

Laplace Transform Inversion through the Theta Algorithm for Power-System EMT Analysis

L.J. Castañón, J. L. Naredo, J. R. Zuluaga, E. Bañuelos-Cabral, Pablo Gómez

Abstract-- Laplace transform analysis of electromagnetic power system transients generally is based on a technique in which the Laplace inversion integral is truncated with a suitable data window. This technique, being referred to as WNLT, is appropriate for most practical cases. Nevertheless, it results inadequate for certain R&D tasks. This paper presents a new technique for numerical Laplace inversion that does not require truncation with a data window; it instead uses Brezinski's theta algorithm to account for the infinite range of the Laplace inversion integral. As opposed to the WNLT, the new technique guarantees consistent and high accuracy levels at low computational costs. Finally, the new technique is applied to the transient analysis of a power-system network. Its results compare favorably well with those from the PSCAD/EMTDC program.

Keywords: Electromagnetic transients, Frequency-domain, Laplace transform, Padé approximants, Shanks transformation, Epsilon algorithm, Theta algorithm.

I. INTRODUCTION

The analysis and simulation of electromagnetic transients (EMTs) are essential for the design and the safe and reliable operation of electric power systems (PS). These activities are usually carried out using time domain (TD) methods based on EMTP methodologies (ATP, PSCAD/EMTDC, EMTP-RV, etc.) [1], [2]. Nevertheless, for many years, progress in EMT analysis has been driven by advances in frequency-domain (FD) analysis. At this respect, a key player among power-system specialists has been the Windowed Numerical Laplace Transform (WNLT) [6], [7]. The WNLT has been adopted both, as an R&D tool and as a reference method to assess time-domain methods, models, and results [3]-[5]

The WNLT discretizes the Laplace inversion integral converting it to an infinite sum that is then truncated for numerical evaluation. [2], [6]. The errors due to this truncation are decreased by applying a suitable data window [2], [6]-[10]. Typical accuracies delivered by the WNLT are within the range of 10^{-3} and 10^{-6} , which is appropriate for most practical situations. Until recently, the WNLT had responded well to the needs of PS-EMT specialists. However, sustained progress in this field is pushing the WNLT beyond its limits. Therefore, new and more advanced tools are needed to support progress in EMT analysis. This paper demonstrates the limitations of the WNLT and has as its main objective the introduction of a new FD tool that is more accurate and reliable than the WNLT. Although the new method is primarily intended as an R&D tool, the general community of PS-EMT specialists may also benefit from its use.

One limitation of the WNLT is that its accuracy is not fixed; that is, if any parameter of a signal to be inverted is modified, the resulting precision changes. Another limitation is that the maximum precision offered by the WNLT is 10^{-9} and this is attained at a very high computational cost; that is, it requires a high number of spectral samples, in the order of 2^{20} (1,048,576). An application of the WNLT requiring a guaranteed accuracy of 10^{-9} or better is the delay identification and extraction from a frequency-domain function prior to applying a rational fit [11], [23]. A poor delay extraction can result in rational fits of an unnecessary high order and with increased possibilities of being non-passive.

Other methods to invert numerically the Laplace transform do not truncate the integration range; instead, these methods use extrapolation techniques to approximate sums of infinite series; these are referred to as sum-acceleration methods and offer high and fixed accuracies with a moderate number of samples and, in consequence, with moderate computational cost. These methods have not been extensively applied to PS-EMT analysis. If anything, only to small networks with single-phase lines with constant parameters [12]. Previously, the authors of this paper have presented one of these methods, the QD algorithm [13].

This paper presents a new numerical inversion technique for the Laplace transform that is based on Brezinski's Theta algorithm [14], [15] to accelerate the convergence of infinite sums. To the best of these authors' knowledge, this is the first application of Brezinski's Theta algorithm in the inversion of the Laplace transform, as well as in the analysis of power-system EMTs. This paper shows that the Theta algorithm far exceeds the limitations of the WNLT at a moderate computational cost

II. NUMERICAL TREATMENT OF THE LAPLACE TRANSFORM

Let $f(t)$ represent a time-domain signal and $F(s)$ its corresponding Laplace transform, both are related by the Laplace inversion integral:

$$f(t) = \frac{1}{\pi j} \int_{c-j\infty}^{c+j\infty} F(s) e^{st} ds, \quad (1)$$

where $s=c+j\omega$ is the Laplace variable with c representing a damping constant and ω the angular frequency. If $f(t)$ is real and causal, a convenient form for (1) is:

$$f(t) = \frac{e^{ct}}{\pi} \operatorname{Re} \left\{ \int_0^\infty F(c+j\omega) e^{j\omega t} d\omega \right\}. \quad (2)$$

