Fast Vectorless Power Grid Verification Under an RLC Model

Nahi H. Abdul Ghani, *Student Member, IEEE,* and Farid N. Najm, *Fellow, IEEE*

***Abstract*—As part of early system design, one must verify that the power grid provides the underlying logic circuitry with voltage levels that are within specified ranges. In this paper, we describe a vectorless verification approach that can be applied early in the design process. We adopt an RLC model of the grid in the framework of current constraints that capture uncertainty about circuit details and activity. With just *a few* linear programs and one linear system solve, our proposed approach provides *tight conservative bounds* on the maximum and minimum worst-case voltage drops at every node of the grid. Results show the accuracy and speed of our technique thus making it practical and scalable.**

***Index Terms*—Integrated circuits, overshoot, power grid, verification, voltage drop.**

I. Introduction

The RISING demand for low voltage modern integrated circuits (ICs) has made efficient analysis of power grids a critical task. A key concern is the fact that a poorly-designed grid will result in excessive fluctuations in the voltage levels supplied to the underlying logic thus slowing it down and putting the overall circuit timing performance at risk. Therefore, in order to ensure correct logic functionality, there is a clear need for efficient grid verification.

There are two main causes of voltage violations; *IR* drop, which is the result of the resistivity of the metal rails, and *Ldi/dt* noise, which is due to the inductance of the rails and the grid-package interconnections. In today's designs, the increasing number of transistors and the high operating frequencies lead to large current demands and fast current transients both of which produce large *IR* drops and considerable *Ldi/dt* noise [1]–[3]. Note, in particular, that such inductive noise can result in either drops or overshoots where voltage levels at the grid nodes exceed that of the supply. Therefore, grid verification should account for both voltage behaviors and grid safety must be stated in terms of the *maximum* and *minimum* worst-case voltage levels at each node on the grid.

Today, grid verification is typically done by simulation. Such an approach requires full knowledge of the current waveforms drawn by every logic block attached to the grid. These waveforms would then be used to simulate the grid and get the voltage drop at every node. Verifying the grid in this manner proves to be problematic. For one thing, the number of current traces needed to cover the space of voltage drops exhibited on the grid is intractable for modern designs where grids can consist of several million nodes. Several attempts have been made to address the problem of searching the current space for the worst-case current waveforms, including the computation of bounds on possible currents [4] or of current statistics [5]. Another major drawback is that a simulation based flow does not allow for *early* grid verification, when grid modifications can be most easily incorporated. The need, then, is for a verification approach that does not depend on simulation, i.e., a *vectorless* approach. Therefore, we adopt the framework of partial current specifications, in the form of *current constraints* [6], which will be detailed in Section II-C. The constraints specify a feasible space in which currents can vary during circuit operation. Under such constraints, RLC grid verification becomes a problem of computing the maximum and minimum voltage drops. In [7], the authors computed these worst-case voltage drops by using a convergent iterative process that requires the solution of linear programs (LPs) while stepping repeatedly through time. This approach becomes prohibitive even for medium size grids.

In this paper, we propose a novel technique to compute *upper* bounds on the maximum worst-case voltage drops and *lower* bounds on the minimum worst-case voltage drops on the grid. These bounds require *a few* LPs and one linear system solve. Experimental results in Section V show that the improvement in runtime of our technique is dramatic while ensuring small error values.

II. Background

A. Power Grid Model

We consider an *RLC* model of the power grid where each branch is represented either by a resistor, referred to as an *r*-branch, or by a resistor in series with an inductor, referred to as an *rl*-branch. Two sets of nodes exist; one is internal to the *rl*-branches. We denote its members as *internal* nodes. The members of the other set are referred to as *external* nodes. Some external nodes may be connected to ideal voltage sources (assuming flip-chip technology, we will refer to an ideal supply voltage source as a C4, with the understanding that any parasitics that are part of a true C4 pad structure have

Manuscript received May 21, 2010; revised August 11, 2010 and September 27, 2010; accepted November 8, 2010. Date of current version April 20, 2011. This work has been supported in part by the Semiconductor Research Corporation. This paper was recommended by Associate Editor P. Li.

The authors are with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 2H8, Canada (e-mail: nahi@eecg.utoronto.ca; f.najm@utoronto.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCAD.2010.2096593

already been modeled and included in the grid description). As a result, the inductance in our grid model represents the inductive component of either the grid interconnect or the pad structure that connects the grid to the external voltage sources. There exists also a capacitor from every external node to ground and some of these nodes have ideal current sources (to ground) representing the currents drawn by the logic circuits tied to the grid. In the derivation of the system equations, we will use the state-variable approach to network analysis [8].

Let the power grid consist of $m + n_1 + p$ nodes, where nodes $1, 2, \ldots, m$ are internal nodes, nodes $(m + 1), (m + 2), \ldots, (m + n_1)$ are external nodes with no voltage sources attached, and the remaining nodes $(m+n_1+1), (m+n_1+2), \ldots, (m+n_1+p)$ are the external nodes where the p voltage sources are connected. Let c_k be the capacitance from every external node k to ground. Let $i_{s,k}(t)$ be the current source connected to external node k, where the direction of positive current is from the node to ground. We also assume that $i_{s,k}(t)$ is defined for every node $k = 1, \ldots, n$ where $n = m + n_1$ so that all external nodes with no current source attached have $i_{s,k}(t) = 0, \forall t$ and all internal nodes have $i_{s,k}(t) = 0, \forall t$. Let $i_s(t)$ be the vector of all $i_{s,k}(t)$ sources. Let $u_k(t)$ be the voltage at every internal or external node k, $k = 1, \ldots, n$ and let $u(t)$ be the vector of all $u_k(t)$ voltage signals. Moreover, let $i_l(t)$ represent the inductive branch currents where $l = 1, \ldots, m$, and let $i(t)$ be the vector of all inductive branch currents. Notice that m also represents the number of inductors in the grid. Fig. 1 is an example of an RLC grid model.

B. System Equations

The time-domain equations that describe the circuit can be derived by applying Kirchoff's current law (KCL) at every node k, where $k = 1, 2, \ldots, n$. In general, applying KCL at every non-C4 node of the grid, we get

$$Gu(t) + Cu'(t) + Mi(t) = -i_s(t) + G_0 v_{dd} \tag{1}$$

where G is an $n \times n$ conductance matrix as in the traditional modified nodal analysis formulation [9]. The G matrix is known to be diagonally-dominant symmetric positive definite $\mathcal{M}$-matrix (so that $G^{-1} \geq 0$). G_0 is an $n \times n$ matrix of the conductance elements connected to V_{dd} sources [6]. C is an $n \times n$ diagonal matrix of node capacitances with the understanding that entries corresponding to internal nodes are 0, and M is an $n \times m$ incidence matrix whose elements are either ± 1 or 0, as in [8]. The term ± 1 occurs in location m_{ij} of the matrix when node i is connected to the jth inductor, else a 0 occurs. The sign of the non-zero terms depends on the positive direction of current in the branch and on the node under consideration. If the current assignment is away from the node, then the sign is positive, else it is negative. Finally, v_{dd} is a constant vector whose entries are all equal to V_{dd} (the supply voltage).

If we set all $i_{s,k}(t) = 0, \forall t$, then no voltage drop should occur on the grid. So, all branch currents will be zero and all node voltages will have a value of V_{dd}. System (1) then becomes

$$Gv_{dd} = G_0 v_{dd}. \tag{2}$$

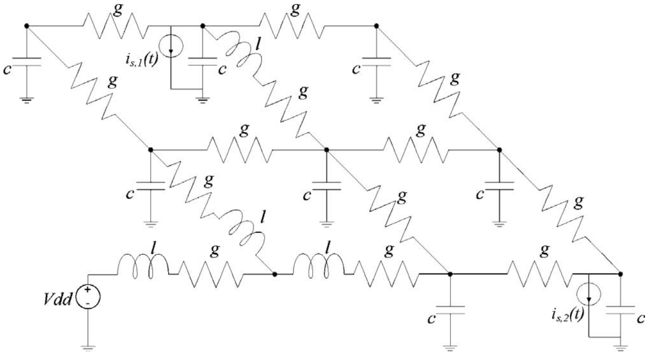


Fig. 1. *RLC* grid model.

Define $v_k(t) = V_{dd} - u_k(t)$ to be the voltage drop at node k and $v(t)$ as the vector of all voltage drops. Then, if we use (1) and (2), the first system equation becomes

$$Gv(t) + Cv'(t) - Mi(t) = i_s(t). \tag{3}$$

Notice that in (3) we do not take into account the relationship between the inductor branch current and the inductor voltage. It remains to model this relationship. The necessary equations were derived in [8] and are simply stated here. Notice that these equations will be applied to all inductances. So, the number of equations will be m. Relating all inductive branch currents to the voltage drop across the respective inductors we get the second system equation

$$M^T v(t) + Li'(t) = 0 \tag{4}$$

where L is an $m \times m$ diagonal matrix of inductance values. Notice that the matrix multiplying the voltage vector is the transpose of the incidence matrix [8]. The pair of equations (3) and (4) represent the complete behavior of the power grid, as a dynamical system. Using a backward finite difference approximation $\left(v'(t) \cong \frac{v(t)-v(t-\Delta t)}{\Delta t}\right)$, a discrete-time version of (3) and (4) can be written as

$$\left(G + \frac{C}{\Delta t}\right) v(t) - Mi(t) = i_s(t) + \frac{C}{\Delta t} v(t - \Delta t) \tag{5}$$

$$M^T v(t) + \frac{L}{\Delta t} i(t) = \frac{L}{\Delta t} i(t - \Delta t). \tag{6}$$

The equation variables are the node voltages and the inductive branch currents. Note that $G + \frac{C}{\Delta t}$ is also a symmetric positive definite $\mathcal{M}$-matrix. For properties of $\mathcal{M}$-matrices, the reader is referred to [10].

