Fast and Stable Circuit Simulation via Interpolation-Supported Numerical Inversion of the Laplace Transform

Emad Gad, *Senior Member, IEEE*, Ye Tao, *Member, IEEE*, and Michel Nakhla, *Life Fellow, IEEE*

***Abstract*— The modified numerical inversion of the Laplace transform (dubbed NILT) has been recently proposed as a fast and provably stable numerical simulation for general circuits. Although it has enabled increasing the simulation time step, it highlighted the need for robust approach that can recover the full waveform in between the time points generated by NILT. This article presents a new approach that addresses this challenge. The proposed approach leverages the NILT framework from a high-order approximation paradigm that computes points on the circuit waveforms to a methodology that computes high-order derivatives of the waveforms at the same points. Using those derivatives, an interpolation approach based on Hermite interpolation is used to construct the circuit waveforms on dense points. Numerical experiments are presented to demonstrate the accuracy of both approaches.**

***Index Terms*— Circuit simulation, differential equations, high-order derivatives, high-order numerical solution of differential equations, interpolation, numerical stability.**

I. Introduction

THE current drive in integrated circuit technology for smaller structure, higher density, and higher operating frequencies has made the modeling and simulation approach a critical stage in the design process. The complexity of the circuits used to model the electronic packages and electrical interconnects structure has increased steadily and has become a significant challenge to circuit simulation. The circuit models are typically developed from a variety of approaches. For example, the partial equivalent element circuit (PEEC) [1], [2] is one such approach where 3-D full-wave electromagnetic (EM) wave propagation in arbitrarily shaped conductors and media is captured using lumped circuit components.

Time-domain simulation is a task commonly invoked in the design cycle of those circuits. This task is typically carried out using differential equations solvers, which approximates the solution of the differential equations that model the circuits, numerically, at a set of discrete-time points [3], [4]. Traditional solvers employed in time-domain circuit simulation belong to a class of methods known as the linear multistep methods (LMS), with Gear's method, trapezoidal rule (TR), and backward Euler (BE) being among the most commonly used in the context of circuit simulations [5], mainly due to the ease with which they can handle the differential equations that model large integrated circuits. Notwithstanding the popularity of these methods, it has been a well-established fact that the LMS family inherently suffers from a conflict between the approximation order and the numerical stability [6]. This conflict essentially forces the simulator to use the low-order, and less efficient, version of LMS method, in order to preserve the numerical stability in the time marching process [7].

One approach that was shown to be free from this conflict is based on the idea of numerical inversion of Laplace transform (NILT) [8, Ch. 10]. NILT is essentially a time-domain approach that is grounded in the frequency domain. As a result of this feature, it offers the rare advantage in being more suitable to handle those circuits whose physics are captured more easily in the frequency domain than in the time domain. For example, NILT is more natural in dealing with multiconductor transmissions lines since they are concisely captured in the frequency domain using the exponential matrix formulation [9], [10]. In addition to this feature, the efficiency of NILT compared to conventional transient time-domain simulators stems from several factors. First, it enables approximating the circuit waveforms with high-degree polynomials, a fact that allows the step taken in marching through time to be larger than the step afforded by conventional simulators, which relies on low-order approximating polynomials to maintain the numerical stability. NILT, on the other hand, is structurally stable regardless of the degree of the underlying approximating polynomial. The stability in this context, mathematically recognized as the L-stability, is another factor that adds to the NILT robustness. L-stability is typically stronger than A-stability [11] and is highly desirable in circuit simulations because it guarantees numerical stability for any type of circuits, including stiff circuits and circuits modeled by differential-algebraic equations [12].

Although NILT was first introduced several decades ago, the recent developments in model-order reduction techniques [13], [14] and the complex full-wave circuit modeling approaches developed using PEEC [2], [15] have shed new

Manuscript received July 7, 2021; revised September 17, 2021 and October 19, 2021; accepted October 21, 2021. Date of publication October 26, 2021; date of current version January 17, 2022. This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada. Recommended for publication by Associate Editor S. Grivet-Talocia upon evaluation of reviewers' comments. *(Corresponding author: Emad Gad.)*

Emad Gad is with the School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON K1N 6N5, Canada (e-mail: egad@eecs.uottawa.ca).

Ye Tao and Michel Nakhla are with the Department of Electronics, Carleton University, Ottawa, ON K1N 6N5, Canada.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCPMT.2021.3122840.

Digital Object Identifier 10.1109/TCPMT.2021.3122840

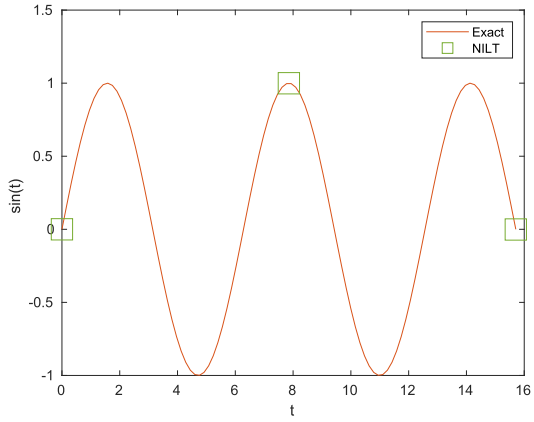

Fig. 1. Illustrating the time stepping using NILT.

lights on its efficiency, usefulness, and renewed interest in employing it as an engine in general circuit simulation. In fact, for PEEC circuits, the NILT approach proved to be more reliable and numerically stable compared to the time-stepping approach, which is frequently affected by instability [16].

In a more recent development [17], a new approach showed how NILT can be reformulated to improve accuracy and, as a result, boost its efficiency. The new approach [17] showed that for roughly the same computational cost of the conventional NILT used with order $N + M$, the local truncation error can be reduced by a factor of $(n + 1)^{N+M}$, where $n \geq 1$. The new approach was therefore dubbed as NILTn to distinguish it from the conventional NILT, which was referred to as NILT0. This feature in NILTn, which has been the key efficiency-boosting factor, coexists with a provably L-stable behavior [11], thereby satisfying both high accuracy and unconditional stability, two features rarely reconciled in traditional circuit simulators.

Nonetheless, the efficiency of the NILT approach, whether it is the conventional NILT0 or the improved NILTn, also highlighted several challenges not seen before with the conventional time-domain marching schemes used in the SPICE-like simulators. More particularly, NILT, by virtue of being a high-order differential equations solver, enables marching in time through stepping on sparsely spaced time points computed on the circuit waveforms without compromising the accuracy or stability. To illustrate this capability more clearly, Fig. 1 shows the time step that NILT can use in stepping through a sinusoidal waveform without losing accuracy. From a computational viewpoint, this large step size is computationally advantageous since it reduces the cost of matrix factorizations typically involved in the time marching process.

Nonetheless, from a user's perspective, the user or designer is oftentimes interested, not only in the value of the waveform at particular points in time but also more so in the temporal shape of the underlying waveform in between those points. This reasonable demand constitutes a challenge for NILT to provide a detailed picture of the waveform supported on a denser set of time points and without losing the NILT efficiency. The goal of the approach proposed in this article is to address this challenge. For example, in the particular instance used in Fig. 1, it will be shown that the proposed approach recovers the entire waveform using only the three points generated by NILT.

The core of the contribution in this work develops an efficient approach to reuse the results obtained from NILT, typically computed at such sparsely separated points in time, to construct an interpolating polynomial that is used to provide a fully detailed and accurate representation of the waveform between those NILT time points. It should be noted here that the objective of using the solution computed at sparsely spaced points to recover a solution on more densely spaced points has been discussed in the literature, e.g., [18]–[20]. In contrast to these approaches, however, the proposed approach leverages the NILT methodology to compute the high-order derivatives and use those derivatives to construct the desired interpolant. To the best of the authors' knowledge, the proposed approach is unique in its interpolation methodology and shows very accurate interpolation results.