A. Discretization

In practice, $F(s)$ usually is given either as a collection of samples or as an analytic function, with the latter being often intractable. These are the two main reasons for the numerical treatment of (1). This is performed substituting ω in (2) by $k\Delta\omega$, where $\Delta\omega$ is the frequency step. Hence:

$$f(t) \cong \tilde{f}(t) = \frac{e^{ct}\Delta\omega}{\pi} \text{Re} \left\{ \sum_{k=0}^{\infty} F(c + jk\Delta\omega) e^{jk\Delta\omega t} \right\}, \quad (3)$$

where $\tilde{f}(t)$ is an approximation to $f(t)$. It follows from (3) that $e^{-ct}\tilde{f}(t)$ is periodic [7]; moreover:

$$\tilde{f}(t) = f(t) + \epsilon_{al} \quad (4)$$

with

$$\epsilon_{al} = \sum_{m=1}^{\infty} f(t + mT) e^{-cmT} \quad (5)$$

and

$$T = 2\pi / \Delta\omega. \quad (6)$$

ϵ_{al} is the aliasing error produced by the discretization of $F(s)$ in (3) and T is the repetition period for $\tilde{f}(t)$. T also is the maximum time span at which a non-periodic $f(t)$ can be effectively approximated by $\tilde{f}(t)$. T is therefore referred to as the observation time. Aliasing error ϵ_{al} and Laplace damping constant c are related through the following expression [10]:

$$c = -\log_e(\epsilon_{rel})/T, \quad (7)$$

where $\epsilon_{rel} = \epsilon_{al}/f_{max}$ and f_{max} is the maximum expected value for $f(t)$ [10]. The aliasing error can thus be controlled by a proper choice of constant c .

Full numerical treatment of (2) requires the discretization also of t in (3); that is, the substitution of t by $n\Delta t$. For convenience, Δt is selected as a multiple of T :

$$\Delta t = T / N. \quad (8)$$

On replacing t by $n\Delta t$ in (3) and on applying (4) and (6):

$$f(n\Delta t) + \epsilon_{al} = \frac{2e^{cn\Delta t}}{\Delta t} \text{Re} \left\{ \frac{1}{N} \sum_{k=0}^{\infty} F(c + jk\Delta\omega) e^{j2\pi nk/N} \right\}, \quad (9)$$

with $n = 0, 1, 2, \dots, N - 1$.

For the sake of clarity, the following notation is now adopted:

$$f_n = f(n\Delta t)$$

and $F_k = F(c + jk\Delta\omega)$.

The evaluation of (9) requires truncating the summation at its right-hand-side to a finite number of terms M :

$$f_n + \epsilon_{al} = \frac{2e^{cn\Delta t}}{\Delta t} \text{Re} \left\{ \frac{1}{N} \sum_{k=0}^{M-1} F_k e^{j2\pi nk/N} \right\} + \epsilon_{tr}, \quad (10)$$

with ϵ_{tr} being the truncation error:

$$\epsilon_{tr} = \frac{2e^{cn\Delta t}}{\Delta t} \text{Re} \left\{ \frac{1}{N} \sum_{k=M}^{\infty} F_k e^{j2\pi nk/N} \right\} \quad (11)$$

The term $e^{j2\pi nk/N}$ in (9) is periodic with respect to running index k and N is its period; (9) can thus be restated as follows:

$$f_n + \epsilon_{al} = \frac{2e^{cn\Delta t}}{\Delta t N} \text{Re} \left\{ \sum_{k=0}^{N-1} \tilde{F}_k e^{j2\pi nk/N} \right\}, \quad (12)$$

with

$$\tilde{F}_k = \sum_{l=0}^{\infty} F_{k+lN}$$

Note that $\tilde{F}_k$ is the aliased version of F_k . This is a consequence of discretizing the time in (9) [19].

B. Windowed Numerical Laplace Transform (WNLT)

For the WNLT, sum truncation in (10) is taken at $M=N$ and truncation error is mitigated by applying a data window σ_k [7], [9], [10]:

$$f_n \cong \frac{2e^{cn\Delta t}}{\Delta t} \text{Re} \left\{ \frac{1}{N} \sum_{k=0}^{N-1} F_k \sigma_k e^{j2\pi nk/N} \right\}. \quad (13)$$

The Von Hann (or Hanning) data window has been found simple and effective [3], [5]:

$$\sigma_k = \begin{cases} [1 + \cos(\pi k/N)]/2 & 0 \leq k \leq N \\ 0 & k < 0, k > N \end{cases} \quad (14)$$

One advantage of truncating at $M=N$ is that the summation inside braces in (13) corresponds to the Discrete Fourier Transform (DFT) which is evaluated with high computational efficiency with the Fast Fourier Transform (FFT) algorithm [8]. For further simplicity and computational efficiency, N is often taken as an integer power of 2 (2^m).

The selection of damping constant c at the WNLT method is based on (7). ϵ_{rel} depends at some point on the number of signal samples N [10]. A value of $\epsilon_{rel} = 10^{-5}$ has been established empirically for an N between 2^9 (512) and 2^{11} (2048) samples [3], [5], [10].

III. INFINITE-SERIES ACCELERATION METHODS

Instead of truncating the summation range of (9) with a data window, other methods resort to extrapolation techniques to accelerate the convergence of the infinite sum of (9). Two of these are the QD algorithm [12], [13], and the Epsilon algorithm [16]. The latter is the precursor of the one being proposed here and, for this reason, it is described next.

A. Epsilon Algorithm.

The infinite sum in (9) is expressed as follows:

$$f(z) = S = \sum_{k=0}^{\infty} c_k z^k \quad (15)$$

where $c_k = F_k$ y $z = e^{2\pi jn/N}$. The definition of S_n is introduced here as the partial sum of the first $n+1$ terms in S :

$$S_n = \sum_{k=0}^n c_k z^k. \quad (16)$$

Now S in (15) is approximated by the following rational function, so that the division of $P_n(z)$ by $Q_k(z)$ generates a series with positive powers of z whose first $n+k+1$ coefficients coincide with their corresponding ones in (15).

$$f(z) \cong \frac{P_n(z)}{Q_k(z)} = \frac{a_0 + a_1 z + \dots + a_n z^n}{1 + b_1 z + \dots + b_k z^k}; \quad (17)$$

then, $P_n(z)/Q_k(z)$ corresponds to the Padé approximant of S

(or of $f(z)$) and is denoted by $[n, k]$ [18]. The approximant provides a value closer than S_n to the infinite series S in (15); additionally, $[n, k]$ tends to extend its validity as a representation of S far beyond that of the partial sums S_{n+k+1} . Nevertheless, its convergence is still slow as for an iterative approach [16].