C. Current Constraints

In our approach, we perform a verification of the grid in the absence of complete information about currents drawn by the underlying circuit, what may be called a *vectorless* approach. We do this because the currents are typically hard to specify, for at least two reasons: 1) there is a large variety of possible current waveforms, so that the worst case is hard to determine up-front and simulation of a large set of waveforms is prohibitively expensive, and 2) grid design and verification cannot wait until the chip design is nearly-complete, and is typically done early in the design flow, so that the details of the underlying circuitry may not yet be available or complete.

Current constraints, originally introduced in [6], capture the uncertainty about the circuit currents arising from both unknown circuit behaviors and the fact that one is uncertain about circuit details early in the design flow. The aim is to verify that the grid is safe (i.e., its voltages remain within certain bounds), under all possible transient current waveforms which satisfy these constraints. Two types of constraints are defined: *local constraints* and *global constraints*.

Local constraints are upper bounds on individual current sources, where a current source can represent a single logic gate or cell, but more typically should represent a larger block. They can be expressed as

$$0 \leq i_s(t) \leq i_L \quad \forall t \geq 0 \tag{7}$$

where i_L is the vector of peak values that the current sources can draw—it is a "DC" upper bound on the transient waveform vector $i_s(t)$. It is important to remember, however, that it is only the constraints that are DC; the currents themselves are transient. To ensure that these constraints are well-defined for every node of the grid, we enforce the condition that any node with no current source connected would have a zero i_L component.

If *only* local constraints are provided, the problem is much simplified, but the results would be overly pessimistic, because it is never the case that all chip components simultaneously draw their maximum currents, hence the need for global constraints, which are upper bounds on the sums of groups of current sources. They represent the peak total power dissipation of a group of circuit blocks. Assuming we have a total of κ global constraints, they can be expressed in matrix form as

$$0 \leq U i_s(t) \leq i_G \quad \forall t \geq 0 \tag{8}$$

where U is a $\kappa \times n$ matrix that consists only of 0s and 1s which indicate which current sources are present in each global constraint. As for the case of local constraints, note that i_G is a fixed time-independent upper bound, a DC constraint, but the currents themselves are transient waveforms over time.

How would one obtain/specify these constraints in practice? If a logic block is available and small enough to simulate, then one can generate the constraints by an "offline" process of simulation, which can be viewed as a characterization of that block. If the block is not yet available or is too large to simulate, then one would need to rely on design expertise and engineering judgment (how big it is, what its power needs were in a previous technology and how scaling would affect those needs, and so on). If, early in the design flow, nothing is known about this block, not even its detailed functionality, one typically is able to come up with an area budget for it. From that, and from the projected power density (watts/μm^2) for the target process technology, one can generate a rough current constraint for it. The bottom line: *something* is typically known about that block, which with good engineering judgment can be formulated into constraints. After all, if truly nothing is known about the circuit currents, then the grid simply cannot be verified.

Another possibility is that the current constraints can be used to implement a "spec-based" design flow; a chip-level designer would simply specify the constraints based on design expertise, and the grid is verified under these constraints. The constraints now become *design guidelines* to be observed in subsequent design activity. If all design teams follow these guidelines and verify their blocks, the end result would be a grid that is *safe by construction*.

Together, the local and global constraints define a *feasible space* of currents, which we denote by $\mathcal{F}$, so that $i_s(t) \in \mathcal{F}$ if and only if it satisfies (7) and (8). Later in the paper, we will define algorithms that operate on a vector $i_s(t)$ and which are applicable at any value of time t. In that context, we will use the shorthand $i_s \in \mathcal{F}$ to denote the fact that $i_s(t)$ is feasible, for any given t.

III. Problem Definition

Before we proceed, we will show how the *exact* RLC grid verification boils down to solving linear programs that depend on a certain matrix inverse and on some matrix-matrix multiplications. The aim is to lay the ground-work for the rest of the paper, in which upper bounds on the maximum worst-case voltage drops and lower bounds on the minimum worst-case voltage drops are computed using a very efficient approach that minimizes the cost of computing the inverse and that limits the number of required matrix-matrix multiplications. The novelty of this paper is two-fold. First, we transform the RLC grid into a reduced circuit to alleviate the problems associated with its system matrices and to allow for fast matrix manipulations. Second, we derive efficient tight upper and lower bounds on the maximum and minimum worst-case voltage drops respectively. However, we do need to provide some detailed background on the RLC grid verification to motivate our contribution.

We are interested in the vectors of maximum and minimum worst-case voltage drops, over all possible currents in $\mathcal{F}$. To simplify the notation, let

$$A = \begin{bmatrix} G + \frac{C}{\Delta t} & -M \\ M^T & \frac{L}{\Delta t} \end{bmatrix} \qquad B = \begin{bmatrix} \frac{C}{\Delta t} & 0 \\ 0 & \frac{L}{\Delta t} \end{bmatrix}$$
$$x(t) = \begin{bmatrix} v(t) \\ i(t) \end{bmatrix} \qquad b(t) = \begin{bmatrix} i_s(t) \\ 0 \end{bmatrix} \tag{9}$$

so that, combining (5) and (6), we can write

$$x(t) = A^{-1}Bx(t-\Delta t) + A^{-1}b(t). \tag{10}$$

Consider the special case where the grid had no stimulus for all $t \leq 0$, so that $x(0) = 0$. At time $t = \Delta t$, we can write

$$\begin{aligned} x(\Delta t) &= A^{-1}Bx(0) + A^{-1}b(\Delta t) \\ &= A^{-1}b(\Delta t). \end{aligned} \tag{11}$$

At $2\Delta t$ and at $3\Delta t$, we have

$$\begin{aligned} x(2\Delta t) &= A^{-1}Bx(\Delta t) + A^{-1}b(2\Delta t) \\ &= A^{-1}BA^{-1}b(\Delta t) + A^{-1}b(2\Delta t) \end{aligned} \tag{12}$$

$$\begin{aligned} x(3\Delta t) &= A^{-1}Bx(2\Delta t) + A^{-1}b(3\Delta t) \\ &= \left(A^{-1}B\right)^2 A^{-1}b(\Delta t) + A^{-1}BA^{-1}b(2\Delta t) \\ &\quad + A^{-1}b(3\Delta t). \end{aligned} \tag{13}$$

This can be repeated, so that at any future time $p\Delta t$, we have

$$x(p\Delta t) = \sum_{k=0}^{p-1} \left(A^{-1}B\right)^k A^{-1}b((p-k)\Delta t). \tag{14}$$

At every point in time $t \in [0, p\Delta t]$, the input vector $b(t)$ must be feasible, i.e., we must have $i_s(t) \in \mathcal{F}$. To simplify the notation, we define $\mathcal{F}_b$ to be the feasible space of $b(t)$ so that $b(t) \in \mathcal{F}_b$ if and only if $i_s(t) \in \mathcal{F}$. Under that condition, we are interested in the maximum and minimum worst-case voltage attained (separately) by each component of the $v(p\Delta t)$ subvector of $x(p\Delta t)$. In order to compactly capture this notion, we introduce the following notation.

Suppose $f(c) : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is a vector function whose components are denoted $f_1(c), \ldots, f_n(c)$, and let $\mathcal{A} \subset \mathbb{R}^n$. Now, define a vector $z \in \mathbb{R}^{2n}$, such that, with $i \in \{1, 2, \ldots, n\}$, z_i is the maximum of $f_i(c)$ over all $c \in \mathcal{A}$ and z_{n+i} is the minimum of $f_i(c)$ over all $c \in \mathcal{A}$. We will denote this by the following operator:

$$z = \underset{\forall c \in \mathcal{A}}{\text{eopt}}(f(c)) \tag{15}$$

where the "eopt$(\cdot)$" notation is introduced to denote the fact that this is an *element-wise* optimization. Notice that each component $z_i, \forall i = 1, \ldots, n$ may be found separately by solving the following maximization problem:

$$\begin{aligned} &\text{maximize } f_i(c) \\ &\text{subject to } c \in \mathcal{A} \end{aligned} \tag{16}$$

and each component $z_{n+i}, \forall i = 1, \ldots, n$ may be found separately by solving the following minimization problem:

$$\begin{aligned} &\text{minimize } f_i(c) \\ &\text{subject to } c \in \mathcal{A}. \end{aligned} \tag{17}$$

Using this notation, we can express the maximum and minimum worst-case voltage drops at all nodes at time $\tau = p\Delta t$ by

$$x_{opt}(\tau) = \underset{\forall b(t) \in \mathcal{F}_b}{\text{eopt}} \left(\sum_{k=0}^{p-1} \left(A^{-1}B\right)^k A^{-1}b((p-k)\Delta t) \right) \tag{18}$$

where the notation "$\forall b(t) \in \mathcal{F}_b$" means that, for every time point $t \in [0, \tau]$, the current $i_s(t)$ satisfies all the (local and global) constraints.

