - \frac{\langle v\rangle_0}{R} - 2\left(\langle q\rangle_1^R\langle i\rangle_1^R + \langle q\rangle_1^I\langle i\rangle_1^I\right)\right), && (4.5)\end{aligned}$$

where $q' = 1 - q$, $\langle\cdot\rangle_1^R$ and $\langle\cdot\rangle_1^I$ are the real part and imaginary part of the index-1 average $\langle\cdot\rangle_1$. Similarly, the index-1 average of the switch model (4.1) can be calculated as

$$\begin{aligned}\frac{d\langle i\rangle_1^R}{dt} &= \omega_s\langle i\rangle_1^I + \frac{1}{L}(-\langle q'\rangle_0\langle v\rangle_1^R + \langle v\rangle_0\langle q\rangle_1^R) && (4.6)\\ \frac{d\langle v\rangle_1^R}{dt} &= \omega_s\langle v\rangle_1^I + \frac{1}{C}\left(\langle q'\rangle_0\langle i\rangle_1^R - \langle i\rangle_0\langle q\rangle_1^R - \frac{\langle v\rangle_1^R}{R}\right) && (4.7)\\ \frac{d\langle i\rangle_1^I}{dt} &= -\omega_s\langle i\rangle_1^R + \frac{1}{L}(-\langle q'\rangle_0\langle v\rangle_1^I + \langle v\rangle_0\langle q\rangle_1^I) && (4.8)\\ \frac{d\langle v\rangle_1^I}{dt} &= -\omega_s\langle v\rangle_1^R + \frac{1}{C}\left(\langle q'\rangle_0\langle i\rangle_1^I - \langle i\rangle_0\langle q\rangle_1^I - \frac{\langle v\rangle_1^I}{R}\right). && (4.9)\end{aligned}$$

Now we have a highly-coupled system of two state variables of index-0 and four state variables of index-1, which define the dynamic behavior of the boost converter. The device non-ideality terms have been dropped due to the fact that they vanish during the small-signal analysis. Notice that (4.4)-(4.9) rely on the index-0 and index-1 averages of the switching function $q(t)$. Substituting (4.3) into (3.5) gives the relation of the index-0 and index-1 averages of the switching function to the duty ratio d as

$$\begin{aligned}\langle q\rangle_0 &= d && (4.10)\\ \langle q\rangle_1^R &= \frac{1}{2\pi}sin(\omega_s t + 2\pi d) && (4.11)\\ \langle q\rangle_1^I &= \frac{1}{2\pi}\left[cos(\omega_s t + 2\pi d) - 1\right]. && (4.12)\end{aligned}$$

Finally, by combining (4.4)-(4.5) with (4.6)-(4.9) and (4.10)-(4.12), the final multi-harmonic averaged model of the boost converter including index-0 and index-1 averages is specified.

4.2 Small-Signal State-Space Model

The small-signal model characterizes the sensitivities of the circuit state variables to the perturbation of parameters such as duty ratio or supply voltage. In this section we will focus on the perturbation of the duty ratio. Let us assume that the boost converter is in steady-state and there is a perturbation in the duty ratio of the form

$$d = \bar{d} + \hat{d}, \tag{4.13}$$

where $\bar{d}$ is the steady-state value of the duty ratio and $\hat{d}$ is the small-signal perturbation. The immediate effect of the duty ratio perturbation is the variation of switching functions. Based on the relations given by (4.10)-(4.13), the variations of the index-0 and index-1 averages of the switching function are

$$\langle \hat{q} \rangle_0 = \hat{d} \tag{4.14}$$

$$\langle \hat{q} \rangle_1^R = \frac{1}{2\pi}\left[sin\left(\omega_s t + 2\pi(\bar{d}+\hat{d})\right) - sin(\omega_s t + 2\pi\bar{d})\right] \tag{4.15}$$

$$\langle \hat{q} \rangle_1^I = \frac{1}{2\pi}\left[cos\left(\omega_s t + 2\pi(\bar{d}+\hat{d})\right) - cos(\omega_s t + 2\pi\bar{d})\right], \tag{4.16}$$

in which the $\langle \hat{q} \rangle_1^R$ and $\langle \hat{q} \rangle_1^I$ are nonlinear functions of the duty ratio perturbation $\hat{d}$. Next, we linearize (4.14)-(4.16) around $d = \bar{d}$ as $\begin{bmatrix} \langle \hat{q} \rangle_0 & \langle \hat{q} \rangle_1^R & \langle \hat{q} \rangle_1^I \end{bmatrix}^T = Q_s \cdot \hat{d}$, where

$$Q_s = \begin{bmatrix} 1 \\ cos\left(\omega_s t + 2\pi\bar{d}\right) \\ -sin\left(\omega_s t + 2\pi\bar{d}\right) \end{bmatrix} \tag{4.17}$$

is the switching function input matrix.

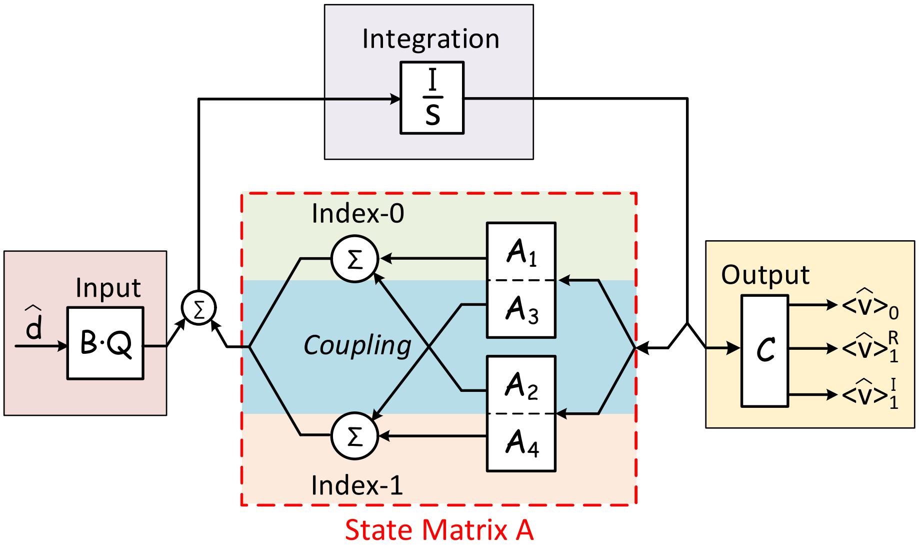


Figure 4.1: The state space of the multi-harmonic small-signal model and the interactions between its subsystems.

The effect of the perturbation continues to propagate to each of the state variables. Based on (4.4)-(4.5), the perturbation of index-0 state variables can be calculated as:

$$\begin{aligned}\frac{d\langle\hat{i}\rangle_0}{dt} &= \frac{1}{L}\left(((\langle\bar{q}\rangle_0 - 1)\langle\hat{v}\rangle_0 + \langle\bar{v}\rangle_0\langle\hat{q}\rangle_0\right) + \frac{2}{L}\left(\langle\bar{q}\rangle_1^R\langle\hat{v}\rangle_1^R + \langle\bar{v}\rangle_1^R\langle\hat{q}\rangle_1^R\right) \\ &\quad + \frac{2}{L}\left(\langle\bar{q}\rangle_1^I\langle\hat{v}\rangle_1^I + \langle\bar{v}\rangle_1^I\langle\hat{q}\rangle_1^I\right) \qquad (4.18)\\ \frac{d\langle\hat{v}\rangle_0}{dt} &= \frac{1}{C}\left((1 - \langle\bar{q}\rangle_0)\langle\hat{i}\rangle_0 + \langle\bar{i}\rangle_0\langle\hat{q}\rangle_0 - \frac{\langle\hat{v}\rangle_0}{R}\right) - \frac{2}{C}\left(\langle\bar{q}\rangle_1^R\langle\hat{i}\rangle_1^R + \langle\bar{i}\rangle_1^R\langle\hat{q}\rangle_1^R\right) \\ &\quad - \frac{2}{C}\left(\langle\bar{q}\rangle_1^I\langle\hat{i}\rangle_1^I + \langle\bar{i}\rangle_1^I\langle\hat{q}\rangle_1^I\right). \qquad (4.19)\end{aligned}$$

Similarly, the perturbation of the index-1 state variables can be calculated as:

$$\frac{d\langle\hat{i}\rangle_1^R}{dt} = \omega_s\langle\hat{i}\rangle_1^I + \frac{1}{L}\left(-\langle\hat{v}\rangle_1^R + \langle\bar{q}\rangle_0\langle\hat{v}\rangle_1^R + \langle\bar{v}\rangle_1^R\langle\hat{q}\rangle_0 + \langle\bar{q}\rangle_1^R\langle\hat{v}\rangle_0 + \langle\bar{v}\rangle_0\langle\hat{q}\rangle_1^R\right) \tag{4.20}$$

$$\frac{d\langle\hat{v}\rangle_1^R}{dt} = \omega_s\langle\hat{v}\rangle_1^I + \frac{1}{C}\left(\langle\hat{i}\rangle_1^R - \frac{\langle\hat{v}\rangle_1^R}{R} - \langle\bar{q}\rangle_0\langle\hat{i}\rangle_1^R - \langle\bar{i}\rangle_1^R\langle\hat{q}\rangle_0 - \langle\bar{q}\rangle_1^R\langle\hat{i}\rangle_0 + \langle\bar{i}\rangle_0\langle\hat{q}\rangle_1^R\right) \tag{4.21}$$

$$\frac{d\langle\hat{i}\rangle_1^I}{dt} = -\omega_s\langle\hat{i}\rangle_1^R + \frac{1}{L}\left(-\langle\hat{v}\rangle_1^I + \langle\bar{q}\rangle_0\langle\hat{v}\rangle_1^I + \langle\bar{v}\rangle_1^I\langle\hat{q}\rangle_0 + \langle\bar{q}\rangle_1^I\langle\hat{v}\rangle_0 + \langle\bar{v}\rangle_0\langle\hat{q}\rangle_1^I\right) \tag{4.22}$$

$$\frac{d\langle\hat{v}\rangle_1^I}{dt} = -\omega_s\langle\hat{v}\rangle_1^R + \frac{1}{C}\left(\langle\hat{i}\rangle_1^I - \frac{\langle\hat{v}\rangle_1^I}{R} - \langle\bar{q}\rangle_0\langle\hat{i}\rangle_1^I - \langle\bar{i}\rangle_1^I\langle\hat{q}\rangle_0 - \langle\bar{q}\rangle_1^I\langle\hat{i}\rangle_0 + \langle\bar{i}\rangle_0\langle\hat{q}\rangle_1^I\right) \tag{4.23}$$