Shanks transformation performs the change of S_n by another series denoted by $e_k(S_n)$ whose convergence is faster than that of the Padé approximants. This transformation can be expressed as a relationship between Hankel determinants [16]:

$$e_k(S_n) = \frac{H_{k+1}^n(S_n)}{H_k^n(\Delta^2 S_n)} \quad (18)$$

where $H_k^n(S_n)$ represents the Hankel determinant of order $k \times k$, whose first row is a simple sequence of partial sums beginning with S_n and ending with S_{n+k-1} . Each successive row consists of a circular rotation of one step to the right of the elements of the previous one. Here Δ represents the forward difference operator on the index “ n ”, and:

$$\begin{aligned} \Delta^2 S_n &= \Delta(\Delta S_n) = \Delta(S_{n+1} - S_n) \\ &= S_{n+2} - 2S_{n+1} + S_n. \end{aligned}$$

A problem with (18) is that it involves the calculation of determinants, so it is impractical for an iterative estimation of S . Wynn's Epsilon algorithm (ϵ -A) substantially improves the Shanks transformation by accelerating its convergence, without requiring the calculation of determinants and providing the successive values of $e_k(S_n)$ recursively [16], [17].

From Sylvester's identity for determinants, the following relation is first obtained [15]:

$$\begin{aligned} \frac{H_k^n(\Delta^3 S_n)}{H_{k+1}^n(\Delta S_n)} - \frac{H_{k-1}^{n+1}(\Delta^3 S_{n+1})}{H_k^{n+1}(\Delta S_{n+1})} \\ = \left[\frac{H_{k+1}^{n+1}(S_{n+1})}{H_k^{n+1}(\Delta^2 S_{n+1})} - \frac{H_{k+1}^n(S_n)}{H_k^n(\Delta^2 S_n)} \right]^{-1}; \end{aligned} \quad (19)$$

then, on replacing S_n by ΔS_n in (18):

$$e_k(\Delta S_n) = \frac{H_{k+1}^n(\Delta S_n)}{H_k^n(\Delta^3 S_n)}; \quad (20)$$

next, on applying (18) and (20) in (19), along with the corresponding index changes, the following expression is obtained:

$$\frac{1}{e_k(\Delta S_n)} = \frac{1}{e_{k-1}(\Delta S_{n+1})} + \frac{1}{e_k(S_{n+1}) - e_k(S_n)}; \quad (21)$$

finally, as the following definitions are introduced in (21):

$$\varepsilon_{2k}^n = e_k(S_n) \quad (22)$$

$$\varepsilon_{2k+1}^n = \frac{1}{e_k(\Delta S_n)}, \quad (23)$$

one obtains:

$$\varepsilon_{k+1}^n = \varepsilon_{k-1}^{n-1} + \frac{1}{\varepsilon_k^{n+1} - \varepsilon_k^n}. \quad (24)$$

Note in this last expression the index change, from “ $2k$ ” to “ k ”, that has been made to merge both subseries, (22) and (23), into (24). According to (22) and (23), when the subscript “ $k+1$ ” is even, (24) provides an approximation to S and, when this subscript is odd, (24) provides an intermediate value.

Expression (24) is the basis for the Epsilon algorithm, which

was applied for the first time in the calculation of the inverse Laplace transform by Crump in 1972 [21]. More recently, Brančik [12] has improved the algorithm by initializing it with the IFFT. He has also applied it to the analysis of circuits with distributed-parameter elements.

B. Theta Algorithm

To further accelerate the convergence of ε_k^n in the Epsilon algorithm, Brezinski derives from (24) the following two expressions [15]:

$$\theta_{2k+1}^m = \theta_{2k-1}^{m+1} + D_{2k}^m \quad (25)$$

$$\theta_{2k+2}^m = \theta_{2k}^{m+1} + w_k D_{2k+1}^m \quad (26)$$

$$\text{with } D_k^m = [\theta_k^{m+1} - \theta_k^m]^{-1}. \quad (27)$$

Note the notation change from ε_k^n to θ_k^m . Also note the introduction of the factor w_k in (26) for accelerating the convergence. It is required that θ_{2k+2}^m converges faster than θ_{2k}^{m+1} ; i.e., [15]:

$$\lim_{m \rightarrow \infty} \frac{\theta_{2k+2}^m}{\theta_{2k}^{m+1}} = 0. \quad (28)$$

