

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
INSTITUTE OF THEORETICAL AND EXPERIMENTAL ELECTRICAL ENGINEERING

ING. LUBOMÍR BRANČÍK, CSc.

**TECHNIQUES OF TIME–DOMAIN SIMULATION
OF TRANSMISSION LINES BASED ON LAPLACE
TRANSFORMATION METHODS**

HABILITATION THESIS

VUTIAM

BRNO 2000

© 2000 L. Brančík
ISBN

CONTENTS

ABOUT AUTHOR.....	4
INTRODUCTION	5
1. FFT-BASED METHODS OF NUMERICAL INVERSION OF LAPLACE TRANSFORMS	7
1.1. ONE-DIMENSIONAL NILT METHOD	7
1.1.1. Theoretical Base	7
1.1.2. FFT and ϵ -Algorithm Application	8
1.1.3. Scalar, Vector and Matrix NILT Versions	9
1.1.4. Experimental Error Analysis	10
1.2. TWO-DIMENSIONAL NILT METHOD	11
1.2.1. Theoretical Base	11
1.2.2. FFT and ϵ -Algorithm Application	12
1.2.3. Experimental Error Analysis	13
2. METHODS OF TIME-DOMAIN SIMULATION OF TRANSMISSION LINES BASED ON ONE-DIMENSIONAL LAPLACE TRANSFORMATION	14
2.1. BASIC MTL EQUATIONS AND THEIR SOLUTION	14
2.2. SIMULATION OF WAVE PROPAGATION IN SIMPLE MULTICONDUCTOR TRANSMISSION LINE SYSTEMS	16
2.2.1. Generalized Thévenin and Norton Equivalents Application	16
2.2.2. Matlab Oriented Computational Method	17
2.2.3. Examples	18
2.3. TIME-DOMAIN SIMULATION OF NETWORKS CONTAINING SECTIONS WITH MULTICONDUCTOR TRANSMISSION LINES	21
2.3.1. MNA Matrix Equation Formulation	21
2.3.2. Examples	22
3. METHODS OF TIME-DOMAIN SIMULATION OF TRANSMISSION LINES BASED ON TWO-DIMENSIONAL LAPLACE TRANSFORMATION	25
3.1. BASIC MTL EQUATIONS AND THEIR TRANSFORMATIONS	25
3.2. BOUNDARY CONDITIONS INCORPORATION	26
3.3. COMPUTATIONAL METHODS	28
3.4. EXAMPLE	29
4. CONCLUSION	30
PRINCIPAL REFERENCES	31
SOUHRN (ABSTRACT IN CZECH)	35

Lubomír Brančík was born in 1961 in Kyjov. In 1985, he obtained Ing. (MSc.) degree in microelectronics, and in 1993, CSc. (PhD.) degree in measuring technique, both at the Brno University of Technology.

From 1987 to 1994 he worked as a research worker at the Institute of Nondestructive Diagnostics and at the Institute of Theoretical and Experimental Electrical Engineering, both at the Brno University of Technology. He dealt with nondestructive testing of metal materials by means of electromagnetic methods focusing his main interest on a computer processing of measured data. Since 1994 he has been working as an assistant professor at the same institute. He is engaged in teaching circuit theory and measuring technique courses. Currently his main research interests include applications of numerical methods in circuit theory and related problems of computer simulation.

INTRODUCTION

Universal mathematical tools suitable for solving large scientific and technological tasks are steadily taking on higher importance this time. This is due to constantly wider and wider availability of sufficiently powerful computers including personal ones which can be used with relatively low costs for the simulations. Nowadays, not only permanently faster processors and larger memories but also advanced computer technologies, e.g. computers with vector processors and/or multiprocessors, are starting to play significant role. These aspects stimulate an effort to developing new and improving existing numerical techniques applicable in the computer simulations.

One of such a universal mathematical tool is e.g. Matlab language of the MathWorks, Inc., which has proven to be very convenient for the development and experimental verification of novel numerical techniques and approaches. It is among others due to the ability of this language to process vectors, matrices and multidimensional arrays very effectively when many mathematical operations can be applied to run in parallel on these data structures. Furthermore, many various advanced mathematical functions are included in the core of the language and others can easily be appended through necessary problem-oriented toolboxes. This is why all the methods developed inside the scope of the habilitation thesis have been programmed, tested and verified using this mathematical language. Besides the Matlab language is becoming increasingly popular not only among scientists and engineers but also among undergraduate students this time, and thus running on widely available PC machines the developed procedures are immediately ready to be introduced into the educational process.

The habilitation thesis has focused its attention in two principal fields. In the first part methods for the numerical inversion of Laplace transforms (NILT), both one-dimensional (1D) and two-dimensional (2D), were elaborated, and the new techniques of their computation have been developed. While the 1D Laplace transformation plays the fundamental role in solving ordinary differential equations, the 2D-LT has its wide applications to the solution of partial differential equations. Therefore, the NILT methods rank among ones frequently used for an analysis of transient behaviour of linear dynamical systems, and their fast and accurate computation has permanently wide theoretical and practical significance.

Although many various techniques were studied, some of them can be found e.g. in [1-16], finally two effective methods have been developed and elaborated in details. The first method is based on the approximation of the exponential function in the definition integral of the inverse Laplace transformation with functions containing hyperbolic cosine or sine. This techniques lead to infinite sums the convergence of which can be accelerated by Euler transformation [17-18]. The second method is based on the trapezoidal rule of integration of the definition integral with properly chosen integration step. This technique leads to infinite sums which can partially be evaluated using FFT algorithm while their convergence is accelerated with the ϵ -algorithm [19-26]. Comparing other existing NILT methods and evaluating results of many experiments the developed procedures seem to rank among the most versatile, accurate and fast. In this reduced thesis extent, however, only the last method is described in more details as just this one has been applied to problems of the consequent thesis part.

The second part of the habilitation thesis sets the task to contribute to techniques of the computer simulation of transmission lines including multiconductor ones. From many various methods, see e.g. [31-40], there are worked out those based on the Laplace transformation (1D and 2D) making allowance for the effective algorithmization just in the universal mathematical language Matlab. The methods related to the analysis and simulation of the multiconductor transmission lines (MTL) attract the great attention this time as they come in useful in the field of high-speed interconnects on microelectronic chips and boards [41-44]. Transmission line effects such as reflection, dispersion, attenuation and crosstalk may trigger false signals which in turn can cause malfunction of all the system. Therefore, sufficiently accurate modeling and transient simulation of both lossy and lossless MTLs are essential in reliable high-speed digital circuit design. The transient analysis of the MTLs is also important in the field of power systems where it can be used to predict the behaviour of long power lines excited by lightning strokes or by short-circuit currents.

Firstly the methods based on the 1D Laplace transformation approach have been elaborated. The developed procedures can handle MTLs considered as lossy, frequency dependent, nonuniform, and under nonzero initial voltage and/or current distributions in general [45-51]. The computation method is basically based on the direct matrix solution of the set of MTL ordinary differential equations in frequency domain. Utilizing Matlab capabilities effectively this approach has the advantages compared with widely used modal analysis technique [31]. To simulate simple MTL systems the generalized Thévenin or Norton equivalents are applied to model MTL's linear terminating networks, and thus to incorporate boundary conditions. For the simulation of more complex systems which contain MTLs as their sections a technique based on the modified nodal admittance (MNA) matrix equation is applied, see latest works [52,53]. As the final computation step the NILT method is always used to get the solution in time domain [19-22].

In the case when only the uniform transmission lines are considered the 2D Laplace transformation approach can be used for the simulation [54-57]. Applying the 2D-LT the originally partial differential equations become the algebraic ones. Then after incorporating boundary conditions the final 2D transform in (q,s) -domain is processed by the 2D-NILT algorithm to get the solution in (x,t) -domain [23-26]. The advantage of the method lies in the possibility to obtain the solution on the whole 2D region in a single calculation step. In the case of the MTL's zero initial conditions the method seems to be simple. To be able to take MTL's nonzero initial conditions into account, however, either the procedure for a partial numerical inversion $\mathbf{L}_x^{-1}[\cdot]$ must be applied or convolution integrals must be precomputed [56]. Beside of this direct 2D-LT method a new approach has also been proposed when certain 2D transforms are inverted separately and finally the results are convolved with respect to time t to get the resultant 2D original [56]. The method can advantageously be used to demonstrate how the waves are propagated along MTL's wires. Utilizing Matlab language capabilities the results can easily be presented in three-dimensional graphs or utilized for an animation process, too.

As the full thesis for habilitation consists of the set of published papers this presented thesis has currently been appended with the chapter 2.3 based on the latest papers [52,53].

1. FFT-BASED METHODS OF NUMERICAL INVERSION OF LAPLACE TRANSFORMS

As the speed of widely available computers is increasingly growing NILT methods based on a relatively simple principle, namely a numerical integration of the Bromwich integral, are gaining a renewed interest, see e.g. [16]. Following papers [8-10] a 1D-NILT method based on the FFT and the ϵ -algorithm was developed [19-22]. It proved to be convenient from point of view of both desired speed and an accuracy. Particularly utilizing capabilities of the Matlab language the numerical method under consideration appears to be very fast to invert not only simple Laplace transforms but also more complicated transforms arising e.g. when multiconductor transmission line systems are simulated. Especially for these purposes the basic (scalar) version of the 1D-NILT algorithm [19,20] was generalized to be usable to invert the Laplace transforms in vector and matrix forms very effectively [21,22]. This is enabled due to the fact that many mathematical operations in the Matlab language (started with the version 5) can run in parallel on multidimensional arrays without necessity to use any outer loop structures. This is why the CPU time is saved considerably in practical applications. The above 1D-NILT method has been generalized to 2D-NILT case in papers [23-26]. Compared with the recently published method [16] the developed one shows to be more accurate about three orders in average.

1.1. ONE-DIMENSIONAL NILT METHOD

1.1.1. Theoretical Base

In principle to find the original $f(t)$ to the Laplace transform $F(s)$ the Bromwich integral can be considered as

$$f(t) = \frac{1}{2\pi j} \int_{c-j\infty}^{c+j\infty} F(s)e^{st} ds \quad (1.1)$$

under the basic assumption $|f(t)| \leq Ke^{\alpha t}$, K real positive, α as the exponential order of the real function $f(t)$, $t \geq 0$, and $F(s)$ defined for $\text{Re}[s] > \alpha$.

The aim is to evaluate this integral numerically not only accurately enough but also very fast on a whole given interval $\langle 0; t_m \rangle$. Applying a trapezoidal rule of the integration then after some manipulations [19,20] the approximate formula in a discrete form $\tilde{f}_k = \tilde{f}(kT)$ can be written as

$$\tilde{f}_k = C_k \{ 2 \text{Re} [\sum_{n=0}^{\infty} F_n z_k^n] - F_0 \}, \quad k = 0, 1, \dots, N-1, \quad (1.2)$$

with

$$C_k = \frac{\Omega}{2\pi} e^{ckT}, \quad F_n = F(c - jn\Omega), \quad z_k = e^{-jkT\Omega}, \quad (1.3)$$

where T and $\Omega = 2\pi/(NT)$ are sampling periods in the original and the transform domain, respectively. It was shown that (1.2) corresponds to the Fourier series approximation of the original $f(t)$ when the error can theoretically be controlled on the interval $t \in \langle 0; NT \rangle$. In

practical computations, however, to avoid increasing the error at the end of this interval the required maximum time is supposed to be $t_m = (M - 1)T$, with $M = N/2$ as the number of resultant computed points. This leads to the condition of choosing the sampling period in the frequency domain as $\Omega = \pi(1 - 1/M)/t_m$. The error analysis [19] has shown that the coefficient c in (1.3) can approximately be set as

$$c \approx \alpha - \frac{\Omega}{2\pi} \ln E_r, \quad (1.4)$$

where E_r denotes the desired relative error. To minimize the error towards this theoretical value the infinite sum in (1.2) must be evaluated as accurately as possible.

1.1.2. FFT and ϵ -Algorithm Application

The solution consists of three steps. Firstly this sum is evaluated using only N first terms when the FFT algorithm can be applied supposing $N = 2^m$, m integer. Secondly after truncating the result of the FFT operation to have only the length M the ϵ -algorithm [27-29] is applied to accelerate the convergence of this series. The ϵ -algorithm uses only a few additional terms above those N used by the FFT, however, the sum becomes as if it were evaluated using much more terms. Finally the result of the ϵ -algorithm is substituted into (1.2) to finish the computation.

The ϵ -algorithm known also as Wynn algorithm ranks among nonlinear ones that are used to speed up the convergence of series. The infinite sum of a complex Fourier series in (1.2) is written in the form of the sum of a power series in a variable z . As it is shown e.g. in [28] just for the partial sums of a power series the ϵ -algorithm can be very efficient as it is equivalent to constructing successive rational Padé approximations to the power series. The principle is explained by using a lozenge diagram in Fig.1.1.

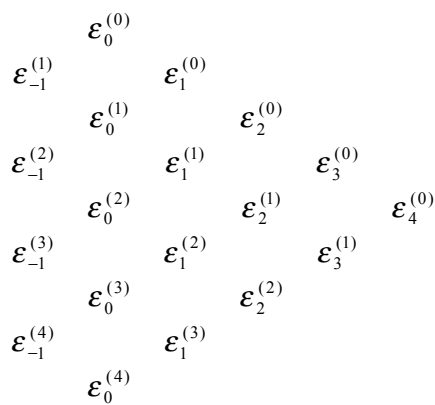


Fig.1.1 The ϵ -algorithm lozenge diagram

The first column is formed with $\tilde{a}_1^{(s)} = 0$, $s = 1, 2, \dots$, the second column with the partial sums computed recurrently as

$$\tilde{a}_0^{(s+1)} = \tilde{a}_0^{(s)} + F_{N+s} z^{N+s}, \quad (1.5)$$

$s = 0, 1, 2, \dots$, when the initial partial sum $\tilde{a}_0^{(0)}$ is gained as the result of a FFT using the first N terms. The remaining columns are created as follows

$$\tilde{a}_{r+1}^{(s)} = \tilde{a}_{r-1}^{(s+1)} + [\tilde{a}_r^{(s+1)} - \tilde{a}_r^{(s)}]^{-1}, \quad (1.6)$$

$r, s = 0, 1, 2, \dots$. Then the sequence of the successive approximations $\tilde{a}_0^{(0)}, \tilde{a}_2^{(0)}, \tilde{a}_4^{(0)}, \dots$ converges usually much more quickly than the original sequence of partial sums. Thus starting the computation with $2P + 1$ partial sums the $\tilde{a}_{2P}^{(0)}$ term is the required result of the ϵ -algorithm. However, this algorithm can be numerically unstable if P is chosen too big. From many experiments it seems $P = 2$ or 3 are good choices.

1.1.3. Scalar, Vector and Matrix NILT Versions

Based on the theory above three versions of the NILT algorithm will be considered as follows [21,22].

A. Scalar version

If the transform to be inverted has the scalar form $F(s)$ just as can be seen in (1.1) then expressing above operations in a vector form the eq. (1.2) can be rewritten as

$$\tilde{\mathbf{f}}^M = \mathbf{C}^M \circ \{2 \operatorname{Re}[\mathbb{E}\{FFT(\mathbf{F}^N)\}] - \mathbf{F}_0^M\}, \quad (1.7)$$

where particular vectors of upper indexed lengths are created according to (1.3), namely for $k = 0, 1, \dots, M-1$, $n = 0, 1, \dots, N-1$. Especially \mathbf{F}_0^M is the M -element constant vector of the values $F(c)$. The $\mathbb{E}\{\cdot\}$ designates an operator of the ε -algorithm (including the operation of $N \rightarrow M$ vector length reduction), and the symbol \circ means so-called Hadamard product of matrices (in Matlab language called as the element-by-element product). In this case the ε -algorithm runs in parallel on all the elements of the vector and it is programmed inside a two-dimensional array (matrix) utilizing Matlab language parallelized array operations effectively.