In practice, we are interested in the steady state solution where the system becomes independent of the initial condition ($b(t) = 0, \forall t \leq 0$). Since the RLC grid model is a dynamical system with a limited memory of its past, then the steady state solution can be obtained by evaluating (18) at points far away from the initial condition, i.e., as $p \rightarrow \infty$. Thus, the general solution to the exact voltage drop maximization and minimization problem is

$$x_{opt}(\tau) = \lim_{p \to \infty} \underset{\forall b(t) \in \mathcal{F}_b}{\text{eopt}} \left(\sum_{k=0}^{p-1} \left(A^{-1}B\right)^k A^{-1}b((p-k)\Delta t) \right). \tag{19}$$

Although the RLC model is dynamic, i.e., its currents and voltages vary with time, the constraints are DC and do not depend on time. Hence, $\mathcal{F}$ is the same for each time step. With this, the components of (19) can be "decoupled," leading to

$$x_{opt}(\tau) = \lim_{p \to \infty} \sum_{k=0}^{p-1} \underset{\forall b \in \mathcal{F}_b}{\text{eopt}} \left[\left(A^{-1}B\right)^k A^{-1}b \right] \tag{20}$$

where b is simply an $(n + m) \times 1$ vector of variables with a subvector i_s that satisfies the (local and global) constraints, without reference to any particular point in time. This is a major simplification of the problem, as it has the advantage that the number of constraints for each optimization is fixed and does not span all previous time points.

Unfortunately, (20) is of theoretical interest only. It cannot be directly computed, as it stands, because it has to be evaluated for a large number of time steps until convergence and because the element-wise optimizations require a matrix inverse A^{-1} and LPs that are proportional to the number of nodes plus the number of inductive branch currents in the grid which for modern designs is in the thousands or even the millions. Also, it includes matrix-matrix multiplications at every time step. As an alternative, and in the rest of the paper, we propose a solution to efficiently verify the voltage drops on the RLC grid. We will find an upper bound on the maximum worst-case voltage drop and a lower bound on the minimum worst-case voltage drop on every node on the grid. These bounds will first require *a few* applications of the "eopt$(\cdot)$" operator. Instead of p applications, only r applications are needed where r will be defined below. The vector resulting from the above operation will then be combined with an *analytical expression* to give the bounds at $p \rightarrow \infty$.

A. Discussion

Before we proceed with our proposed solution, we will provide some discussion to highlight the impact of this paper and the significance of grid verification under an RLC grid model. The discussion will focus on a small 6-node RLC grid.

One of the difficulties in analyzing RLC grid models, especially the portions around a C4 pad, is that only certain temporal arrangements of the circuit currents may lead to the worst-case voltage drops on the grid. An example is shown in Fig. 2, which illustrates the current stimulus that induces the maximum worst-case voltage drop on one of

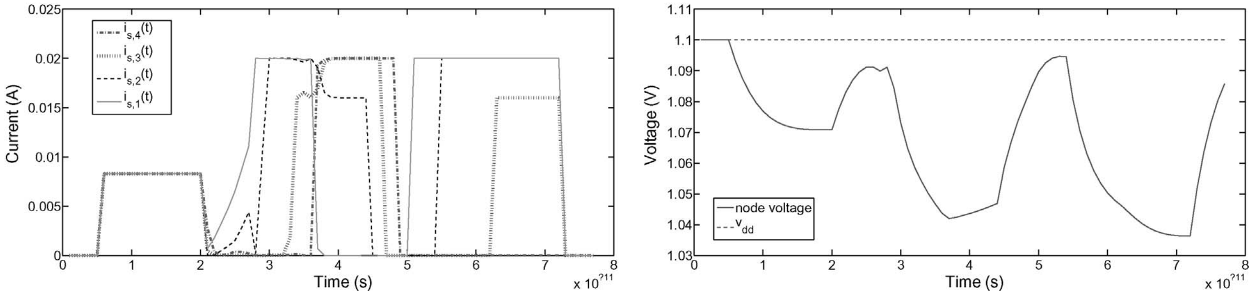


Fig. 2. Current configuration leading to the maximum worst-case voltage drop and the corresponding voltage waveform obtained by HSPICE simulation.

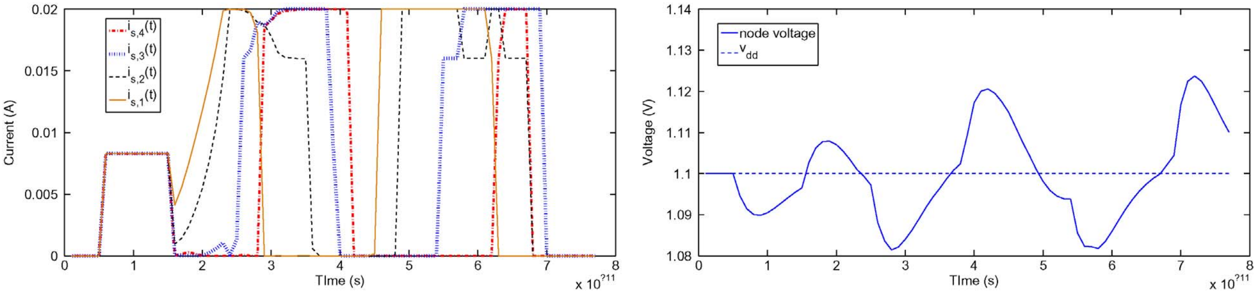


Fig. 3. Current configuration leading to the minimum worst-case voltage drop (overshoot) and the corresponding voltage waveform obtained by HSPICE simulation.

the grid nodes. The figure also shows the voltage waveform resulting from simulating the grid, using HSPICE, subject to that current stimulus. This current stimulus was found by running the exact verification approach in (20). The approach also reported a maximum worst-case voltage drop of 63.68884 mV. This worst-case value was confirmed by the voltage waveform generated by HSPICE as it showed a maximum voltage drop of 63.67949 mV. Another example is shown in Fig. 3, which illustrates the current stimulus that causes the maximum overshoot (minimum worst-case voltage drop) on the same grid node. This stimulus, along with a maximum worst-case overshoot of 23.7449925 mV, is also found by running the exact verification technique. Again, HSPICE was used to confirm the results by simulating the grid subject to that current stimulus. In this case, HSPICE reported a maximum overshoot of 20.96346 mV. The HSPICE-generated voltage waveform is also shown in Fig. 3. An important aspect that was revealed by simulation using HSPICE is that the worst-case voltage waveforms gradually build up in magnitude until they reach their peaks.

As one can see, the current waveforms leading to the worst-case drops are quite complex even for this small 6-node grid. Such complexity arises from the transient system behavior and is compounded by imposing local and global constraints on the power grid current sources under the vectorless verification framework. Note that the global constraints are the reason some currents do not transition from zero to their peak values at certain points in time. As a result, to uncover such current loads, along with the worst-case voltage drops, we employed the novel and unique approach that is derived in (20). From Table I in Section V, it is also clear that using this exact approach becomes computationally prohibitive even for small-size grids. Therefore, our proposed upper and lower bound approach, is a fast verification technique that can uncover the fact that *there exists* a combination of feasible current waveforms which can result in voltage violations, be they overshoots or voltage drops.

IV. Proposed Solution

A. Grid Transformation

An alternative way of writing (6) is

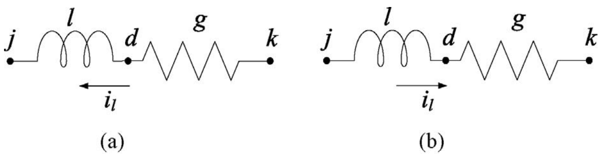
$$i(t-\Delta t) = \left(\frac{L}{\Delta t}\right)^{-1} M^T v(t) + i(t) \tag{21}$$

where $i(t)$ and $i(t-\Delta t)$ are the inductive branch currents at time t and $t-\Delta t$, respectively. Note that these are the currents in what we previously defined as *rl*-branches. Fig. 4 shows a sample branch with two external nodes j and k, with voltage drops v_j and v_k, an internal node d, with voltage drop v_d, and a branch current i_l. Depending on the assigned direction of the inductive branch current, two cases arise. Fig 4(a) shows the case where i_l flows away from the internal node d. The KCL equation at node d at time t can be written as

$$g v_d(t) - g v_k(t) = i_l(t). \tag{22}$$

(Recall that v_d and v_k are *voltage drops*). Equation (22) is precisely the row in (3) corresponding to node d. Let the vector g_d^T denote the same row of the matrix G. Then, a compact form of (22) becomes

$$g_d^T v(t) = i_l(t). \tag{23}$$



(a) (b)

Fig. 4. Typical *rl*-branch. (a) First case. (b) Second case.