To determine w_k , first apply the operator Δ on both sides of (26):

$$\Delta \theta_{2k+2}^m = \Delta \theta_{2k}^{m+1} + w_k \Delta D_{2k+1}^m;$$

then divide by θ_{2k}^{m+1} :

$$\frac{\Delta \theta_{2k+2}^m}{\Delta \theta_{2k}^{m+1}} = 1 + w_k \frac{\Delta D_{2k+1}^m}{\Delta \theta_{2k}^{m+1}}; \quad (29)$$

next, take the limit as $m \rightarrow \infty$ and consider relation (28) to obtain:

$$w_k \lim_{m \rightarrow \infty} \frac{\Delta D_{2k+1}^m}{\Delta \theta_{2k}^{m+1}} = -1;$$

or,

$$w_k = - \lim_{m \rightarrow \infty} \frac{\Delta \theta_{2k}^{m+1}}{\Delta D_{2k+1}^m}. \quad (30)$$

Given the difficulty of evaluating this limit algorithmically, Brezinski replaces w_k by the following factor in (26):

$$w_k^m = - \frac{\Delta \theta_{2k}^{m+1}}{\Delta D_{2k+1}^m} = - \frac{[\theta_{2k}^{m+2} - \theta_{2k}^{m+1}]}{[D_{2k+1}^{m+1} - D_{2k+1}^m]}; \quad (31)$$

so,

$$\theta_{2k+2}^m = \theta_{2k}^{m+1} + \frac{[\theta_{2k}^{m+2} - \theta_{2k}^{m+1}][\theta_{2k+1}^{m+2} - \theta_{2k+1}^{m+1}]}{\theta_{2k+1}^{m+2} - 2\theta_{2k+1}^{m+1} + \theta_{2k+1}^m}. \quad (32)$$

Expressions (25) and (32) form the basis of the Theta algorithm (θ -A). These expressions are applied alternately in the recursive update of θ_i^m ; (25) when “ i ” is odd ($i = 2k + 1$) and (32) when it is even ($i = 2k + 2$).

Fig. 1a illustrates the structure of the Theta algorithm, both for the estimation of S in (15) and for the inversion of the Laplace transform. This figure consists of a two-dimensional arrangement of the terms θ_i^m whose subscripts “ i ” and “ m ” indicate its column and its diagonal, respectively. For the numerical inversion of the Laplace transform, the terms θ_i^m are vectors of dimension “ N ”; i.e., the number of time samples time to be obtained. The terms in the first column are initialized with zero vectors:

$$\theta_{-1}^m = \mathbf{0}; m = 1, 2, \dots, 3J + 3,$$

where J is the number of iterations (or refinements) of the algorithm. Fig. 1 illustrates, for example, the case $J = 2$. The first term of the second column θ_0^0 is initialized with the vector of partial sums $\mathbf{S}_N$ which is obtained in a highly convenient way with the inverse fast Fourier transform of N samples (ifft_N):

$$\theta_0^0 = \mathbf{S}_N = N \times \text{ifft}_N(\mathbf{F}),$$

where $\mathbf{F}$ is the vector of N samples of the function $F(s)$ that is to be inverted:

$$\mathbf{F} = [F_0/2, F_1, \dots, F_{N-1}]^T.$$

The additional terms in the second column of Fig. 1a are obtained iteratively as follows:

$$\theta_0^{m+1} = \theta_0^m + \mathbf{E}^m \times F_{N+m}; m = 0, 1, \dots, 3J + 2$$

where $\mathbf{E}$ is the vector with the N samples of the complex exponential:

$$\mathbf{E} = [1, e^{2\pi j/N}, e^{4\pi j/N}, \dots, e^{2\pi j(N-1)/N}]^T$$

Once θ_{-1}^m and θ_0^m have been initialized (i.e., the first two columns of the diagram in Fig. 1a), the elements θ_i^m at the other columns on the right are obtained with (25) when " i " is odd and with (32) when " i " is even. Note that Fig. 1b shows the dependence of odd-column element θ_{2k+1}^m to be updated with those elements of the previously available columns $2k$ and $2k-1$. Note also that Fig. 1c shows the dependence of even-column element θ_{2k+2}^m to be updated with those elements of the previously available columns $2k+1$ and $2k$.

At Fig. 1a, the columns with an odd index are marked in blue and those with an even index in black. The elements θ_{2k}^0 of even index on the upper diagonal of the diagram are the ones providing the estimates for S and, the higher their subscript is, the better approximation these are to the expression between braces in (9).

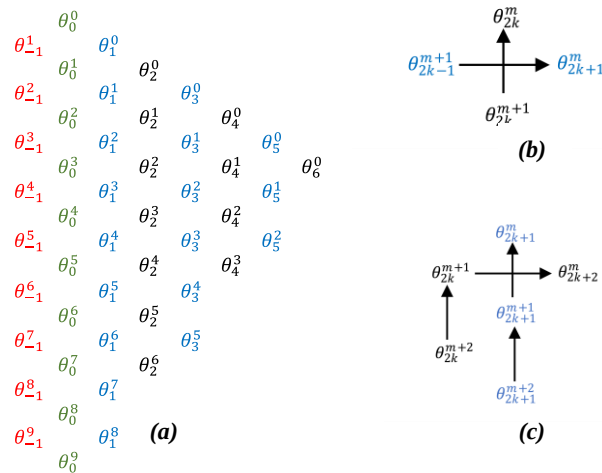


Fig. 1. **a)** Theta algorithm diagram. **b)** Diagram for equation (25). **c)** Diagram for equation (32).