The rest of this article is organized as follows. Section II presents a well-rounded review of the background and notations needed to present the proposed approaches. Section III describes the proposed interpolation approach. Section IV provides the numerical experiments that validate the accuracy of the proposed approach.

II. NILT0 and NILTn: Background, Notations, and Problem Formulation

A. Notations and Mathematical Background

In what follows, $\mathbb{N}$, $\mathbb{R}$, and $\mathbb{C}$ define the sets of nonnegative integer numbers and real and complex numbers, respectively. $n \in \mathbb{N}$ will stand for nonnegative integer, $n = 0, 1, 2, \ldots,$ and p is defined throughout this article as $p = n + 1$.[1] A multi-index in the context of this article defines a vector in $\mathbb{N}^M$, i.e., a vector whose M components are nonnegative integers. A boldfont lowercase letter, e.g., $\mathbf{l}$, will be used to denote a given multi-index, while its individual components will be denoted by a subscripted italic (nonbold) lowercase of the same letter, i.e., $l_i, i = 1, 2, \ldots, M$, where $l_i \in \mathbb{N}$. A set, Λ_p^M, is a set of multi-indices $\mathbf{l} \in \mathbb{N}^M$ restricted by the definition $\Lambda_p^M := \{\mathbf{l}\colon \sum_{i=1}^M l_i = p\}$. The factorial of a multi-index will be defined as the product of the factorials of its components, $\mathbf{l}! := \prod_{i=1}^M l_i!$. A multi-index, $\mathbf{l}$, when used as the power of another complex vector $\mathbf{d} \in \mathbb{C}^M$ defines the operation $\mathbf{d}^{\mathbf{l}} := \prod_{i=1}^M d_i^{l_i}$. The unity complex number $(-1)^{1/2}$ is denoted as $\jmath$.

The developments in this work employ the notation $\mathbf{x}(t_0)^{(i)}$ to denote the ith-order derivative of $\mathbf{x}(t)$ with respect to t computed at $t = t_0$. $\mathcal{O}(h^m)$ will be utilized in referring to series with terms proportional to h^q with $q \geq m$.

The NILT approach is built around the idea of Padé rational approximation $[N/M]$ [21] of the complex exponential function e^z, $z \in \mathbb{C}$, $\xi_{N,M}(z)$, which is given by

$$\xi_{N,M}(z) = \frac{\sum_{i=0}^{N} \alpha_{i,M,N} z^i}{\sum_{i=0}^{M} (-1)^i \alpha_{i,N,M} z^i} \tag{1}$$

[1] p and $n + 1$ will be interchangeably used throughout this article to simplify some mathematical expressions.

where

$$\alpha_{i,N,M} = \frac{(M+N-i)!}{(M+N)!}\binom{M}{i}. \tag{2}$$

It is well known that $e^z - \xi_{N,M}(z) = \mathcal{O}(z^{N+M+1})$ [12], [21], [22]. Of particular importance to NILT are the residues and poles of the rational function $\xi_{N,M}(z)$ denoted with the duo (k_i, z_i). For even values of M, (k_i, z_i) appear in complex-conjugate pairs, while for odd values of M, there will be only one real pair that will be denoted by (k_0, z_0), with the rest $M-1$ pairs appearing in complex-conjugate pairs.

B. Circuit Formulation

A general linear circuit is represented in the time or Laplace domain using the MNA formulation [23]

$$\mathbf{G}\mathbf{x}(t) + \mathbf{C}\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = \mathbf{b}(t) \tag{3}$$

$$(\mathbf{G} + s\mathbf{C})\mathbf{X}(s) = \mathbf{B}(s) + \mathbf{C}\mathbf{x}(0) \tag{4}$$

where $\mathbf{G}, \mathbf{C} \in \mathbb{R}^{K\times K}$ are matrices representing the memoryless and memory elements in the circuit, $\mathbf{b}(t) \in \mathbb{R}^K$ ($\mathbf{B}(s) \in \mathbb{C}^K$) is a vector that captures the independent sources of the circuits in the time (Laplace) domain, and $\mathbf{x}(t)$ and $\mathbf{X}(s)$ are vectors of the (unknown) circuit voltage waveforms, appended by waveforms for currents in inductors and voltage sources, in the time and Laplace domains, respectively.

The time-domain $\mathbf{x}(t)$ can be recovered from the Laplace-domain $\mathbf{X}(s)$ via the inverse Laplace transformation, which is obtained from the infinite complex integral

$$\mathbf{x}(t) = \mathscr{L}^{-1}\{\mathbf{X}(s)\} = \frac{1}{2\pi\jmath}\int_{c-\jmath\infty}^{c+\jmath\infty}\mathbf{X}(s)e^{st}\mathrm{d}s \tag{5}$$

where c is a constant chosen such that all singularities in $\mathbf{X}(s)$ lie on the left of the vertical line $\Re(s) = c$, with $\Re(\cdot)$ denoting the real part of a complex number.

The key idea in the NILT0 or NILTn approaches is to obtain the time-domain response of the circuit $\mathbf{x}(t)$ through replacing the integral in (5) with a finite sum of the integrand evaluated at prespecified values of s. This is possible if e^{st} is first replaced by the $[N/M]$ rational approximation $\xi_{N,M}(st)$, for $N = M-2$. A summary of the fundamental steps involved in both approaches leading up to approximation of $\mathbf{x}(t)$ is presented in Sections II-C and II-D. A deeper review of the theoretical grounding for both approaches can be found in [17] and [24, ch. 10].

C. NILT0: The Computational Steps

The basic idea in the NILT0 aims at approximating the circuit time-domain waveforms $\mathbf{x}(t)$ from its Laplace-domain representation $\mathbf{X}(s)$ in (3) and (4). The NILT0 approximations to $\mathbf{x}(t)$, which will be denoted as $\tilde{\mathbf{x}}^{[0]}(t)$, are obtained through applying the Cauchy theorem of residues on the line integral in (5) [25] to obtain the following expressions:

$$\tilde{\mathbf{x}}^{[0]}(t) = -\frac{1}{t}\sum_{i=1}^{M/2} 2\Re\left\{k_i\mathbf{X}\Big(\frac{z_i}{t}\Big)\right\} \tag{6}$$

for even M or

$$\tilde{\mathbf{x}}^{[0]}(t) = -\frac{1}{t}\sum_{i=1}^{\frac{M-1}{2}} 2\Re\left\{k_i\mathbf{X}\Big(\frac{z_i}{t}\Big)\right\} - \frac{1}{t}k_0\mathbf{X}\Big(\frac{z_0}{t}\Big) \tag{7}$$

for odd M, with k_i and z_i, $i = 1, \ldots, M$ being, respectively, the residues and poles of $\xi_{N,M}(z)$, introduced above, and $\mathbf{X}(z_i/t)$ is the solution of (4) at $s = (z_i/t)$.

In what follows, M will be taken as even assuming that odd values can be treated in a similar manner.

NILT0 can also be implemented as a time marching scheme, in the sense that the result obtained at, say, $t = (q-1)h$, $\tilde{\mathbf{x}}^{[0]}((q-1)h)$, is used to move a step forward (march) to $t = qh$, where h is an appropriate step size and $q = 1, 2, \ldots$. This is done by shifting the time origin to $(q-1)h$, leading to

$$\tilde{\mathbf{x}}^{[0]}(qh) = -\frac{1}{h}\sum_{i=1}^{M/2} 2\Re\left\{k_i\widehat{\mathbf{X}}\Big(\frac{z_i}{h}\Big)\right\} \tag{8}$$

where $\widehat{\mathbf{X}}(z_i/h)$ solves the system

$$(\mathbf{G} + s\mathbf{C})\widehat{\mathbf{X}}(s) = \mathbf{B}(s) + \mathbf{C}\tilde{\mathbf{x}}^{[0]}((q-1)h) \tag{9}$$

for $s = (z_i/h)$, $i = 1, \ldots, M/2$. In that sense, $\tilde{\mathbf{x}}^{[0]}((q-1)h)$ is used in (9) to obtain $\widehat{\mathbf{X}}(s)$, which in turn is used in (8) to carry the one-step move forward in time.