Applying (23) at time $t - \Delta t$, we get

$$g_d^T v(t - \Delta t) = i_l(t - \Delta t). \tag{24}$$

Fig 4(b) shows the case where i_l flows toward the internal node d. In this case, the KCL equation at node d at time $t - \Delta t$ gives

$$-g_d^T v(t - \Delta t) = i_l(t - \Delta t). \tag{25}$$

Depending on the current assignment in every *rl*-branch, we can either apply (24) or (25) to all the internal nodes to obtain

$$\hat{G} v(t - \Delta t) = i(t - \Delta t) \tag{26}$$

where $\hat{G}$ is an $m \times n$ matrix whose kth row is that of either G or $-G$ and it corresponds to the internal node of the kth *rl*-branch. Combining (21) and (26) eliminates $i(t - \Delta t)$ and leads to

$$\hat{G} v(t - \Delta t) = \left(\frac{L}{\Delta t}\right)^{-1} M^T v(t) + i(t). \tag{27}$$

Then, we substitute $i(t)$ from (27) into (5) to get

$$\left(G+\frac{C}{\Delta t}+M\left(\frac{L}{\Delta t}\right)^{-1} M^T\right) v(t) = \left(\frac{C}{\Delta t}+M\hat{G}\right) v(t-\Delta t)+i_s(t). \tag{28}$$

It is clear that (28) technically eliminates the inductive branch currents from the equation variables thus drastically reducing size of the system under verification. A similar transformation technique was employed in [11] for a different grid model and using the trapezoidal finite difference approximation. In our paper, the RLC grid represented by (5) and (6) is now transformed into a circuit with similar properties as the RC grid model [12]. As is the case for the system matrix in the RC case, and unlike A in (9), $G + \frac{C}{\Delta t} + M(\frac{L}{\Delta t})^{-1} M^T$ is sparse, symmetric, positive definite, and a banded $\mathcal{M}$-matrix. For such a matrix, it is well known that its inverse has entries whose values decay exponentially as one moves away from the diagonal [13]. Such matrices are often referred to as *practically sparse* and their inverses can be efficiently computed, with a high degree of precision, by using a sparse approximate inverse technique (SPAI) [14]. As we will see in Section IV-B, computing the matrix inverse efficiently and accurately is crucial for our approach.

B. Upper and Lower Bounds

1) *Overview:* Now we will show how the upper and lower bounds can be obtained. The derivation of the main result will proceed in two steps. First, we will prove that the bounds at time t can be obtained from the bounds at time $t - r\Delta t$, where r is a crucial parameter that we will define below and which represents a small number of time steps. Second, we will show that this means that the true upper and lower bounds at infinity can be written as a matrix-vector product. The matrix is the inverse of $(I - R)$, where R is a matrix to be introduced below, that depends on r. The vector will be obtained by r applications of the "eopt(·)" operator.

2) *Bounds on Voltage Drops That Are $r\Delta t$ Steps Apart:* We use (28) to write an equation analogous to (10) as follows:

$$v(t) = D^{-1} E v(t - \Delta t) + D^{-1} i_s(t) \tag{29}$$

where $D = G+\frac{C}{\Delta t}+M(\frac{L}{\Delta t})^{-1} M^T$ and $E = \frac{C}{\Delta t}+M\hat{G}$. Because the RLC grid is a stable linear system and because the backward difference approximation used in (5) and (6) is absolutely stable [15], it follows that for $i_s(t) = 0$ $\forall t$ and any bounded initial state $v(0)$, (29) converges to 0 as $t \to \infty$. One can show that this means that $\rho(D^{-1}E) < 1$, where $\rho(D^{-1}E)$ is the magnitude of the largest eigenvalue of $D^{-1}E$, called the spectral radius of the matrix. Following the same reasoning as in Section III, the voltage on the grid at any time $p\Delta t$ can be written as

$$v(p\Delta t) = \sum_{k=0}^{p-1} \left(D^{-1}E\right)^k D^{-1} i_s((p-k)\Delta t) \tag{30}$$

where $i_s(t) \in \mathcal{F}$ for $t \in [0, p\Delta t]$. Let r be a positive integer such that p/r is an integer, then grouping every r consecutive terms in (30) leads to

$$v(p\Delta t) = \sum_{k=0}^{p/r-1} \left(D^{-1}E\right)^{kr} \left[\sum_{q=0}^{r-1} \left(D^{-1}E\right)^q D^{-1} i_s((p-q-kr)\Delta t)\right]. \tag{31}$$

As we will see in Section IV-B4, it is possible to choose an appropriate choice of r, which allows the computation of the upper and lower bound vectors. Define

$$N = \left(D^{-1}E\right)^r \tag{32}$$

$$s(t) = \sum_{q=0}^{r-1} \left(D^{-1}E\right)^q D^{-1} i_s(t - q\Delta t) \tag{33}$$

keeping in mind that both N and $s(t)$ are functions of r.

Claim 1: Equation (31) can be represented as a recursive relation as follows:

$$v(t) = Nv(t - r\Delta t) + s(t) \quad \forall t = r\Delta t, 2r\Delta t, \ldots, p\Delta t \tag{34}$$

where $v(0) = 0$ for all $t \leq 0$.

Proof: Let $v^{\star}(t)$ be the value of (31), and $v^{\dagger}(t)$ be that of (34). It is obvious that the case at time $t = r\Delta t$ is satisfied, i.e., $v^{\star}(r\Delta t) = v^{\dagger}(r\Delta t) = \sum_{q=0}^{r-1}(D^{-1}E)^{q}D^{-1}i_s(r\Delta t - q\Delta t)$. Then, the claim is true by induction if we prove the following for all $t = p\Delta t$:

$$v^{\star}(p\Delta t - r\Delta t) = v^{\dagger}(p\Delta t - r\Delta t) \Rightarrow v^{\star}(p\Delta t) = v^{\dagger}(p\Delta t). \tag{35}$$

Assuming the left-hand side of (35) is true, we get

$$\begin{aligned} v^{\dagger}(p\Delta t) &= Nv^{\dagger}(p\Delta t - r\Delta t) + s(p\Delta t) \\ &= Nv^{\star}(p\Delta t - r\Delta t) + s(p\Delta t). \end{aligned} \tag{36}$$

Using (31) at $t = (p\Delta t - r\Delta t)$, we obtain

$$\begin{aligned} &v^{\star}(p\Delta t - r\Delta t) = \\ &\sum_{k=0}^{\frac{p-r}{r}-1}(D^{-1}E)^{kr}\left[\sum_{q=0}^{r-1}(D^{-1}E)^{q}D^{-1}i_s\left((p-r-q-kr)\Delta t\right)\right]. \end{aligned} \tag{37}$$

Combining (36) and (37), and using (32) and (33) leads to

$$\begin{aligned} &v^{\dagger}(p\Delta t) = \\ &\sum_{k=-1}^{\frac{p-r}{r}-1}(D^{-1}E)^{(k+1)r}\left[\sum_{q=0}^{r-1}(D^{-1}E)^{q}D^{-1}i_s\left((p-r-q-kr)\Delta t\right)\right]. \end{aligned} \tag{38}$$

Let $j = k+1$, then we can re-write equation (38) as follows:

$$\begin{aligned} &v^{\dagger}(p\Delta t) = \\ &\sum_{j=0}^{p/r-1}(D^{-1}E)^{jr}\left[\sum_{q=0}^{r-1}(D^{-1}E)^{q}D^{-1}i_s\left((p-q-jr)\Delta t\right)\right]. \end{aligned} \tag{39}$$

Therefore, $v^{\star}(p\Delta t) = v^{\dagger}(p\Delta t)$. This completes the proof. ■

We will now show that (34) leads to upper and lower bound vectors on the worst-case voltage drop on the grid. Define

$$\begin{bmatrix} s_{max}^{(r)}(t) \\ s_{min}^{(r)}(t) \end{bmatrix} = \underset{\forall i_s \in \mathcal{F}}{\text{eopt}}\; s(t). \tag{40}$$

Notice from (33) that (40) depends on the last r time steps. But we know that the current constraints are DC and that $\mathcal{F}$ is the same for each time point, then the result of running "eopt($s(t)$)" would be the *same* for any t. We can therefore decouple the components of (40) leading to

$$\begin{bmatrix} s_{max}^{(r)}(t) \\ s_{min}^{(r)}(t) \end{bmatrix} = \underset{\forall i_s(t) \in \mathcal{F}}{\text{eopt}} \sum_{q=0}^{r-1}\left(D^{-1}E\right)^{q}D^{-1}i_s(t-q\Delta t) \tag{41}$$

$$= \sum_{q=0}^{r-1}\underset{\forall i_s \in \mathcal{F}}{\text{eopt}}\left[\left(D^{-1}E\right)^{q}D^{-1}i_s\right] \triangleq \begin{bmatrix} s_{max}^{(r)} \\ s_{min}^{(r)} \end{bmatrix} \tag{42}$$

where i_s is simply an $n \times 1$ vector of variables that satisfies both local and global constraints. Thus, the result of applying eopt($s(t)$) for any consecutive r time steps is the vector composed of $s_{max}^{(r)}$ and $s_{min}^{(r)}$.