From all the above, the inverse d Laplace transform is as follows:

$$\mathbf{f} = \left(\frac{2}{T}\right) \times \mathbf{C} \circ \Re\{\theta_{2J+2}^0\}, \quad (33)$$

where $\mathbf{f}$ is the vector of N samples for the signal being obtained in the time domain; $\mathbf{C}$ is the vector of order N whose elements C_n are of the form

$$C_n = e^{cn\Delta t}, \quad n = 0, 1, 2, \dots, N - 1;$$

finally, the operator " $\circ$ " represents the Hadamard product of two vectors resulting in another vector formed with the multiplication between corresponding elements. In the experience of these authors, the theta algorithm often converges at the first iteration; i.e., at $J = 0$.

IV. VALIDATION

One way to evaluate the precision and performance of the method proposed here is to apply it to s-domain functions whose inverse transforms are already analytically determined. The numerical analysis community has already established a set of 35 test functions to evaluate numerical methods for inverting Laplace transforms [18], [20]. The proposed Theta algorithm has been tested with all these functions and, except for one case that is addressed below, this algorithm shows practically the same satisfactory performance. For the sake of paper-space economy, only one representative case is presented below in which the WNLt, Epsilon, and Theta algorithms are tested and compared.

Consider the following function of s , along with its inverse transform:

$$F(s) = \left[1/s + \frac{s + 0.5}{(s + 0.5)^2 + (2\pi f)^2}\right] e^{-0.2s} \quad (34)$$

and

$$f(t) = [1 + \cos(2\pi f(t - 0.2))]e^{-0.5(t-0.2)}u(t - 0.2), \quad (35)$$

where $u(t)$ is the unit step (Heaviside function). The three abovementioned methods are now applied in the numerical inversion of (34) for three different values of the oscillation frequency: $f = 2$ Hz, $f = 10$ Hz and $f = 20$ Hz. The observation time for the inverted function is $T = 1.2$ s and $N = 1024$ samples are used. According to subsection II.B, the Laplace damping constant c is determined by means of (7) with a relative error $\epsilon_{rel} = 10^{-5}$ for the WNLt, while $\epsilon_{rel} = 10^{-9}$ is used for determining c in the Epsilon and the Theta algorithms.

Fig. 2a shows the plots of $f(t)$ as obtained with the WNLt for the three values of frequency being considered. All the differences between the algorithmically obtained plots and their corresponding ones from (35) cannot be differentiated by eye. For this reason, the plots of the base 10 logarithm of the relative errors for the WNLt are provided in Fig. 2b. Relative errors are calculated as follows:

$$\epsilon_{rel}(n) = \frac{|f(n\Delta t) - \tilde{f}_n|}{\max\{f(t)\}}, \quad n = 0, 1, 2, \dots, 1023.$$

where $f(n\Delta t)$ is the corresponding value of the analytic function in (35), $\tilde{f}_n$ is the n -th value being obtained numerically and $\max\{f(t)\}$ is the maximum value of $|f(t)|$ in (35). Note that the precision level of 10^{-5} is obtained only for the case with $f = 2$ Hz. By modifying f in (34) the precision does not remain fixed; in fact, it is degraded.

Figs. 3a and 3b show the respective relative errors of the results with the Epsilon and Theta algorithms when applied to (34). It can be observed that the precisions of both methods are similar and remain fixed despite the changes in f . In terms of

computational speed, the Theta algorithm is two times faster than the Epsilon one and 4.2 times slower than the WNLT. On a personal computer with processor Core i9, 2.3 GHz, 16 GB RAM and running in MatLab, the test case function (35) was executed with the WNLT in 0.04 ms, with the Theta algorithm in 0.17 ms and with the Epsilon algorithm in 0.32 ms. From all the above, it follows that the accuracies of the Theta and Epsilon algorithms are practically the same, Nevertheless, the Theta algorithm is preferred for its higher computational efficiency.

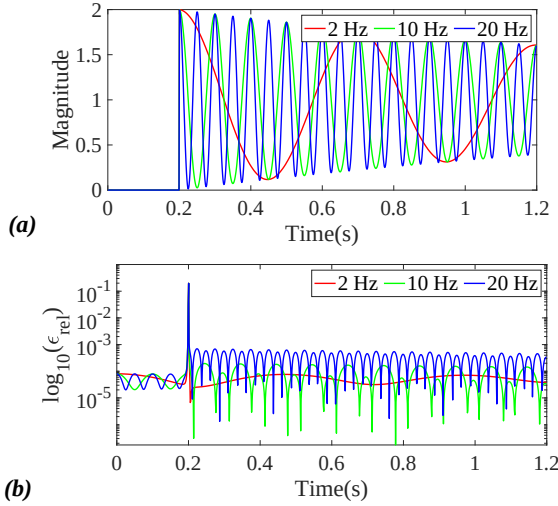


Fig. 2. **a)** Plots of $f(t)$ being obtained with the WNLT. **b)** Plots of errors for the obtained $f(t)$ with three different oscillation frequencies.

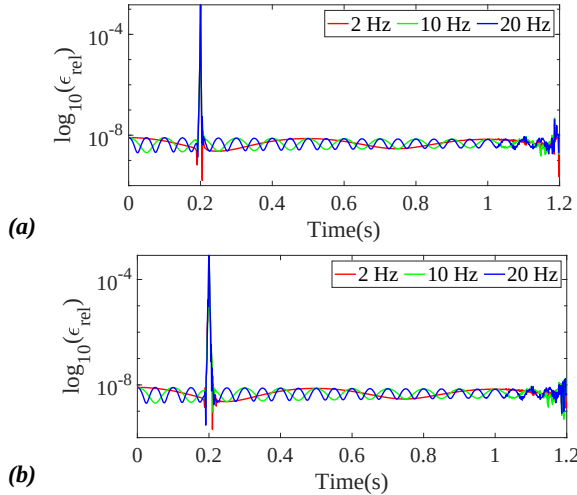


Fig. 3. **a)** Plots of errors for the $f(t)$ being obtained with the Epsilon algorithm. **b)** Plots of errors for the $f(t)$ being obtained with the Theta algorithm.