B. Vector version

In the case when the transform has the form of a vector $\mathbf{F}^J(s) = [F_1(s), F_2(s), \dots, F_J(s)]^T$ then an approximate formula in a matrix form corresponding to (1.2) can be written as

$$\tilde{\mathbf{f}}^{J \times M} = \mathbf{C}^{J \times M} \circ \{2 \operatorname{Re}[\mathbb{E}\{FFT_{\langle 2 \rangle}(\mathbf{F}^{J \times N})\}] - \mathbf{F}_0^{J \times M}\}, \quad (1.8)$$

where all the terms are matrices of upper indexed sizes computed according to (1.3), but formed for all the components $F_j(s)$, $j = 1, 2, \dots, J$, of the vector $\mathbf{F}^J(s)$. The subscript $\langle 2 \rangle$ means that the FFT operation runs along the 2nd dimension (columns) but in parallel for all the rows. In this case the ε -algorithm runs in parallel on all the matrix elements when programmed inside a three-dimensional array. The computation runs faster compared to a component-wise inversion by means of the scalar algorithm running within an outer loop structure.

C. Matrix version

In the case when the transform has the form of a matrix

$$\mathbf{F}^{J \times L}(s) = \begin{bmatrix} F_{11}(s) & F_{12}(s) & \cdots & F_{1L}(s) \\ F_{21}(s) & F_{22}(s) & \cdots & F_{2L}(s) \\ \vdots & \vdots & & \vdots \\ F_{J1}(s) & F_{J2}(s) & \cdots & F_{JL}(s) \end{bmatrix},$$

then an approximate formula can be written by means of the three-dimensional arrays of

upper indexed sizes as

$$\tilde{\mathbf{f}}^{J \times M \times L} = \mathbf{C}^{J \times M \times L} \circ \{2 \operatorname{Re}[\mathbb{E}\{FFT(\mathbf{F}^{J \times N \times L})\}] - \mathbf{F}_0^{J \times M \times L}\}. \quad (1.9)$$

Here the FFT operation runs along the second dimension, but in parallel over the first and the third dimensions along which the transform matrix to be inverted is situated. The respective 3D arrays are again created according to (1.3). Because the ε -algorithm must now act on the values stored in the three-dimensional array it runs inside a four-dimensional array to ensure the computation in parallel saving the CPU time considerably.

1.1.4. Experimental Error Analysis

As it was experimentally verified the method can successfully be applied to invert very large class of Laplace transforms $F(s)$. This covers not only rational but also irrational and transcendental functions including transforms leading to originals delayed in time. In all the experiments the relative error under 10^{-10} was possible to achieve with the exception of the origin of time interval and a vicinity of discontinuities. The desired relative error can be adjusted according to (1.4) in advance. Three examples are presented below in Tab.1.1 and Fig.1.2.

Tab.1.1 Laplace transforms with their originals

i	1	2	3
$F_i(s)$	$1/\sqrt{s^2+1}$	$e^{-\sqrt{s}}/s$	e^{-s}/s
$f_i(t)$	$J_0(t)$	$\operatorname{erfc}(1/(2\sqrt{t}))$	$1(t-1)$

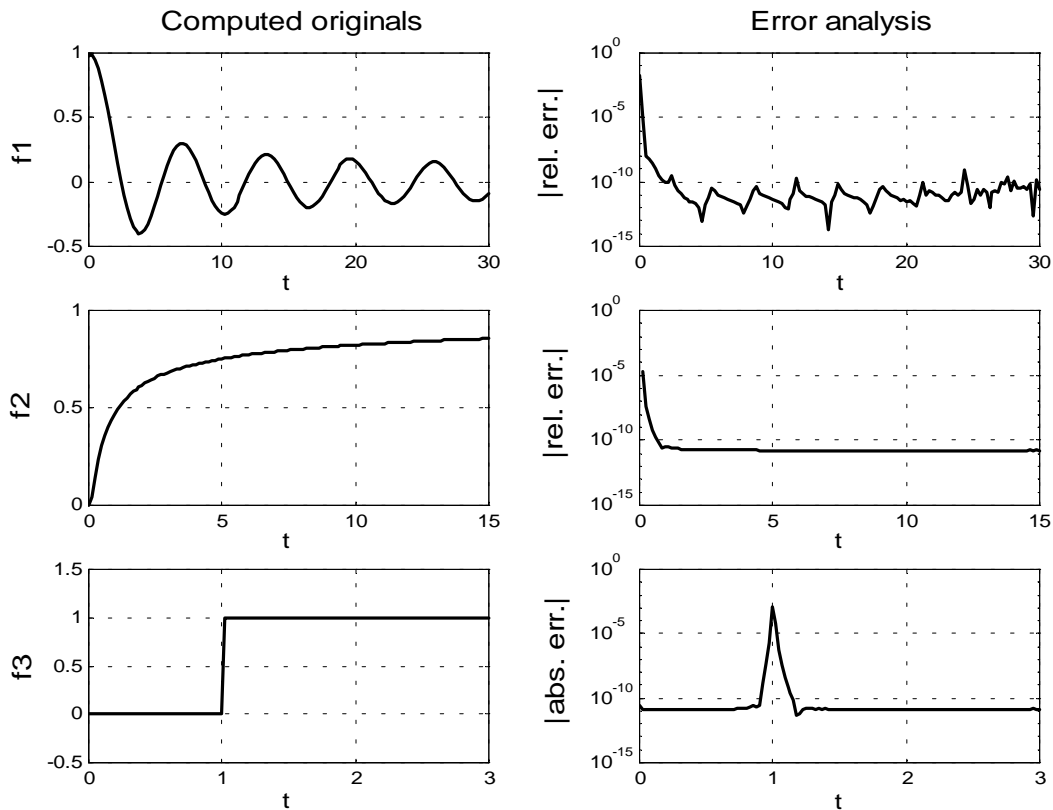


Fig.1.2 Computed originals with determined errors (M=128, P=3)

1.2. TWO-DIMENSIONAL NILT METHOD

1.2.1. Theoretical Base

Consider the definition formula of the 2D-ILT in the form of the improper double-integral as

$$f(t_1, t_2) = -\frac{1}{4\pi^2} \int_{c_1 - j\infty}^{c_1 + j\infty} \int_{c_2 - j\infty}^{c_2 + j\infty} e^{s_1 t_1 + s_2 t_2} F(s_1, s_2) ds_1 ds_2, \quad (1.10)$$

under basic assumption $|f(t_1, t_2)| < Ke^{\alpha_1 t_1 + \alpha_2 t_2}$, with K, α_1, α_2 as positive real constants, and $F(s_1, s_2)$ defined on a region $\{[s_1, s_2] \in \mathbf{C}^2 : \text{Re}[s_1] > \alpha_1 \wedge \text{Re}[s_2] > \alpha_2\}$, see [30] for details. The aim is to evaluate the double-integral numerically not only accurately enough but also to do it fast for a whole given 2D original region on a grid of discrete points. Then following results in [23] the discrete form of the approximate formula can be derived after a trapezoidal rule of the integration and some arrangements were applied as

$$\begin{aligned} \tilde{f}(k_1 T_1, k_2 T_2) = C^{k_1, k_2} \{ & 2 \text{Re} [\sum_{n_1=0}^{\infty} \sum_{n_2=0}^{\infty} F_{-n_1, -n_2} E_{-n_1, -n_2}^{k_1, k_2} + \sum_{n_1=0}^{\infty} (\sum_{n_2=0}^{\infty} F_{-n_1, n_2} E_{n_2}^{k_2}) E_{-n_1}^{k_1} - \\ & - \sum_{n_1=0}^{\infty} F_{-n_1, 0} E_{-n_1}^{k_1} - \sum_{n_2=0}^{\infty} F_{0, -n_2} E_{-n_2}^{k_2}] + F_{0,0} \} \end{aligned}, \quad (1.11)$$

with $k_i = 0, 1, \dots, N_i - 1$, $i = 1, 2$. For simplicity new designations were introduced as

$$\begin{aligned} F_{n_1, n_2} = F(c_1 + jn_1 \Omega_1, c_2 + jn_2 \Omega_2), \quad E_{n_1, n_2}^{k_1, k_2} = e^{jk_1 T_1 n_1 \Omega_1 + jk_2 T_2 n_2 \Omega_2} = E_{n_1}^{k_1} E_{n_2}^{k_2}, \\ C^{k_1, k_2} = \frac{\Omega_1 \Omega_2}{4\pi^2} e^{c_1 k_1 T_1 + c_2 k_2 T_2} = C^{k_1} C^{k_2}, \end{aligned} \quad (1.12)$$

with $\Omega_i = 2\pi/N_i T_i$ and T_i , $i = 1, 2$, as sampling periods in transform and original domains, respectively. As was shown in [23] the eq. (1.11) corresponds to two-dimensional Fourier series approximation of the original $f(t_1, t_2)$, and theoretically the error can be controlled on the region $\langle 0, N_1 T_1 \rangle \times \langle 0, N_2 T_2 \rangle$. Also the approximate formula for choosing c_i coefficients was derived in the form

$$c_i \approx \alpha_i - \frac{\ln(E_r/2)}{N_i T_i}, \quad i = 1, 2, \quad (1.13)$$

where E_r denotes the desired relative error. In practice, however, the error increases rapidly near far ends of 2D original region. Therefore the required maximum 2D interval is supposed to be $\langle 0, t_{1m} \rangle \times \langle 0, t_{2m} \rangle$, with $t_{im} = (M_i - 1)/T_i$, and $M_i = N_i/2$, $i = 1, 2$, are numbers of resultant computed points. This leads to the conditions of choosing $\Omega_i = \pi(1 - 1/M_i)/t_{im}$, $i = 1, 2$. To minimize error towards its theoretical value the infinite sums in (1.11) must be evaluated as accurately as possible.

1.2.2. FFT and ϵ -Algorithm Application

To be able to evaluate (1.11) very fast only $N_i = 2^{m_i}$, m_i integer, $i=1,2$, terms will first be summed. In such a case the incomplete sums for $n_i = 0,1,\dots,N_i - 1$, $i=1,2$, can be calculated using the FFT algorithm to save the CPU time considerably. This is enabled due to the integration steps which were chosen properly and led to the exponential terms according to (1.12). Besides the ϵ -algorithm taking next terms above N_i , $i=1,2$ **Chyba! Chybné propojení.**, into account is integrated into the computational process to give a precision to the result. Expressing these operations in the matrix form the approximate formula can be written as

$$\hat{\mathbf{f}}^{k_1,k_2} = \mathbf{C}^{k_1,k_2} \circ \{2 \operatorname{Re}[\hat{\mathbf{G}}_{12} + \hat{\mathbf{G}}_{21} - \hat{\mathbf{G}}_{11} - \hat{\mathbf{G}}_{22}] + \mathbf{G}_{00}\} . \quad (1.14)$$

Here all terms are $M_1 \times M_2$ matrices. The $\hat{\mathbf{f}}^{k_1,k_2}$ matrix corresponds to (1.11), the \mathbf{C}^{k_1,k_2} matrix is defined by (1.12) and can be calculated as

$$\mathbf{C}^{k_1,k_2} = \mathbf{C}^{k_2} \otimes \mathbf{C}^{k_1} , \quad (1.15)$$

where \mathbf{C}^{k_1} and \mathbf{C}^{k_2} are $M_1 \times 1$ and $1 \times M_2$ matrices (column and row vectors), respectively, and \otimes means Kronecker tensor product of matrices. The \mathbf{G}_{00} matrix is calculated as follows

$$\mathbf{G}_{00} = F_{0,0} [\mathbf{I}_{M_2} \otimes \mathbf{I}_{M_1}] , \quad (1.16)$$

where $F_{0,0} = F(c_1, c_2)$ is given by the eq. (1.12), and \mathbf{I}_{M_1} and \mathbf{I}_{M_2} are $M_1 \times 1$ and $1 \times M_2$ matrices (column and row vectors) whose all elements are equal to 1. Further, the $\hat{\mathbf{G}}_{ij}$, $i, j=1,2$, $i \neq j$, matrices result from the operations as follows

$$\mathbb{E} \{ \underset{\langle 1 \rangle}{FFT}(\mathbf{F}_{-n_1, n_2}) \} \rightarrow \hat{\mathbf{G}}_1 \quad \text{and} \quad \mathbb{E} \{ \underset{\langle 2 \rangle}{FFT}(\hat{\mathbf{G}}_1^*) \} \rightarrow \hat{\mathbf{G}}_{12} , \quad (1.17)$$

$$\mathbb{E} \{ \underset{\langle 2 \rangle}{FFT}(\mathbf{F}_{-n_1, -n_2}) \} \rightarrow \hat{\mathbf{G}}_2 \quad \text{and} \quad \mathbb{E} \{ \underset{\langle 1 \rangle}{FFT}(\hat{\mathbf{G}}_2) \} \rightarrow \hat{\mathbf{G}}_{21} , \quad (1.18)$$

where \mathbf{F}_{-n_1, n_2} and $\mathbf{F}_{-n_1, -n_2}$ are matrices created according to (1.12), the subscript $\langle k \rangle$, $k=1,2$, means that the N_k -points FFT operation is performed along the k^{th} dimension running in parallel over all the remaining ones. Such a choosing of a specific dimension is enabled in Matlab language environment where no loops are needed to perform these FFT operations. The symbol $*$ designates the operation of complex conjugation, the symbol $\mathbb{E}\{\}$ represents an operator of the ϵ -algorithm (including the operation of $N \rightarrow M$ matrix dimension reduction) which is applied on the result of each FFT operation, see later for details. Finally, the $\hat{\mathbf{G}}_{ii}$, $i=1,2$, matrices are computed according to formulae

$$\hat{\mathbf{G}}_{11} = \mathbf{I}_{M_2} \otimes \hat{\mathbf{G}}_1^{(:,1)} \quad (1.19) \quad \text{and} \quad \hat{\mathbf{G}}_{22} = \mathbf{I}_{M_1} \otimes \hat{\mathbf{G}}_2^{(1,:)} , \quad (1.20)$$

where $\hat{\mathbf{G}}_1^{(:,1)}$ and $\hat{\mathbf{G}}_2^{(1,:)}$ are the 1st column and the 1st row of matrices $\hat{\mathbf{G}}_1$ and $\hat{\mathbf{G}}_2$ according to equations (1.17) and (1.18), respectively. In the eq. (1.14) the symbol \circ designates the Hadamard product of matrices (in Matlab language called as the element-by-element product). Using the ϵ -algorithm the matrices resulted from particular FFT

operations are made more accurate as if they were computed using many more terms than only N_i , $i=1,2$. Here all components of the lozenge diagram in Fig.1.1 are matrices whose orders depend on which matrix is to be treated (eqs. (1.17), (1.18)). More detailed explanations about these procedures can be found in [24]. Again the number of terms entering into the ϵ -algorithm should not be chosen too big to avoid numerical instability.