Employing the idea of marching through time, as described by (8) and (9), enables NILT0 to be used for a longer period of time without losing accuracy.

D. NILTn: The Computational Steps

NILTn was introduced in [17] providing a new methodology that boosted the accuracy and, as a consequence, the efficiency of the conventional NILT (NILT0). In the NILTn approach for taking steps in time $t = h, 2h, \ldots$, the approximations to $\mathbf{x}(t)$ at $t = qh$, denoted by $\tilde{\mathbf{x}}^{[n]}(qh)$, are computed using (10), as shown at the bottom of the next page, where $p = n+1$ (n being an integer satisfying $n \geq 1$), $\mathbf{k} = [k_1, k_2, \ldots, k_M]^\top$ and

$$\widehat{\mathbf{X}}\Big(\frac{pz_i}{t}\Big)^{(\mu)} \equiv \left.\frac{\mathrm{d}^\mu\widehat{\mathbf{X}}(s)}{\mathrm{d}s^\mu}\right|_{s=\frac{pz_i}{t}} \tag{11}$$

being the μth-order derivative, with respect to s, of $\widehat{\mathbf{X}}(s)$ computed by recursively solving the systems

$$(\mathbf{G} + s\mathbf{C})\widehat{\mathbf{X}}(s)^{(0)} = \mathbf{B}(s) + \mathbf{C}\tilde{\mathbf{x}}((q-1)h) \tag{12a}$$

$$(\mathbf{G} + s\mathbf{C})\widehat{\mathbf{X}}(s)^{(\mu)} = \frac{\mathrm{d}^\mu}{\mathrm{d}s^\mu}\mathbf{B}(s) - \mu\mathbf{C}\widehat{\mathbf{X}}(s)^{(\mu-1)},\ \mu > 0 \tag{12b}$$

for $s = (pz_i/h)$.

It should be straightforward to verify that setting $n = 0$ in (10) recovers $\tilde{\mathbf{x}}^{[0]}(qh)$ of (8).

III. Generating Dense Points via Interpolation

The NILT approach, when employed as a scheme for marching through time, is essentially akin to the IVP-based numerical solvers of the MNA-formulated circuit differential equations that are commonly deployed in SPICE-like circuit simulators. However, NILT differs from the traditional

numerical solvers adopted in those simulators in key features. Foremost among those features is the fact that the order of $N + M$ approximation is typically higher than that of the traditional solvers within the same class of stability. For example, the A-stable TR method used in the SPICE do have "order 2," which means that the polynomial used to approximate the waveform cannot have a degree higher than 2 if A-stability is to be maintained. By contrast, NILT is A-stable of order $N + M$ (arbitrarily high) so long as $N = M - 2$. The high-order approximation enables increasing the step size without incurring significant approximation errors.

Nevertheless, the high order of NILT poses a new challenge not seen with the traditional solvers: the need to have a detailed picture for the circuit temporal waveform in between the points generated by the NILT on that waveform. The need here arises from the fact that points computed by NILT are widely separated in time as a result of being powered by the high-order capability of the NILT engine, leaving out, from the user's viewpoint, what could be behavioral important details such as an overshoot or a peak. This need instigates the following question: can the points computed by NILT, sparsely distributed as they are, be used to reconstruct the circuit waveform on more densely distributed points without requiring significant additional computational cost?

The above question is addressed by the contribution in this work. More particularly, a new approach that leverages the NILT framework from a high-order approximation paradigm that computes points on the circuit waveforms to a methodology that computes high-order derivatives of the waveforms at the same points is developed in this article. The new NILT-based computations of high-order derivatives is a scalar operation that can be performed on a single, or a selected few, waveforms of the circuit, and therefore, its computation does not scale in proportion to the circuit size K. The high-order derivatives, computed through closed-form formulas derived for that purpose, are then used to construct the interpolants through a Hermite interpolation approach.

A. NILT-Based Computation of High-Order Derivatives

The proposed interpolation approach relies on three main ingredients.

1) A set of discrete-time points $t_1, t_2, \ldots$ at which the values of the waveform to be interpolated, say scalar waveform $x_k(t)$, are computable. Although the interpolation approach is general enough to handle nonuniformly separated time points, we will assume, for notational simplicity, that those points are equidistant at $t = qh$, with $q = 1, 2, \ldots$.
2) The values of the waveforms at $t = qh$. Those values are made available through the NILTn and are given by $\tilde{\mathbf{x}}^{[n]}(qh)$, obtained from (10) [or (8) for NILT0].
3) The high-order derivatives of $x_k(t)$ at $t = qh$.

Unfortunately, numerical approximation of high-order derivatives is inherently an ill-conditioned process. The proposed approach provides an alternative approach that does not rely on numerical approximations but, instead, leverages the NILT methodology to compute the high-order derivatives. The remainder of this section illustrates this approach.

To avoid making the growth in the mathematical complexity hide the fundamental ideas, NILT0 will be chosen first, due to its relative simplicity, as a demonstration vehicle for this approach. In addition, a slight expansion of notation is needed to facilitate the new developments, where $\tilde{\mathbf{x}}^{[n]}(qh)^{(\beta)}$ will be used to mark the βth-order derivative of $\mathbf{x}(t)$ evaluated at $t = qh$ and approximated using the NILTn approach, with $n = 0$ implying NILT0 and $n > 0$ referring to NILTn. Although the developed operations can be performed componentwise, we will pursue a vector-based description to avoid the notational complexity needed in that regard.

B. Computing $\tilde{\mathbf{x}}^{[0]}(qh)^{(\beta)}$ With the NILT0

The starting point in this process is the differentiation identity of the Laplace tranformation, that is,

$$\begin{aligned}\frac{\mathrm{d}^\beta \mathbf{x}(qh)}{\mathrm{d}t^\beta} &= \mathscr{L}^{-1}\left\{ s^\beta \mathbf{X}(s) - \sum_{\vartheta=1}^{\beta} \mathbf{x}(0)^{(\vartheta-1)} s^{\beta-\vartheta} \right\}\Big|_{t=qh} \\ &= \mathscr{L}^{-1}\left\{ s^\beta \mathbf{X}(s) - \sum_{\vartheta=1}^{\beta-1} \mathbf{x}(0)^{(\vartheta-1)} s^{\beta-\vartheta} \right\}\Big|_{t=qh} \\ &\quad - \underbrace{\mathscr{L}^{-1}\left\{ \mathbf{x}(0)^{(\beta-1)} \right\}\Big|_{t=qh}}_{\mathbf{x}(0)^{\beta-1}\delta(t)\big|_{t=qh}=0} \end{aligned} \tag{13}$$

where $\mathbf{x}(0)^{(\vartheta)}$, as stated in the notations, denote the ϑth-order derivative of $\mathbf{x}(t)$ at the initial point $t = 0$.