Out of the previously mentioned 35 test functions, for function 34 as listed in [18] the accuracy of both algorithms, the Epsilon and the Theta, falls to levels like those of the WNLT. This function is given as follows and corresponds to a square wave:

$$f_{34}(s) = \frac{2e^{s/2}}{s} \operatorname{cosech}\left(\frac{s}{2}\right).$$

It is apparent from this case that the three methods being tested here (WNLT, Epsilon, and Theta) cannot satisfactorily handle signals and functions that include multiple discontinuities. This could well be a topic for future research.

V. POWER SYSTEM EMT TEST CASE

Fig. 4 shows the one-line diagram of an electrical power network consisting of three generators, three transformers, eight overhead transmission lines, four R-XL loads and one capacitive load. Table I provides the data for the generators and transformers, while Table II provides the values for the loads. The base voltage is 200 kV. All the lines have the same transversal geometry and electrical properties of materials. Fig. 5 shows the line geometry and provides its material data. Table III provides the lengths of the lines. Fig. 4 network is based on a test case provided in references [24] and [25] and has been adapted for the purposes of this paper.

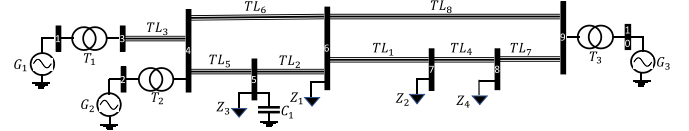


Fig. 4. Network one-line diagram.

For the calculation of transient responses (overvoltages), the network is represented in nodal form in the Laplace domain. Generators are incorporated into this representation by their Norton equivalents, transformers are represented by series R-L branches. The transmission lines are included through their three-phase nodal representations with full frequency-dependent AB parameters [1], [2], [22].

TABLE I
GENERATOR AND TRANSFORMER DATA.

	Generator			Transformer		
	G_1	G_2	G_3	T_1	T_2	T_3
V_{pu}	$1.03 \angle 20.2^\circ$	$1.01 \angle 10.5^\circ$	$1.01 \angle -17^\circ$	-	-	-
Z						
R	1.2Ω	1.1Ω	0.8Ω	1.5Ω	0.8Ω	0.6Ω
L	38.98mH	45.52mH	35.23mH	23.4mH	29.5mH	35.23mH

TABLE II
LOAD DATA.

Load	Z_1	Z_2	Z_3	Z_4	C_1
R	250Ω	350Ω	250Ω	420Ω	
L	25mH	60mH	25mH	30mH	
C	-	-	-	-	20uF

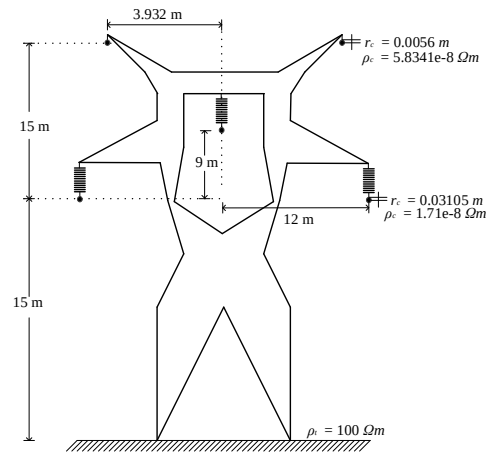


Fig. 5. Line transversal geometry and material data.

TABLE III
LINE LENGTHS.

Line	TL_1	TL_2	TL_3	TL_4	TL_5	TL_6	TL_7	TL_8
Length (km)	220	35	10	35	65	133	42	375

The simultaneous energization of the entire network is now simulated. To do this, the following relationship is used:

$$\mathbf{V}(s) = \mathbf{Y}(s)^{-1} \mathbf{I}(s) \quad (36)$$

where $\mathbf{V}(s)$ is the vector of nodal voltages of the network, $\mathbf{I}(s)$ is the vector of currents injected into the nodes and $\mathbf{Y}(s)$ is the nodal matrix of the network. Once (36) is solved, the waveforms of nodal voltages are determined with the inverse Laplace transform:

$$\mathbf{v}(t) = \mathcal{L}^{-1}(\mathbf{V}(s))$$

For the Laplace numerical inversion, the Epsilon and Theta algorithms are used with 1024 samples and the observation time is set at $T = 40$ ms. Figures 6a and 6b show the respective voltage responses for nodes 6 and 9. These figures also include the simulations being obtained with the PSCAD / EMTDC program with the same number of time steps $N=1024$. Note that the results with the three methods apparently are in good agreement. Fig. 7a provides plots of the logarithm base 10 of absolute differences for the voltage waveforms of phase a at node 6. In blue color is the plot of differences between PSCAD/EMTDC and the Theta algorithm, both with $N=1024$ time steps. The plot in red color corresponds to the differences as the number of time steps in the PSCA/EMTDC is increased to $N=8192$, and the Theta algorithm remains with $N=1024$ steps. Note that one segment of the differences in blue color is above 1 %. Also note that, at higher resolutions, the results from PSCAD/EMTDC approach those of the Theta algorithm.