Two versions of 2D-NILT procedure have been programmed in the Matlab language – the scalar version intended to invert simple 2D Laplace transforms $F(s_1, s_2)$ and the vector version enabling to invert transforms $\mathbf{F}^J(s_1, s_2) = [F_1(s_1, s_2), F_2(s_1, s_2), \dots, F_J(s_1, s_2)]^T$ in parallel. To guarantee the high speed of computation three- and four-dimensional arrays were utilized when programming scalar and vector 2D-NILT procedures, respectively.

1.2.3. Experimental Error Analysis

The experiments have proven the method can successfully be used to invert relatively large class of Laplace transforms $F(s_1, s_2)$. This covers not only rational but also irrational and transcendental functions including transforms leading to shifted originals. For many transforms the relative errors about 10^{-8} can be achieved with the exception of the origins of the original 2D interval and vicinities of discontinuities. The desired relative error can be adjusted according to (1.13) in advance. Two examples are presented in Fig.1.3 (the errors are expressed in logarithmic scales).

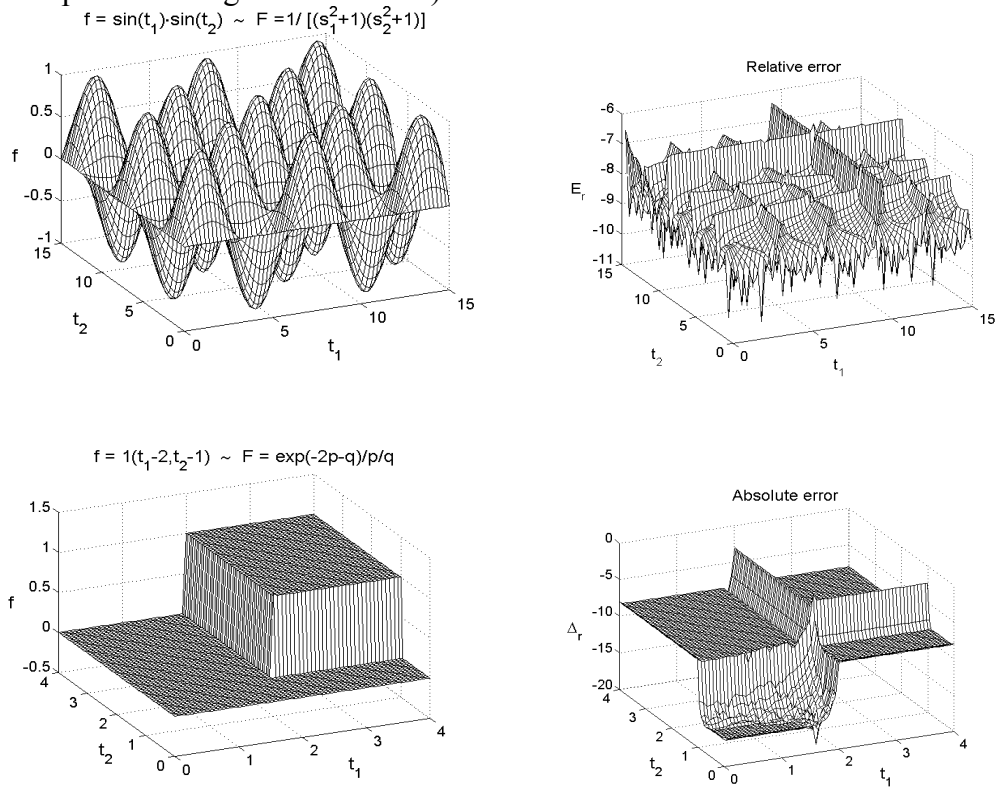


Fig.1.3 Computed originals with determined errors ($M_1=M_2=256$, $P=2$)

2. METHODS OF TIME-DOMAIN SIMULATION OF TRANSMISSION LINES BASED ON ONE-DIMENSIONAL LAPLACE TRANSFORMATION

In the first part the attention will be focussed on MTLs terminated at both ends with linear networks that can be modeled by the generalized Thévenin or Norton equivalents. Generally lossy, nonuniform and frequency dependent MTLs under nonzero initial voltage and/or current distributions can be considered. The aim is to be able to determine not only voltage and/or current waveforms at the MTL's ends but also to show how the waves are propagating along respective wires of the MTL [45-51].

In the second part whole linear systems containing MTLs as their sections are analysed. Here a method based on the modified nodal admittance (MNA) matrix equation is used while the same general cases of MTLs as mentioned above can be considered [52,53].

2.1. BASIC MTL EQUATIONS AND THEIR SOLUTION

Consider $(n+1)$ -conductor transmission line of the length l with per-unit-length $n \times n$ matrices $\mathbf{R}_0(x)$, $\mathbf{L}_0(x)$, $\mathbf{G}_0(x)$, $\mathbf{C}_0(x)$, i.e. the MTL can be nonuniform in general. The MTL equations represent a set of $2n$, coupled, first-order, partial differential equations with varying coefficients which can be expressed in a compact matrix form as [32]

$$\frac{\partial}{\partial x} \begin{bmatrix} \mathbf{v}(x,t) \\ \mathbf{i}(x,t) \end{bmatrix} = \begin{bmatrix} \mathbf{0} & -\mathbf{R}_0(x) \\ -\mathbf{G}_0(x) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{v}(x,t) \\ \mathbf{i}(x,t) \end{bmatrix} - \begin{bmatrix} \mathbf{0} & \mathbf{L}_0(x) \\ \mathbf{C}_0(x) & \mathbf{0} \end{bmatrix} \cdot \frac{\partial}{\partial t} \begin{bmatrix} \mathbf{v}(x,t) \\ \mathbf{i}(x,t) \end{bmatrix}, \quad (2.1)$$

where $\mathbf{v}(x,t)$ and $\mathbf{i}(x,t)$ are $n \times 1$ column vectors of voltage and current waveforms of the n active conductors at a distance x from MTL's left end, respectively, $\mathbf{0}$ means $n \times n$ zero matrix. After Laplace transformation with respect to t is performed and possible frequency dependences of per-unit-length matrices are incorporated the MTL equations take the form

$$\frac{d}{dx} \begin{bmatrix} \mathbf{V}(x,s) \\ \mathbf{I}(x,s) \end{bmatrix} = \begin{bmatrix} \mathbf{0} & -\mathbf{Z}(x,s) \\ -\mathbf{Y}(x,s) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{V}(x,s) \\ \mathbf{I}(x,s) \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{L}_0(x,s) \\ \mathbf{C}_0(x,s) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{v}(x,0) \\ \mathbf{i}(x,0) \end{bmatrix}, \quad (2.2)$$

where $\mathbf{v}(x,0)$ and $\mathbf{i}(x,0)$ are column vectors of initial voltage and current distributions, respectively, and further

$$\mathbf{Z}(x,s) = \mathbf{R}_0(x,s) + s\mathbf{L}_0(x,s) \quad (2.3) \quad , \quad \mathbf{Y}(x,s) = \mathbf{G}_0(x,s) + s\mathbf{C}_0(x,s) \quad (2.4)$$

denote per-unit-length series impedance and shunt admittance matrices, respectively. Obviously, the eq. (2.2) can be written formally as

$$\frac{d}{dx} \mathbf{W}(x,s) = \mathbf{M}(x,s)\mathbf{W}(x,s) + \mathbf{N}(x,s)\mathbf{w}(x,0) \quad , \quad (2.5)$$

with the solution [58-60]

$$\mathbf{W}(x,s) = \Phi_{x_0}^x(s)\mathbf{W}(x_0,s) + \int_{x_0}^x \Phi_{\xi}^x(s)\mathbf{N}(\xi,s)\mathbf{w}(\xi,0)d\xi \quad . \quad (2.6)$$

In the last equation $\Phi_{x_0}^x(s)$ is so-called integral matrix (matrizant) which can be expressed as the infinite series

$$\Phi_{x_0}^x(s) = \mathbf{E} + \int_{x_0}^x \mathbf{M}(\xi_1, s) d\xi_1 + \sum_{i=1}^{\infty} \int_{x_0}^x \mathbf{M}(\xi_1, s) \int_{x_0}^{\xi_1} \mathbf{M}(\xi_2, s) \int_{x_0}^{\xi_2} \mathbf{M}(\xi_3, s) \cdots \int_{x_0}^{\xi_i} \mathbf{M}(\xi_{i+1}, s) d\xi_{i+1} d\xi_i \cdots d\xi_1 \quad (2.7)$$

where \mathbf{E} means the identity matrix of order $2n$, $\mathbf{W}(x_0, s)$ is the solution at $x = x_0$. In the case of a uniform MTL the matrix \mathbf{M} is independent of the variable x and the integral matrix becomes the transition matrix expressed by means of a matrix exponential function as

$$\Phi_{x_0}^x(s) \Big|_{\mathbf{M}(x,s)=\mathbf{M}(s)} = \ddot{\mathbf{O}}(x - x_0, s) = e^{\mathbf{M}(s)[x-x_0]} \quad (2.8)$$

In general case, however, only an approximate integral matrix can be calculated. It can be made by dividing the interval of the variable x into several subintervals supposing \mathbf{M} to be constant in each of them. As the integral matrix (2.7) can now be written in the form

$$\Phi_{x_0}^x(s) = \Phi_{x_{m-1}}^x(s) \Phi_{x_{m-2}}^{x_{m-1}}(s) \cdots \Phi_{x_1}^{x_2}(s) \Phi_{x_0}^{x_1}(s) \quad (2.9)$$

where m is the number of subintervals, then taking the formula (2.8) into account the approximate integral matrix can be expressed as

$$\tilde{\Phi}_{x_0}^x(s) = \prod_{k=1}^m e^{\mathbf{M}(\zeta_k, s) \Delta x_k} \quad (2.10)$$

where $\Delta x_k = x_k - x_{k-1}$, $\zeta_k \in (x_{k-1}, x_k)$, $k = 1, 2, \dots, m$, and $x_m = x$.

In practical computations it can be more advantageous to use a recurrent formula as

$$\tilde{\Phi}_0^{x_k}(s) = e^{\mathbf{M}(\zeta_k, s) \Delta x_k} \cdot \tilde{\Phi}_0^{x_{k-1}}(s) \quad \text{with} \quad \tilde{\Phi}_0^0(s) = \mathbf{E} \quad (2.11)$$

Besides, to speed up the calculation only first two terms of Taylor series of the matrix exponential function can be taken into account. This leads to the alternative formula

$$\hat{\Phi}_{x_0}^x(s) = \prod_{k=1}^m [\mathbf{E} + \mathbf{M}(\zeta_k, s) \Delta x_k] \quad (2.12)$$

or in a recurrent form as

$$\hat{\Phi}_0^{x_k}(s) = [\mathbf{E} + \mathbf{M}(\zeta_k, s) \Delta x_k] \cdot \hat{\Phi}_0^{x_{k-1}}(s) \quad \text{with} \quad \hat{\Phi}_0^0(s) = \mathbf{E} \quad (2.13)$$

Considering the limit case $m \rightarrow \infty$ the equation (2.12) leads to so-called product-integral introduced by Volterra [58]. Thus besides the formula (2.7) the integral matrix can also be expressed as

$$\Phi_{x_0}^x(s) = \lim_{m \rightarrow \infty} \tilde{\Phi}_{x_0}^x(s) = \lim_{m \rightarrow \infty} \hat{\Phi}_{x_0}^x(s) = \int_{x_0}^x [\mathbf{E} + \mathbf{M}(x, s) dx] \quad (2.14)$$

The application of the recurrent formula (2.11) is also convenient for uniform MTLs when waves propagation along the line is simulated. In such a case the transition matrix $\Phi(x - x_0, s)$ is often evaluated for a whole set $\Delta x_k = x_k - x_0 = k\Delta x$ with a regular step Δx . Then the matrix exponential function (2.8) can be applied only once to determine

$$\Phi(\Delta x, s) = e^{\mathbf{M}(s) \Delta x} \quad (2.15)$$

and the recurrent formula has the form as

$$\Phi(k\Delta x, s) = \Phi(\Delta x, s) \cdot \Phi((k-1)\Delta x, s), \quad k = 1, 2, \dots, m, \quad \Phi(0, s) = \mathbf{E}. \quad (2.16)$$

2.2. SIMULATION OF WAVE PROPAGATION IN SIMPLE MULTICONDUCTOR TRANSMISSION LINE SYSTEMS

2.2.1. Generalized Thévenin and Norton Equivalent Application

Suppose a simple linear system containing an MTL terminated at both ends with linear networks according to Fig.2.1.

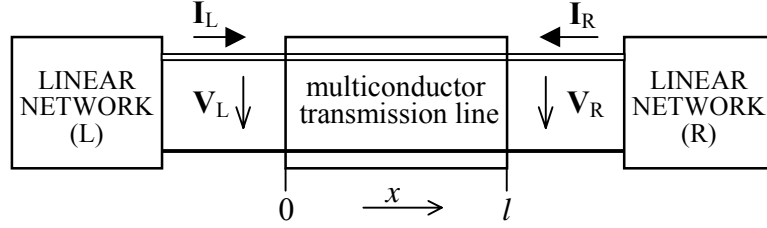


Fig.2.1 Simple multiconductor transmission line system

To incorporate boundary conditions into the solution (2.6) the terminating networks are regarded as multiports and modeled with their generalized Thévenin or Norton equivalents [32]. Then after Laplace transformation the boundary conditions can be expressed as

$$\mathbf{V}_{L,R}(s) = \mathbf{V}_{iL,R}(s) - \mathbf{Z}_{iL,R}(s)\mathbf{I}_{L,R}(s) \quad (2.17)$$

or

$$\mathbf{I}_{L,R}(s) = \mathbf{I}_{iL,R}(s) - \mathbf{Y}_{iL,R}(s)\mathbf{V}_{L,R}(s) \quad (2.18)$$

for Thévenin or Norton equivalent, respectively. Here $\mathbf{V}_i(s)$ and $\mathbf{I}_i(s)$ are $n \times 1$ internal voltage and current vectors, $\mathbf{Z}_i(s)$ and $\mathbf{Y}_i(s)$ are $n \times n$ internal impedance and admittance matrices, and subscripts L and R correspond to the left and right MTL's ends, respectively.

Putting $x_0 = 0$ (MTL's left end) and denoting ${}^{ij}\Phi_0^l(s)$, $i, j = 1, 2$, as square submatrices of the whole integral matrix $\Phi_0^l(s)$ the voltage and current vectors at $x=l$ (MTL's right end) can be determined according to (2.6) in a decomposed form as

$$\begin{bmatrix} \mathbf{V}_R(s) \\ -\mathbf{I}_R(s) \end{bmatrix} = \begin{bmatrix} {}^{11}\Phi_0^l(s) & {}^{12}\Phi_0^l(s) \\ {}^{21}\Phi_0^l(s) & {}^{22}\Phi_0^l(s) \end{bmatrix} \cdot \begin{bmatrix} \mathbf{V}_L(s) \\ \mathbf{I}_L(s) \end{bmatrix} + \begin{bmatrix} \mathbf{V}_0^l(s) \\ \mathbf{I}_0^l(s) \end{bmatrix}, \quad (2.19)$$

where

$$\mathbf{V}_0^l(s) = \int_0^l [{}^{11}\Phi_\xi^l(s)\mathbf{L}_0(\xi, s)\mathbf{i}(\xi, 0) + {}^{12}\Phi_\xi^l(s)\mathbf{C}_0(\xi, s)\mathbf{v}(\xi, 0)] d\xi \quad (2.20)$$

and

$$\mathbf{I}_0^l(s) = \int_0^l [{}^{21}\Phi_\xi^l(s)\mathbf{L}_0(\xi, s)\mathbf{i}(\xi, 0) + {}^{22}\Phi_\xi^l(s)\mathbf{C}_0(\xi, s)\mathbf{v}(\xi, 0)] d\xi \quad (2.21)$$

are voltage and current components, respectively, existing due to nonzero initial voltage and/or current distributions on the MTL.