Claim 2: Let Q be the matrix of the element-wise absolute values of the entries of N. Let S be the element-wise non-positive matrix $S = \frac{1}{2}(N-Q)$. If $v_{lb}(0) = v_{ub}(0) = 0$, then for $v_{lb}(t)$ and $v_{ub}(t)$ computed as

$$\begin{bmatrix} v_{ub}(t) \\ v_{lb}(t) \end{bmatrix} = \begin{bmatrix} N-S & S \\ S & N-S \end{bmatrix}\begin{bmatrix} v_{ub}(t-r\Delta t) \\ v_{lb}(t-r\Delta t) \end{bmatrix} + \begin{bmatrix} s_{max}^{(r)} \\ s_{min}^{(r)} \end{bmatrix} \tag{43}$$

we have $v_{lb}(t) \leq v(t) \leq v_{ub}(t)$, $\forall t = r\Delta t, 2r\Delta t, \ldots, p\Delta t$.

Proof: Since $v(0) = 0$ for all $t \leq 0$, then the base case at time $t = 0$ is satisfied as an equality, i.e., $v_{lb}(0) = v(0) = v_{ub}(0) = 0$. Then, the claim is true by induction if we prove the following for all $t = r\Delta t, 2r\Delta t, \ldots, p\Delta t$:

$$\left.\begin{aligned} v_{ub}(t-r\Delta t) \geq v(t-r\Delta t) \\ v_{lb}(t-r\Delta t) \leq v(t-r\Delta t) \end{aligned}\right\} \Rightarrow \begin{aligned} v_{ub}(t) \geq v(t) \\ v_{lb}(t) \leq v(t). \end{aligned} \tag{44}$$

Assuming the left-hand side of (44) is true, and by making use of the fact that S is a non-positive matrix and $N-S$ is a non-negative matrix, we get

$$\begin{aligned} (N-S)v_{lb}(t-r\Delta t) &\leq (N-S)v(t-r\Delta t) \\ &\leq (N-S)v_{ub}(t-r\Delta t) \\ Sv_{ub}(t-r\Delta t) \leq Sv(t-r\Delta t) &\leq Sv_{lb}(t-r\Delta t). \end{aligned} \tag{45}$$

Adding the above two equations leads to

$$\begin{aligned} (N-S)v_{lb}(t-r\Delta t) + Sv_{ub}(t-r\Delta t) &\leq Nv(t-r\Delta t) \\ &\leq (N-S)v_{ub}(t-r\Delta t) + Sv_{lb}(t-r\Delta t). \end{aligned} \tag{46}$$

Adding (46) to $s_{min}^{(r)} \leq s(t) \leq s_{max}^{(r)}$, then utilizing (34) and (43), we obtain

$$v_{lb}(t) \leq v(t) \leq v_{ub}(t).$$

This completes the proof. ■

Therefore, given bounds on $v(t-r\Delta t)$, bounds on $v(t)$ exist and can be computed using claim 2.

3) *Bounds in the Steady State:* Even though one can only compute bounds on voltage drops that are r time steps ahead in time, we are interested in upper and lower bounds in the steady state case, i.e., when $t = p\Delta t \to \infty$. Let $y(t) = [v_{ub}(t) \quad v_{lb}(t)]^{T}$ and $w = [s_{max}^{(r)} \quad s_{min}^{(r)}]^{T}$. Then, in matrix notation, (43) yields

$$y(t) = Ry(t-r\Delta t) + w \tag{47}$$

where R is a $2n \times 2n$ matrix defined as

$$R = \begin{bmatrix} N - S & S \\ S & N - S \end{bmatrix}. \tag{48}$$

Writing (47) at every time step $r\Delta t, 2r\Delta t, \ldots, p\Delta t$, yields

$$\begin{aligned} y(r\Delta t) &= w \\ y(2r\Delta t) &= Rw + w \\ y(3r\Delta t) &= R^2 w + Rw + w \\ &\vdots \\ y(p\Delta t) &= (R^{p-1} + \ldots + R + I)w. \end{aligned} \tag{49}$$

It is therefore clear that the convergence of the upper/lower bound depends on the convergence of the matrix series $(R^p + \ldots + R + I)$ as $p \to \infty$. It is well known [10] that the series $(R^p + \ldots + R + I)$ converges as $p \to \infty$ if and only if $\rho(R) < 1$, under which condition the limit of the series is $(I - R)^{-1}$. Consequently, as $p \to \infty$, the series in (49) converges to

$$\begin{bmatrix} v_{ub}(\infty) \\ v_{lb}(\infty) \end{bmatrix} = \lim_{p \to \infty} y(p\Delta t) = (I - R)^{-1} w \tag{50}$$

provided that $\rho(R) < 1$, a condition which we now explore.

4) *Convergence:* Denote the set of all eigenvalues of a matrix X by $\sigma(X)$. We mentioned earlier that the spectral radius is the magnitude of the largest eigenvalue and since $\rho(D^{-1}E) < 1$, then we can say

$$\max_{\forall \lambda \in \sigma(D^{-1}E)} |\lambda| < 1. \tag{51}$$

The *spectral mapping theorem* [16] provides that, if X is a square matrix and k is an integer, then $\sigma(X^k) = \{\lambda^k : \lambda \in \sigma(X)\}$. From this and by using (32), it follows that

$$\rho(N) = \max_{\forall \lambda \in \sigma(D^{-1}E)} |\lambda^r| < 1. \tag{52}$$

So, in order to show that $\rho(R) < 1$, it will be useful to relate the spectral radius of R to that of N.

Claim 3: The set of the eigenvalues of R, called the spectrum of R, is the union of the spectrums of N and Q.

Proof: Recall that the eigenvalues of R are the roots of the following characteristic polynomial:

$$\det(R - \lambda I) = 0 \tag{53}$$

where $\lambda \in \mathbb{C}$. Replacing R by its block-matrix representation from (48), and using $|X|$ to denote the $\det(X)$, then

$$\det(R - \lambda I) = \begin{vmatrix} N - S - \lambda I & S \\ S & N - S - \lambda I \end{vmatrix}. \tag{54}$$

Since the determinant of a matrix is invariant under elementary row additions, and by using the block-matrix determinant [17], we manipulate (54) to obtain

$$\begin{aligned} \det(R - \lambda I) &= \begin{vmatrix} N - \lambda I & N - \lambda I \\ S & N - S - \lambda I \end{vmatrix} \\ &= |N - \lambda I| \times \left| N - S - \lambda I - S(N - \lambda I)^{-1}(N - \lambda I) \right|. \end{aligned} \tag{55}$$

Using (53) and (55), it is easy to see that the eigenvalues of R should satisfy

$$\det(N - \lambda I) = 0 \tag{56}$$

or

$$\det(N - 2S - \lambda I) = 0. \tag{57}$$

The spectrum of R is thus the union of the spectrums of N and $N - 2S$. Recall from claim 2 that $2S = N - Q$. Hence, $Q = N - 2S$. This completes the proof. ■

As a result, the series in (49) converges, as $p \to \infty$, if and only if $\rho(N) < 1$ and $\rho(Q) < 1$. We already saw that the spectral radius of N is less than 1, but that of Q is not trivial and is expensive to compute. In this paper, we employ a conservative criterion to bound $\rho(Q)$ based on the fact that, for any matrix X and any matrix p-norm, we have $\rho(X) \leq \|X\|_p$ [10]. More specifically, we will bound $\rho(Q)$ by the minimum of the $\|Q\|_\infty$ and the $\|Q\|_1$ norms that are defined as

$$\|Q\|_\infty \triangleq \max_{1 \leq i \leq n} \sum_{j=1}^{n} |q_{ij}| \tag{58}$$

$$\|Q\|_1 \triangleq \max_{1 \leq j \leq n} \sum_{i=1}^{n} |q_{ij}|. \tag{59}$$

Since (58) and (59) are the maximum row/column sum of the absolute value of the entries of Q, and since Q is the matrix of element-wise absolute value of the entries of N, we have $\|N\|_1 = \|Q\|_1$ and $\|N\|_\infty = \|Q\|_\infty$. Therefore, a sufficient condition for the convergence of the series in (49), as $p \to \infty$, is when the minimum of $\|N\|_\infty$ and $\|N\|_1$ is less than unity.

This can be achieved by making use of the fact that for any matrix X such that $\rho(X) < 1$, we have $\lim_{k \to \infty} \|X^k\|_p = 0$ for any p-norm [10]. This implies that $\|X^k\|_p$ asymptotically decreases with increasing k. From this, and from the fact that $N = (D^{-1}E)^r$ and $\rho(D^{-1}E) < 1$, we can say that

$$\exists\, r_0 \in \mathbb{N} \text{ such that } \|(D^{-1}E)^{r_0}\|_p < 1 \tag{60}$$

where $\mathbb{N}$ is the set of integers. Therefore, our approach will be to find such an r_0 for either $p = 1$ or $p = \infty$, set $r = r_0$, so that either $\|N\|_1 < 1$ or $\|N\|_\infty < 1$, which ensures convergence.

In Section V, we show that the value r_0 required to guarantee convergence is small in practice.