The underlying concept of the NILT0 methodology can be invoked here to replace the integral operation, which is required for the inverse Laplace transformation of (13), with a summation over the residues of the integrand [24]. The process followed here is similar to the process used to obtain (6) from (5). In the current context, it will provide, at its termination, the NILT0 version of the βth-order derivative of

$$\tilde{\mathbf{x}}^{[n]}(qh) = -\frac{pp!}{t} \sum_{\mathbf{l}\in\Lambda_p^M} \sum_{\substack{i=1 \\ l_i\neq 0}}^{M} \frac{\mathbf{k}^{\mathbf{l}}}{\mathbf{l}!} \left[\frac{1}{(l_i-1)!} \sum_{\mu=0}^{l_i-1} \binom{l_i-1}{\mu} \left(\frac{p}{t}\right)^{\mu} \widehat{\mathbf{x}}\left(\frac{pz_i}{t}\right)^{(\mu)} \cdot \left(\sum_{\boldsymbol{\nu}\in\Lambda_{l_i-1-\mu}^{M-1}} \frac{(l_i-1-\mu)!}{\boldsymbol{\nu}!} \prod_{\substack{j=1 \\ j\neq i}}^{M} \frac{(-1)^{\nu_j} \prod_{\theta=0}^{\nu_j-1} (l_j+\theta)}{(z_i-z_j)^{l_j+\nu_j}} \right) \right] \tag{10}$$

$\mathbf{x}(t)$, i.e., or $\tilde{\mathbf{x}}^{[0]}(t)^{(\beta)}$, as follows:

$$\tilde{\mathbf{x}}^{[0]}(t)^{(\beta)} = \frac{-1}{t}\sum_{i=1}^{M/2} 2\Re\left\{ z_i^{\beta} k_i \mathbf{X}\Big(\frac{z_i}{h}\Big)\right\} + \frac{1}{t}\sum_{i=1}^{M/2}\sum_{\vartheta=1}^{\beta-1} 2\Re\left\{ z_i^{\beta-\vartheta}\, \mathbf{x}(0)^{(\vartheta)}\right\}. \quad (14)$$

In an analogous manner to using the notion of marching through time [given by (8) and (9)] that was used to compute $\tilde{\mathbf{x}}^{[0]}(qh)$ from $\tilde{\mathbf{x}}^{[0]}((q-1)h)$, the derivatives, $\tilde{\mathbf{x}}^{[0]}(qh)^{(\beta)}$, $q = 1, 2, \ldots$, can also be obtained from $\tilde{\mathbf{x}}^{[0]}((q-1)h)^{(\beta)}$ in the following manner:

$$\tilde{\mathbf{x}}^{[0]}(qh)^{(\beta)} = \frac{-1}{h}\sum_{i=1}^{M/2} 2\Re\left\{ z_i^{\beta} k_i \widehat{\mathbf{X}}\Big(\frac{z_i}{h}\Big) - \sum_{\vartheta=1}^{\beta-1} z_i^{\beta-\vartheta}\tilde{\mathbf{x}}^{[0]}((q-1)h)^{(\vartheta)}\right\}. \quad (15)$$

One needs to stress here that (15) provides a closed-form expression for computing the βth-order derivative of $\mathbf{x}(t)$ based on its Laplace-domain version $\mathbf{X}(s)$ and the past time derivatives up to order $\beta - 1$. It is also obvious that this computation can be performed componentwise on a single or, if desired, few selected components of the vector $\mathbf{x}(t)$. More importantly, to note here is the fact that this computation does not require additional computations since $\widehat{\mathbf{X}}(z_i/h)$ is readily available upon computing the NILT0 of $\tilde{\mathbf{x}}^{[0]}(qh)$, which corresponds to $\beta = 0$.

C. Computing $\tilde{\mathbf{x}}^{[n]}(qh)^{(\beta)}$ With the NILTn

The generalization to NILTn is considered here to compute $\tilde{\mathbf{x}}^{[n]}(qh)^{(\beta)}$. Set $p = n + 1$ and define $\mathbf{w}(t)$ as

$$\mathbf{w}(t) := \mathbf{x}\Big(\frac{t}{p}\Big) \quad (16)$$

so that

$$\mathbf{x}(t) = \mathbf{w}(pt) \quad (17)$$

$$\mathbf{x}(t)^{(\beta)} = p^{\beta}\,\mathbf{w}(pt)^{(\beta)}. \quad (18)$$

Using the scaling property of the Laplace transform, $\mathbf{w}(t)$ can be expressed as

$$\mathbf{w}(t) = \mathscr{L}^{-1}\{p\mathbf{X}(ps)\}. \quad (19)$$

The Laplace transform identity used to write (13) can be invoked again to express the βth-order derivative of $\mathbf{w}(t)$ with respect to t, or $\mathbf{w}(t)^{(\beta)}$, as

$$\mathbf{w}(t)^{(\beta)} = \mathscr{L}^{-1}\Big\{ p\, s^{\beta}\mathbf{X}(ps)\Big\} - \mathscr{L}^{-1}\Big\{\sum_{\vartheta=1}^{\beta-1}\mathbf{w}(0)^{(\vartheta-1)}s^{\beta-\vartheta}\Big\}. \quad (20)$$

Substituting from (20) into (18) and expressing the inverse of the Laplace transform in its known integral form results in showing that $\mathbf{x}(t)^{(\beta)}$ can be represented as the sum of two separate waveforms, $\mathbf{\Theta}_1(t)^{(\beta)}$ and $\mathbf{\Theta}_2(t)^{(\beta)}$

$$\mathbf{x}(t)^{(\beta)} = \overbrace{\frac{p}{2\pi_J}\int_{c-J\infty}^{c+J\infty} s^{\beta}\mathbf{X}(ps)\,e^{\mathrm{pst}}\mathrm{d}s}^{\mathbf{\Theta}_1(t)^{(\beta)}} - \underbrace{\frac{1}{2\pi_J}\int_{c-J\infty}^{c+J\infty}\sum_{\vartheta=1}^{\beta-1}\mathbf{x}(0)^{(\vartheta-1)}s^{\beta-\vartheta}e^{\mathrm{pst}}\mathrm{d}s}_{\mathbf{\Theta}_2(t)^{(\beta)}}. \quad (21)$$

In the following, the NILTn process is invoked to express both line integrals given in (21) as a finite summations over the residues of the respective integrands around its poles in the right-half-plane. We consider the first integral first, i.e., $\int_{c-J\infty}^{c+J\infty} s^{\beta}\mathbf{X}(ps)e^{\mathrm{pst}}\mathrm{d}s$.

The first integral has a similar form[2] to the starting integral that was used in [17] to derive the NILTn time-domain response shown by (10). The only difference between the integrand of the first integral in (21) and that which was used to derive (10) in [17] is that instead of having $\mathbf{X}(ps)e^{\mathrm{pst}}$, we now have $s^{\beta}\mathbf{X}(ps)e^{\mathrm{pst}}$ as the integrand. This difference, however, does not impede following through the same derivations that were used in [17] and were conducive to the result of (10). Indeed, a closer look would show that the only modification that needs to be enacted in following those derivations in the present context will eventually result in having the $\widehat{\mathbf{X}}(\cdot)^{(\mu)}$ term of (10) replaced by the (μ)th-order derivative of the term $s^{\beta}\mathbf{X}(ps)$ [denoted as $(s^{\beta}\mathbf{X}(ps))^{(\mu)}$)], with respect to s, and evaluated at $s = (pz_i/t)$, $i = 1, \ldots, M$. The new term, $(s^{\beta}\mathbf{X}(ps))^{(\mu)}$, can be expressed using the generalized Leibniz formula [26] for the μth-order derivative as follows:

$$\big(s^{\beta}\mathbf{X}(ps)\big)^{(\mu)} = \sum_{r=0}^{\min(\beta,\mu)}\binom{\mu}{r}\frac{\beta!\,s^{\beta-r}}{(\beta-r)!}\mathbf{X}(ps)^{(\mu-r)}.$$

Utilizing the right-hand side of the above expression, for $s = (pz_i/t)$, in place of $\widehat{\mathbf{X}}(pz_i/t)^{(\mu)}$ in (10), should provide the NILTn version of the integral $\mathbf{\Theta}_1(t)^{(\beta)}$. To follow the same notational conventions adopted so far, that version is denoted as $\tilde{\mathbf{\Theta}}_1^{[n]}(t)^{(\beta)}$ and is given by (23), shown at the bottom of the next page, where

$$\mathcal{W}_{i,\mu} = \sum_{\boldsymbol{\nu}\in\Lambda_{l_i-1-\mu}^{M-1}}\frac{(l_i-1-\mu)!}{\boldsymbol{\nu}!}\prod_{\substack{j=1\\ j\neq i}}^{M}\frac{(-1)^{\nu_j}\prod_{\theta=0}^{\nu_j-1}(l_j+\theta)}{(z_i-z_j)^{l_j+\nu_j}}. \quad (22)$$

The second integral part of (21), namely, $\mathbf{\Theta}_2(t)^{(\beta)}$, can be treated with the NILTn approach in a likewise manner but with an analogous modification to arrive at a similar expression to (10). The ramification to this modification in this case will see to $\widehat{\mathbf{X}}(\cdot)^{(\mu)}$ of (10) being replaced by the μth-order derivative of the term $s^{\beta-\vartheta}$, with respect to s, and computed at $s = (pz_i/t)$. This replacement will provide the NILTn version of the second integral $\Theta_2^{[n]}(t)^{(\beta)}$ [NILTn approximation

[2]See [17, eq. (35)] and the ensuing derivations.

of $\mathbf{\Theta}_2(t)^{(\beta)}$], which is captured by (24) shown at the bottom of the page for $\beta > 1$. For $\beta \leq 1$, $\Theta_2^{[n]}(t)^{(\beta)} = 0$.