From the equations above, according to equivalents used to model the terminating linear networks the expression for either current or voltage vectors at the left end of the MTL can be derived as follows

Thévenin (left) – Thévenin (right)

$$\mathbf{I}_L(s) = \{ [\mathbf{\Phi}'_0(s) - \mathbf{Z}_{iR}(s) \mathbf{\Phi}'_0(s)] \mathbf{Z}_{iL}(s) + \mathbf{Z}_{iR}(s) \mathbf{\Phi}'_0(s) - \mathbf{\Phi}'_0(s) \}^{-1} \cdot \{ [\mathbf{\Phi}'_0(s) - \mathbf{Z}_{iR}(s) \mathbf{\Phi}'_0(s)] \mathbf{V}_{iL}(s) - \mathbf{V}_{iR}(s) + \mathbf{V}'_0(s) - \mathbf{Z}_{iR}(s) \mathbf{I}'_0(s) \} , \quad (2.22)$$

Norton (left) – Norton (right)

$$\mathbf{V}_L(s) = \{ [\mathbf{\Phi}'_0(s) - \mathbf{Y}_{iR}(s) \mathbf{\Phi}'_0(s)] \mathbf{Y}_{iL}(s) + \mathbf{Y}_{iR}(s) \mathbf{\Phi}'_0(s) - \mathbf{\Phi}'_0(s) \}^{-1} \cdot \{ [\mathbf{\Phi}'_0(s) - \mathbf{Y}_{iR}(s) \mathbf{\Phi}'_0(s)] \mathbf{I}_{iL}(s) + \mathbf{I}_{iR}(s) + \mathbf{I}'_0(s) - \mathbf{Y}_{iR}(s) \mathbf{V}'_0(s) \} , \quad (2.23)$$

Thévenin (left) – Norton (right)

$$\mathbf{I}_L(s) = \{ [\mathbf{\Phi}'_0(s) - \mathbf{Y}_{iR}(s) \mathbf{\Phi}'_0(s)] \mathbf{Z}_{iL}(s) + \mathbf{Y}_{iR}(s) \mathbf{\Phi}'_0(s) - \mathbf{\Phi}'_0(s) \}^{-1} \cdot \{ [\mathbf{\Phi}'_0(s) - \mathbf{Y}_{iR}(s) \mathbf{\Phi}'_0(s)] \mathbf{V}_{iL}(s) + \mathbf{I}_{iR}(s) + \mathbf{I}'_0(s) - \mathbf{Y}_{iR}(s) \mathbf{V}'_0(s) \} , \quad (2.24)$$

Norton (left) – Thévenin (right)

$$\mathbf{V}_L(s) = \{ [\mathbf{\Phi}'_0(s) - \mathbf{Z}_{iR}(s) \mathbf{\Phi}'_0(s)] \mathbf{Y}_{iL}(s) + \mathbf{Z}_{iR}(s) \mathbf{\Phi}'_0(s) - \mathbf{\Phi}'_0(s) \}^{-1} \cdot \{ [\mathbf{\Phi}'_0(s) - \mathbf{Z}_{iR}(s) \mathbf{\Phi}'_0(s)] \mathbf{I}_{iL}(s) - \mathbf{V}_{iR}(s) + \mathbf{V}'_0(s) - \mathbf{Z}_{iR}(s) \mathbf{I}'_0(s) \} . \quad (2.25)$$

The remaining voltage or current vectors are always given by the necessary boundary conditions (2.17) or (2.18), respectively.

2.2.2. Matlab Oriented Computational Method

To solve the voltage and current vectors at a given coordinate x the basic equation (2.6) for $x_0 = 0$ is used in the form

$$\mathbf{W}(x, s) = \mathbf{\Phi}^x(s) \mathbf{W}_L(s) + \mathbf{\Gamma}^x(s) , \quad (2.26)$$

with $\mathbf{W}(x, s) = [\mathbf{V}(x, s), \mathbf{I}(x, s)]^T$, $\mathbf{W}_L(s) = [\mathbf{V}_L(s), \mathbf{I}_L(s)]^T$ and $\mathbf{\Gamma}^x(s) = [\mathbf{V}_0^x(s), \mathbf{I}_0^x(s)]$, where $\mathbf{V}_L(s)$ and $\mathbf{I}_L(s)$ are determined from the equations in the chapter 2.2.1. As it can be seen from (2.22) – (2.25), however, the $\mathbf{V}'_0(s)$ and $\mathbf{I}'_0(s)$ values for the right end of the MTL must be computed first to be able to continue with solving (2.26). Because the integral expression $\mathbf{\Gamma}_0^x(s)$ is that of a convolution type the solution can be done as follows [49,50].

a. A three-dimensional array of a cumulative product of matrices according to (2.11) or (2.13) is computed, with $m = 2^N$, N integer. In the case of a uniform line it is better to use the eqs. (2.15) and (2.16). In the Matlab language a built-in matrix exponential function (marked as **expm**) is executed very effectively using Padé approximation method without necessity to compute eigenvalues and eigenvectors of the matrix. This array is denoted as

$$\mathbf{\Omega}_{\ddot{o}} = \langle 3 \rangle \{ \mathbf{\Phi}_0^{x_k}(s) \}_{k=1}^m , \quad (2.27)$$

with $x_k = k\Delta x$, $x_m = l$ and Δx as a discretization step, the superscript $\langle 3 \rangle$ means that the array is formed along to 3rd dimension. Such a created array is saved and will be used to determine $\Phi_0^x(s)$ in (2.26).

b. After designation $\Psi(x,s) = \mathbf{N}(x,s)\mathbf{w}(x,0)$, see eqs. (2.5) and (2.6), another 3D array is created as follows

$$\Omega_\Psi = \langle 3 \rangle \left\{ \Psi^T(x_k, s) \otimes \mathbf{1}_{2n}^T \right\}_{k=1}^m, \quad (2.28)$$

where $\mathbf{1}_{2n} = [1, 1, \dots, 1]$ is $1 \times 2n$ row vector with all elements equal to 1 and the symbol \otimes means Kronecker tensor product of matrices implemented like **kron** function in the Matlab language.

c. Thus the values of the integral expression $\Gamma_0^x(s)$ in (2.26) can be determined from the array created as follows

$$\Omega_\Gamma = \frac{l}{m} \text{IFFT} \left(\sum_{\langle 2 \rangle} \left(\text{FFT}(\Omega_\Phi) \circ \text{FFT}(\Omega_\Psi) \right) \right). \quad (2.29)$$

Here FFT and IFFT denote the $2m$ -point fast Fourier transformation operation and its inversion, respectively, subscripts $\langle 2 \rangle$ and $\langle 3 \rangle$ determine dimensions along which necessary operations are performed. A symbol \circ means so-called element-by-element product (as it is defined in Matlab language) when corresponding elements of both arrays are multiplied. Also **fft** and **sum** functions rank among the class of built-in functions in the Matlab language.

Finally the m -th column of the array Ω_Γ is the $\Gamma_0^l(s)$ needed in the calculation $\mathbf{V}_L(s)$ or $\mathbf{I}_L(s)$ according to eqs. (2.22) – (2.25). The solution continues with the application of the method for the numerical inversion of Laplace transforms. Especially the matrix form of the NILT function is very convenient to be used, see the chapter 1.1.3.

2.2.3. Examples

Example 2.1 – MTL Driven from External Source

Consider the simple linear system with the uniform (3+1)-conductor transmission line in Fig.2.2.

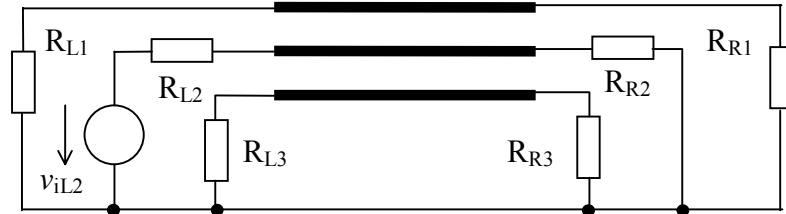


Fig.2.2 The (3+1)-conductor transmission line system
The terminating resistors are $R_{Li} = R_{Ri} = 1\Omega$, $i = 1, 2, 3$. The MTL has the length $l = 1m$, the per-unit-

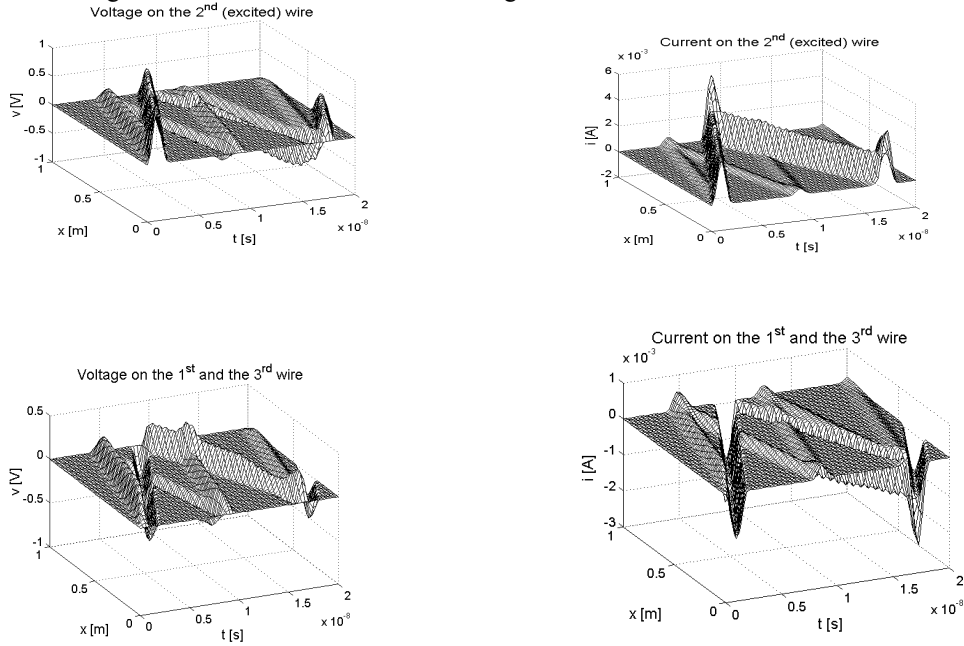
length matrices are [35] $\mathbf{R}_0 = \begin{bmatrix} 41.7 & 0 & 0 \\ 0 & 41.7 & 0 \\ 0 & 0 & 41.7 \end{bmatrix} \frac{\hat{v}}{m}$, $\mathbf{L}_0 = \begin{bmatrix} 2.4 & 0.69 & 0.64 \\ 0.69 & 2.36 & 0.69 \\ 0.64 & 0.69 & 2.4 \end{bmatrix} \frac{\hat{i}H}{m}$

$$\mathbf{G}_0 = \begin{bmatrix} 0.6 & 0 & 0 \\ 0 & 0.6 & 0 \\ 0 & 0 & 0.6 \end{bmatrix} \frac{\text{mS}}{\text{m}}, \quad \mathbf{C}_0 = \begin{bmatrix} 21 & -12 & -4 \\ -12 & 26 & -12 \\ -4 & -12 & 21 \end{bmatrix} \frac{\text{pF}}{\text{m}}.$$

The input voltage source driving the central wire of the MTL has the waveform $v_{iL2}(t) = \sin^2(\pi/2 \cdot 10^{-9})$ if $0 \leq t \leq 2 \cdot 10^{-9}$, $v_{iL2}(t) = 0$ otherwise, with the Laplace transform

$$V_{iL2}(s) = \frac{1 - \exp(-2 \cdot 10^{-9} s)}{2s \cdot [(10^{-9} s/\pi)^2 + 1]}.$$

The resultant voltage and current waves are shown in Fig.2.3.



Example 2.2 – MTL Excited with Nonzero Initial Voltage Distribution

Consider the same MTL system as in the *Example 2.1* with the exceptions as follows: $R_{Li} = R_{Ri} = 10k\Omega$, $i = 1, 2, 3$, and $v_{iL2}(t) = 0$. In this case the MTL is excited with the nonzero initial voltage distribution on its central wire of the form $v_2(x, 0) = \sin^2(\pi[3x/l - 1])$ if $l/3 \leq x \leq 2l/3$, and $v_2(x, 0) = 0$ otherwise. On the remaining MTL's wires zero initial voltage and current distributions are considered. The resultant voltage and current waves are shown in Fig.2.4.

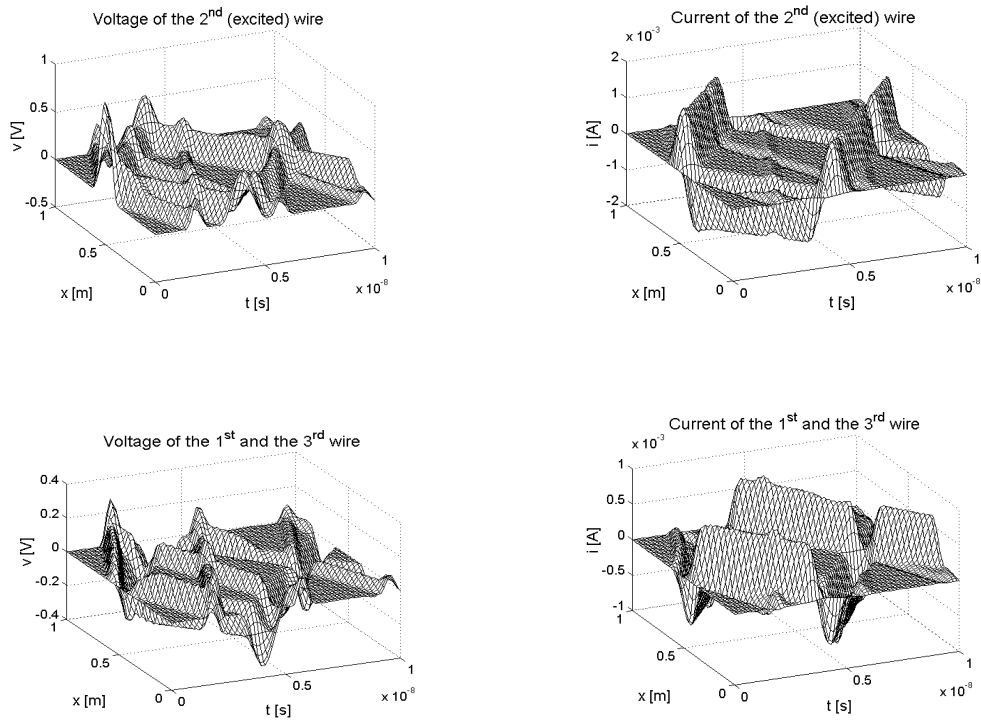


Fig.2.4 Voltage and current waves on the wires

Example 2.3 – Parabolic RG-21 Cable Transient Response Analysis Consider RG-21 cable with the parameters $l = 96.8m$, $R_0 = 0.35\Omega/m$, $G_0 = 0S/m$, $L_0 = 265nH/m$ and $C_0 = 94.34pF/m$ [36]. The parabolic inhomogeneity and frequency dependence due to the skin effect are involved as $Z(x, s) = [R_0 + sL_0 + K\sqrt{s}] \cdot (1 + ax)^2$, $Y(x, s) = sC_0 / (1 + ax)^2$. There is considered $a = (\sqrt{2} - 1)/l$ with $K = 2.5 \cdot 10^{-4} \Omega\sqrt{s}/m$ or $K = 0$ (frequency independent cable). The cable is driven with a unit-step voltage $\mathbf{1}(t)$ and terminated with $R_{iL} = R_{iR} = 53\Omega$. The voltage transient responses are shown in Fig.2.5.