Fig. 7b shows in blue color the base 10 logarithms of the differences between the results of the WNLT and those from the Theta algorithm both using the same number of frequency steps $N=1024$. The red color plot corresponds to the differences between the WNLT with $N=8192$ frequency steps and the Theta algorithm remaining with $N=1024$. Also note that as the number of samples is increased in the WNLT its results become closer to those of the Theta algorithm.

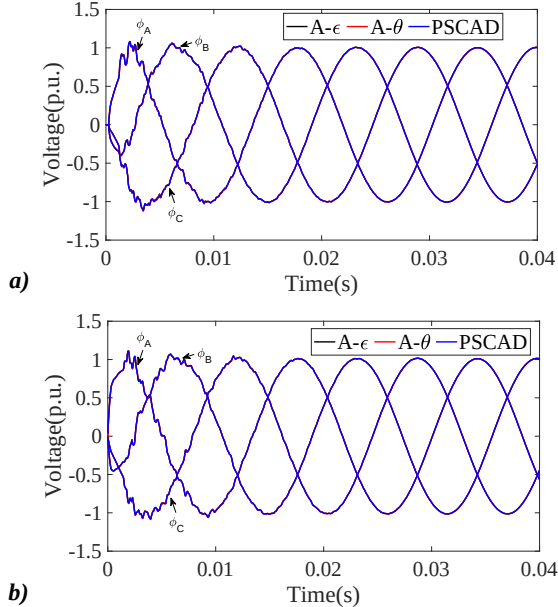


Fig. 6. **a)** Voltage waveforms at node 6. **b)** Voltage waveforms at node 9.

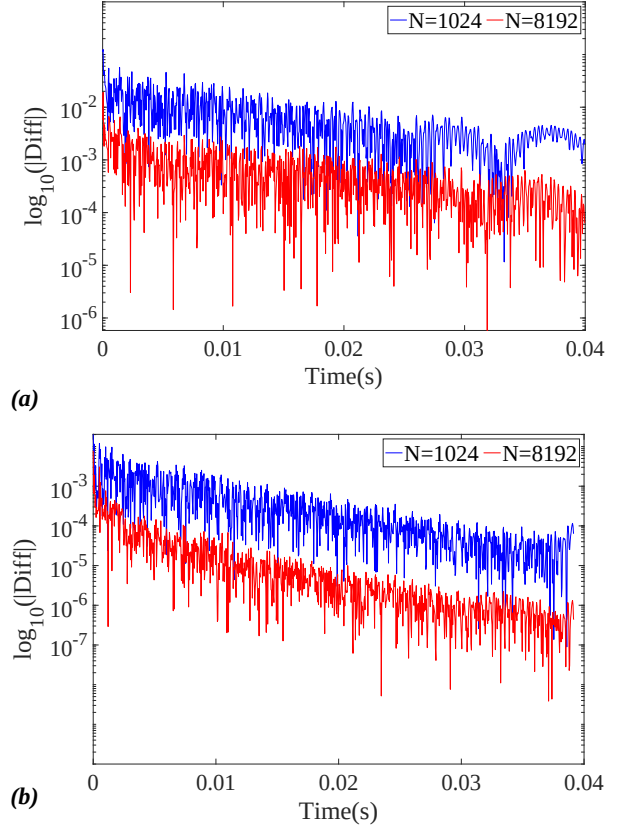


Fig. 7. **a)** Comparing PSCAD with the Theta algorithm. **b)** Comparing the WNLT with the Theta algorithm.

VI. CONCLUSIONS

The windowed numerical Laplace transform (WNLT) is commonly used for frequency domain analysis of electromagnetic transients in power systems [3], [5]-[10]. This technique is appropriate for many practical applications and often is adopted as reference to assess newly developed time-domain methods. However, the authors of this article have found that it presents serious limitations as reference method, as well as for research and development tasks. One of its limitations is that the WNLT does not guarantee a fixed precision. This has been shown with an example in this paper. Another limitation is that its accuracy is limited to values between 10^{-5} and 10^{-6} , and, in addition, an accuracy up to 10^{-9} can be attained at an excessive computational cost. For these reasons, the authors of this paper have dedicated themselves to the search for other methods of numerical inversion of the Laplace transform that can achieve high accuracies at a moderate computational cost, and that their precision levels are fixed.

In this article, a new method of numerical inversion of the Laplace transform based on Brezinski's Theta algorithm [14], [15], [17] has been proposed and described. Apparently, this is the first time that this algorithm has been used for the numerical inversion of the Laplace transform and for the analysis of EMTs in power systems. Along with the theta algorithm, the description of Wynn's Epsilon algorithm [12], [16] has also been included, since this is a precursor to the first one. The proposed algorithm has been evaluated first by its application to a set of 35 test functions that has been established by the Numerical

Analysis community [18], [20]. Except for one function which corresponds to a square wave with several discontinuities, this algorithm shows a high performance. The Theta algorithm has been then applied to a transient study in a 10-node electric network with three generators, three transformers, eight overhead lines and five loads. The results of this study have been compared satisfactorily with those obtained with the PSCAD/EMTDC program.

Finally, it has been demonstrated that the proposed method offers a guaranteed high precision at a moderate computational cost; therefore, it is recommended both, as reference method and as a tool for research and development activities in the field of EMT analysis.