Computing $\tilde{\mathbf{x}}^{[n]}(t)^{(\beta)}$ using the NILTn framework is thus obtained by adding $\tilde{\mathbf{\Theta}}_1^{[n]}(t)^{(\beta)}$ and $\tilde{\mathbf{\Theta}}_2^{[n]}(t)^{(\beta)}$.

The time marching version of $\tilde{\mathbf{x}}^{[n]}(t)^{(\beta)}$, i.e., $\tilde{\mathbf{x}}^{[n]}(qh)^{(\beta)}$, can be obtained in the manner used for NILT0 to compute $\tilde{\mathbf{x}}^{[0]}(qh)^{(\beta)}$ from $\tilde{\mathbf{x}}^{[n]}((q-1)h)^{(\beta)}$. In this case, $\tilde{\mathbf{x}}^{[n]}(qh)^{(\beta)}$ is obtained by adding time marching versions of $\tilde{\mathbf{\Theta}}_1^{[n](\beta)}$ and $\tilde{\mathbf{\Theta}}_2^{[n](\beta)}$, that is,

$$\tilde{\mathbf{x}}^{[n]}(qh)^{(\beta)} = \tilde{\mathbf{\Theta}}_1^{[n]}(qh)^{(\beta)} + \tilde{\mathbf{\Theta}}_2^{[n]}(qh)^{(\beta)}. \tag{25}$$

Here, $\tilde{\mathbf{\Theta}}_1^{[n]}(qh)^{(\beta)}$ is obtained from (23) simply by replacing $\mathbf{X}(pz_i/t)^{(\mu-r)}$ with $\widehat{\mathbf{X}}((pz_i/h))^{(\mu-r)}$ (the one that solves (12a) and (12b)). On the other hand, $\tilde{\mathbf{\Theta}}_2^{[n]}(qh)^{(\beta)}$ will be obtained from (24) by replacing $\mathbf{x}(0)^{(\vartheta-1)}$ with $\tilde{\mathbf{x}}^{[n]}((q-1)h)^{(\vartheta-1)}$. In both cases, t will be replaced with h.

It should be mentioned here that the comments made at the end of Section III-B in regard to the efficiency of computing for NILT0-based derivatives apply equally to the NILTn-based derivatives, with the obvious addition that NILTn is more accurate than NILT0.

D. Explicit Hermite Interpolation Using Cycle Index Polynomials

Hermite interpolation is based on constructing a polynomial that satisfies a set of values along with the first few derivatives at a set of prespecified points. It is a well-known fact that there is a unique polynomial that can achieve this task. For example, there is a unique polynomial in x of degree 6 [27] that can be made equal to two prespecified values at two points, say $x = x_1$ and $x = x_2$, while satisfying prespecified derivatives of orders 1 and 2 at x_1 and orders 1, 2, and 3 at x_2.

The task of constructing the Hermite interpolating polynomial is simply the task of computing the polynomial coefficients. Computing the polynomial coefficients has been typically carried out numerically through constructing and inverting a matrix that is determined based on the displacement between the points of interpolation. However, a relatively recent technique that appeared in [27] showed that this task needs not be performed numerically. Rather, the interpolation coefficients can be analytically computed from the discrete-time points and the values of the derivatives at those points.

The previous sections have set the stage for constructing the Hermite interpolation polynomial by showing how the NILTn approach can be used to compute the values of the circuit waveform and its first few derivatives at discrete-time points. This section carries this idea further by demonstrating how those results obtained from NILT can be used in the construction of an interpolant polynomial through the approach presented in [27].

The approach presented in [27] is based on the notion of cycle index polynomials. To introduce this notion, we use the notation $y_n(x_i|i \in [n])$ to represent the multinomial $y_n(x_1, x_2, \ldots, x_n)$ and use that to describe the cycle index polynomials $z_n(x_i|i \in [n])$ through the recurrence relation

$$\begin{aligned} z_0 &= 1 \\ nz_n(x_k|k \in [n]) &= \sum_{k=1}^{n} x_k z_{n-k}(x_i|i \in [n-k]). \end{aligned} \tag{26}$$

Next, define the following time points $t_1, t_2, \ldots, t_n$ as being different ($t_m \neq t_n$, if $m \neq n$) and not necessarily equidistant. Furthermore, assume that a given function $\theta(t)$ is known through its derivatives: $\theta(t_\alpha)^{(\beta)}$ at those points, where $\alpha = 1, 2, \ldots, H$ and $\beta = 0, \ldots, m_\alpha - 1$. The Hermite interpolation problem is finding the polynomial $\Gamma(t)$ such that

$$\Gamma(t_\alpha)^{(\beta)} = \theta(t_\alpha)^{(\beta)}, \quad \alpha = 1, \ldots, H, \quad \beta = 0, \ldots, m_\alpha - 1. \tag{27}$$

The work presented in [27] has shown that this problem can be expressed analytically in terms of $\theta(t_\alpha)^{(\beta)}$ using the cycle index polynomials as illustrated by (28), shown at the bottom of the next page.

It is to be noted here that the interpolant uses the coefficients of the cycle index polynomial along with the time points and the values/derivatives of the waveform at those points to construct the desired interpolant.

Fig. 2 revisits the issue illustrated in Fig. 1 and shows the results of applying the interpolant formula (28) to generate dense points for the values/derivatives obtained from NILTn for the sinusoidal waveform, highlighting the points obtained by NILT and the points generated through the interpolant formula.

E. Observations and Considerations for Implementations

This section highlights some of the details pertinent to the implementation of the proposed interpolation approach. One observation that needs to be made here is that setting $\beta = 0$

$$\tilde{\mathbf{\Theta}}_1^{[n]}(t)^{(\beta)} = -\frac{pp!}{t} \sum_{\mathbf{l} \in \Lambda_p^M} \sum_{\substack{i=1 \\ l_i \neq 0}}^{M} \frac{\mathbf{k}^{\mathbf{l}}}{\mathbf{l}!} \left[\frac{1}{(l_i-1)!} \sum_{\mu=0}^{l_i-1} \binom{l_i-1}{\mu} \left(\frac{p}{t}\right)^{\mu} \left(\sum_{r=0}^{\min(\beta,\mu)} \binom{\mu}{r} \frac{\beta!}{(\beta-r)!} \left(\frac{pz_i}{t}\right)^{\beta-r} \mathbf{X}\left(\frac{pz_i}{t}\right)^{(\mu-r)} \right) \mathcal{W}_{i,\mu} \right] \tag{23}$$

$$\tilde{\mathbf{\Theta}}_2^{[n]}(t)^{(\beta)} = -\frac{pp!}{t} \sum_{\vartheta=1}^{\beta-1} \mathbf{x}(0)^{(\vartheta-1)} \sum_{\mathbf{l} \in \Lambda_p^M} \sum_{\substack{i=1 \\ l_i \neq 0}}^{M} \frac{\mathbf{k}^{\mathbf{l}}}{\mathbf{l}!} \left[\frac{1}{(l_i-1)!} \sum_{\mu=0}^{l_i-1} \binom{l_i-1}{\mu} \left(\left(\frac{(\beta-\vartheta)!}{(\beta-\vartheta-\mu)!} \left(\frac{z_i}{t}\right)^{\beta-\vartheta-\mu} \right) \right) \mathcal{W}_{i,\mu} \right], \quad \beta > 1 \tag{24}$$