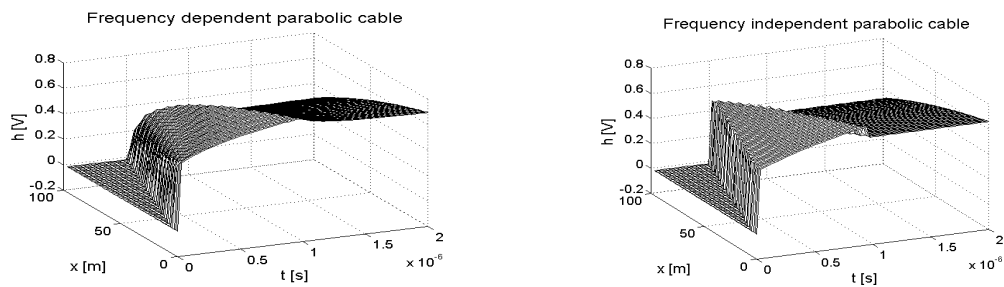


Fig.2.5 Transient response of frequency dependent/independent cab

2.3. TIME-DOMAIN SIMULATION OF NETWORKS CONTAINING SECTIONS WITH MULTICONDUCTOR TRANSMISSION LINES

One of the method which is general enough to be used to analyse networks containing sections with MTLs is the modified nodal admittance (MNA) matrix equation method [42,43]. The MNA matrix equation will be formulated to enable to take the general case of MTLs into account, i.e. the MTLs can be regarded as nonuniform and under nonzero initial voltage and/or current distributions.

2.3.1. MNA Matrix Equation Formulation

Consider a linear network containing a section with lumped-parameter components and P multiconductor transmission lines, see Fig.2.6.

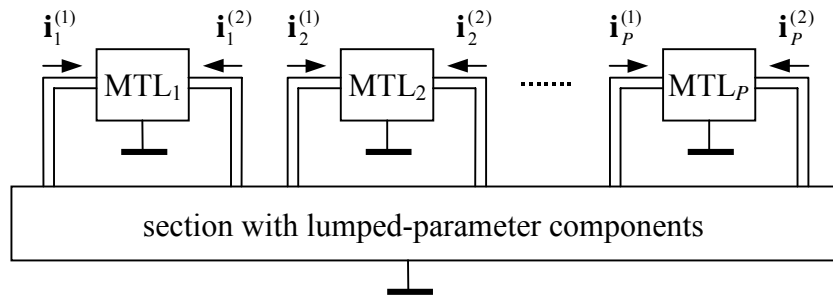


Fig.2.6 Linear network containing MTLs

According to [43] the modified nodal admittance matrix equation in the time domain has the form

$$\mathbf{C}_M \frac{d\mathbf{v}_M(t)}{dt} + \mathbf{G}_M \mathbf{v}_M(t) + \sum_{k=1}^P \mathbf{D}_k \mathbf{i}_k(t) = \mathbf{i}_M(t) \quad , \quad (2.30)$$

where \mathbf{C}_M and \mathbf{G}_M are $N \times N$ constant matrices with entries determined by the lumped memory and memoryless components respectively, $\mathbf{v}_M(t)$ is the $N \times 1$ vector of node voltages appended by currents of independent voltage sources and inductors, $\mathbf{i}_M(t)$ is the $N \times 1$ vector of source waveforms, $\mathbf{i}_k(t)$ is the $n_k \times 1$ vector of currents entering the k -th MTL, and \mathbf{D}_k is the $N \times n_k$ selector matrix with entries $d_{i,j} \in \{0,1\}$ mapping the vector $\mathbf{i}_k(t)$ into the node space of the network. After Laplace transformation the frequency-domain representation has the form

$$[\mathbf{G}_M + s\mathbf{C}_M] \mathbf{V}_M(s) + \sum_{k=1}^P \mathbf{D}_k \mathbf{I}_k(s) = \mathbf{I}_M(s) + \mathbf{C}_M \mathbf{v}_M(0) \quad (2.31)$$

The MTLs consist of $N_k = n_k/2$ active conductors, i.e. they can be regarded as $2N_k$ -ports. Then the $\mathbf{I}_k(s)$ in (2.31) is formed to contain vectors of currents entering the input and output ports as $\mathbf{I}_k(s) = [\mathbf{I}_k^{(1)}(s), \mathbf{I}_k^{(2)}(s)]^T$ and will be expressed from the basic MTL matrix equation (2.6). In terms of the multiport theory the integral matrix $\Phi_0^l(s)$ acts as the chain matrix $\Phi(s)$. Thus denoting $\mathbf{W}(0,s) = [\mathbf{V}^{(1)}(s), \mathbf{I}^{(1)}(s)]^T$, $\mathbf{W}(l,s) = [\mathbf{V}^{(2)}(s), -\mathbf{I}^{(2)}(s)]^T$ and $\Gamma_0^l(s) = [\mathbf{V}^{(\Gamma)}(s), \mathbf{I}^{(\Gamma)}(s)]^T$ the k -th MTL can be described by the equation

$$\begin{bmatrix} \mathbf{V}_k^{(2)}(s) \\ -\mathbf{I}_k^{(2)}(s) \end{bmatrix} = \begin{bmatrix} \Phi_{11}(s) & \Phi_{12}(s) \\ \Phi_{21}(s) & \Phi_{22}(s) \end{bmatrix}_k \cdot \begin{bmatrix} \mathbf{V}_k^{(1)}(s) \\ \mathbf{I}_k^{(1)}(s) \end{bmatrix} + \begin{bmatrix} \mathbf{V}_k^{(\Gamma)}(s) \\ \mathbf{I}_k^{(\Gamma)}(s) \end{bmatrix}. \quad (2.32)$$

After some manipulations the admittance equations taking nonzero initial conditions into account can be written as [53]

$$\begin{bmatrix} \mathbf{I}_k^{(1)}(s) \\ \mathbf{I}_k^{(2)}(s) \end{bmatrix} = \begin{bmatrix} \mathbf{Y}_{11}(s) & \mathbf{Y}_{12}(s) \\ \mathbf{Y}_{21}(s) & \mathbf{Y}_{22}(s) \end{bmatrix}_k \cdot \begin{bmatrix} \mathbf{V}_k^{(1)}(s) \\ \mathbf{V}_k^{(2)}(s) \end{bmatrix} - \begin{bmatrix} \mathbf{Y}_{12}(s) & \mathbf{0} \\ \mathbf{Y}_{22}(s) & \mathbf{E} \end{bmatrix}_k \cdot \begin{bmatrix} \mathbf{V}_k^{(\Gamma)}(s) \\ \mathbf{I}_k^{(\Gamma)}(s) \end{bmatrix}, \quad (2.33)$$

where $\mathbf{Y}_{11}(s) = -\Phi_{12}^{-1}(s)\Phi_{11}(s)$, $\mathbf{Y}_{22}(s) = -\Phi_{22}(s)\Phi_{12}^{-1}(s)$, $\mathbf{Y}_{12}(s) = \Phi_{12}^{-1}(s)$, and due to the reciprocity of the MTL, $\mathbf{Y}_{21}(s) = \mathbf{Y}_{12}^T(s)$. Besides in the case of the uniform MTL the equality $\mathbf{Y}_{11}(s) = \mathbf{Y}_{22}(s)$ would also be valid. Further $\mathbf{0}$ and \mathbf{E} mean the zero and identity matrix, respectively. In a compact matrix form the equation (2.33) can be written as

$$\mathbf{I}_k(s) = \mathbf{Y}_k(s)\mathbf{V}_k(s) - \mathbf{X}_k(s)\Gamma_k(s). \quad (2.34)$$

Finally after substituting the equation (2.34) into the (2.31) the resultant MNA matrix equation can be written in the form

$$\mathbf{V}_M(s) = \left[\mathbf{G}_M + s\mathbf{C}_M + \sum_{k=1}^P \mathbf{D}_k \mathbf{Y}_k(s) \mathbf{D}_k^T \right]^{-1} \left[\mathbf{I}_M(s) + \mathbf{C}_M \mathbf{v}_M(0) + \sum_{k=1}^P \mathbf{D}_k \mathbf{X}_k(s) \Gamma_k(s) \right]. \quad (2.35)$$

In this case the method for the numerical inversion of Laplace transforms in the vector form according to the chapter 1.1.3. is very useful.

2.3.2. Examples

Example 2.4 – MTL Network Driven with External Source

Consider a linear network with three uniform (2+1)-conductor transmission lines in Fig.2.7 [37,43].

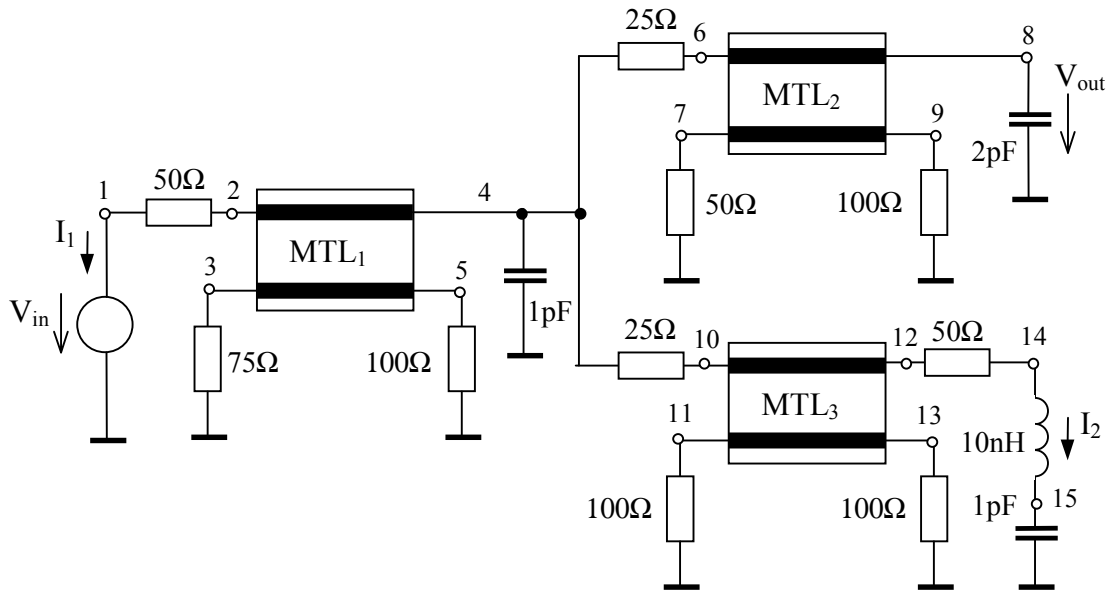


Fig.2.7 Linear network with three uniform MTLs

The MTLs differ only in their lengths as follows: $l_1 = 0.05m$, $l_2 = 0.04m$ and $l_3 = 0.03m$, per-unit-length matrices for all three MTLs are

$$\mathbf{R}_0 = \begin{bmatrix} 75 & 15 \\ 15 & 75 \end{bmatrix} \frac{\Omega}{m}, \mathbf{L}_0 = \begin{bmatrix} 494.6 & 63.3 \\ 63.3 & 494.6 \end{bmatrix} \frac{nH}{m}, \mathbf{G}_0 = \begin{bmatrix} 0.1 & -0.01 \\ -0.01 & 0.1 \end{bmatrix} \frac{S}{m}, \mathbf{C}_0 = \begin{bmatrix} 62.8 & -4.9 \\ -4.9 & 62.8 \end{bmatrix} \frac{pF}{m}$$

As it is marked in Fig.2.7, 15 node voltages and 2 currents are the variables to be solved by the MNA method, i.e. the system of 17 equations is solved. Using the Matlab language the frequency-domain solution $\mathbf{V}_M(s)$ according to (2.35) has been programmed in the form of the M-file function. Finally this function is called by the **niltv** function for the numerical inversion of vector Laplace transforms, see the chapter 1.1.3. Using this function the time-domain solution $\mathbf{v}_M(t) = \mathcal{L}^{-1}\{\mathbf{V}_M(s)\}$ is computed for all the vector components in parallel saving the CPU time considerably. Running on the PC with the Pentium II 266MHz/64MB the CPU time was under 6 seconds. In Fig.2.8 only the input $v_{in}(t)$ and output $v_{out}(t)$ voltage waveforms are shown as the example.

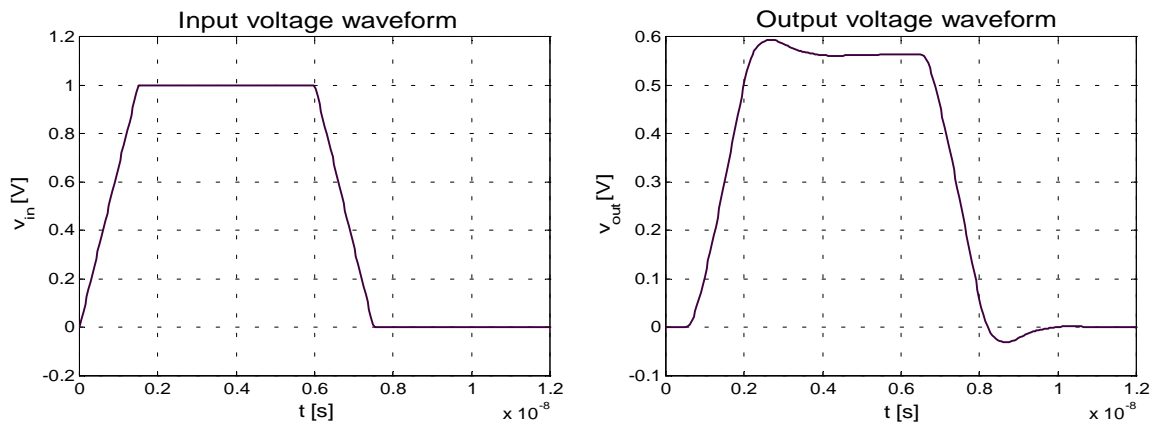


Fig.2.8 Computed input and output voltage waveforms

The obtained results agree very well with those computed by both the NILT method in [37] and the asymptotic waveform evaluation technique (AWE) in [43].

Example 2.5 – MTL Network Excited with MTL's Nonzero Initial Condition

Consider a simple linear network with two identical (2+1)-conductor transmission lines in Fig.2.9 .