VII. REFERENCES

- [1] Hermann W. Dommel, *EMTP-Theory Book*, Bonneville Power Administration, Portland, OR, USA, 1996.
- [2] Martinez-Velasco, Juan A., *Transient Analysis of Power Systems: Solution Techniques, Tools and Applications*, John Wiley & Sons, 2014.
- [3] F. A. Uribe, J. L. Naredo, P. Moreno, L. Guardado, "Electromagnetic Transients in Underground Transmission Systems Through the Numerical Laplace Transform", *International Journal of Electrical Power and Energy Systems*, Vol. 24/3, pp 215-221, March 2002.
- [4] B. Gustavsen, "Validation of frequency-dependent transmission line models," *IEEE Transactions on Power Delivery*, vol. 20, no. 2, pp. 925-933, April 2005.
- [5] Pablo Moreno, Pablo Gómez, José L. Naredo, J. L. Guardado, "Frequency domain transient analysis of electrical networks including non-linear conditions", *International Journal of Electrical Power and Energy Systems*, Vol. 27, Issue 2, pp. 139-146, Feb. 2005.
- [6] P. Moreno and A. Ramirez, "Implementation of the Numerical Laplace Transform: A Review", *IEEE Trans. on Power Delivery*, vol. 23, no. 4, pp. 2599-2609, Oct. 2008.
- [7] N. Mullineux and J.R. Reed, "Developments in Obtaining Transient Response Using Fourier Transforms: Part IV-Survey of the Theory", *Int. J. Elect. Engng. Educ.*, Vol. 10, pp. 259-265, 1973.
- [8] A. Ametani, "The Application of the Fast Fourier Transform to Electrical Transient Phenomena", *Int. J. Elect. Engng. Educ.*, Vol. 10, Issue 4, pp. 277-287, October 1973.
- [9] D.J. Wilcox, "Numerical Laplace Transformation and Inversion", *Int. J. Elect. Engng. Educ.*, Vol. 15, pp. 247-265, 1978.
- [10] L.M. Wedepohl, "Power System Transients: Errors Incurred in the Numerical Inversion of the Laplace Transform", in Proc. of The IEEE Twenty-Sixth Midwest Symposium on Circuits and Systems, Puebla, Mexico, August 1983.
- [11] Martin G. Vega, J. L. Naredo, O. Ramos-Leaños, "Minimum Delay Systems for the Modeling of Transmission Lines at EMT Power System Studies", in Proc. of The 12th International Conference on Power Systems Transients, IPST'2017, Seoul, Korea, June 2017.
- [12] Brančík, L., "Programs for fast numerical inversion of Laplace transforms in Matlab language environment." Proceedings of 7th Conference MATLAB'99, pp. 27-39, ISBN 80-7080-354-1, Prague, Czech Republic, November 3, 1999
- [13] L. J. Castañón, J. L. Naredo, J. R. Zuluaga, M. G. Vega-Grijalva, "Electromagnetic-Transient Analysis in the Laplace-Domain Through the QD Algorithm", in Proc. of the 13th International Conference on Power Systems Transients, IPST'2019, ISSN 2434-9739, Perpignan, France, June, 2019.
- [14] Brezinski, Claude, "Convergence acceleration during the 20th century.", *Journal of Computational and Applied Mathematics*, 122.1, pp. 1-21, 2000.
- [15] Brezinski, Claude, *Accélération de la convergence en analyse numérique.*, Vol. 584. Springer, 2006.
- [16] P. Wynn, "On a Device for Computing the $e_m(S_n)$ Transformation", *Mathematical Tables and Other Aids to Computation*, Vol. 10, No. 54, pp. 91-96, Apr. 1956.
- [17] Claude Brezinski, Yi He, Xing Biao Hu, Michela Redivo-Zaglia, Jian-Quing Sun, "Multistep ϵ -algorithm, Shanks' transformation, and the Lotka-Volterra system by Hirota's method." *Mathematics of Computation*, 81.279, pp. 1527-1549, 2012.
- [18] Cohen, A.M, *Numerical methods for Laplace transform inversion*, Springer Science, New York, U.S.A., 2007.
- [19] John G. Proakis, Dimitris G. Manolakis, *Digital Signal Processing*", 4th Ed., Prentice Hall, 2007.
- [20] Davies, Brian, and Brian Martin. "Numerical inversion of the Laplace transform: a survey and comparison of methods.", *Journal of computational physics*, 33.1, (1979): 1-32.
- [21] Crump, Kenny S. "Numerical inversion of Laplace transforms using a Fourier series approximation.", *Journal of the ACM (JACM)*, 23.1 (1976): 89-96.
- [22] J.A. Martinez-Velasco, *Power System Transients: Parameter Determination*, CRC Press, Boca Raton FL, 2009.
- [23] B. Gustavsen, "Optimal Time Delay Extraction for Transmission Line Modeling," in *IEEE Transactions on Power Delivery*, vol. 32, no. 1, pp. 45-54, Feb. 2017.
- [24] Prabha Kundur, "Power System Stability and Control", MacGraw-Hill, 1994.
- [25] M. Matar and R. Iravani, "The Reconfigurable-Hardware Real-Time and Faster-Than-Real-Time Simulator for the Analysis of Electromagnetic Transients in Power Systems," in *IEEE Transactions on Power Delivery*, vol. 28, no. 2, pp. 619-627, April 2013.