Combining (4.18)-(4.19) with (4.20)-(4.23) and rewriting the equations into a state-space representation gives:

$$\frac{d}{dt}\begin{bmatrix}\langle\vec{x}\rangle_0 \\ \langle\vec{x}\rangle_1\end{bmatrix} = \begin{bmatrix}A_1 & A_2 \\ A_3 & A_4\end{bmatrix}\begin{bmatrix}\langle\vec{x}\rangle_0 \\ \langle\vec{x}\rangle_1\end{bmatrix} + \begin{bmatrix}B_1 \\ B_2\end{bmatrix}\cdot Q_s\cdot\hat{d}, \tag{4.24}$$

where $\langle\vec{x}\rangle_0 = \begin{bmatrix}\langle\hat{i}\rangle_0 & \langle\hat{v}\rangle_0\end{bmatrix}^T$ is a two dimensional vector of index-0 state variables and $\langle\vec{x}\rangle_1 = \begin{bmatrix}\langle\hat{i}\rangle_1^R & \langle\hat{v}\rangle_1^R & \langle\hat{i}\rangle_1^I & \langle\hat{v}\rangle_1^I\end{bmatrix}^T$ is a four dimensional vector of index-1 state variables. The state matrix A and input matrix B are partitioned to form two subsystems. A_1 and A_4 represent the subsystems of the index-0 components and index-1 components. A_2 and A_3 represent the interaction between those two subsystems. Similarly, B_1 and B_2 represent the input to the index-0 and index-1 subsystems. The submatrices in state matrix A and the input matrix B are

$$A_1 = \begin{bmatrix} 0 & \frac{\langle\bar{q}\rangle_0 - 1}{L} \\ \frac{1-\langle\bar{q}\rangle_0}{C} & -\frac{1}{RC} \end{bmatrix} \tag{4.25}$$

$$A_2 = \begin{bmatrix} 0 & \frac{2\langle\bar{q}\rangle_1^R}{L} & 0 & \frac{2\langle\bar{q}\rangle_1^I}{L} \\ 0 & \frac{-2\langle\bar{q}\rangle_1^R}{C} & 0 & \frac{-2\langle\bar{q}\rangle_1^I}{C} \end{bmatrix} \tag{4.26}$$

$$A_3 = \begin{bmatrix} 0 & \frac{\langle\bar{q}\rangle_1^R}{L} \\ \frac{-\langle\bar{q}\rangle_1^R}{C} & 0 \\ 0 & \frac{\langle\bar{q}\rangle_1^I}{L} \\ \frac{-\langle\bar{q}\rangle_1^I}{C} & 0 \end{bmatrix} \tag{4.27}$$

$$A_4 = \begin{bmatrix} 0 & \frac{\langle\bar{q}\rangle_0 - 1}{L} & \omega_s & 0 \\ \frac{1-\langle\bar{q}\rangle_0}{C} & -\frac{1}{RC} & 0 & \omega_s \\ -\omega_s & 0 & 0 & \frac{\langle\bar{q}\rangle_0 - 1}{L} \\ 0 & -\omega_s & \frac{1-\langle\bar{q}\rangle_0}{C} & -\frac{1}{RC} \end{bmatrix} \tag{4.28}$$

$$B_1 = \begin{bmatrix} \frac{\langle\bar{v}\rangle_0}{L} & \frac{2\langle\bar{v}\rangle_1^R}{L} & -\frac{2\langle\bar{v}\rangle_1^I}{L} \\ -\frac{\langle\bar{i}\rangle_0}{C} & \frac{-2\langle\bar{i}\rangle_1^R}{C} & \frac{2\langle\bar{i}\rangle_1^I}{L} \end{bmatrix} \tag{4.29}$$

$$B_2 = \begin{bmatrix} \frac{\langle\bar{v}\rangle_1^R}{L} & \frac{\langle\bar{v}\rangle_0}{L} & 0 \\ \frac{-\langle\bar{i}\rangle_1^R}{C} & \frac{\langle\bar{i}\rangle_0}{C} & 0 \\ \frac{\langle\bar{v}\rangle_1^I}{L} & 0 & -\frac{\langle\bar{v}\rangle_0}{L} \\ \frac{-\langle\bar{i}\rangle_1^I}{C} & 0 & -\frac{\langle\bar{i}\rangle_0}{C} \end{bmatrix}. \tag{4.30}$$

The system described in (4.24) is a complete characterization of the multi-harmonic small-signal model for the boost converter. The system diagram is shown in Fig. 4.1, with an output matrix C that selects all voltage harmonic components as the outputs. Compared to the conventional small-signal model, which is a small-signal DC averaged model with a state matrix equal to A_1, the multi-harmonic small-signal model provides more accurate circuit behavior by modeling the index-0 and index-1 components, as well as the interactions between them. This model can be easily extended to include harmonic components of an arbitrary degree.

4.3 Small-signal AC model

In this section, we cast the proposed small-signal state-space model to the frequency domain to derive a linear time-varying AC model . We show how our AC model can immediately capture important high-frequency characteristics and lead to useful design insights.

4.3.1 Frequency-domain small-signal model

The state-space model of Fig. 4.1 outputs the index-0, and the real and imaginary parts of the index-1 components of the converter output voltage. These components can be combined to form the total voltage response

$$\hat{v}(t) = \langle \hat{v} \rangle_0(t) + \langle \hat{v} \rangle_1(t) e^{j\omega_s t} + \langle \hat{v} \rangle_{-1}(t) e^{-j\omega_s t}, \tag{4.31}$$

where $v(t)$ is a combined output, and for convenience we have represented the index-1 component using complex exponentials. Applying the Laplace transform to (4.31) gives the frequency-domain representation

$$\hat{v}(s) = \int_0^\infty \langle \hat{v} \rangle_0(t) e^{-st} dt + \int_0^\infty \langle \hat{v} \rangle_1(t) e^{(j\omega_s - s)t} dt + \int_0^\infty \langle \hat{v} \rangle_{-1}(t) e^{-(j\omega_s + s)t} dt, \tag{4.32}$$

which can be further simplified to

$$\hat{v}(s) = \langle \hat{v} \rangle_0(s) + \langle \hat{v} \rangle_1(s - j\omega_s) + \langle \hat{v} \rangle_{-1}(s + j\omega_s). \tag{4.33}$$

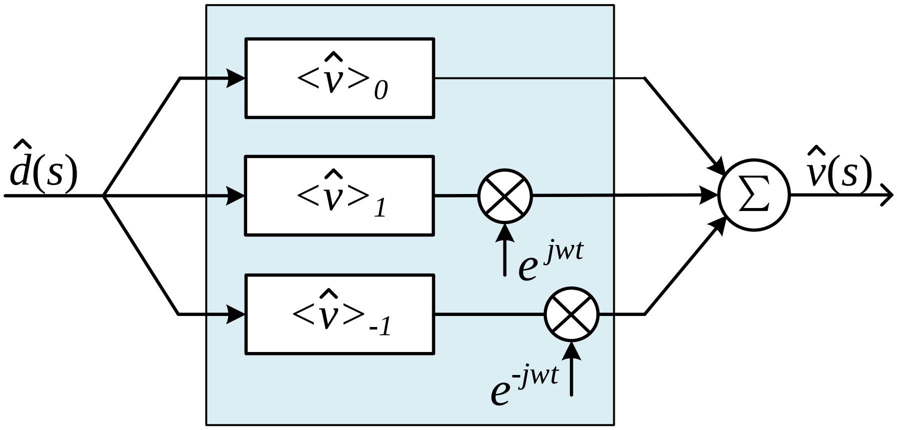


Figure 4.2: Illustration of the linear time-variant (LTV) system with the frequency-shift effect.

The small signal model of (4.33) is a linear time-variant (LTV) system, which is illustrated in Fig. 4.2. Each of $\langle\hat{v}\rangle_0$, $\langle\hat{v}\rangle_1$, and $\langle\hat{v}\rangle_{-1}$ can be characterized using a scalar LTI transfer function derived based on the small-signal state-space model. Note that the index-1 components $\langle\hat{v}\rangle_1(t)$ and $\langle\hat{v}\rangle_{-1}(t)$ are modulated by periodic signals $e^{j\omega_s t}$ and $e^{-j\omega_s t}$, which have frequency-shift effects in the frequency-domain response (4.33).

4.3.2 Parametric dependencies of high-frequency behavior

It is worth noticing that the proposed model captures the important high-frequency characteristics based on two mechanisms. First, as a model with six state variables, the proposed model accounts for high-frequency poles and zeros that are missing from traditional second-order models such as the ones presented in [39]. Second, the inclusion of the index-1 components takes into account additional high-frequency behaviors due to the frequency-shift effects.