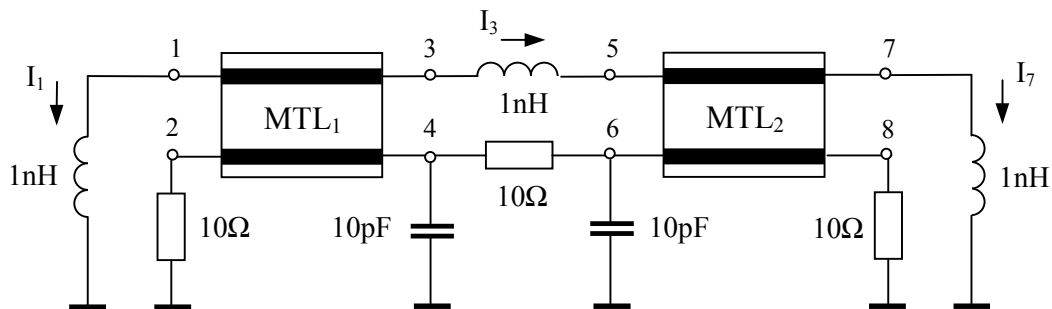


Fig.2.9 Linear network with initially excited MTL₁

The MTLs are uniform of the length $l = 0.05m$ and with the per-unit-length matrices as in the *Example 2.4*. On the first wire of the MTL₁ the nonzero initial voltage distribution exists as follows

$$v_1(x,0) = \sin^2\left(\pi\left[\frac{4x}{l} - \frac{3}{2}\right]\right) \text{ if } \frac{3}{8}l \leq x \leq \frac{5}{8}l, \quad v_1(x,0) = 0 \text{ otherwise,}$$

while the MTL₂ is considered under zero initial conditions. As it is shown in Fig.2.9, 8 node voltages and 3 currents are the variables to be solved by the MNA method, i.e. the system is described by 11 equations. In the Matlab language the frequency-domain solution $\mathbf{V}_M(s)$ according to (2.35) has been programmed in the form of the M-file function which is called by the **niltv** function for the numerical inversion of vector Laplace transforms as is described in the chapter 1.1.3. Using the PC with the Pentium II 266MHz/64MB the CPU time was about one minute. In this case the computation is much more time-consuming compared to the *Example 2.4* as the matrix convolution integral must be evaluated, see the chapter 2.2.2. In Fig.2.10 there are presented some node voltage and current waveforms as examples.

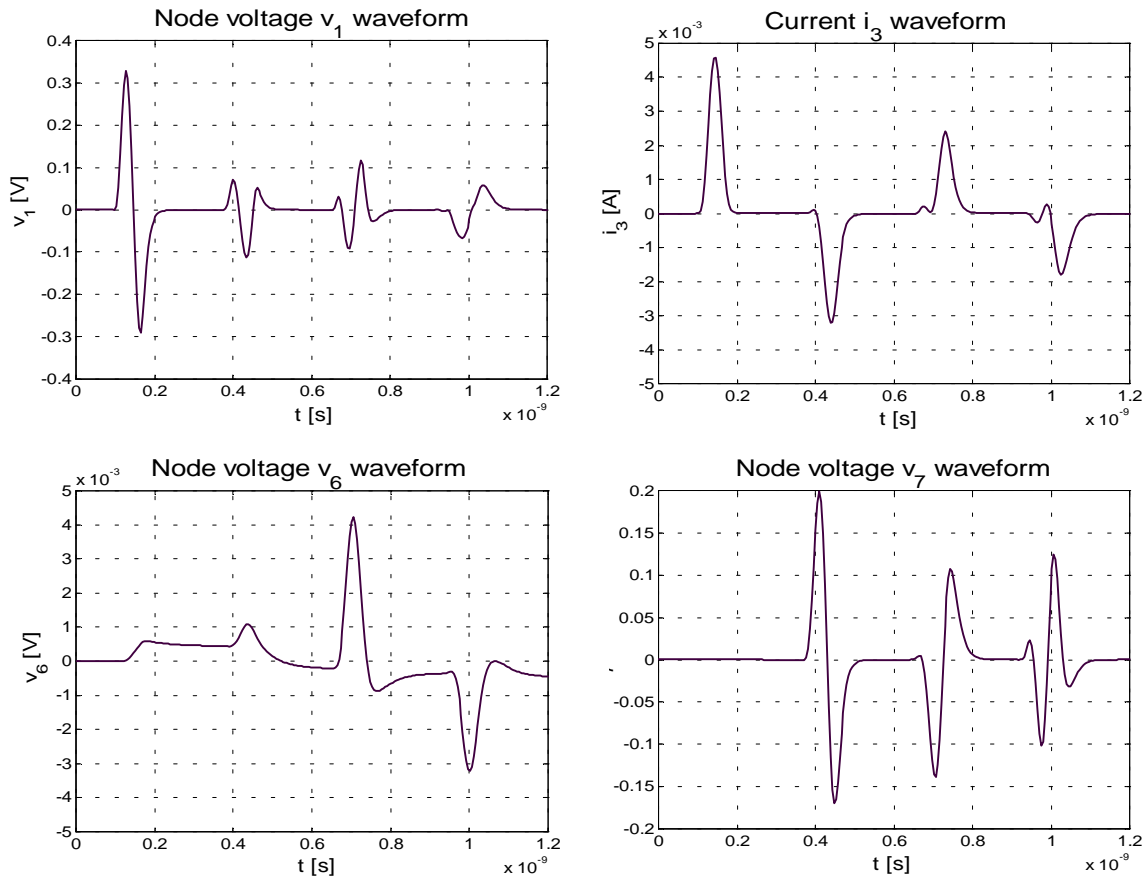


Fig.2.10 Computed node voltage and current waveforms

The results have been verified by comparing results obtained by the method based exclusively on the multiport theory when the cascade connection of multiports was considered and terminating networks of the MTLs were modeled with the generalized Norton equivalents, see the chapter 2.2.1.

3. METHODS OF TIME-DOMAIN SIMULATION OF TRANSMISSION LINES BASED ON TWO-DIMENSIONAL LAPLACE TRANSFORMATION

In the case when the aim is to determine not only the voltage and current waveforms at the ends of the line but also to show how the waves propagate along the line it can be more advantageous to use the method based on the two-dimensional Laplace transformation. Thus two partial Laplace transformations are performed – with respect to time t and spatial coordinate x – to transform the transmission line's partial differential equations into the algebraic ones. Then taking boundary conditions into account two-dimensional transforms in (q,s) -domain are derived and subsequently a method for the numerical inversion of Laplace transforms in two variables is applied to obtain the solution in (x,t) -domain. To be able to use this method generally, however, only uniform transmission lines have to be considered [54-57].

3.1. BASIC MTL EQUATIONS AND THEIR TRANSFORMATIONS

Consider a uniform $(n+1)$ -conductor transmission line of a length l with per-unit-length $n \times n$ matrices \mathbf{R}_0 , \mathbf{L}_0 , \mathbf{G}_0 , \mathbf{C}_0 . The MTL equations can be expressed in compact matrix form as [32]

$$\frac{\partial}{\partial x} \begin{bmatrix} \mathbf{v}(x,t) \\ \mathbf{i}(x,t) \end{bmatrix} = \begin{bmatrix} \mathbf{0} & -\mathbf{R}_0 \\ -\mathbf{G}_0 & \mathbf{0} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{v}(x,t) \\ \mathbf{i}(x,t) \end{bmatrix} - \begin{bmatrix} \mathbf{0} & \mathbf{L}_0 \\ \mathbf{C}_0 & \mathbf{0} \end{bmatrix} \cdot \frac{\partial}{\partial t} \begin{bmatrix} \mathbf{v}(x,t) \\ \mathbf{i}(x,t) \end{bmatrix}, \quad (3.1)$$

where $\mathbf{v}(x,t)$ and $\mathbf{i}(x,t)$ are $n \times 1$ column vectors of instantaneous voltages and currents of n active conductors at a distance x from MTL's left end, respectively, $\mathbf{0}$ denotes $n \times n$ zero matrix. After performing partial Laplace transformation with respect to time t and taking frequency dependences of primary parameters into account the equation (3.1) leads to

$$\frac{d}{dx} \begin{bmatrix} \mathbf{V}(x,s) \\ \mathbf{I}(x,s) \end{bmatrix} = \begin{bmatrix} \mathbf{0} & -\mathbf{Z}(s) \\ -\mathbf{Y}(s) & \mathbf{0} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{V}(x,s) \\ \mathbf{I}(x,s) \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{L}_0(s) \\ \mathbf{C}_0(s) & \mathbf{0} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{v}(x,0) \\ \mathbf{i}(x,0) \end{bmatrix}, \quad (3.2)$$

where

$$\mathbf{Z}(s) = \mathbf{R}_0(s) + s\mathbf{L}_0(s) \quad , \quad \mathbf{Y}(s) = \mathbf{G}_0(s) + s\mathbf{C}_0(s) \quad (3.3)$$

are the per-unit-length series impedance and shunt admittance matrices, respectively, and $\mathbf{v}(x,0)$ and $\mathbf{i}(x,0)$ are column vectors of initial voltage and current distributions, respectively. Obviously, the equation (3.2) can be written formally as

$$\frac{d}{dx} \mathbf{W}(x,s) = \mathbf{M}(s)\mathbf{W}(x,s) + \mathbf{N}(s)\mathbf{w}(x,0) \quad , \quad (3.4)$$

with the solution [58-60]

$$\mathbf{W}(x,s) = e^{\mathbf{M}(s)[x-x_0]} \mathbf{W}(x_0,s) + \int_{x_0}^x e^{\mathbf{M}(s)[x-\xi]} \mathbf{N}(s)\mathbf{W}(\xi,0) d\xi \quad , \quad (3.5)$$

where $\mathbf{W}(x_0,s)$ is the solution given at a coordinate x_0 . Hereafter we put $x_0 = 0$ as the left end of the line is considered at this coordinate. Thus we can write

$$\mathbf{W}(x,s) = \mathbf{\Phi}(x,s)\mathbf{W}(0,s) + \int_0^x \mathbf{\Phi}(x-\xi,s)\mathbf{N}(s)\mathbf{W}(\xi,0) d\xi \quad , \quad (3.6)$$

where $\Phi(x, s)$ denotes a transition matrix. Evidently, the integral expressions in last two equations are those of a convolution type. Therefore after applying partial Laplace transformation with respect to coordinate x we have

$$\mathbf{W}(q, s) = [q\mathbf{E} - \mathbf{M}(s)]^{-1} [\mathbf{W}(0, s) + \mathbf{N}(s)\mathbf{W}(q, 0)] , \quad (3.7)$$

where \mathbf{E} means an identity matrix of the order $2n$. Here a matrix analogy of the well-known relations for transforms of the exponential function and the convolution integral was used. To prove this result by another way let us apply Laplace transformation w. r. to x directly on the equation (3.4) which leads to

$$q\mathbf{W}(q, s) - \mathbf{W}(0, s) = \mathbf{M}(s)\mathbf{W}(q, s) + \mathbf{N}(s)\mathbf{W}(q, 0) , \quad (3.8)$$

and from this the equation (3.7) is already evident [56].

The above used partial Laplace transformations are defined as

$$\mathbb{L}_t[\mathbf{f}(x, t)] = \int_0^{\infty} \mathbf{f}(x, t)e^{-st} dt \quad (3.9) , \quad \mathbb{L}_x[\mathbf{f}(x, t)] = \int_0^l \mathbf{f}(x, t)e^{-qx} dx \quad (3.10)$$

as the MTL has a finite length l and for $x > l$ the function $\mathbf{f}(x, t)$ is supposed to be zero.

3.2. BOUNDARY CONDITIONS INCORPORATION

Consider a simple MTL system according to Fig.2.1 as is shown in the chapter 2. The eq. (3.7) can be written in a decomposed form as

$$\begin{bmatrix} \mathbf{V}(q, s) \\ \mathbf{I}(q, s) \end{bmatrix} = \begin{bmatrix} q\mathbf{E} & \mathbf{Z}(s) \\ \mathbf{Y}(s) & q\mathbf{E} \end{bmatrix}^{-1} \left\{ \begin{bmatrix} \mathbf{V}_L(s) \\ \mathbf{I}_L(s) \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{L}_0(s) \\ \mathbf{C}_0(s) & \mathbf{0} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{V}_0(q) \\ \mathbf{I}_0(q) \end{bmatrix} \right\} , \quad (3.11)$$

where $\mathbf{V}_L(s) = \mathbf{V}(0, s)$ and $\mathbf{I}_L(s) = \mathbf{I}(0, s)$ are coupled by the equations (2.17) or (2.18), $\mathbf{V}_0(q) = \mathbf{V}(q, 0)$ and $\mathbf{I}_0(q) = \mathbf{I}(q, 0)$ are transforms of initial voltage and current distributions on the MTL, respectively, \mathbf{E} denotes the identity matrix of the order n . Applying rules for the inverse of a partitioned matrix and performing some manipulations the inverse matrix in (3.11) can be expressed as follows [56]

$$\mathbf{D}(q, s) = \begin{bmatrix} q\mathbf{E} & \mathbf{Z}(s) \\ \mathbf{Y}(s) & q\mathbf{E} \end{bmatrix}^{-1} = \begin{bmatrix} q\mathbf{A}(q, s) & -\mathbf{A}(q, s)\mathbf{Z}(s) \\ -\mathbf{B}(q, s)\mathbf{Y}(s) & q\mathbf{B}(q, s) \end{bmatrix} , \quad (3.12)$$

where

$$\mathbf{A}(q, s) = [q^2\mathbf{E} - \mathbf{Z}(s)\mathbf{Y}(s)]^{-1} , \quad \mathbf{B}(q, s) = [q^2\mathbf{E} - \mathbf{Y}(s)\mathbf{Z}(s)]^{-1} . \quad (3.13)$$

Due to the diagonal symmetry of the per-unit-length matrices it is also valid

$$\mathbf{B}(q, s) = \mathbf{A}^T(q, s) . \quad (3.14)$$

Thus taking (3.12) into account the equation (3.11) can be written in the form

$$\begin{bmatrix} \mathbf{V}(q, s) \\ \mathbf{I}(q, s) \end{bmatrix} = \begin{bmatrix} \mathbf{D}_{11}(q, s) & \mathbf{D}_{12}(q, s) \\ \mathbf{D}_{21}(q, s) & \mathbf{D}_{22}(q, s) \end{bmatrix} \cdot \begin{bmatrix} \mathbf{V}_L(s) \\ \mathbf{I}_L(s) \end{bmatrix} + \begin{bmatrix} \mathbf{V}_0(q, s) \\ \mathbf{I}_0(q, s) \end{bmatrix} , \quad (3.15)$$

where

$${}^{\Gamma} \mathbf{V}_0(q, s) = \mathbf{D}_{11}(q, s) \mathbf{L}_0(s) \mathbf{I}_0(q) + \mathbf{D}_{12}(q, s) \mathbf{C}_0(s) \mathbf{V}_0(q) , \quad (3.16)$$

$${}^{\Gamma} \mathbf{I}_0(q, s) = \mathbf{D}_{21}(q, s) \mathbf{L}_0(s) \mathbf{I}_0(q) + \mathbf{D}_{22}(q, s) \mathbf{C}_0(s) \mathbf{V}_0(q) . \quad (3.17)$$

When the partial inverse Laplace transformation of the equation (3.15) w. r. to x is performed and $x = l$ substituted the result is

$$\begin{bmatrix} \mathbf{V}_R(s) \\ -\mathbf{I}_R(s) \end{bmatrix} = \begin{bmatrix} \Phi_{11}(s) & \Phi_{12}(s) \\ \Phi_{21}(s) & \Phi_{22}(s) \end{bmatrix} \cdot \begin{bmatrix} \mathbf{V}_L(s) \\ \mathbf{I}_L(s) \end{bmatrix} + \begin{bmatrix} \mathbf{V}'_0(s) \\ \mathbf{I}'_0(s) \end{bmatrix} , \quad (3.18)$$

where $\Phi_{ij}(s)$, $i, j=1,2$, are square submatrices of the whole transition matrix $\Phi(s) = \Phi(l, s)$ as it follows from the equation (3.6). As the transition matrix is only the special case of the integral matrix the terms

$$\mathbf{V}'_0(s) = \mathbb{L}_x^{-1} \left[{}^{\Gamma} \mathbf{V}_0(q, s) \right]_{x=l} \quad (3.19) \quad \text{and} \quad \mathbf{I}'_0(s) = \mathbb{L}_x^{-1} \left[{}^{\Gamma} \mathbf{I}_0(q, s) \right]_{x=l} \quad (3.20)$$

can be expressed from (2.20) and (2.21), respectively, replacing ${}^{ij} \Phi'_0(s)$ with $\Phi_{ij}(s)$, i.e.