5) *Choice of Time Step:* We digress briefly to mention that, when transient analysis is used, the exact worst-case voltage drop values, as well as their upper and lower bounds, depend on the choice of time step Δt. To see why, observe that in all of

Algorithm 1 RLC_BOUND

```
Input: RLC circuit matrices in (5) and (6)
Output: $y$ in (50)
 1: Transform the RLC grid into the form described by equation (28)
 2: Compute $D$ and $E$
 3: Compute $D^{-1}$ using SPAI with some $\epsilon$ as the stopping criterion
    for all the columns
 4: $r = 1$
 5: Compute $w = \underset{\forall i \in \mathcal{F}}{\text{eopt}}\, D^{-1} i$
 6: $(w, N, r)$ = VAR_STEP_OPTIMIZATION($D^{-1}$, $E$, $w$)
 7: Construct $R$ using (48)
 8: Compute $y$ using (50):
        LU-factorize $(I - R)$ : $(I - R) = L \cdot U$
        Forward solve: $Lz = w$
        Backward solve: $Uy = z$
```

the above derivations, we are implicitly assuming that currents may change arbitrarily in their feasibility region $\mathcal{F}$ within Δt time. This would allow currents to switch from zero to their local constraint values, or vice versa, within a single time step. This is clearly not true when Δt is very small because the switching activity of the currents is dictated by the speed of the underlying logic circuit. Note that our implicit assumption, when used with a small Δt, is conservative because obviously all slower current scenarios are covered. On the other hand, Δt has to be small enough to capture the true transient behavior of the grid voltages. As a result, design expertise should guide the choice of Δt, by striking a balance between the dynamics of the current loads and the grid transient behavior. We also caution the reader that too small a Δt might introduce noise in the computations and render them useless due to roundoff error issues that arises when Δt is close to the machine epsilon [18].

Note that the convergence of our approach, as evident from Section IV-B4, is immune to the choice of time step Δt. There always exists a value of r for which the matrix series in (49) converges.

C. Upper and Lower Bound Algorithm

In this section, we present a high-level description of our proposed approach that finds upper and lower bounds on the worst-case voltage drops on the grid. Algorithm 1, RLC_BOUND, takes as input an RLC circuit along with the local and global constraints. It returns upper and lower bounds on the voltage drop on every node on the grid as the final output.

The algorithm starts by constructing the reduced system as described by (28), and it computes the corresponding matrices D and E. Note that such computation is cheap as it only requires matrix additions. The $M\left(\frac{L}{\Delta t}\right)^{-1} M^T$ submatrix of D is extracted from the circuit topology. It is basically the "G matrix" of the inductive elements. Denote by d_{ij} the entry of D in the ith row and jth column. Then, an inductor l_c between nodes a and b contributes to four locations in D. A value of $\frac{\Delta t}{l_c}$ is added to d_{aa} and d_{bb}, and a value of $-\frac{\Delta t}{l_c}$ is added to d_{ab} and d_{ba}. As for the $M\hat{G}$ submatrix of E, it is constructed as follows. Consider the *rl*-branch shown in Fig. 4. If the current assignment is away from node d, as shown in Fig 4(a), then g_d^T is added to the dth row of $M\hat{G}$ and $-g_d^T$ is added to its jth row. If the current assignment is toward node j, as shown in Fig 4(b), then $-g_d^T$ is added to the dth row of $M\hat{G}$ and g_d^T is added to its jth row.

Procedure 1 VAR_STEP_OPTIMIZATION

```
1: $N = D^{-1}E$
2: while $(min(\|N\|_\infty, \|N\|_1) \geq 1)$ do
3:     Increment $r$: $r = r + 1$
4:     $P = ND^{-1}$
5:     Update $w$: $w = w + \underset{\forall i_s \in \mathcal{F}}{\text{eopt}}\ Pi_s$
6:     Update $N$: $N = N \times (D^{-1}E)$
7: end while
8: return $(w, N, r)$
```

After that, D^{-1} is efficiently computed (using SPAI [14]) and stored in memory. By capturing *only* the significant entries of the inverse, the SPAI method guarantees a *sparse* yet very accurate approximation of D^{-1}. For more information on SPAI, the reader is referred to Appendix A. Once D^{-1} is computed, the algorithm then computes w in (42) for $r = 1$ which requires n maximizations and n minimizations. The fact that D^{-1} is sparse allows us to reduce the size of each of these optimizations resulting in quite fast optimization runs. Then, RLC_BOUND calls Procedure 1, VAR_STEP_OPTIMIZATION. The purpose of this procedure is to find an r, and the corresponding w, such that either $\|N\|_1 < 1$ or $\|N\|_\infty < 1$. It first computes N for $r = 1$ (step 1) and stores it in memory. The procedure then iterates while checking $\|N\|_1$ and $\|N\|_\infty$. As long as the condition fails, r is incremented and an *update* of w is carried out as follows. Based on (42), we can express w for a given r as

$$w = \sum_{q=0}^{r-2} \underset{\forall i_s \in \mathcal{F}}{\text{eopt}} \left[(D^{-1}E)^q D^{-1} i_s\right] + \underset{\forall i_s \in \mathcal{F}}{\text{eopt}} \left[(D^{-1}E)^{r-1} D^{-1} i_s\right].$$

The first term on the right-hand side is simply w at $r - 1$, and $(D^{-1}E)^{r-1}$ in the second term is N, also at $r - 1$. Since both quantities were computed and stored *a priori*, the update simply requires a matrix-matrix multiplication (step 4), and one "eopt" (step 5). Procedure 1 also keeps on updating N using (32). As soon as an r is reached such that either $\|N\|_1 < 1$ or $\|N\|_\infty < 1$, Procedure 1 exits and returns the current r, N, and w values to Algorithm 1. The latter resumes execution at step 7. As discussed in Section IV-B4, the series in (49) is now guaranteed to converge and the closed form solution of the upper and lower bounds is computed in steps 7–8 via an LU factorization and a forward/backward solve.

Note that the multiplications in steps 1, 4, and 6 of Procedure 1 drop any entry in the result whose value is below some user-defined threshold. This maintains the sparsity structure of N and P as long as r is small, thus allowing the reduction in the size of the LPs required for updating w is step 5.

V. Experimental Results

Algorithm 1, along with Procedure 1, has been implemented in C++. The algorithm uses the Mosek optimization package [19] to solve the required linear programs. We carried out several experiments on a set of randomly-generated power

TABLE I

Speed and Accuracy Comparisons of the Proposed Approach with the Exact Approaches

Power Grid	**Our Approach**		**Comparison to Exact Approach**		**Cost of Exact Methods**		
Nodes	r	cpu Time	Upper Bound Error	Lower Bound Error	cpu Time of the Approach in (20)	cpu Time of the Approach in [7]	Time Steps
51	3	1.55 s	1.4627 mV	1.32 mV	19.81 s	38.49 min	73
88	5	7.87 s	3.764 mV	5.07 mV	133.17 s	5.36 h	110
150	2	3.38 s	1.007 mV	1.1658 mV	4.09 min	6.62 h	51
187	2	4.47 s	3.923 mV	4.13 mV	16.74 min	3.08 days	105
250	1	3.716 s	0.5 mV	2.058 mV	59.2 min	–	129
295	5	18.204 s	7.34 mV	6.35 mV	35.97 min	–	62
2325	3	2.16 min	3.40 mV	1.47 mV	4.3 h	–	95
8835	3	39.93 min	9.9 mV	5.2 mV	3.45 days	–	101
13 527	1	1.22 h	–	–	–	–	–
19 613	3	2.91 h	–	–	–	–	–
42 897	4	4.16 h	–	–	–	–	–
76 328	4	9.83 h	–	–	–	–	–
118 971	7	22.32 h	–	–	–	–	–
170 193	5	36.19 h	–	–	–	–	–

TABLE II

Runtime Breakdown of the Proposed Approach

Power Grid	**Our Approach Runtime Breakdown**			
Nodes	Approximate Inverse (Step 3 in Algorithm 1)	Matrix Multiplication (Steps 1, 4, and 6 in Procedure 1)	Optimization (Step 5 in Algorithm 1 and Step 5 in Procedure 1)	System Solve (Step 8 in Algorithm 1)
8835	12.39 min	15.37 min	8.73 min	3.42 min
13 527	15.45 min	32.80 min	18.67 min	6.27 min
19 613	22.68 min	1.59 h	47.20 min	9.33 min
42 897	1.4 h	1.67 h	35.86 min	29.52 min
76 328	2.91 h	4.28 h	1.16 h	1.47 h
118 971	3.82 h	12.39 h	3.12 h	3 h
170 193	5.77 h	19.55 h	5.36 h	5.51 h

grids, using a 2.6 GHz Linux machine with 8 GB of memory. The grids were based on user specifications, including grid dimensions, metal layers, pitch and width per layer, percentage of *rl*-branches, and C4 and current source distribution. These C4s and current sources were randomly placed on the grid. Moreover, the circuits generated include user-defined RLC models for the package-grid interconnections and all the experiments were performed on grids with up to ten global constraints and 1.1 V supply voltage.

To assess the quality of our results, we computed the maximum and minimum worst-case voltage drops on the grid using two exact verification approaches. The first approach was proposed by the authors of [7] where these worst-case voltage drops at every node are computed by solving a set of LPs at consequent time steps until convergence. The drawback of this approach is that the number of constraints is multiplied by the number of time steps which, even for small grids, can lead to potentially very large optimization problems. The second exact method was formulated in this paper for the RLC case as given in (20). Even though the method also uses LPs to compute the worst-case voltage drops, its advantage over the previous exact technique is that the number of constraints at each time step is fixed and does not span previous time-points. Still, the method requires a matrix inverse that is computed once, in addition to matrix-matrix multiplications and solving linear programs at every time step until convergence.