Recall that the state matrix A from (4.24) is a 6×6 matrix that has four submatrices, where A_1 and A_4 correspond to the subsystem of the index-0 and index-1 harmonic components. As pointed out in [42], there is a distinctive model separation in system A and we can closely approximate the eigenvalues of A using those of the submatrices A_1 and A_4. From (4.28), we evaluate the eigenvalues of A_4 analytically [18]

$$\lambda_{1,2}(A_4) = \quad -\alpha \pm j(-\omega_s + \sqrt{{\omega_n}^2 - \alpha^2}) \tag{4.34}$$

$$\lambda_{3,4}(A_4) = \quad -\alpha \pm j(\omega_s + \sqrt{{\omega_n}^2 - \alpha^2}), \tag{4.35}$$

where $\omega_n = \frac{1-d}{\sqrt{LC}}$, $\alpha = \frac{1}{2R_LC}$ and ω_s is the switching frequency of the converter. In our case, system eigenvalues are actually the poles for the corresponding transfer functions. For each pair of complex conjugates we derive the quality factor Q, which indicates the magnitude of the resonant overshoot, and the angular corner frequency ω_0, which indicates the location of the corresponding overshoot. For boost converters, we have

$$\begin{aligned} Q &= \frac{1}{2\alpha}\sqrt{\alpha^2 + \left(\omega_s \pm \sqrt{{\omega_n}^2 - \alpha^2}\right)^2} \\ &\approx \frac{1}{2\alpha} \cdot |\omega_s \pm \omega_n| \end{aligned} \tag{4.36}$$

and

$$\begin{aligned} \omega_0 &= \sqrt{\alpha^2 + \left(\omega_s \pm \sqrt{{\omega_n}^2 - \alpha^2}\right)^2} \\ &\approx |\omega_s \pm \omega_n|. \end{aligned} \tag{4.37}$$

The effects of Q and ω_0 on resonant overshoot are illustrated in Fig. 4.3. From (4.37), it is clear that the overshoots appear around the switching frequency ω_s, and hence have significant effect on the converter's high-frequency response. For converter systems with Q larger than 1, there will be large overshoots/spikes around the switching frequency, which can potentially lead to stability problems. In the following converter design examples, we will show how high-frequency overshoots eventually lead to instability of the closed-loop system and how this can be fixed by adjusting converter design parameter to reduce the

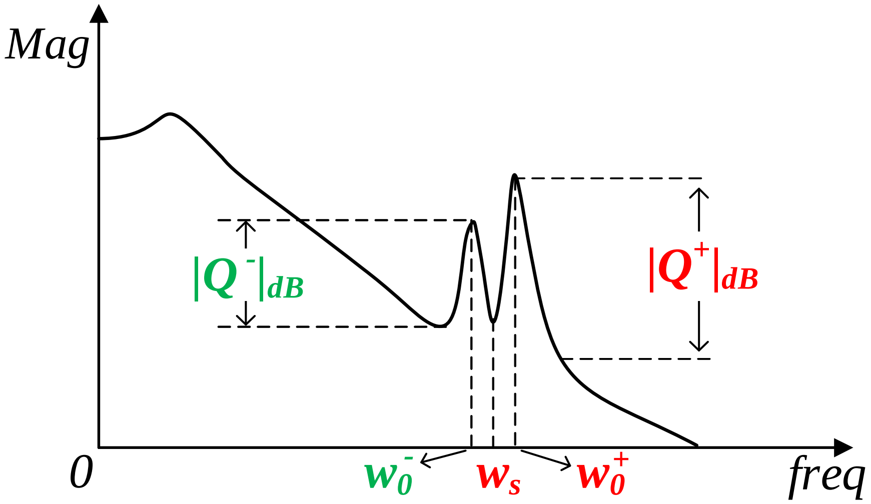


Figure 4.3: The effect of Q and ω_0 on a boost converter frequency response.

value of Q.

4.4 Experimental Results

We demonstrate the application of the proposed model using two different converter examples. The two converter types selected are the boost converter shown in Fig. **??** and the buck-boost converter shown in Fig. 4.4. The nominal duty ratio for both converters is 0.4 with the full set of parameters for both converter types as shown in Table I. We analyze the frequency response of each converter circuit using the proposed small-signal model and compare the results with the frequency response obtained using the conventional small-signal model which is based on the average model that only considers the DC component of the Fourier series of the signals [39]. We also compared our proposed model with some reference values, which are the actual gain values obtained from a transient analysis. These reference values were calculated by first perturbing the duty ratio by a small amount (0.001 or 0.1$\%$ in both cases) and then measuring the resulting perturbation of the average of the output voltage. The perturbation is a sinusoidal waveform and the commercial tool

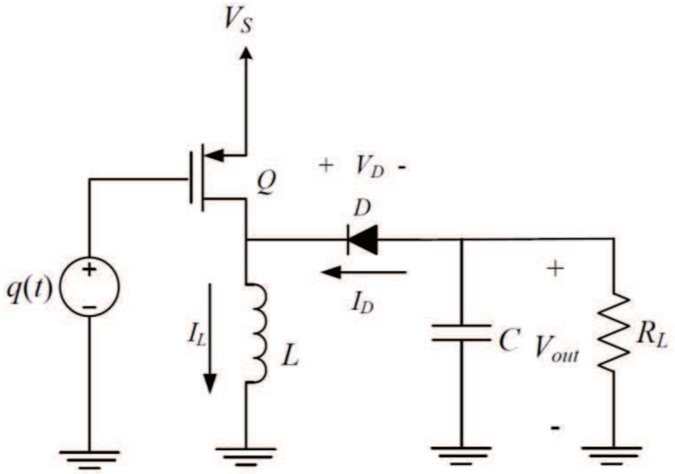
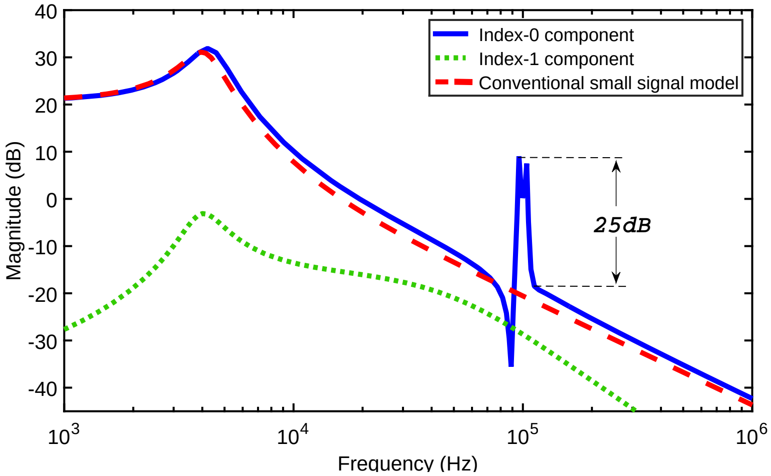


Figure 4.4: The Buck-Boost Converter

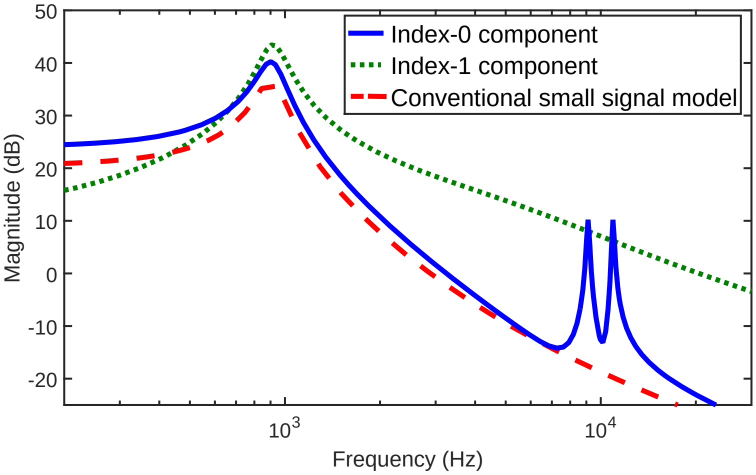
Table 4.1: Circuit parameters

	$R[\Omega]$	$C[F)]$	$L[H]$	$f_s[Hz]$	$V_{in}[V]$	d [ratio]
Boost converter	20	50μ	75μ	100k	2	0.4
Buckboost converter	4	220μ	50μ	10k	4	0.4

used for performing the transient analysis is Cadence Virtuoso. Finally, we demonstrate how the conventional small-signal model can lead to a false understanding of the closed-loop stability and how our proposed model can aid in the compensator design to regain stability of the two converters. Note that the method for analyzing stability using our proposed model is only empirical since the proposed model forms a linear time-variant (LTV) system.

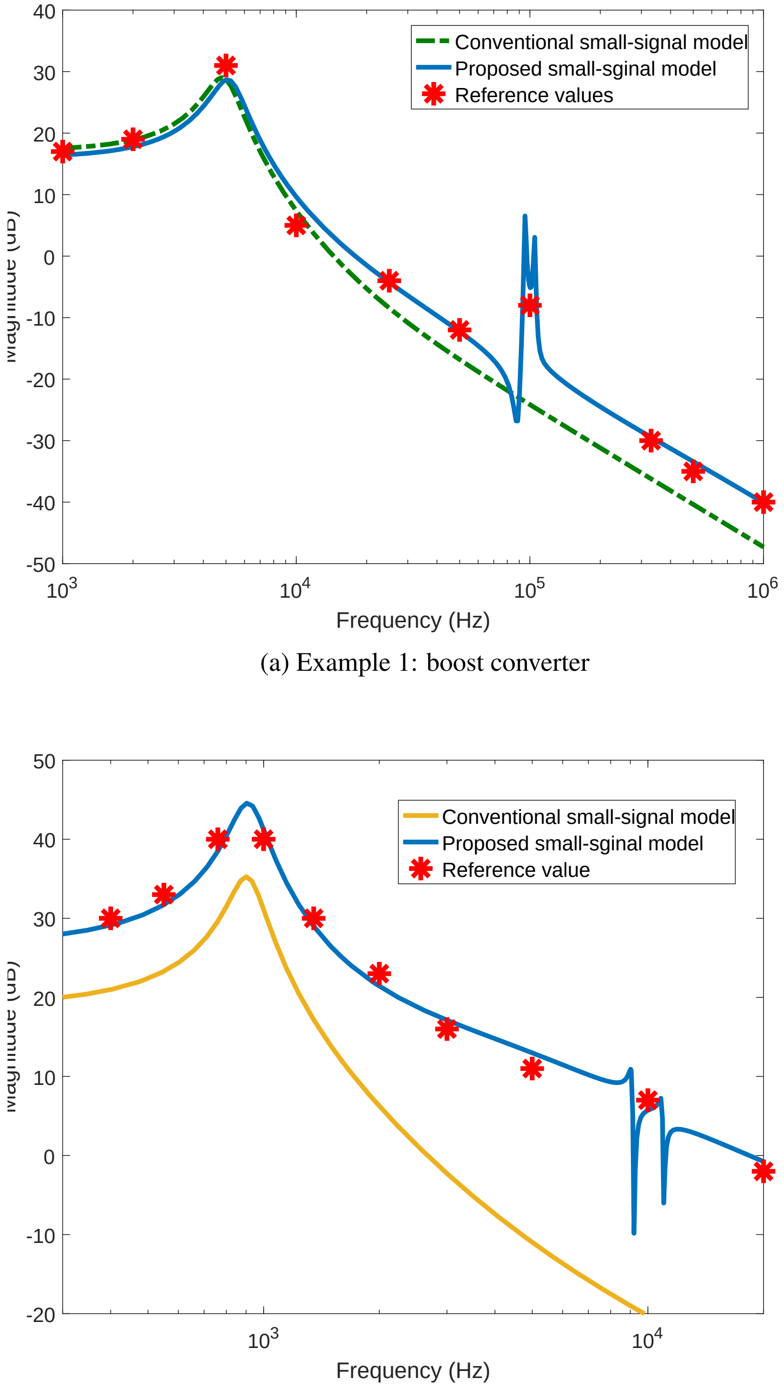


(a) Example 1: boost converter



(b) Example 2: buck-boost converter

Figure 4.5: The control-to-output transfer functions of the DC-DC converters modeled by the proposed small-signal model(index-0 and index-1) and the conventional small-signal model.