$$\mathbf{V}'_0(s) = \int_0^l [\Phi_{11}(l-\xi, s) \mathbf{L}_0(s) \mathbf{i}(\xi, 0) + \Phi_{12}(l-\xi, s) \mathbf{C}_0(s) \mathbf{v}(\xi, 0)] d\xi , \quad (3.21)$$

$$\mathbf{I}'_0(s) = \int_0^l [\Phi_{21}(l-\xi, s) \mathbf{L}_0(s) \mathbf{i}(\xi, 0) + \Phi_{22}(l-\xi, s) \mathbf{C}_0(s) \mathbf{v}(\xi, 0)] d\xi . \quad (3.22)$$

Similarly according to terminating equivalents the eqs. (2.22) – (2.25) can be utilized to express either current or voltage vectors at the left end of the MTL as

Thévenin (left) – Thévenin (right)

$$\begin{aligned} \mathbf{I}_L(s) = \{ & [\Phi_{11}(s) - \mathbf{Z}_{iR}(s) \Phi_{21}(s)] \mathbf{Z}_{iL}(s) + \mathbf{Z}_{iR}(s) \Phi_{22}(s) - \Phi_{12}(s) \}^{-1} \\ & \cdot \{ [\Phi_{11}(s) - \mathbf{Z}_{iR}(s) \Phi_{21}(s)] \mathbf{V}_{iL}(s) - \mathbf{V}_{iR}(s) + \mathbf{V}'_0(s) - \mathbf{Z}_{iR}(s) \mathbf{I}'_0(s) \} \} , \end{aligned} \quad (3.23)$$

Norton (left) – Norton (right)

$$\begin{aligned} \mathbf{V}_L(s) = \{ & [\Phi_{22}(s) - \mathbf{Y}_{iR}(s) \Phi_{12}(s)] \mathbf{Y}_{iL}(s) + \mathbf{Y}_{iR}(s) \Phi_{11}(s) - \Phi_{21}(s) \}^{-1} \\ & \cdot \{ [\Phi_{22}(s) - \mathbf{Y}_{iR}(s) \Phi_{12}(s)] \mathbf{I}_{iL}(s) + \mathbf{I}_{iR}(s) + \mathbf{I}'_0(s) - \mathbf{Y}_{iR}(s) \mathbf{V}'_0(s) \} \} , \end{aligned} \quad (3.24)$$

Thévenin (left) – Norton (right)

$$\begin{aligned} \mathbf{I}_L(s) = \{ & [\Phi_{21}(s) - \mathbf{Y}_{iR}(s) \Phi_{11}(s)] \mathbf{Z}_{iL}(s) + \mathbf{Y}_{iR}(s) \Phi_{12}(s) - \Phi_{22}(s) \}^{-1} \\ & \cdot \{ [\Phi_{21}(s) - \mathbf{Y}_{iR}(s) \Phi_{11}(s)] \mathbf{V}_{iL}(s) + \mathbf{I}_{iR}(s) + \mathbf{I}'_0(s) - \mathbf{Y}_{iR}(s) \mathbf{V}'_0(s) \} \} , \end{aligned} \quad (3.25)$$

Norton (left) – Thévenin (right)

$$\begin{aligned} \mathbf{V}_L(s) = \{ & [\Phi_{12}(s) - \mathbf{Z}_{iR}(s) \Phi_{22}(s)] \mathbf{Y}_{iL}(s) + \mathbf{Z}_{iR}(s) \Phi_{21}(s) - \Phi_{11}(s) \}^{-1} \\ & \cdot \{ [\Phi_{12}(s) - \mathbf{Z}_{iR}(s) \Phi_{22}(s)] \mathbf{I}_{iL}(s) - \mathbf{V}_{iR}(s) + \mathbf{V}'_0(s) - \mathbf{Z}_{iR}(s) \mathbf{I}'_0(s) \} \} . \end{aligned} \quad (3.26)$$

Finally the $\mathbf{V}_L(s)$ and $\mathbf{I}_L(s)$ vectors – given completely by the respective equation above and boundary condition (2.17) or (2.18) – are substituted into the eq. (3.15).

3.3. COMPUTATIONAL METHODS

To find the solution in the whole two-dimensional (x,t) -domain the equation (3.15) has to be inverted using a procedure for two-dimensional numerical inversion of Laplace transforms (2D-NILT). In the case of the MTL's zero initial conditions the method seems to be straightforward and simple. To be able to take MTL's nonzero initial conditions into account, however, either the procedure for partial numerical inversion $\mathcal{L}_x^{-1}[\cdot]$ according to (3.19) and (3.20) must be applied or the convolution integrals according to (3.21) and (3.22) must be precomputed. The first method can be faster if this runs as a part of the 2D-NILT algorithm.

As is shown in [56], however, another method can be proposed to be usable in the case of MTL's nonzero initial conditions. In this method voltage and current 2D Laplace transform vectors are expressed in the unified form as

$$\mathbf{F}(q,s) = \mathbf{F}_1(q,s) \cdot \mathcal{L}_x^{-1}[\mathbf{F}_0(q,s)]_{x=l} + \mathbf{F}_2(q,s) , \quad (3.27)$$

where detailed forms of $\mathbf{F}_i(q,s)$, $i = 0,1,2$, depend on the terminating equivalents under consideration according to eqs. (3.23)–(3.26). Especially $\mathbf{F}_0(q,s)$ is $n \times 1$ vector depending on the MTL's nonzero initial conditions, $\mathbf{F}_2(q,s)$ is $n \times 1$ vector depending moreover on terminating driving sources and $\mathbf{F}_1(q,s)$ is $n \times n$ matrix which depends only on passive quantities.

Now to find the original $\mathbf{f}(x,t) = \mathcal{L}_{xt}^{-1}[\mathbf{F}(q,s)]$ we can write due to a linearity

$$\mathbf{f}(x,t) = \mathcal{L}_{xt}^{-1}\{\mathbf{F}_1(q,s) \cdot \mathcal{L}_x^{-1}[\mathbf{F}_0(q,s)]_{x=l}\} + \mathcal{L}_{xt}^{-1}[\mathbf{F}_2(q,s)] . \quad (3.28)$$

Taking a basic property of the 2-D Laplace transformation into account as [30]

$$\mathcal{L}_{xt}^{-1}[\mathbf{F}(q,s)] = \mathcal{L}_x^{-1}\{\mathcal{L}_t^{-1}[\mathbf{F}(q,s)]\} = \mathcal{L}_t^{-1}\{\mathcal{L}_x^{-1}[\mathbf{F}(q,s)]\} , \quad (3.29)$$

we have

$$\mathbf{f}(x,t) = \mathcal{L}_t^{-1}\{\mathcal{L}_x^{-1}[\mathbf{F}_1(q,s)] \cdot \mathcal{L}_x^{-1}[\mathbf{F}_0(q,s)]_{x=l}\} + \mathcal{L}_{xt}^{-1}[\mathbf{F}_2(q,s)] , \quad (3.30)$$

and subsequently

$$\mathbf{f}(x,t) = \mathcal{L}_{xt}^{-1}[\mathbf{F}_1(q,s)] * \mathcal{L}_{xt}^{-1}[\mathbf{F}_0(q,s)]_{x=l} + \mathcal{L}_{xt}^{-1}[\mathbf{F}_2(q,s)] . \quad (3.31)$$

Here the symbol $*$ means the convolution operation with respect to time t . Finally expressing this equation in (x,t) -domain we can write

$$\mathbf{f}(x,t) = \mathbf{f}_1(x,t) * \mathbf{f}_0(l,t) + \mathbf{f}_2(x,t) , \quad (3.32)$$

or

$$\mathbf{f}(x,t) = \mathbf{f}_2(x,t) + \int_0^t \mathbf{f}_1(x,t-\xi) \mathbf{f}_0(l,\xi) d\xi . \quad (3.33)$$

In this method only the basic 2D-NILT procedure is applied three times, and finally the matrix convolution integral is evaluated. This can be performed with the similar method as was described in the chapter 2.2.2.

3.4. EXAMPLE

Consider a simple linear system with the uniform (2+1)-conductor transmission line according to Fig.3.1.

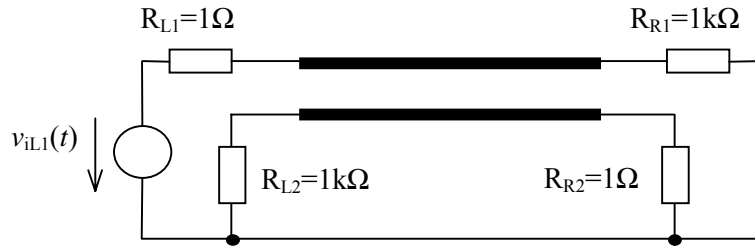


Fig.3.1 The (2+1)-conductor transmission line system

The MTL has the length $l = 0.3m$, and it is considered as lossless ($\mathbf{R}_0 = \mathbf{G}_0 = \mathbf{0}$), with the remaining per-unit-length matrices [34]

$$\mathbf{L}_0 = \begin{bmatrix} 494.6 & 63.3 \\ 63.3 & 494.6 \end{bmatrix} \frac{nH}{m}, \quad \mathbf{C}_0 = \begin{bmatrix} 62.8 & -4.9 \\ -4.9 & 62.8 \end{bmatrix} \frac{pF}{m}.$$

The input voltage waveform is $v_{iL1}(t) = \sin^2(\pi/2 \cdot 10^{-9})$ if $0 \leq t \leq 2 \cdot 10^{-9}$, and $v_{iL1}(t) = 0$ otherwise. The resultant voltage and current wave propagations are computed by the vector form of the 2D-NILT procedure as is described in the chapter 1.2.

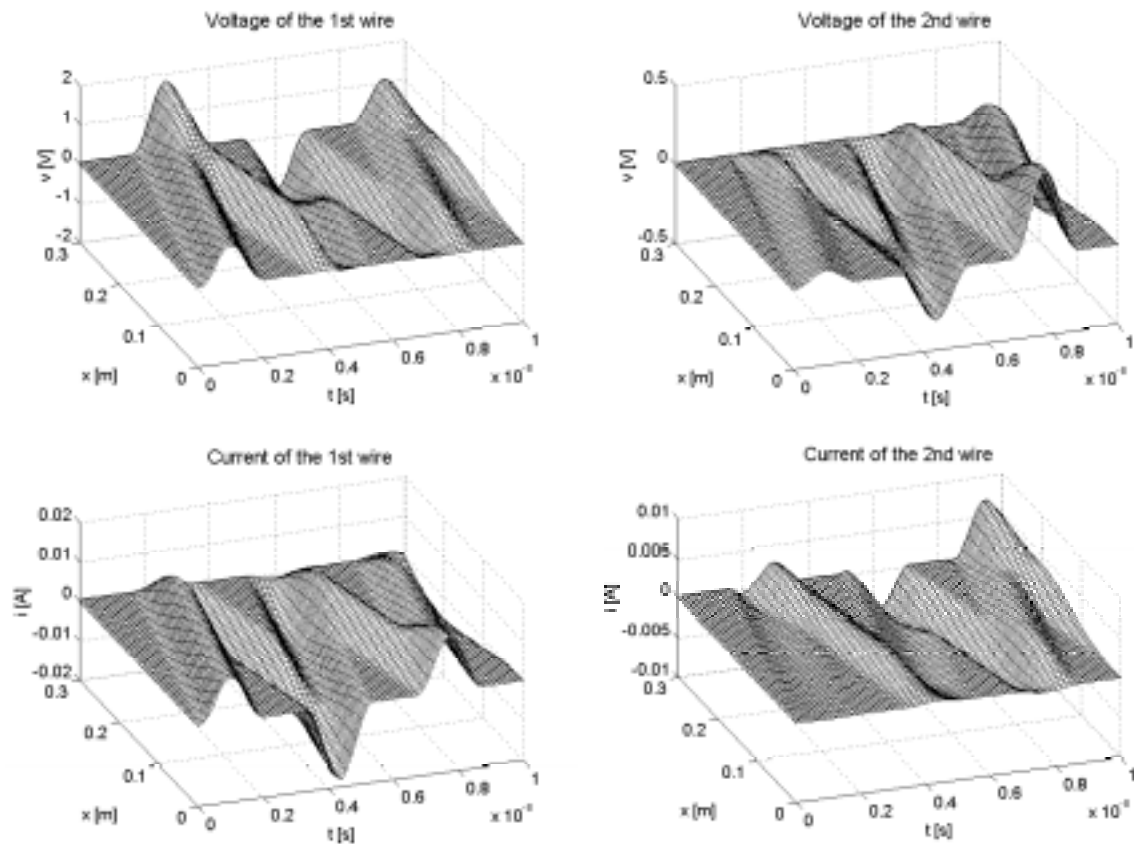


Fig.3.2 Voltage and current waves on the MTL's wires

4. CONCLUSION

On more general level the thesis for habilitation has set the task to contribute to development of the computer simulation of systems with distributed parameters including multiconductor transmission lines. The methods under consideration are intended to simulate linear systems and they are based on the Laplace transformation approach, both one- and two-dimensional.

In concrete terms the thesis is a contribution to two relatively separate areas. Firstly the methods for the numerical inversion of Laplace transforms in both one and two variables were elaborated and new techniques of their computation have been developed. There were worked out two different methods. The first method is based on the approximation of the exponential function in the ILT definition integral with expressions containing hyperbolic functions when, finally, Euler transformation is used to accelerate convergence of infinite series. More explanations about the principle including 1D-NILT method in details are, however, described only in the full habilitation thesis. The generalization to 2D-NILT case was done in [15]. The second method is based on direct numerical integration of the ILT definition integral in conjunction with the FFT and ϵ -algorithm application. It is especially suitable to be used to get the whole required interval in the original domain very fast. Comparing other often used NILT methods and evaluating results of many experiments the developed ones seem to rank among the most versatile, accurate and fast.

Secondly methods of simulation of MTL systems based on both 1D and 2D Laplace transformation were worked out making allowance for the effective algorithmization in the mathematical language Matlab. In the case of the 1D-LT the developed procedures can handle MTLs considered as lossy, frequency dependent, nonuniform, and under nonzero initial voltage and/or current distributions in general. To simulate simple MTL systems the generalized Thévenin or Norton equivalents are applied to model MTL's linear terminating networks. For more complex systems containing MTLs as their sections a technique based on the modified nodal admittance matrix equation is used. For the purposes of taking nonzero initial conditions into account the novel Matlab oriented FFT-based method of matrix convolution integral calculation was developed. Using advanced Matlab's matrix exponential function in the solution the eigenvalues and eigenvectors need not be found as it is required in well-known modal analysis technique. For the case of uniform MTLs a method based on the 2D-LT were worked out. The advantage of the method lies in the possibility to obtain the solution on the whole 2D region in a single calculation step. Beside of the direct 2D-LT method a novel approach has been proposed when certain 2D transforms are inverted separately, and finally the results are convolved with respect to time to get the resultant 2D original.