Note that we used the same Δt value for the exact methods as well as our upper/lower bound technique. We assumed the underlying logic circuitry of all the grids under study can run at a maximum frequency of 10 GHz, so that, as was done in [20], we use a Δt value of 1 ps.

Table I shows the speed and accuracy of our proposed technique for computing the upper and lower bound voltage drop vectors. We compare our results with the vectors of maximum and minimum worst-case voltage drops resulting from the two exact approaches we discussed earlier. As an accuracy measure, we report the maximum absolute error between the upper and lower bound vectors obtained by our technique and the vectors resulting from the exact verification. Column 4 shows the maximum absolute error between the upper bound voltage drop vector and the exact maximum voltage drop vector, and column 5 shows the maximum absolute error between the lower bound voltage drop vector and the exact minimum voltage drop vector. The number of grid nodes is reported in column 1, and the value of r for which our approach converges is shown in column 2. The table also reports the runtimes of the three methods under consideration as well as the number of time steps required by the two exact approaches to converge. Results show that our proposed method resulted in a maximum absolute error of around $10mV$ across all nodes on the grids under study. Note that this error comes from two main sources. The first one is the sparse approximation of D^{-1}. We keep this to a minimum by using a very small ϵ_k in (67), in the range of 10^{-9} to 10^{-7}. This means that the maximum difference between an entry in the kth column of the exact inverse and the corresponding entry in

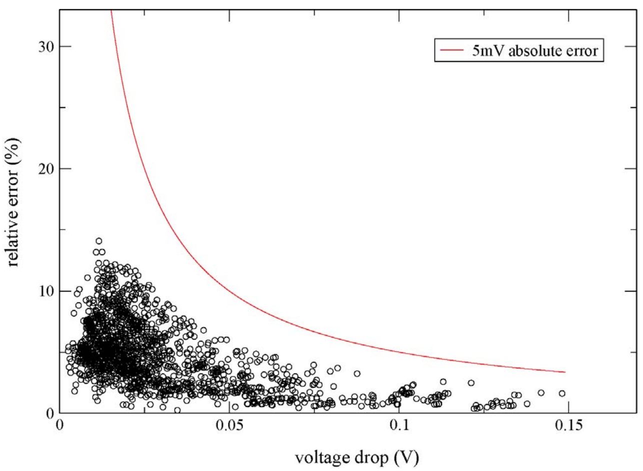


Fig. 5. Accuracy of the proposed approach.

the kth column of the approximate inverse is between 10^{-9} and 10^{-7}. The second, and main, source of error comes from the computation of the upper/lower bound vectors via (50). The reason an error is incurred is because our method performs optimizations for the first r time steps only, where r is a very small number of time steps compared to what is required in the exact approach.

The error comparisons could not be extended to larger grids and we were able to perform these comparisons for grids with a maximum of 8835 nodes since the exact approaches are computationally expensive for larger grids. The exact technique of [7] took 3.08 days for the 187-node grid, while that presented in (20) took 3.45 days for the 8835-node grid. Results show that our approach runs several orders of magnitude faster than the exact computations which exhibit impractical times even for small grids. For instance, computing the upper and lower bound vectors for the 8835-node grid took 39.93 minutes while computing the exact worst-case voltage drops using (20) took 3.45 days. This is over 124X speed up! Moreover, our technique allows one to verify grids with a relatively large number of nodes; verification of a 170 193-node grid requires 36.19 h. This would have taken several days or even weeks in the exact case. Note also that, for all the grids under consideration, the maximum value of r for which (49) converges is *very* small thus limiting the number of matrix multiplications and the number of linear programs. Thus, our method makes checking an RLC power grid a feasible and practical solution.

One might argue that such grids are still small when compared to full-chip grids containing millions of nodes. Nevertheless, our approach is important for at least two reasons: 1) it is an approach to rigorously check the safety of an *RLC* grid in a truly vectorless approach, and 2) our method can be applied to the top-level main feeder network of the grid that is not as large as grids at the end of the design flow. The ability to test the main feeder network early in the design flow is a major advantage of our technique. We believe that it can lead to practical methods for early vectorless grid verification.

Table II shows a breakdown of the runtime of the major components of our approach for the largest seven grids we investigated. Column 2 represents the time it takes to compute D^{-1} using SPAI. Column 3 shows the runtime of the matrix-matrix multiplications required, and columns 4 and 5 report the runtimes of the optimization component and the linear system solve, respectively. Even though the matrix-matrix multiplications comprise a significant portion of the total runtime of the algorithm, there is scope to achieve considerable speedup in computing D^{-1} as well as in performing the linear optimization step, by exploiting parallelism inherent in SPAI and "eopt." From Appendix A, the columns of the approximate inverse can be computed by *n independent* least-square problems. Also, recall from Section III that "eopt" is an element-wise optimization where each element can be found *separately*. As a result, given a cluster of multi-core machines, the runtime of these two components, if parallelized, would be reduced by a factor equal to the total number of cores in the cluster. This represents a significant speedup given that the combined runtime of SPAI and "eopt" for the grids under study represents 30% to 55% of the total runtime. Therefore, with parallelism exploited, our algorithm can be fast enough to verify power grids with a larger number of nodes.

To further showcase the accuracy of our algorithm, Fig. 5 shows a scatter plot of the relative errors, in percent, versus the maximum worst-case voltage drops on a 2325-node grid. The figure also shows the curve corresponding to an absolute error of 5 mV where a point on the curve represents a node voltage drop that is over-estimated by exactly 5 mV. On the other hand, a point with an over-estimation greater than 5 mV will lie in the region above the curve. From the resulting scatter plot, it is clear that such absolute errors are very small meaning that our algorithm is very accurate. Note that for some maximum worst-case voltage drops, especially those with small magnitudes, the relative error can be high. This, however, is of no concern in practice, as the critical voltage drops are those that are relatively large. In that case, the relative error incurred is small. A similar behavior is observed for the relative errors versus the minimum worst-case voltage drops.

VI. Conclusion

With the rising demand of low voltage designs, power grid verification is becoming increasingly important. We introduced a fast and efficient inductance-aware early verification technique in the framework of current constraints. Our approach transforms the RLC grid into a reduced circuit to alleviate the problems associated with its system matrices then, it computes conservative bounds on the worst-case voltage drops on the grid. The approach requires only a few linear programs and one linear system solve thus making early grid verification under the RLC model practical and scalable. Prior art simply could not handle grids of more than a couple thousand nodes, and even for such systems runtimes were prohibitive. Our method showed a drastic improvement in runtime; verifying a grid of about 170 000 nodes took just over 36 h. Results also showed that, for the girds under study, the errors incurred by our technique were relatively very small, never exceeding 10 mV.

Appendix

It is obvious from Section IV-B that D^{-1} is essential for the computation of the upper and lower bounds and needs to be computed in a fast and accurate manner so as not to compromise the speed and accuracy of the overall technique. In Section IV-A, we stated that in addition to eliminating the inductive branch currents and reducing the system size, the advantage of the RLC grid transformation is that D^{-1} is practically sparse and thus can be efficiently computed by a sparse approximate inverse technique.

In what follows, we will briefly explain the SPAI given in [14]. Let M be an $n \times n$ matrix and let the vector m_k denote the kth column of M. Note that this M is different from the incidence matrix defined in Section II-B. Let e_k be the $n \times 1$ vector consisting of all zeros, except for its kth component which is 1. Then, consider the $n \times 1$ vector, called the kth *residual*, defined by

$$r_k = Dm_k - e_k \qquad \forall k = 1, 2, \ldots, n. \tag{61}$$

It is clear that, if we choose $M = D^{-1}$, then $r_k = 0, \forall k$. In general, the norm of the residual $\|r_k\|_2$ (the Euclidean or $l-2$ norm in this case) is positive, and becomes zero only when $M = D^{-1}$. Thus, the norms of the residuals, for all k, provide a measure of how far M is from being equal to the desired matrix inverse D^{-1}. Let $M^\star \triangleq D^{-1}$, so that when M takes the value $M^\star$, then $\|r_k\|_2 = 0, \forall k$.

Given some user-defined tolerance η, SPAI computes an approximation $\hat{M}$ to the actual inverse matrix $M^\star$. It operates by finding an $\hat{M}$ matrix that gives a very small residual norm, as an approximation to the unknown $M^\star$ (which, as we saw above, would give a zero residual). The technique starts with an arbitrary initial matrix M and iteratively refines its columns by minimizing the Frobenius norm $\|DM - I\|_F$ subject to η. Recall that for any $n \times n$ matrix H, the Frobenius norm is defined as [10]

$$\|H\|_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} |h_{ij}|^2} \tag{62}$$

where h_{ij} are the entries of H, which can be re-written as

$$\|H\|_F = \sqrt{\sum_{k=1}^{n} \|He_k\|_2^2} = \sqrt{\sum_{k=1}^{n} \|h_k\|_2^2} \tag{63}$$

where h_k is the kth column of H. Obviously, the minimum of (63) (which is zero) occurs when $H = 0$. Likewise, the minimum of $\|DM - I\|_F$ (which is zero) occurs when $M = D^{-1} = M^\star$. By reducing $\|DM - I\|_F$ until it is less than η, SPAI finds an $\hat{M}$ that is a good approximation to $M^\star = D^{-1}$. To see how SPAI works, consider that

$$\begin{aligned}\|DM - I\|_F^2 &= \sum_{k=1}^{n} \|(DM - I)e_k\|_2^2 \\ &= \sum_{k=1}^{n} \|Dm_k - e_k\|_2^2 = \sum_{k=1}^{n} \|r_k\|_2^2.\end{aligned} \tag{64}$$

In this way, the problem of finding an approximate inverse $\hat{M}$ separates into n *independent* least-squares problems

> For each $k = 1, 2, \ldots, n$, vary m_k so as to minimize $\|r_k\|_2^2$ until an m_k is found for which $\|r_k\|_2 \leq \eta$.