(a) Example 1: boost converter

(b) Example 2: buck-boost converter

Figure 4.6: Comparison of our proposed model, traditional small signal model and reference values.

4.4.1 Analysis of the Frequency-Domain Responses

Fig. 4.5 shows the index-0 and index-1 harmonic components of each converter obtained by the proposed small-signal model. Compared with the transfer function obtained by the conventional small-signal model, our proposed model demonstrates enhanced accuracy and reveals important response characteristics and design insights. Fig. 4.6 also shows a direct comparison with the ground truth reference values. As can be seen, the proposed method again demonstrates significantly improved accuracy relative to the reference values when compared with the traditional model.

The first converter we analyzed is the boost converter with the resulting plots as shown in Fig. 4.5(a) and in Fig. 4.6(a). The index-1 component is negligible in this example. Because our proposed model is a six-state model that also captures the index-0 and index-1 interactions, the computed index-0 component captures several spikes around the switching frequency. The peak magnitude of the spikes is $25dB$, which means that the switching noise is essentially amplified by the boost converter. As we will show later in the closed-loop simulation, the amplified switching noise affects the stability of the system by disrupting the behavior of the pulse-width modulator [39]. On the other hand, the conventional small-signal model only demonstrates a typical two-pole low-pass filter characteristic, without showing any sign of high-frequency spikes or instability.

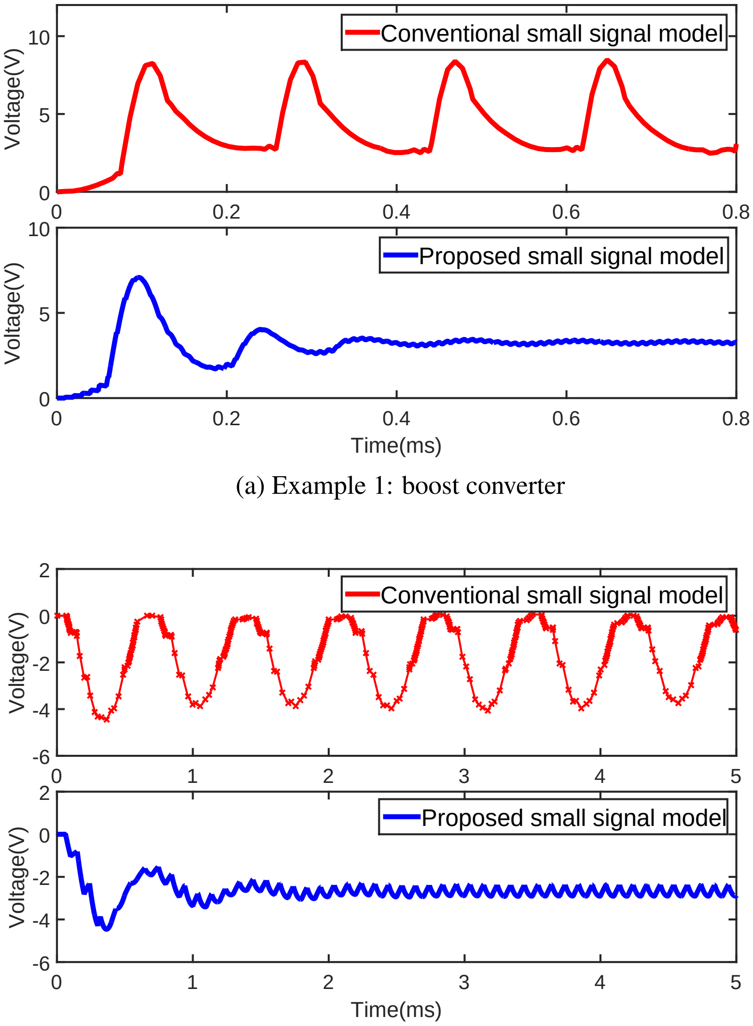
The second example converter we analyzed is the buck-boost converter with the resulting plots as shown in Fig. 4.5(b) and in Fig. 4.6(b). The frequency responses obtained by the proposed small-signal model shows that the index-0 component has a large low-frequency gain with two spikes around the switching frequency. The index-1 harmonic component has a significant magnitude at high frequency, which indicates that the harmonic component of the converter response is sensitive to the perturbation of the duty ratio. On the other hand, the conventional small-signal model only captures behavior of the index-0 component, which provides no information on the harmonics of the response. As will be shown next in the compensator design example, large harmonic components can possibly lead to an unstable closed-loop design.

4.4.2 Stability analysis and closed-loop design

For each converter circuit, we design a lead compensator based on the gain crossover frequency and the open-loop phase margin obtained from the frequency response of the conventional small-signal model which is based on the average model that only considers the DC component of the Fourier series of the signals [39]. Table II shows the lead compensator transfer functions for both converters. Fig. 4.7 shows the transient simulation of the closed-loop systems using the converters and the designed compensators. It is clear that the compensators based on the conventional small-signal model fail to stabilize either converter circuit in the start-up transient simulation.

The failure of the compensator design demonstrates the limitation of the conventional small-signal model in modeling the high-frequency behavior of DC-DC converters. On the other hand, our proposed model can accurately capture high-frequency circuit behaviors and provide truthful responses since it is based on a multi-harmonic averaged model, which considers the DC component, the 1st order component and the interactions between them. In fact, the index-0 component of the proposed model predicts the instability of the previously designed closed-loop systems. According to the revised bode stability criterion [43], both compensated systems are unstable with negative gain margins of $-15dB$ and $-11dB$.

Taking things a step further, we will use the proposed model to improve the closed-loop design and correct the compensator designs, which eventually regains the stability of the the closed-loop systems. In the first example, the causes of the instability of the boost converter are the spikes around the switching frequency. As discussed in Section



(a) Example 1: boost converter

(b) Example 2: buck-boost converter

Figure 4.7: Comparison of the transient responses of the closed-loop systems designed by the conventional small-signal model and the proposed small-signal model.

Table 4.2: Lead compensator designs based on the conventional small-signal model and the proposed small-signal model

	Conventional small-signal model	Proposed small-signal model
Boost converter	$\frac{2.61s+1.47\times10^4}{s+3.85\times10^4}$	$\frac{4.22s+1.22\times10^5}{s+5.17\times10^5}$
Buckboost converter	$\frac{4.41s+1.02\times10^5}{s+4.53\times10^5}$	$\frac{42.3s+2.73\times10^4}{s+1.16\times10^4}$

4.3.2, these spikes are essentially high-frequency overshoots of the index-0 component. To reduce the magnitude of the spikes, we reduce the quality factors. According to (4.36), the maximum quality factor of the boost converter is $Q = 26.3 = 28dB$, which is a good estimation of the amplitude of the spike shown in Fig. 4.5(a). By changing the value of the output capacitance C from $50\mu F$ to $10\mu F$, the value of the quality factor is reduced to 1.5 and the magnitude of the spike is reduced to $2.3dB$, which is less than 8% of the original spike magnitude. Then, we design a new lead compensator using the frequency response of the improved converter design. As shown by the transient simulation in 4.7(a), the closed-loop system becomes stable.

In the second example, the conventional small-signal model, which is a simple second-order model, fails to reveal the potential design problems due to the large magnitude of the index-1 component. To properly design a stable closed-loop system, we suppress the amplitude of the index-1 component. By adjusting the converter parameters to $L = 800\mu H$ and $C = 10\mu F$, we are able to limit the maximum magnitude of the index-1 component in all frequencies under $2dB$. With the new converter design, we re-design the lead compensator based on the gain crossover frequency obtained from our index-0 model. The transfer function of the lead compensator is shown in Table II. With the new lead compensator, the output voltage of the buck-boost converter settles down as shown in Fig. 4.7(b), which validates the stability of the closed-loop system.

4.5 Summary

In this chapter, a novel multi-harmonic small-signal model that accurately accounts for the high-frequency responses of the DC-DC converters is presented. Compared with existing small-signal models, the proposed model considers both the DC averages as well as the higher-order harmonic components in addition to the interactions between them, thereby providing a complete small-signal characterization of the converter circuits. Two converter design examples are presented which demonstrate the significant improvements of the proposed model on frequency-domain response analysis, stability analysis and closed-loop design.

5. CONVERGENCE-BOOSTED GRAPH PARTITIONING USING MAXIMUM SPANNING TREES FOR POWER DISTRIBUTION NETWORKS *

Large IC designs with multi-million or even multi-billion devices make the power grid analysis very challenging due to time and memory constraints. In this chapter, we present a novel preconditioning technique by integrating two families of methods, the support graph preconditioners and the partition-based preconditioners. The result is an elegant yet powerful partition-based preconditioner that targets optimizing the numerical convergence property of the partitioned system. To do this, we extract the maximum spanning tree, which is an optimized preconditioner for fast convergence, from the underlying circuit graph. Then we use the maximum spanning tree to guide the partitioning algorithm to create a partition of the system matrix such that degradation of the numerical properties brought by the support graph is minimized. Within each partitioned block, we add all edges from the original graph back to the blocks to build a block-Jacobi-like preconditioner. This MST-guided preconditioning process inherits the numerical convergence property from support graph and the divide-and-conquer nature from a partition-based preconditioner. We verify the efficiency and accuracy of the MST-guided preconditioner using 6 industrial and 3 synthetic power grid benchmark circuits. Our proposed method demonstrates distinctive advantages over existing preconditioners as well as direct solvers in factorization time and convergence speed. Compared to the state-of-the-art direct solver CHOLMOD, our proposed method achieves a runtime speedup up to 11.5X for DC analysis.