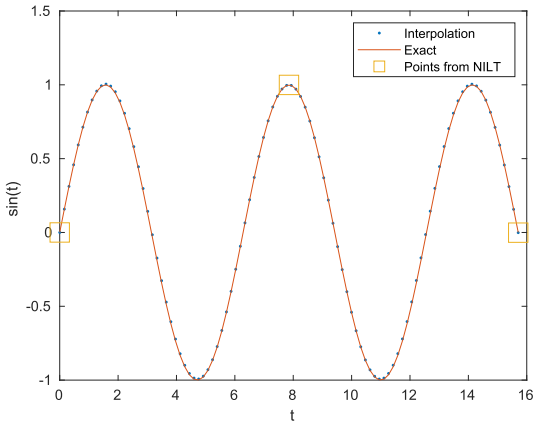

Fig. 2. Interpolation using (28) from three points computed by NILTn using the high-order derivatives at those points.

in (21), (24), and (25) recovers the formula of (10), indicating that (21), (24), and (25) are the generalization of NILTn from the 0th-order derivative (simple plain value) of $\mathbf{x}(t)$ to the higher order derivatives.

1) Computing Initial Derivatives at $t = 0$: It is to be noted from (15) that in the first step, where $q = 1$, computing the derivatives, $\tilde{\mathbf{x}}^{[0]}(h)^{(\beta)}$, requires the initial $t = 0$ derivatives (beginning with order 0 and up to order $\beta - 1$) to kickstart the process of marching the derivatives through time. Those derivatives are supplied by a one-time process launched at the beginning of the march. In the subsequent steps, where $q > 1$, $\tilde{\mathbf{x}}^{[0]}(h)^{(\beta)}$ is used to compute $\tilde{\mathbf{x}}^{[0]}(2h)^{(\beta)}$ and so forth. The initial computation of the derivatives at $t = 0$ has been described in [28].

2) Computational Complexity: In determining the computational complexity of the interpolation, it is key to observe that the process of computing the high-order derivatives, and constructing interpolating polynomial, is essentially a componentwise process, in the sense that it can be performed on a single circuit waveform at a time, although the above presentation described it for all circuit variables. Given that it is typically the case that the user's intent is to inspect or visualize a single or few selected waveforms on the circuit variables, the associated computational complexity in implementing this approach does not have to scale in proportion to the circuit MNA formulation K and is therefore negligible in the large scheme of implementing NILTn. This observation is easily confirmed through seeing that (21), (24), and (25) require only those user-selected components to be taken from the vector $\hat{\mathbf{X}}(pz_i/h)$ to proceed with computing the derivatives and constructing the interpolating polynomial of (28).

3) Selection of NILTn Parameters: Selection of the points generated by NILT can be done by estimating the error resulting from given values for M and N and step length h using the truncation error estimation described in [24, Sec. 10.2] .

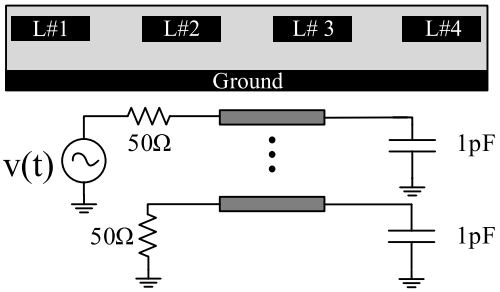

Fig. 3. Four-coupled interconnect circuit (Example 2).

It was seen through various experimental simulations that NILTn is most accurate for $n \leq 4$. For $n = 1, 2, 3$, and 4, the most accurate results are obtained for values of (M, N) that are equal to $(8, 6)$, $(6, 4)$, $(5, 3)$, and $(4, 2)$, respectively.

As to the selection of the degree of the interpolating polynomial $\Gamma(t)$, it can be seen that $\Gamma(t)$ as expressed in (28) is a polynomial in t of degree given by $\sum_{\alpha=1}^{n}(m_\alpha + 1) - 1$. It was seen experimentally that high-degree (with degree >15) polynomials suffer from numerical accuracy problems. Numerical experimentation has shown that the best results are obtained for polynomials with degree not exceeding 12.

IV. Experimental Results

Numerical results are presented in this section to validate the proposed approaches. The main focus in those experiments is to show the accuracy of the proposed interpolation in capturing the circuit waveforms between the points computed by NILTn and the reliability of the proposed error estimation in predicting the closeness of the NILTn approximation to the exact circuit response. The CPU time efficiency of NILTn has been previously demonstrated on those examples in [17].

A. Example 1: Interpolation of NILT2

This example considered the transmission line (TL) circuit shown in Fig. 3. The TL in this circuit had four coupled conductors, with length 10 cm each, above a ground plane. The TL structure was modeled with lumped RLCK [29] components giving rise to an MNA formulation (3) with $N = 1204$ variables. The circuit is excited on one conductor (active line) using a 1-V trapezoidal pulse voltage source with 0.1-ns rise/fall times and a pulsewidth of 5 ns. The other conductors (victim lines) are all terminated with 50-Ω resistor and 1-pF capacitor at the source (near) end and load (far) end, respectively.

The NILTn response of the circuit to the piecewise linear (PWL) pulse is constructed through shifting and scaling [13] the response to a ramp source, $u(t) = t$ up to $t = 10$ ns. Fig. 4 shows the response of the circuit to the ramp source

$$\Gamma(t) = \sum_{\alpha=1}^{H}\sum_{\beta=0}^{m_\alpha-1}\frac{\theta(t_\alpha)^{(\beta)}}{\beta!}\prod_{\substack{\gamma=1\\ \gamma\neq\alpha}}^{H}\left(\frac{t-t_\gamma}{t_\alpha-t_\gamma}\right)^{m_\gamma}\sum_{k=\beta}^{m_\alpha-1}(t-t_\alpha)^k z_{k-\beta}\left(\sum_{\gamma\neq\alpha}\frac{m_\gamma}{(t_\gamma-t_\alpha)^r}\middle| r\in[k-\beta]\right) \tag{28}$$

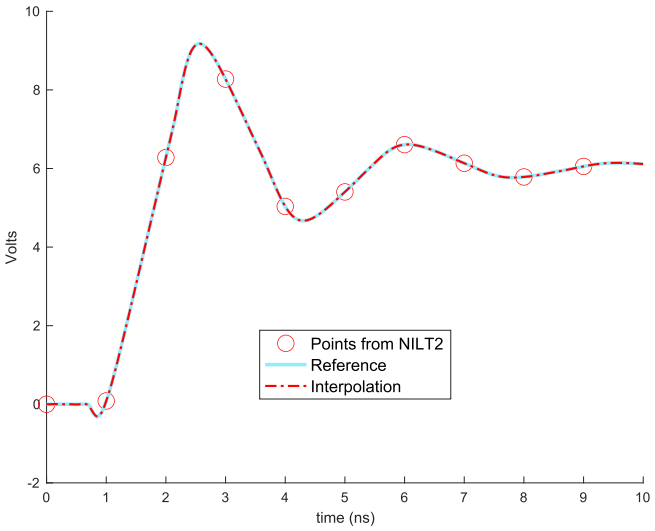

Fig. 4. Response of the circuit to a time ramp source $u(t) = t$ observed at the far end of the second victim line.