While $\|r_k\|_2 > \eta$, the iterative refinement of m_k proposed by [14] attempts to improve on the column by appending the most profitable entries, i.e., the ones that will result in the largest reduction in $\|r_k\|_2$. In this manner, the technique automatically captures the sparsity pattern by including the significant entries of the inverse while avoiding fill-ins.

The numerical study in [14] shows that SPAI is stable and it captures the main entries of D^{-1} extremely well. Moreover, it is an extremely efficient technique. It is inherently parallel as the columns of $\hat{M}$ can be computed independently of one another. Moreover, the computation of a column is cheap; it requires a matrix-vector product and several QR factorizations of small submatrices of the original sparse matrix D.

Normally, SPAI terminates when $\|r_k\|_2$ is smaller than a user-supplied η, for every k. If $m_k^\star$ is the kth column of $M^\star = D^{-1}$, then clearly $Dm_k^\star = e_k$, so that $r_k = Dm_k - e_k = D(m_k - m_k^\star)$. This leads to

$$\|m_k - m_k^\star\|_\infty = \|D^{-1}r_k\|_\infty \leq \|D^{-1}\|_\infty \|r_k\|_\infty \tag{65}$$

where $\|\cdot\|_\infty$ is the infinity norm. We define the error tolerance $\epsilon_k > 0$, where $\epsilon_k \in \mathbb{R}$ is a scalar. As the authors did in [21], we modify SPAI so that it stops when, for every k, we have

$$\|r_k\|_\infty \leq \frac{\epsilon_k}{\|D^{-1}\|_\infty} \tag{66}$$

which achieves the condition that, for every k

$$\|\hat{m}_k - m_k^\star\|_\infty \leq \epsilon_k. \tag{67}$$

It remains to explain how $\|D^{-1}\|_\infty$ is found, when D^{-1} is unknown! This is easy to do, in fact, because D is an $\mathcal{M}$-matrix, as follows. Let a be the $n \times 1$ vector of all 1s, so that $a^T = [1\ 1\ \cdots\ 1]$. Then, $D^{-1}a$ is a vector whose every component is the sum of all the entries in the corresponding row of D^{-1}. Because $D^{-1} \geq 0$, then each component of $D^{-1}a$ is actually the sum of the absolute values of all the entries in that row of D^{-1}. Thus, the component with the largest value in $D^{-1}a$ is, in fact, the "largest absolute row sum" of D^{-1} which is its infinity norm. In other words, if b is a vector such that $Db = a$, then it is clear that

$$\|D^{-1}\|_\infty = \|D^{-1}a\|_\infty = \|b\|_\infty. \tag{68}$$

Finding this b is easy, by performing one LU-factorization of D, followed by a backward and a forward solve starting with a.

References

[1] W. H. Lee, S. Pant, and D. Blaauw, "Analysis and reduction of on-chip inductance effects in power supply grids," in *Proc. IEEE Int. Symp. Quality Electron. Design*, Mar. 2004, pp. 131–136.

[2] A. Muramatsu, M. Hashimoto, and H. Onodera, "Effects of on-chip inductance on power distribution grid," in *Proc. ACM Int. Symp. Phys. Design*, Apr. 2005, pp. 63–69.

[3] N. Srivastava, X. Qi, and K. Banerjee, "Impact of on-chip inductance on power distribution network design for nanometer scale integrated circuits," in *Proc. IEEE Int. Symp. Quality Electron. Design*, Mar. 2005, pp. 341–351.

[4] H. Kriplani, F. N. Najm, and I. N. Hajj, "Pattern independent maximum current estimation in power and ground buses of CMOS VLSI circuits: Algorithms, signal correlation, and their resolution," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 14, no. 8, pp. 998–1012, Aug. 1995.

[5] S. Pant, D. Blaauw, S. Sundareswaran, and R. Panda, "A stochastic approach to power grid analysis," in *Proc. ACM/IEEE DAC*, Jun. 2004, pp. 171–176.

[6] D. Kouroussis and F. N. Najm, "Static pattern-independent technique for power grid voltage integrity verification," in *Proc. IEEE/ACM Design Autom. Conf.*, Jun. 2003, pp. 99–104.

[7] N. H. A. Ghani and F. N. Najm, "Handling inductance in early power grid verification," in *Proc. IEEE/ACM ICCAD*, Nov. 2006, pp. 127–134.

[8] T. R. Bashkow, "The A matrix, new network description," *IRE Trans. Circuit Theory*, vol. 4, no. 3, pp. 117–119, Sep. 1957.

[9] L. T. Pillage, R. A. Rohrer, and C. Visweswariah, *Electronic Circuit and System Simulation Methods*. New York: McGraw-Hill, 1995.

[10] Y. Saad, *Iterative Methods for Sparse Linear Systems*. Philadelphia, PA: SIAM, 2003.

[11] T. Chen and C. C. Chen, "Efficient large-scale power grid analysis based on preconditioned Krylov-subspace iterative methods," in *Proc. IEEE/ACM DAC*, Jun. 2001, pp. 559–562.

[12] M. Nizam, F. N. Najm, and A. Devgan, "Power grid voltage integrity verification," in *Proc. IEEE/ACM Int. Symp. Low Power Electron. Design*, Aug. 2005, pp. 239–244.

[13] S. Demko, W. F. Moss, and P. W. Smith, "Decay rates for inverses of band matrices," *Math. Comput.*, vol. 43, no. 168, pp. 491–499, Oct. 1984.

[14] M. J. Grote and T. Huckle, "Parallel preconditioning with sparse approximate inverse," *SIAM J. Sci. Comput.*, vol. 18, no. 3, pp. 838–853, May 1997.

[15] J. D. Lambert, *Numerical Methods for Ordinary Differential Systems: The Initial Value Problem*. Chichester, U.K.: Wiley, 1991.

[16] S. Lang, *Algebra*. Menlo Park, CA: Addision-Wesley, 1993.

[17] F. R. Gantmacher, *The Theory of Matrices*. New York: Chelsea, 1960.

[18] R. L. Burden and J. D. Faires, *Numerical Analysis*. Pacific Grove, CA: Brooks/Cole, 2001.

[19] Mosek [Online]. Available: http://www.mosek.com

[20] J. N. Kozhaya, S. R. Nassif, and F. N. Najm, "A multigrid-like technique for power grid analysis," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 21, no. 10, pp. 1148–1160, Oct. 2002.

[21] N. H. A. Ghani and F. N. Najm, "Fast vectorless power grid verification using an approximate inverse technique," in *Proc. IEEE/ACM DAC*, Jul. 2009, pp. 184–189.

Nahi H. Abdul Ghani (S'04) received the B.E. degree in computer and communications engineering (with high distinction) from the American University of Beirut, Beirut, Lebanon, in 2005, and the M.A.Sc. degree in electrical and computer engineering from the University of Toronto, Toronto, ON, Canada, in 2007, where he is currently working toward the Ph.D. degree from the Department of Electrical and Computer Engineering.

He interned with Altera Corporation, Toronto, in 2010. His current research interests include computer-aided design for integrated circuits with a focus on the verification and analysis of power grids.

Farid N. Najm (S'85–M'89–SM'96–F'03) received the B.E. degree in electrical engineering from the American University of Beirut, Beirut, Lebanon, in 1983, and the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign (UIUC), Urbana, in 1989.

He is currently a Professor with the Department of Electrical and Computer Engineering (ECE), University of Toronto, Toronto, ON, Canada. From 1989 to 1992, he was with Texas Instruments, Dallas. He then joined the ECE Department, UIUC, as an Assistant Professor and became an Associate Professor in 1997. In 1999, he joined the ECE Department, University of Toronto, where he is currently a Professor and Chair. In 2010, he authored the book, *Circuit Simulation* (New York: Wiley). His current research interests include computer-aided design for very large scale integration, with an emphasis on circuit level issues related to power, timing, variability, and reliability.

Dr. Najm is a Fellow of the Canadian Academy of Engineering. He is an Associate Editor for the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS. He received the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN Best Paper Award, the NSF Research Initiation Award, the NSF CAREER Award, and was an Associate Editor for the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION SYSTEMS from 1997 to 2002. He serves on the executive committee of the International Symposium on Low-Power Electronics and Design (ISLPED), and has served on the technical committees of various conferences, including ICCAD, DAC, CICC, ISQED, and ISLPED.