*Wang, Ya, et al. "Convergence-Boosted Graph Partitioning using Maximum Spanning Trees for Iterative Solution of Large Linear Circuits." *Proceedings of the 54th Annual Design Automation Conference 2017.* ©2017 ACM, Inc. Reprinted by permission. http://doi.acm.org/10.1145/3061639.3062215

5.1 Background

5.1.1 Power gird analysis

Power grid analysis targets the problem of analyzing the supply noise of power distribution networks (PDNs) in integrated circuits. A power grid can be modeled as a large linear network with resistors, energy storage elements including capacitors and inductors, and voltage/current sources. Typical power grid analysis include DC and transient analysis. In DC analysis the power grid is modeled as a resistive network and simulation captures the steady state behavior of the network. In transient analysis the energy storage devices are also considered when evaluating power supply noise. It can be solved by a series of equivalent DC analysis at different time points, where the capacitors and inductors are replaced with companion models that only include resistors and voltage/current sources. As such, we focus on DC analysis, which can be formulated using Nodal Analysis [44] as the following linear system problem

$$Ax = b, \tag{5.1}$$

where A is a $n \times n$ symmetric positive definite matrix (SPD) representing the resistance network, b is a $n \times 1$ vector representing the external input current sources, and x is the $n \times 1$ vector of unknown node voltages. A voltage source can be converted to an equivalent current source using Norton's therorm [45], thus keeping the positive-definite property of matrix A.

There are mainly two types of methods used to solve the linear SPD system described in (5.1). Direct methods like Cholesky factorization [23] decompose the matrix A into lower and upper triangular matrices L and L^T. Then the solution is found using forward/backwards substitution. However, the superlinear cost of matrix factorization makes direct methods unscalable for large power grid analysis. On the other hand, iterative solvers avoid direct matrix factorization by iteratively reducing the error of the solution step by step. With a good preconditioner, an iterative solver may achieve better time and memory efficiency than direct solvers.

5.1.2 Preconditioned Iterative Solvers

The convergence of iterative solvers such as conjugate gradient depends on the condition number of the matrix A, which is defined by

$$\kappa(A) = \frac{\lambda_{max}(A)}{\lambda_{min}(A)} \tag{5.2}$$

where $\lambda_{max}(A)$ and $\lambda_{min}(A)$ are the largest and smallest eigenvalue of the matrix A. When the condition number of a matrix is large, the matrix is considered ill-conditioned and the convergence for such matrices is slow. In such cases, preconditioners are applied to reduce the number of iterations. In preconditioned iterative solvers, instead of solving the original system (5.1), we find a preconditioner matrix P such that we can efficiently solve the alternative problem

$$P^{-1}Ax = P^{-1}b. \tag{5.3}$$

The efficiency of solving the preconditioned system depends on the condition number, which is defined by

$$\kappa(A, P) = \frac{\lambda_{max}(A, P)}{\lambda_{min}(A, P)}, \tag{5.4}$$

where $\lambda_{max}(A, P)$ and $\lambda_{min}(A, P)$ are the extreme generalized eigenvalues of the matrix pencil (A, P) [26]. In practice, we never explicitly formulate the inverse of the preconditioner P or try to compute $P^{-1}A$. Instead, we use the preconditioner to solve another problem that can help reduce the errors in our current solution. For example, in conjugate

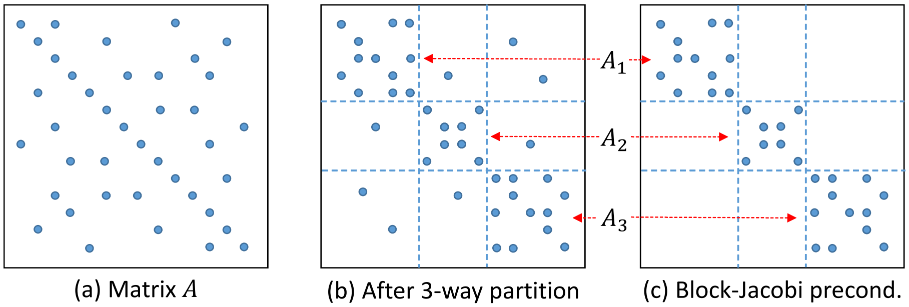


Figure 5.1: An example of building a block-Jacobi preconditioner from a matrix A based on 3-way partitioning.

gradient, we need to solve the linear system

$$Py = r, \tag{5.5}$$

where $r = b - Ax$ is the residue of the original system and y is used to update the current solution x. To summarize, a good preconditioner should be both effective in significantly reducing the condition numbers and efficient to compute. In the rest of this section, two types of preconditioners that our proposed method builds upon are briefly introduced.

5.1.2.1 Block-Jacobi preconditioner

Many power grid designs exhibit a hierarchical structure where several local power grids are connected to a global power grid. Each of the local power grids are either loosely connected or disjoint from each other. The block-Jacobi preconditioner exploits this structure by finding a partitioning of the matrix A into several loosely coupled sub-matrices. Fig. 5.1 shows an example of applying a 3-way partition on a matrix A. With a good permutation, most of the nonzero entries in A are placed into the block matrices A_i on the diagonal, leaving the rest of the matrix very sparse. To construct the block-Jacobi preconditioner, only the entries in the diagonal block are kept and the other nonzero en-

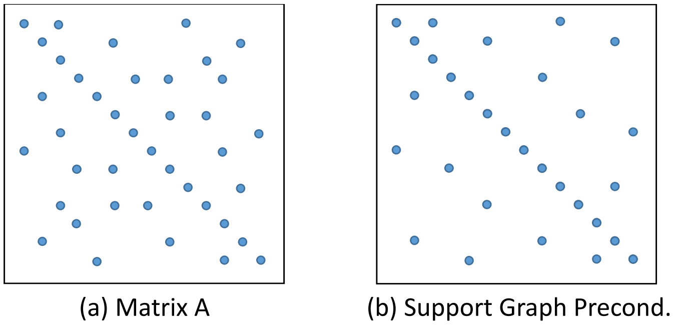


Figure 5.2: An example of building a support graph preconditioner from a matrix A.

tries outside each diagonal block matrix A_i are removed. The result is the block-Jacobi preconditioner as shown in Fig. 5.1(c). Since each block matrices A_i are disjoint from the rest of the matrix P, we can solve (5.5) by solving each block matrix separately, and combine the solutions to form the complete solution of (5.5). Due to the super-linear time complexity of factorization with respect to the matrix size [23], the sum of the cost of factorizing each block matrix Ai is less than the cost of factorizing the unpartitioned matrix A. Furthermore, parallelism can be easily exploited to further improve the efficiency.

5.1.2.2 Support graph preconditioner

While block-Jacobi preconditioning sparsifies the matrix A by ignoring the nonzero entries outside the diagonal blocks, the support graph preconditioning reduces the matrix in a more systemic way. It first constructs an undirected weighted graph G from the conductance matrix A, where each node corresponds to a row in the matrix and the weight of each edge between two nodes w_{ij} stands for the conductance value on the corresponding entry a_{ij} in the matrix. Then a subgraph called the support graph is extracted from G and the preconditioner P is constructed according to the subgraph. An example of constructing the support graph preconditioner is shown in Fig.5.2.

Recall (5.4) shows that for preconditioned system (5.3) the number of iterations is proportional to $\kappa(A, P)$, which is the ratio of the largest and smallest generalized eigenvalue of matrix pencil (A, P). We define the support of pencil (A, P), denoted by $\sigma(A, P)$, by

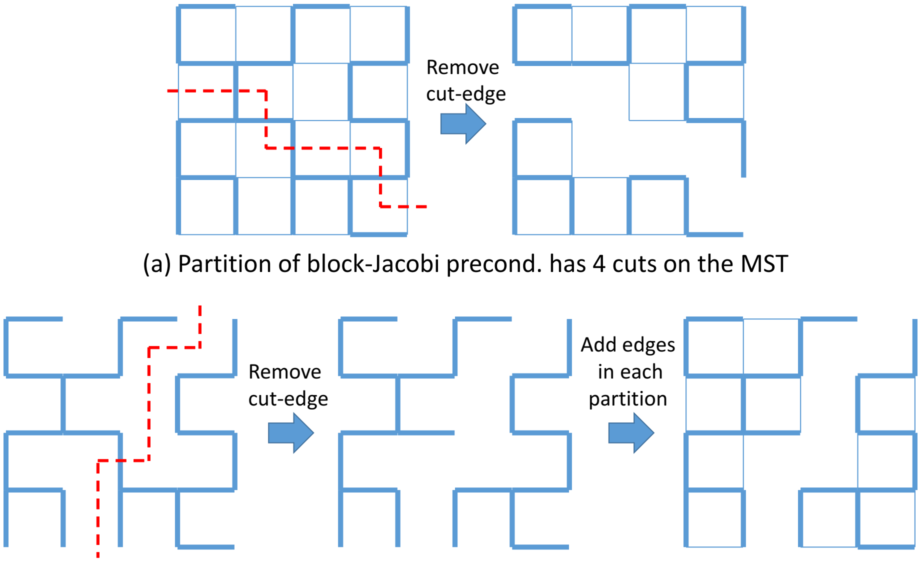
$$\sigma(A, P) = min\{\tau : \tau P - A \text{ is positive semidefinite}\}. \tag{5.6}$$

According to the support lemma [25], the support $\sigma(A, P)$ enforces an upper bound on the condition number $\kappa(A, P)$. Consequently, the support $\sigma(A, P)$ also bounds the number of iterations for preconditioned system to reach convergence. Thus using the support graph as the preconditioner can significantly reduce the condition number of the system and accelerate convergence [27, 46].

5.2 MST-guided Preconditioner

Both block-Jacobi and support graph preconditioners are successful preconditioning techniques. The strength of the block-Jacobi method is that it exploits the divide-and-conquer approach to solve each block matrix individually. With the help of parallelism, the cost of factorization and solve can be significantly reduced. However, the method depends on a good partition strategy to limit the number of iterations to reach convergence. On the other hand, The strength of the support graph method lies in its theoretical upper bound on the condition number and fast convergence. The bottle-neck of this method is its inability to incorporate parallel processing. For power grids of size larger than a million, direct factorization of the preconditioner is computationally expensive.

In this section, we explore the possibility of bridging these two different methods, with the goal to build a preconditioner that has both divide-and-conquer nature and fast-convergence property.



(a) Partition of block-Jacobi precond. has 4 cuts on the MST

(b) Partition of MST-guided precond. has 1 cut on the MST

Figure 5.3: Comparison between partition strategies of (a) the block-Jacobi preconditioner and (b) the MST-guided preconditioner. The thick lines represent the maximum spanning tree of the graph.