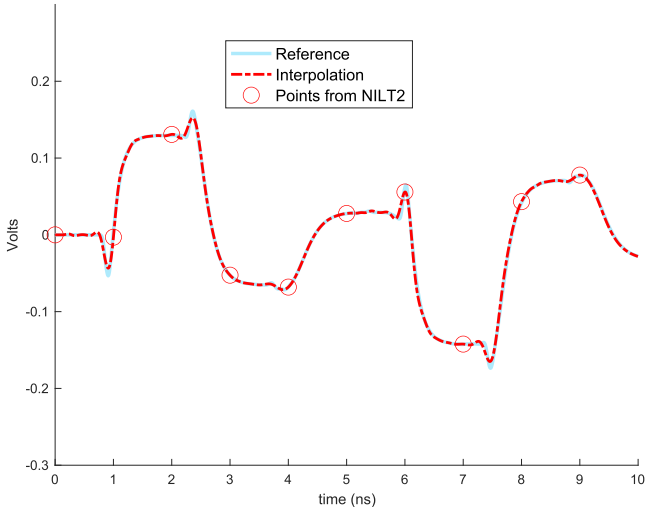

Fig. 5. Response of the circuit to the trapezoidal PWL input observed at the far end of the second victim line.

obtained from NILT2 at steps of $h = 1$ ns (circles in the figure).

Fig. 4 shows the results of the interpolation obtained through the Hermite interpolation polynomials with the coefficients constructed as described in Section III. The coefficients were obtained by using derivatives up to the fifth-order derivative at the points obtained from NILT.

Fig. 4 also uses the results obtained from NILT2 at a very small step size $h = 1$ ps as a reference against which the interpolation results are compared in order to demonstrate the accuracy of the interpolating polynomial in capturing the full details of the waveform.

The NILT2, interpolation, and reference responses (at the far end of the second victim line) of the circuit to the PWL input are shown in Fig. 5.

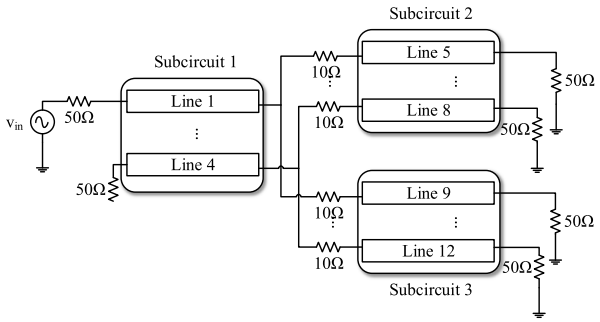

Fig. 6. Three interconnected subcircuits (Example 2).

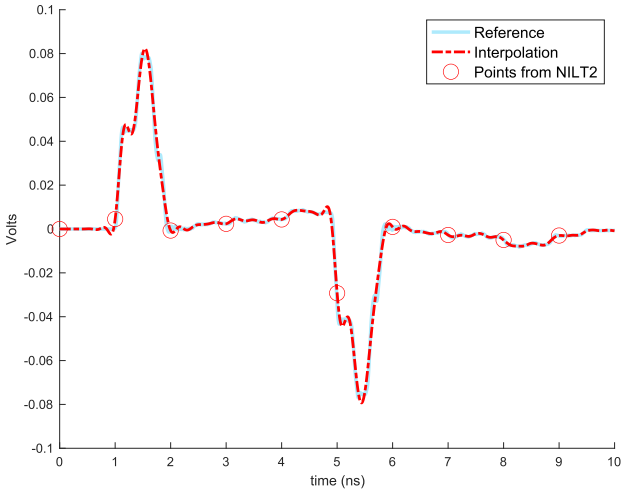

Fig. 7. Response at the far end of the second line in the second TL, as obtained from NILT2 (circles) and the interpolating polynomials compared with a reference results.

B. Example 2: Interpolation of NILT3

The circuit in this example is a network of three sets of coupled TLs connected, as shown in Fig. 6. Each TL set in this network is constructed from a set four coupled conductors with a structure similar to the one used in the previous example. Using lumped RLCK segmentation to model the TLs resulted in MNA formulation (3) with $K = 3612$ variables. The network is excited at the top conductor in the first TL set (as shown in Fig. 6) with a 1-V trapezoidal pulse voltage source with 0.1-ns rise/fall times and a pulsewidth of 5 ns.

The responses at the end nodes of the active line (top conductor) in the first TL and the second victim line in the second TL are considered to demonstrate the performance of the proposed approach. Fig. 7 shows the response at the second line of the second TL network, through the points obtained by NILT2 ($N = 3$ and $M = 5$) at steps of $h = 1$ ns and the dense interpolation constructed by the proposed approach. This figure also provides the results obtained using a much smaller step size of $h = 1$ ps to serve as a reference demonstrating the accuracy of the interpolation. Fig. 8 reproduces those results again for the response at the far end of the first line of the first TL network.

Table I shows the advantage of the proposed approach through comparing the number of LU factorizations needed

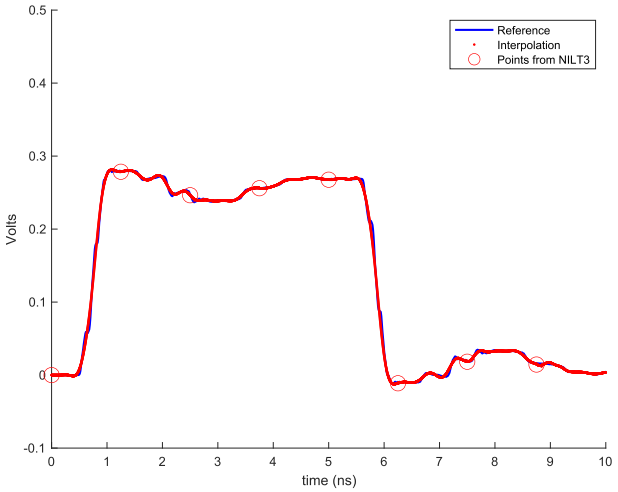

Fig. 8. Response at the top conductor of the first TL, as obtained from NILT3, interpolation, and compared with reference results obtained in using small step size 1 ps.

TABLE I
SAVINGS IN THE LU DECOMPOSITIONS AND CPU TIME THROUGH INTERPOLATION

	No interpolation		Proposed		speedup	max
	LUs	CPU (sec)	LUs	CPU (sec)		error %
Ex. 1	1551	3.9	30	0.07	56×	5.3
Ex. 2	2571	11.7	27	0.12	98×	9.8

had the NILT approach been required to produce the full details of the waveforms, versus the number of LU factorizations normally needed in using NILT to solve those same waveforms. The table also shows the measured CPU times in both approaches highlighting the speedup in using the proposed approach. The last column of the table reveals the maximum error computed by subtracting the interpolated waveforms from the reference ones at the same points, dividing by the reference results for normalization, and reporting the maximal absolute value of the result in percent. A comparison between the NILT*n* and the traditional SPICE simulation has been presented in [17].

V. CONCLUSION

This article presented a new contribution that aims at supporting the recently introduced modified numerical inversion of the Laplace transform (NILT*n*) for simulating general circuits in the time domain. The contribution described a new interpolation approach to construct a Hermite interpolation polynomial that interpolates the circuit waveforms in between the time points computed by the NILT*n*. This approach reproduces the circuit waveforms on a set of points that is much denser than that the points generated by NILT*n* with negligible computational cost. It, therefore, maintains the same efficiency of NILT*n* while enabling it to provide, to the user, a detailed description of the circuit output waveform. Numerical experimental results are presented showing the accuracy and robustness of the proposed approach.