5.2.1 Minimum-cut partition of maximum spanning tree

In the traditional block-Jacobi method, the goal of partition is to minimize the total value of nonzero entries outside the diagonal blocks. This can be done by mapping the matrix to an undirected weighted graph using the same method described in Section 5.1.2.2 and then applying graph partitioning to find a minimum-cut partition of the graph. The sum of the weights of the edges which are cut across is denoted as the cut cost. It reflects the quality of the partition. In the ideal case where the graph exhibits a natural separation, a partition of zero cut cost can be found and the block-Jacobi preconditioner converges in a single step. However, when the cut cost is not zero, there is no guarantee that minimizing the cut cost leads to the best convergence rate. Fig. 5.3(a) shows a example of such partition of which the cut cost is minimized.

Now we reexamine the partitioning strategy of the block-Jacobi preconditioner from the support graph point of view. A common choice of support graph is the maximum spanning tree. By the discussion of Section 5.1.2.2, the maximum spanning tree preconditioner guarantees an upper bound on the number of iterations. Since the maximum spanning tree connects every node of the graph, any partition of the graph will also remove edges from the maximum spanning tree and thus destroy the bound of the condition number. In that sense, in order to achieve the best convergence rate in a partition-based preconditioner, we shall keep the edges of the maximum spanning tree as much as possible. This suggests that the goal of partition in a block-Jacobi preconditioner should be minimizing the cut-cost of the maximum spanning tree instead of the full circuit graph. This also suggests that the actual weights of the edges in maximum spanning tree does not matter and the cut-cost should be calculated by the number of edges removed from maximum spanning tree. The maximum spanning tree can be regarded as an underlying structure of the graph that is crucial for convergence and every edge in the maximum spanning tree is of equal importance. In some cases, the less damage caused to it during partition, the faster the convergence would be.

Partitioning the maximum spanning tree produces multipliable disjoint sub-trees which corresponds to a set of partitioned blocks. To facilitate a faster convergence, we add the edges of the original graph that fall into each partitioned block back to that block. The result is in accordance with the block-Jacobi method, where only the edges on the cut are removed. This process of building the MST-guided preconditioner is illustrated in Fig.5.3(b), where the maximum spanning tree is highlighted in the graph. In this example, both methods remove 7 edges from the graph to form a 2-way partition. The MST-guided partition only cuts one edge on the maximum spanning tree while the block-Jacobi preconditioner cuts four edges on the maximum spanning tree. As a result, the convergence of the block-Jacobi preconditioner is expected to be much slower than the MST-guided preconditioner. This will be demonstrated later based on real industrial benchmarks by comparing the actual condition numbers of each method.

5.2.2 Algorithm flow

The flow of constructing an MST-guided preconditioner from an input SPD matrix A is stated as follows.

1. Map matrix A to an undirected weighted graph $G = (V, E)$. The size of V is equal to the dimension of A. For each off-diagonal entry in A, a corresponding edge is connected between node i and j with weight $w_{ij} = a_{ij}$.

2. Find the maximum spanning tree from the graph G.

3. Find the minimum-cut partition of the maximum spanning tree.

4. Refill all non-MST edges from G within each partition to form the preconditioner P.

5.3 Experimental Results

We verify the efficiency and accuracy of the proposed MST-guided preconditioner by comparing it to several existing preconditioners [3, 46] and an state-of-the-art direct solver. All algorithms have been implemented and integrated into an in-house circuit simulator written in C++. For the direct solver we use a public-domain solver CHOLMOD [23]. For graph partitioning, we used the widely adopted METIS package[47]. Circuit simulations are performed on a single 64-bit 16-Core 2.20GHz AMD Opteron Processor with 32GB RAM running the Red Hat Linux. We measure the performance of our method using six industrial benchmarks from the IBM Austin Research Lab[48] and three syntectic power grid circuits generated from industrial power grid designs. The node count and device count of each circuit are shown in Table. 5.2. Conversion from voltage sources to the Norton equivalent current sources is applied during circuit parsing, ensuring the conductance matrix to be SPD for each benchmark. The parallelization is implemented using OpenMP.

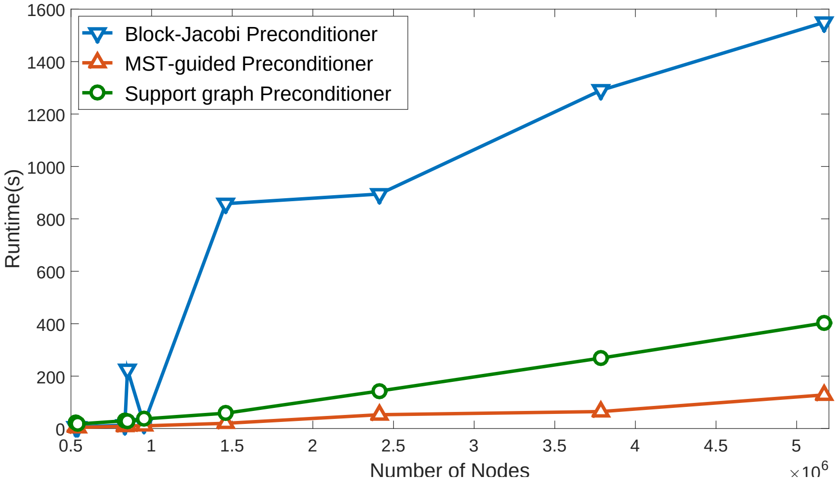


Figure 5.4: DC simulation runtime comparison between block-Jacobi preconditioner, support graph preconditioner and the proposed MST-guided preconditioner.

5.3.1 Analysis of results

Fig. 5.4 shows the DC simulation runtime comparison of a conjugate gradient iterative solver using the block-Jacobi preconditioner, support graph preconditioner and the proposed MST-guided preconditioner. The block-Jacobi preconditioner is constructed using the method mentioned in Section 5.1.2.1. The support graph preconditioner is constructed by first extracting the maximum spanning tree from the circuit graph, then gradually adding edges back to the maximum spanning tree until the best performance is reached. The details of building such a preconditioner can be found in [46]. For the block-Jacobi and MST-guided methods, the results are based on 8-way partitioning. For the largest benchmark, the proposed method achieves a speedup of 12.1X compared to the block-Jacobi preconditioner and a speedup of 2.6X compared to the support graph preconditioner. The runtime speedups can be explained by two reasons which are elaborated as follows.

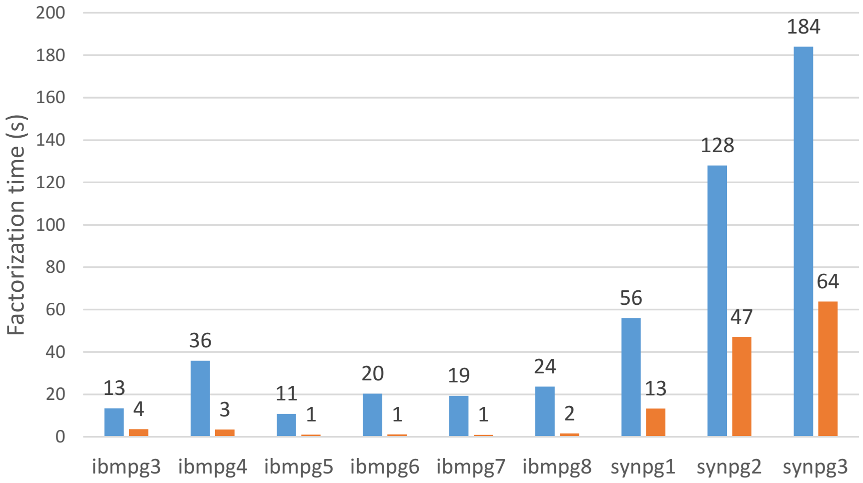


Figure 5.5: Factorization time of support-graph preconditioner (left column) and parallel MST-guided preconditioner (right column).

5.3.1.1 *Efficient factorization*

One of the advantages of the MST-guided preconditioner against support graph preconditioner is the efficient factorization. In the MST-guided method, the matrix is partitioned into equal-sized sub-matrices, which takes full advantage of parallelism in factorization process. However the support-graph preconditioner can only be factorized in a serial fashion. Fig. 5.5 shows the matrix factorization time of the support-graph preconditioner and the parallel MST-guided preconditioner, which performs 8-way matrix partition and factorize each block matrix in a single thread. The results demonstrate the advantage of parallel implementation of MST-guided preconditioner, which achieves speedups up to 21X with respect to the support-graph preconditioner in factorization time.

5.3.1.2 *Accelerated convergence*

The fundamental strength of the MST-guided method lies in its exceptional partition quality and fast convergence. Again, we compare it with the block-Jacobi preconditioner and the support graph preconditioner.

Table 5.1: Comparison of iteration numbers between the support graph preconditioner (SG), block-Jacobi preconditioner (Block) and MST-guided preconditioner (MST). Rd_{Block} and Rd_{SG} are the reduction rate of iteration numbers of MST-guided method compared to the block-Jacobi and support graph method, respectively. '-' indicates converge failure within 5000 iterations.

Circuit	SG	Block	MST	Rd_{SG}	Rd_{Block}
ibmpg3	68	1080	38	1.8X	26.3X
ibmpg4	2	40	30	0.1X	1.3X
ibmpg5	46	47	26	1.8X	1.8X
ibmpg6	34	61	42	0.8X	1.4X
ibmpg7	13	30	39	0.3X	0.8X
ibmpg8	93	2,283	40	2.3X	57.1X
synpg1	86	4,131	34	2.5X	121.5X
synpg2	94	-	34	2.8X	-
synpg3	94	-	38	2.5X	-

Table 5.1 reports the number of iterations required to reach convergence for each power grid benchmark. In 6 out of all 9 cases, the MST-guided preconditioner out performs both the block-Jacobi and support-graph perconditioner. For benchmarks of size larger than 1 million, the MST-guided preconditioner shows distinctive advantages by achieving as much as 121.5X reduction in the number of iterations compared with the block-Jacobi method. The block-Jacobi preconditioner fails to reach convergence within 5000 iterations for the last two benchmarks. This shows that the maximum spanning tree of the circuit graph indeed provides better partitioning insights than the circuit graph itself.

It is also worth noticing that the MST-guided method shows 2X to 3X improvement in iteration numbers even compared with the support graph method. This results may seem counter-intuitive at first glance. The following analysis well justifies this improvement. It is true that the MST-guided preconditioner partitions the graph and removes edges from the maximum spanning tree, which might destroy the theoretical bound on the iteration numbers. However, two actions are taken to compensate for the loss caused by the parti-

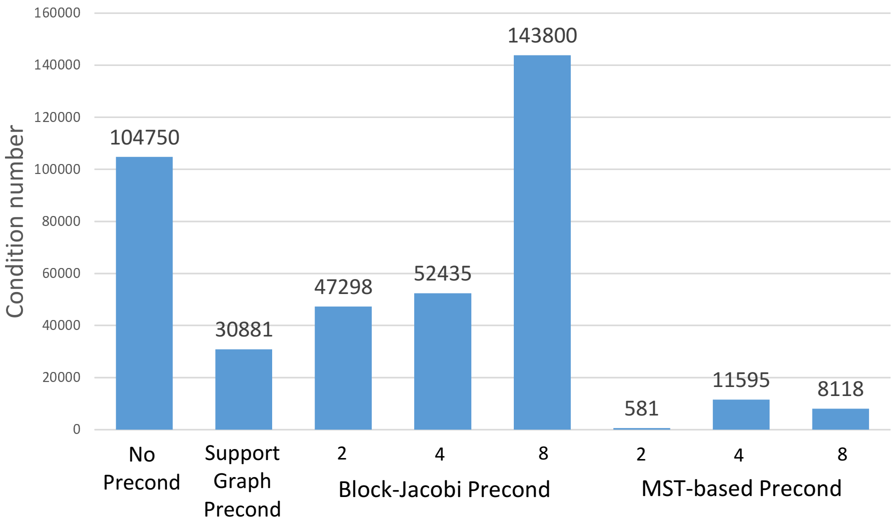


Figure 5.6: Condition numbers of a benchmark using different preconditioners. Numbers above block-Jacobi and MST-guided preconditioner indicate the number of partitions.

tion. First of all, the potential negative impact on the convergence is minimized by finding the minimum-cut partition of the maximum spanning tree. Secondly, all non-MST edges from the full circuit graph are refilled to the preconditioner within each partitioned block, which improves the convergence speed.

To better illustrate this, we compute the actual condition number of each preconditioned system of an industrial power grid benchmark [48]. Due to the high computational cost of calculating the condition number, we consider a small benchmark with $16,328$ nodes and $40,901$ devices. The condition numbers corresponding to different methods with different partition size are shown in Fig. 5.6. The MST-guided preconditioner achieves the lowest condition number of 581, which is 53X smaller than that of the support graph method. This confirms our analysis that keeping all edges within each partition indeed improves the convergence. In the meantime, the block-Jacobi preconditioner, which also keeps all edges within each partition, fails to significantly reduce the condition numbers. This confirms the understanding that partitioning on the maximum spanning tree

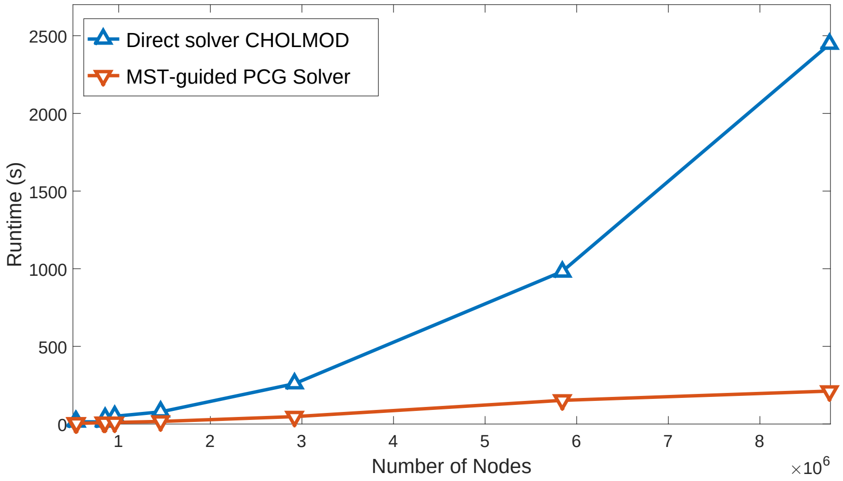


Figure 5.7: The runtime comparison of direct solver CHOLMOD and the MST-guided preconditioned conjugated gradient (PCG) Solver.

leads to improved circuit partitions which retain excellent numerical properties of support graph based methods.

5.3.2 Comparison with direct solver

Table 5.2 demonstrates the DC simulation runtime comparison between the preconditioned conjugated gradient (PCG) solver using our proposed preconditioner and the direct solver CHOLMOD. For the best performance, the last four benchmark are partitioned into 16 blocks. The total runtime of the direct solver is dominated by the Cholesky factorization, which can be extremely slow for large conductance matrices with lots of fill-ins. One of the significant advantages of our proposed MST-guided preconditioner is its divide-and-conquer nature and low-cost factorization. For example, factorizing benchmark $ibmpg8$ using our proposed method only takes $1.5s$, which is 52X faster than the factorization cost of the direct solver. In the meanwhile, partitioning the MST instead of the full circuit graph makes the partitioner more sensitive about cutting the edges of the MST and guides the partitioning process towards including the most critical MST edges in the resulting circuit blocks, leading to improved convergence. For all benchmarks, with the chosen partition size shown in the table, our MST-guided PCG solver is able to converge within 54 iterations. Figure 5.7 compares the runtime trends of the direct solver and our proposed MST-guided PCG solver. While the runtime of the direct solver increases drastically in circuit size, the runtime of the proposed method grows almost only linearly with a small slope. For the largest benchmark, our proposed method achieves a 11.5X runtime speedup with respect to the direct solver, while keeping the average error on nodal voltages below 0.33mV.

5.4 Summary

In the chapter, we present a novel preconditioner based on MST-guided partition. The preconditioner is constructed by first extracting the maximum spanning tree (MST) from the circuit graph, followed by the minimum-cut partition of the maximum spanning tree, which largely retains the good numerical properties of the support graph based methods. Within each partition, the edges from the original circuit graph are added back to the maximum spanning tree to build the final block-Jacobi-like preconditioner. By doing so, the proposed method maximizes the benefit inherited from the MST-guided preconditioner while producing a disjoint block structure which can be easily leveraged for parallel preconditioning. The DC simulation results on IBM and synthetic power grid benchmarks show that our proposed method achieves up to 11.5X runtime speedup compared with the state-of-the-art direct solution methods.

Table 5.2: Summary of DC simulation runtimes of direct solver and our proposed solver. #Node: the number of nodes. #Dev: the number of devices. Fact: Cholesky decomposition runtime. Solve: solving runtime. Total: total runtime. Iter: number of iterations to convergence. Part: number of partitions. Error: the average error of node voltages in V. All the runtimes are in seconds.

Circuit Info			CHOLMOD			MST-guided method						
Name	#Node	#Dev	Fact	Solve	Total	Part	Fact	Solve	Total	Iter	Error	Speedup
ibmpg3	850,627	1,603,120	38.82	0.25	39.07	8	3.64	7.67	11.31	38	2.89e-4	3.5X
ibmpg4	952,619	1,837,933	49.44	0.29	49.73	8	3.43	6.84	10.27	30	1.52e-4	4.8X
ibmpg5	540,398	1,617,925	14.28	0.15	14.43	8	1.02	3.27	4.29	26	1.00e-4	3.3X
ibmpg6	834,253	2,410,618	13.42	0.21	13.63	8	1.06	7.63	8.69	42	3.30e-4	1.5X
ibmpg7	530,823	1,781,254	18.84	0.17	19.01	8	0.92	5.08	6.00	39	2.12e-4	3.1X
ibmpg8	1,460,084	2,710,779	78.10	0.45	78.55	16	1.50	15.18	16.67	54	2.69e-4	4.7X
synpg1	2,920,167	4,548,712	259.01	0.75	259.76	16	13.34	34.46	47.80	40	2.97e-4	5.4X
synpg2	5,840,333	8,577,255	980.11	1.55	981.66	16	69.93	82.51	152.44	44	1.56e-4	6.4X
synpg3	8,760,499	11,961,629	2,443.4	2.30	2,445.7	16	101.89	110.36	212.25	48	2.27e-4	11.5X

6. CONCLUSION

The dissertation presents efficient and accurate analysis methods for two important component of power delivery circuitry in modern IC design, DC-DC converters, and power distribution networks.

The simulation and modeling of DC-DC converters are challenging due to the multirate characteristics. Chapter 2 presents a robust and efficient envelope following method for time-domain analysis of DC-DC converters based upon a numerically robust time-delayed phase condition to track the envelopes of circuit states under a varying switching frequency. At the core of the algorithm are a novel time-delayed equal-phase condition and a mechanism that smoothly tracks the transitions of the circuit state. The implementation of three fast simulation technique significantly improves the efficiency of the algorithm without degrading the accuracy level. We verify the robustness, generality, and efficiency of the proposed technique using several test circuits for which our technique offers excellent simulation speedups and robustness.

Using a different approach, in chapter 3 we present a multi-harmonic model, which captures the DC response as well as higher-order harmonics of PWM DC-DC converters. As a full order model, it retains the inductor current as a state variable and is accurate even when the converter is in the transient state. Our model seamlessly transitions between CCM and DCM during the simulation. Moreover, the efficiency of simulating the proposed model is boosted due to two system decoupling techniques with minimum impact on model accuracy. Our model was tested on two different DC-DC converters and speedups of one order of magnitude were achieved with respect to transistor-level simulations.

Small-signal models of DC-DC converters are widely used for analyzing stability and play an important role in converter design and control. In Chapter 4, we proposed a small-signal model based on the multi-harmonic large-signal model proposed in Chapter 3. As a result, our proposed small signal model accurately accounts for the high-frequency responses of the DC-DC converters. In two converter examples, the proposed model accurately captures important high-frequency overshoots and undershoots of the converter response, which are otherwise unaccounted for by the existing techniques.

In Chapter 5, we tackle the challenges of simulating power distribution networks by presenting a parallel partition-based iterative solver guided by support graph theory. The proposed method maximizes the benefit inherited from the support graph preconditioner while producing a disjoint block structure which can be easily leveraged for parallel preconditioning. The DC simulation results on IBM and synthetic power grid benchmarks show that our proposed method achieves up to 11.5X runtime speedup compared with the state-of-the-art direct solution methods.

REFERENCES

[1] “International technology roadmap for semiconductors. 2011 overall roadmap technology characteristics (ortc) tables.." http://www.itrs.net

[2] J. White and S. B. Leeb, “An envelope-following approach to switching power converter simulation,” *IEEE Transactions on Power Electronics*, vol. 6, no. 2, pp. 303–307, 1991.