REFERENCES

[1] A. E. Ruehli, "Equivalent circuit models for three-dimensional multiconductor systems," *IEEE Trans. Microw. Theory Techn.*, vol. MTT-22, no. 3, pp. 216–221, Mar. 1974.
[2] A. E. Ruehli, G. Antonini, and L. Jiang, *Circuit Oriented Electromagnetic Modeling Using the PEEC Techniques*. Hoboken, NJ, USA: Wiley, 2017.
[3] E. Hairer, S. P. Nøsett, and G. Wanner, *Solving Ordinary Differential Equations I Nonstiff Problems*, 3rd ed. Berlin, Germany: Springer, 2008.
[4] J. C. Butcher, *Numerical Methods for Ordinary Differential Equations*. Hoboken, NJ, USA: Wiley, 2003.
[5] F. N. Najm, *Circuit Simulation*. Hoboken, NJ, USA: Wiley, 2010.
[6] G. G. Dahlquist, "A special stability problem for linear multistep methods," *BIT Numer. Math.*, vol. 3, no. 1, pp. 27–43, Mar. 1963.
[7] K. Kundert, *The Designer's Guide to SPICE and Spectre*. Norwell, MA, USA: Kluwer, 1995.
[8] J. Vlach and K. Singhal, *Computer Methods for Circuit Analysis and Design*. New York, NY, USA: Van Nostrand Reinhold, 1983.
[9] R. Griffith and M. S. Nakhla, "Mixed frequency/time domain analysis of nonlinear circuits," *IEEE Trans. Comput.-Aided Design Integr.*, vol. 11, no. 8, pp. 1032–1043, Aug. 1992.
[10] J. R. Griffith and M. S. Nakhla, "Time-domain analysis of lossy coupled transmission lines," *IEEE Trans. Microw. Theory Techn.*, vol. 38, no. 10, pp. 1480–1487, Oct. 1990.
[11] A. Iserles and S. P. Nørsett, *Order Stars*. Englewood Cliffs, NJ, USA: Chapman & Hall, 1991.
[12] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II, Stiff and Differential-Algebraic Problems*, vol. 2. Springer, 1996. [Online]. Available: https://link.springer.com/book/10.1007/978-3-642-05221-7
[13] Y. Tao, B. Nouri, M. S. Nakhla, M. A. Farhan, and R. Achar, "Variability analysis via parameterized model order reduction and numerical inversion of Laplace transform," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 7, no. 5, pp. 678–686, May 2017.
[14] L. Lombardi, Y. Tao, B. Nouri, F. Ferranti, G. Antonini, and M. S. Nakhla, "Parameterized model order reduction of delayed PEEC circuits," *IEEE Trans. Electromagn. Compat.*, vol. 62, no. 3, pp. 859–869, Jun. 2020.
[15] H. Heeb and A. E. Ruehli, "Three-dimensional interconnect analysis using partial element equivalent circuits," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 39, no. 11, pp. 974–982, Nov. 1992.
[16] A. E. Ruehli, L. Lombardi, G. Antonini, Y. Tao, M. S. Nakhla, and F. Ferranti, "Impulse response for full wave PEEC models avoiding late time instability," in *Proc. IEEE 28th Conf. Electr. Perform. Electron. Packag. Syst. (EPEPS)*, Oct. 2019, pp. 1–3.
[17] Y. Tao, E. Gad, and M. Nakhla, "Fast and stable time-domain simulation based on modified numerical inversion of the Laplace transform," *IEEE Trans. Compon., Packag. Manuf., Technol.*, vol. 11, no. 5, pp. 848–858, May 2021.
[18] E. Hairer and A. Ostermann, "Dense output for extrapolation methods," *Numerische Math.*, vol. 58, no. 1, pp. 419–439, Dec. 1990.
[19] L. F. Shampine and L. O. Jay, "Dense output," in *Encyclopedia of Applied and Computational Mathematics*. Berlin, Germany: Springer, 2015, pp. 339–345.
[20] L. Jay, "Dense output for extrapolation based on the semi-implicit midpoint rule," *ZAMM J. Appl. Math. Mech./Zeitschrift Angew. Math. Mech.*, vol. 73, no. 11, pp. 325–329, 1993.
[21] G. A. Baker and P. Graves-Morris, "Padé approximants," in *Encyclopedia of Mathematics*, G. C. Rota, Ed. New York, NY, USA: Cambridge Univ. Press, 1995.
[22] J. C. Butcher, "General linear methods," *Acta Numerica*, vol. 15, pp. 157–256, May 2006.
[23] C.-W. Ho, A. E. Ruehli, and P. A. Brennan, "The modified nodal approach to network analysis," *IEEE Trans. Circuits Syst.*, vol. CS-22, no. 6, pp. 504–509, Jun. 1975.
[24] J. Vlach and K. Singhal, *Computer Methods for Circuit Analysis and Design*, 2nd ed. New York, NY, USA: Wiley, 1993.
[25] A. Jeffrey, *Complex Analysis and Applications*. Boca Raton, FL, USA: CRC Press, 1992.
[26] C. J. Coe, "The generalized Leibniz formula," *Amer. Math. Monthly*, vol. 57, no. 7P1, pp. 459–466, Aug. 1950.

[27] F. J. Hickernell and S. Yang, "Explicit Hermite interpolation polynomials via the cycle index with applications," *Int. J. Numer. Anal. Model.*, vol. 5, no. 3, pp. 457–465, 2008.
[28] E. Gad, M. Nakhla, R. Achar, and Y. Zhou, "*A*-stable and *L*-stable high-order integration methods for solving stiff differential equations," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 28, no. 9, pp. 1359–1372, Sep. 2009.
[29] C. Paul, *Analysis of Multiconductor Transmission Lines*. Hoboken, NJ, USA: Wiley, 1994.

Emad Gad (Senior Member, IEEE) received the Ph.D. degree from Carleton University, Ottawa, ON, Canada, in 2003.

He is currently an Associate Professor with the School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa. His current research interests include numerical simulation and modeling approaches of high-speed and RF circuits.

Dr. Gad was a co-recipient of the 2002 IEEE Microwave Prize for a significant contribution in a field of endeavor of the IEEE Microwave and Theory Techniques Society and also a recipient of several honorary awards, including the Governor General Gold Medal, the Carleton University Medal for the Outstanding Academic Performance at the graduate level, and the Ottawa Center for Research and Innovation 2003 as a Student Researcher of the year. He is a Practicing Professional Engineer in the province of Ontario, Canada.

Ye Tao (Member, IEEE) received the B.Eng. and M.A.Sc. degrees in electrical and computer engineering from Carleton University, Ottawa, ON, Canada, in 2014 and 2016, respectively, where he is currently pursuing the Ph.D. degree in electrical and computer engineering with the Department of Electronics.

His research interests include numerical techniques, simulation and modeling of high-speed interconnect, the computer-aided design of VLSI circuits, and uncertainty quantification.

Michel Nakhla (Life Fellow, IEEE) received the Ph.D. degree in electrical engineering from the University of Waterloo, Waterloo, ON, Canada, in 1975.

From 1976 to 1988, he was with Bell-Northern Research, Ottawa, ON, as the Senior Manager of the Computer-Aided Engineering Group. In 1988, he joined Carleton University, Ottawa, as a Professor and a Holder of the Senior Computer-Aided Engineering Industrial Chair established by Bell-Northern Research and the Natural Sciences and Engineering Research Council of Canada. He is the Founder of the High-Speed CAD Research Group, Carleton University, where he is currently a Chancellor's Professor of Electrical Engineering. He has authored more than 370 peer-reviewed research articles. His current research interests include modeling and simulation of high-speed circuits and interconnects, uncertainty quantification, model-order reduction, parallel processing, and nonlinear and microwave circuits.

Dr. Nakhla is a fellow of the Canadian Academy of Engineering. He is currently an Associate Editor of the IEEE TRANSACTIONS ON COMPONENTS, PACKAGING AND MANUFACTURING TECHNOLOGY and served as an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS. He is on various international committees, including the Standing Committee of the IEEE Signal and Power Workshop. He also served on several Canadian and International Government-Sponsored Research Grants Selection Panels.