[3] K. Sun, Q. Zhou, K. Mohanram, and D. C. Sorensen, “Parallel domain decomposition for simulation of large-scale power grids,” in *2007 IEEE/ACM International Conference on Computer-Aided Design*, pp. 54–59, IEEE, 2007.

[4] T. Mei and J. Roychowdhury, “An efficient and robust technique for tracking amplitude and frequency envelopes in oscillators,” in *Proceedings of the 2005 IEEE/ACM International conference on Computer-aided design*, pp. 599–603, IEEE Computer Society, 2005.

[5] L. M. Silveira, J. White, and S. Leeb, “A modified envelope-following approach to clocked analog circuit simulation,” in *Computer-Aided Design, 1991. ICCAD-91. Digest of Technical Papers., 1991 IEEE International Conference on*, pp. 20–23, IEEE, 1991.

[6] Z. Zeng, P. Li, and Z. Feng, “Parallel partitioning based on-chip power distribution network analysis using locality acceleration,” in *2009 10th International Symposium on Quality Electronic Design*, pp. 776–781, IEEE, 2009.

[7] X.-X. Liu, S. X.-D. Tan, H. Wang, and H. Yu, “A gpu-accelerated envelope-following method for switching power converter simulation,” in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2012*, pp. 1349–1354, IEEE, 2012.

[8] A. G. Gheorghe, F. Constantinescu, M. Marin, and M. Nitescu, “The envelope following analysis of a buck converter with closed loop control,” in *Electrical and Electronics Engineering (ISEEE), 2013 4th International Symposium on*, pp. 1–5, IEEE, 2013.

[9] T. Kato, K. Inoue, and Y. Kanda, “Envelope following analysis of an autonomous power electronic system,” in *Computers in Power Electronics, 2006. COMPEL’06. IEEE Workshops on*, pp. 29–33, IEEE, 2006.

[10] T. Kato, “Multi-rate transient analysis of power electronic circuits by the envelope-following method with sensitivities of switch timings,” in *Power Electronics Specialists Conference, PESC’94 Record., 25th Annual IEEE*, pp. 1277–1281, IEEE, 1994.

[11] R. D. Middlebrook and S. Ćuk, “A general unified approach to modelling switching-converter power stages,” *International Journal of Electronics Theoretical and Experimental*, vol. 42, no. 6, pp. 521–550, 1977.

[12] A. Davoudi, J. Jatskevich, and T. De Rybel, “Numerical state-space average-value modeling of pwm dc-dc converters operating in dcm and ccm,” *Power Electronics, IEEE Transactions on*, vol. 21, no. 4, pp. 1003–1012, 2006.

[13] J. Sun, D. M. Mitchell, M. F. Greuel, P. T. Krein, and R. M. Bass, “Averaged modeling of pwm converters operating in discontinuous conduction mode,” *Power Electronics, IEEE Transactions on*, vol. 16, no. 4, pp. 482–492, 2001.

[14] V. Vorperian, “Simplified analysis of PWM converters using model of PWM switch part I: Continuous conduction mode,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 26, pp. 490–496, May 1990.

[15] V. Vorpérian, “Simplified analysis of pwm converters using model of pwm switch. ii. discontinuous conduction mode,” *Aerospace and Electronic Systems, IEEE Transac-*

tions on, vol. 26, no. 3, pp. 497–505, 1990.

[16] Y. Amran, F. Huliehel, and S. Ben-Yaakov, “A unified SPICE compatible average model of PWM converters,” *IEEE Transactions Electronics*, pp. 585–694, 1991.

[17] S. R. Sanders, J. M. Noworolski, X. Z. Liu, and G. C. Verghese, “Generalized averaging method for power conversion circuits,” *Power Electronics, IEEE Transactions on*, vol. 6, no. 2, pp. 251–259, 1991.

[18] V. A. Caliskan, O. Verghese, and A. M. Stankovic, “Multifrequency averaging of dc/dc converters,” *IEEE Transactions on Power Electronics*, vol. 14, no. 1, pp. 124–133, 1999.

[19] D. Maksimović, A. M. Stanković, V. J. Thottuvelil, and G. C. Verghese, “Modeling and simulation of power electronic converters,” *Proceedings of the IEEE*, vol. 89, no. 6, pp. 898–912, 2001.

[20] J. M. Noworolski and S. R. Sanders, “Generalized in-plane circuit averaging,” in *Applied Power Electronics Conference and Exposition, 1991. APEC’91. Conference Proceedings, 1991., Sixth Annual*, pp. 445–451, IEEE, 1991.

[21] Z. Zeng, S. Lai, and P. Li, “Ic power delivery: Voltage regulation and conversion, system-level cooptimization and technology implications,” *ACM Transactions on Design Automation of Electronic Systems*, vol. 18, no. 2, pp. 1–15, 2013.

[22] L. Scandola, L. Corradini, and G. Spiazzi, “Multi-harmonic small-signal modeling of digitally controlled dc-dc series resonant converters,” in *IEEE Workshop on Control and Modeling for Power Electronics (COMPEL)*, pp. 1–8, IEEE, 2015.

[23] Y. Chen, T. A. Davis, W. W. Hager, and S. Rajamanickam, “Algorithm 887: Cholmod, supernodal sparse cholesky factorization and update/downdate,” *ACM Transactions on Mathematical Software (TOMS)*, vol. 35, no. 3, p. 22, 2008.

[24] T. A. Davis and E. Palamadai Natarajan, “Algorithm 907: Klu, a direct sparse solver for circuit simulation problems,” *ACM Transactions on Mathematical Software (TOMS)*, vol. 37, no. 3, p. 36, 2010.

[25] M. Bern, J. R. Gilbert, B. Hendrickson, N. Nguyen, and S. Toledo, “Support-graph preconditioners,” *SIAM Journal on Matrix Analysis and Applications*, vol. 27, no. 4, pp. 930–951, 2006.

[26] E. G. Boman and B. Hendrickson, “Support theory for preconditioning,” *SIAM Journal on Matrix Analysis and Applications*, vol. 25, no. 3, pp. 694–717, 2003.

[27] X. Zhao, J. Wang, Z. Feng, and S. Hu, “Power grid analysis with hierarchical support graphs,” in *Proceedings of the International Conference on Computer-Aided Design*, pp. 543–547, IEEE Press, 2011.

[28] Z. Feng and Z. Zeng, “Parallel multigrid preconditioning on graphics processing units (gpus) for robust power grid analysis,” in *Proceedings of the 47th Design Automation Conference*, pp. 661–666, ACM, 2010.

[29] H. K. Thornquist, E. R. Keiter, R. J. Hoekstra, D. M. Day, and E. G. Boman, “A parallel preconditioning strategy for efficient transistor-level circuit simulation,” in *Proceedings of the 2009 International Conference on Computer-Aided Design*, pp. 410–417, ACM, 2009.

[30] W.-R. Liou, M.-L. Yeh, and Y. L. Kuo, “A high efficiency dual-mode buck converter IC for portable applications,” *IEEE Transactions on Power Electronics*, vol. 23, no. 2, pp. 667–677, 2008.

[31] J. Roychowdhury, “Analyzing circuits with widely separated time scales using numerical pde methods,” *Circuits and Systems I: IEEE Transactions on Fundamental Theory and Applications*, vol. 48, no. 5, pp. 578–594, 2001.

[32] T. J. Aprille Jr and T. N. Trick, "Steady-state analysis of nonlinear circuits with periodic inputs," *Proceedings of the IEEE*, vol. 60, no. 1, pp. 108–114, 1972.

[33] P. Maffezzoni, "A versatile time-domain approach to simulate oscillators in rf circuits," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 56, no. 3, pp. 594–603, 2009.

[34] A. Brambilla and P. Maffezzoni, "Envelope-following method to compute steady-state solutions of electrical circuits," *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, vol. 50, no. 3, pp. 407–417, 2003.

[35] D. Levy, "Introduction to numerical analysis," 2010.

[36] T. Mei and J. Roychowdhury, "A robust envelope following method applicable to both non-autonomous and oscillatory circuits," in *Proceedings of the 43rd annual Design Automation Conference*, pp. 1029–1034, ACM, 2006.

[37] A. S. Limjoco, "Measuring output ripple and switching transients in switching regulators," `http://www.analog.com/media/en/technical-documentation/application-notes/AN-1144.pdf`.

[38] C. D. S. Inc, "Spectre circuit simulator," 2015. `http://www.cadence.com/products/cic/spectre_circuit/pages/default.aspx`.

[39] R. W. Erickson and D. Maksimovic, *Fundamentals of power electronics*. Springer Science & Business Media, 2007.

[40] Y. Wang, D. Gao, D. Tannir, and P. Li, "Multi-harmonic nonlinear modeling of low-power pwm dc-dc converters operating in ccm and dcm," in *Proceedings of the 2016 Design, Automation & Test in Europe Conference & Exhibition*, EDA Consortium, 2016.

[41] D. Tannir, Y. Wang, and P. Li, “Accurate modeling of nonideal low-power pwm dc-dc converters operating in ccm and dcm using enhanced circuit averaging techniques,” *ACM Transactions on Design Automation of Electronic Systems*, vol. 21, no. 4, pp. 1–15, 2016.

[42] I. J. Pérez-Arriaga, G. C. Verghese, F. L. Pagola, J. Sancha, and F. C. Schweppe, “Developments in selective modal analysis of small-signal stability in electric power systems,” *Automatica*, vol. 26, no. 2, pp. 215–231, 1990.

[43] J. Hahn, T. Edison, and T. F. Edgar, “A note on stability analysis using bode plots,” 2001.

[44] L. Pillage, *Electronic Circuit & System Simulation Methods (SRE)*. McGraw-Hill, Inc., 1998.

[45] J. Brittain, “Thevenin’s theorem,” *Ieee Spectrum*, vol. 27, no. 3, p. 42, 1990.

[46] X. Zhao and Z. Feng, “Gpscp: a general-purpose support-circuit preconditioning approach to large-scale spice-accurate nonlinear circuit simulations,” in *2012 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 429–435, IEEE, 2012.

[47] G. Karypis and V. Kumar, “Metis–unstructured graph partitioning and sparse matrix ordering system, version 2.0,” 1995.

[48] S. R. Nassif, “Power grid analysis benchmarks,” in *Proceedings of the 2008 Asia and South Pacific Design Automation Conference*, pp. 376–381, IEEE Computer Society Press, 2008.