In the field of NILT techniques future effort can be oriented to find more efficient and numerically stable methods of accelerating convergence of infinite series. Especially for the case of the FFT-based NILT methods a quotient-difference algorithm of Rutishauser will be tested to compare with the ϵ -algorithm application. In the field of MTL systems time-domain simulation the aim is to enable to include MTL's nonlinear terminating networks into the solution.

Acknowledgements:

The work on this thesis was supported by Research Programme CEZ: J22/98:262200011.

PRINCIPAL REFERENCES

- [1] Piessens, R.: A Bibliography on Numerical Inversion of the Laplace Transform. TW20, Appl. Math. & Progr., 1974.
- [2] Bellman, R., Kalaba, R. E., Locket, J.: Numerical Inversion of the Laplace Transform: Application to Biology, Economics, Engineering, and Physics. American Elsevier Publishing Company, Inc., New York, 1966.
- [3] Wing, O.: Recent Computer Experiments on the Numerical Inversion of the Laplace transform. In: Summer School on Circuit Theory SSCT'68, Prague 1968, pp. 1-18.
- [4] Zakian, V.: Optimization of Numerical Inversion of Laplace Transforms. Electronic Letters, 6, 1970, pp. 677-679.
- [5] Singhal, K., Vlach, J.: Computation of Time Domain Response by Numerical Inversion of the Laplace Transform. J. of the Franklin Inst., Vol. 299, No. 2, Feb. 1975, pp. 109-126.
- [6] Valsa, J.: One Method of Numerical Inversion of Laplace Transforms (in Czech), Technique of Electrical Machines - Theoretical Issue, Brno, 1975, pp. 9-18.
- [7] Hosono, T.: Numerical Inversion of Laplace Transforms and Some Applications to Wave Optics. Radio Science, Vol. 16, Nr. 6, 1981, pp. 1015-1019.
- [8] Dubner, H., Abate, J.: Numerical Inversion of Laplace Transforms by Relating them to the Finite Fourier Cosine Transform. Journal of the Association for Computing Machinery, Vol. 15, No. 1, January 1968, pp. 115-123.
- [9] Crump, K. S.: Numerical Inversion of Laplace Transforms Using a Fourier Series Approximation. J. ACM, 23, Jan. 1976, pp. 89- 96.
- [10] DeHoog, F. R., Knight, J. H., Stokes, A. N.: An Improved Method for Numerical Inversion of Laplace Transforms. SIAM J. Sci. Stat. Comput., Vol. 3, No. 3, Sept. 1982, pp. 357-366.
- [11] Valsa, J., Brančík, L.: A Comparative Study of Several Algorithms for the Numerical Inversion of Laplace Transforms. In: Proc. of 21th Seminar on Fundamentals of Electro-technics and Circuit Theory SPETO'98, Gliwice-Ustroń, Poland, May 1998, pp. 307-310.
- [12] Berger, B. S.: Inversion of the N-dimensional Laplace Transform. Math. of Comp., 20, 1966, pp. 418-421.
- [13] Singhal, K., Vlach, J., Vlach, M.: Numerical Inversion of Multidimensional Laplace Transform. Proc. of IEEE, 1975, pp. 1627-1628.
- [14] Choudhury, G.L., Lucantoni, D.M., Whitt, W.: Multidimensional Transform Inversion with Applications to the Transient M/G/1 Queue. The Annals of Applied Probability, Vol. 4, No. 3, 1994, pp. 719-740.
- [15] Valsa, J.: Numerical Inversion of Multivariable Laplace Transforms Proc. of 5th EDS'98, Brno, Czech Republic, June 1998, pp. 337-340.
- [16] Hwang, Ch., Lu, M. J.: Numerical Inversion of 2-D Laplace Transforms by Fast Hartley Transform Computations. Journal of the Franklin Institute, 336, 1999, pp. 955-972.
- [17] Valsa, J., Brančík, L.: Approximate Formulae for Numerical Inversion of Laplace Transforms. International Journal of Numerical Modelling: Electronic Networks, Devices and Fields, Vol. 11, No. 3, 1998, pp. 153-166.

- [18] Valsa, J., Brančík, L.: Approximate Formulae for Numerical Inversion of Laplace Transforms. In: Proc. of 20th Seminar on Fundamentals of Electrotechnics and Circuit Theory SPETO'97, Ustroń, Poland, May 1997, pp. 249-254.
- [19] Brančík, L.: The Fast Computing Method of Numerical Inversion of Laplace Transforms Using FFT Algorithm. In: Proc. of 5th Electronic Devices and Systems International Conference EDS'98, Brno, Czech Republic, June 1998, pp. 97-100.
- [20] Brančík, L.: An Improvement of FFT-based Numerical ILT Procedure by Application of ϵ -algorithm. In: Sborník přednášek Moderní směry výuky elektrotechniky a elektroniky STO-7, Brno, září 1999, str. 196-199.
- [21] Brančík, L.: Programs for Fast Numerical Inversion of Laplace Transforms in Matlab Language Environment. In: Sborník 7. roč. MATLAB'99, Praha, listopad 1999, str. 27-39.
- [22] Brančík, L.: Matlab Oriented Numerical Inversion of Matrix Laplace Transforms Applied to Simulation of Multiconductor Transmission Lines. In: Proc. of 5th Scientific-Technical Conference Computer Applications in Electrical Engineering ZkwE'2000, Poznań/Kiekrz, Poland, April 2000, pp. 33-36.
- [23] Brančík, L., Valsa, J.: A Fast Computing Method of Numerical Inversion of Two-Dimensional Laplace Transforms Using FFT Algorithms. In: Proc. of Advances in Systems, Signals, Control and Computers SSCC'98, Vol. II, Durban, South Africa, September 1998, pp. 102-106.
- [24] Brančík, L.: An Improvement of FFT-based Numerical Inversion of Two-Dimensional Laplace Transforms by Means of ϵ -algorithm. In: Proc. of IEEE International Symposium on Circuits and Systems ISCAS'2000, Geneva, Switzerland, May 2000, pp. 581-584.
- [25] Brančík, L.: An Improved Numerical Inversion of Two-Dimensional Laplace Transforms with Application to Transient Analysis of Transmission Lines. Electronic Devices and Systems 1999 - Proceedings, Brno, Czech Republic, 1999, pp. 261-267.
- [26] Brančík, L.: Programs for Fast Numerical Inversion of Two-Dimensional Laplace Transforms in Matlab Language Environment with some Applications. In: Proc. of 23rd Seminar on Fundamentals of Electrotechnics and Circuit Theory SPETO'98, Gliwice-Ustroń, Poland, May 2000, pp. 197-200.
- [27] Wynn, P.: On a device for computing the $e_m(S_n)$ transformation. MTAC, 10, 1956, pp. 91-96.
- [28] Macdonald, J. R.: Accelerated convergence, divergence, iteration, extrapolation, and curve fitting. J. Appl. Phys., 10, 1964, pp. 3034-3041.
- [29] Skelboe, S.: Computation of the Periodic Steady-State Response of Nonlinear Networks by Extrapolation Methods. IEEE Trans. Circuits Syst., Vol. CAS-27, March 1980, pp. 161-175.
- [30] Ditkin, V. A., Prudnikov, A. P.: Operational Calculus in Two Variables and its Applications. Pergamon, New York 1962.
- [31] Faria, J. A. B.: Multiconductor Transmission-Line Structures: Modal Analysis Techniques. John Wiley & Sons, New York, 1993.
- [32] Paul, C. R.: Analysis of Multiconductor Transmission Lines. John Wiley & Sons, New York, 1994.

- [33] Djordjevič, A. R., Sarkar, T. K., Harrington, R. F.: Time-Domain Response of Multiconductor Transmission Lines. Proceedings of the IEEE, Vol. 75, No. 6, June 1987, pp. 743-764.
- [34] Griffith, J. R., Nakhla, M. S.: Time-Domain Analysis of Lossy Coupled Transmission Lines. IEEE Transactions on Microwave Theory and Techniques, Vol. 38, No. 10, Oct. 1990, pp. 1480-1487.
- [35] Chang, E. C., Kang, S. M.: Transient Simulation of Lossy Coupled Transmission Lines Using Iterative Linear Least Square Fitting and Piecewise Recursive Convolution. IEEE Transactions on Circuits and Systems – I: Fundamental Theory and Applications, Vol. 43, No. 11, November 1996, pp. 923-932.
- [36] Chang, F. Y.: Waveform Relaxation Analysis of Nonuniform Lossy Transmission Lines Characterized with Frequency-Dependent Parameters. IEEE Transactions on Circuit and Systems, Vol. 38, No. 12, December 1991.
- [37] Lum, S., Nakhla, M. S., Zhang, Q. J.: Sensitivity Analysis of Lossy Coupled Transmission Lines. IEEE Transactions on Microwave Theory and Techniques, Vol. 39, No. 12, December 1991, pp. 2089-2099.
- [38] Chang, F. Y.: Transient Analysis of Lossy Transmission Lines with Arbitrary Initial Potential and Current Distributions, IEEE Trans. on Circuits and Systems – I, Vol. 39, No. 3, March 1992, pp. 180-198.
- [39] Valsa, J., Brančík, L.: Time Domain Analysis of Lossy Transmission Lines. In: Sborník přednášek mezinárodní vědecké konference VŠB-TU Ostrava, sekce 21 Teoretická elektrotechnika, září 1995, str. 48-51.
- [40] Brančík, L., Valsa, J.: A Critical Comparison of Computer Simulation Methods for Systems Containing Sections of Transmission Lines. In: Proc. of 6th National Scientific Conference with International Participation Radioelektronika'96, Vol. I, Brno, April 1996, pp. 58-61.
- [41] Goel, A. K.: High-Speed VLSI Interconnections: Modeling, Analysis, and Simulation. John Wiley & Sons, New York, 1994.
- [42] Nakhla, M. S., Zhang, Q. J.: Modeling and Simulation of High Speed VLSI Interconnects. A Special Issue of Analog Integrated Circuits and Signal Processing - An International Journal, Vol. 5, No. 1, 1994, Kluwer Academic Publishers, Boston, 1994.
- [43] Chiprout, E., Nakhla, M. S.: Asymptotic Waveform Evaluation and Moment Matching for Interconnect Analysis. Kluwer Academic Publishers, Boston, 1994.
- [44] Radu, S., Onofrei, V., Albulet, M.: Interconnection Problems in Modern Computers. IEEE Design & Test of Computers, Spring 1996, pp. 36-45.
- [45] Brančík, L.: Application of Numerical Inversion of Laplace Transforms in Transmission Line Time-Domain Analysis. In: Proc. of 3rd International Conference on Advanced Methods in the Theory of Electrical Engineering Applied to Power Systems AMTEE'97, Plzeň, September 1997, pp. 213-216.
- [46] Brančík, L.: Aplikace numerické inverzní Laplaceovy transformace při časové analýze a modelování přenosového vedení. In: Sborník přednášek Moderní směry výuky elektrotechniky a elektroniky STO-6, Brno, září 1997, str. 48-51 (in Czech).

- [47] Brančik, L.: Transient Analysis in Multiconductor Transmission Line Systems Using Numerical ILT. In: Proc. of 8th International Czech-Slovak Scientific Conference Radioelektronika'98, Vol. 2, Brno, April 1998, pp. 344-347.
- [48] Brančik, L.: One Possibility How to Compute Voltage and Current Distributions along the Multiconductor Transmission Line. In: Proc. of Workshop CAD&CAE'98, Praha, listopad 1998, str. 6-9.
- [49] Brančik, L.: Time-Domain Simulation of Transmission Lines under Nonzero Initial Conditions Using Convolution Integral and Numerical ILT. In: Proc. of 9th International Czech-Slovak Scientific Conference Radioelektronika'99, Brno, Czech Republic, April 1999, pp. 344-347.
- [50] Brančik, L.: Time-Domain Simulation of Nonuniform Multiconductor Transmission Lines under Nonzero Initial Conditions Using Matlab Language. Proc. of European Conference on Circuit Theory and Design ECCTD'99, Vol. 2, Stresa, Italy, August-September 1999, pp.1135-1138.
- [51] Brančik, L.: Transient Analysis of Nonuniform Lossy Transmission Lines with Frequency-Dependent Parameters. In: Proc. of 4th International Conference on Advanced Methods in the Theory of Electrical Engineering Applied to Power Systems, Plzeň, September 1999, pp. E01-E04.
- [52] Brančik, L.: Time-Domain Simulation of Networks Containing Sections with Distributed Parameters Using Matlab Language. In: Proc. of 10th Int. Czech-Slovak Scientific Conf. Radioelektronika'2000, Bratislava, Slovak Republic, Sept. 2000, pp. III.108-III.111.
- [53] Brančik, L.: Simulation of Transient Phenomena in Networks Containing Initially Excited Multiconductor Transmission Lines. In: Proc. of 23rd International Conference Telecommunications and Signal Processing TSP'2000, Brno, Czech Republic, September 2000, pp. 93-96.
- [54] Valsa, J., Brančik, L.: Time-Domain Simulation of Lossy Transmission Lines with Arbitrary Initial Conditions. In: Proc. of Advances in Systems, Signals, Control and Computers SSCC'98, Vol. III, Durban, South Africa, September 1998, pp. 305-307.
- [55] Valsa, J., Brančik, L., Steinbauer, M.: Simulation of Lumped-Distributed Linear Networks with Nonzero Initial Conditions. In: Zborník prednášok Seminár katedier z oblasti teoretickej elektrotechniky Českej a Slovenskej republiky, Vrátna, Slovenská republika, September 1998, str. 10-12.
- [56] Brančik, L.: Simulation of Transient Phenomena in Distributed Linear Networks by the Two-Dimensional Laplace Transformation. In: Proc. of 22th International Conference on Fundamentals of Electrotechnics and Circuit Theory SPETO'99, Gliwice-Ustroń, Poland, May 1999, pp. 275-278.
- [57] Brančik, L., Steinbauer, M.: Two Approaches how to Simulate Wave Propagation on Transmission Lines under Nonzero Initial Conditions. In: Proc. of 8th Scientific Conference Theoretical Electrical Engineering and Electrical Measurement, Košice, Slovak Republic, September 1999, pp. 22-27.
- [58] Gantmacher, F. R.: The Theory of Matrices. Chelsea, New York, 1977.
- [59] Mayer, D.: Úvod do teorie elektrických obvodů. SNTL/ALFA, Praha 1978 (in Czech).
- [60] Mann, H.: Využití počítače při elektrotechnických návrzích. SNTL/ALFA, Praha 1984 (in Czech).

SOUHRN (ABSTRACT IN CZECH)

Habilitační práce si v obecnější rovině klade za cíl přispět k rozvoji metod počítačové simulace systémů obsahujících části s rozprostřenými parametry, včetně vícevodičových přenosových vedení. Uvažované metody jsou určeny pro simulaci lineárních soustav a jsou založeny na aplikaci Laplaceovy transformace, a to jednorozměrné i dvojrozměrné.

V konkrétní rovině je práce příspěvkem ke dvěma relativně samostatným oblastem. Do první oblasti patří vývoj dostatečně přesných a rychlých metod pro numerickou inverzní Laplaceovu transformaci (NILT), a to funkcí jedné i dvou proměnných. Druhá oblast je pak věnována vývoji algoritmů pro simulaci lineárních soustav s rozprostřenými parametry využívající metod numerické inverze Laplaceových obrazů. Při rozpracování vhodných numerických postupů je přitom přihlíženo k možnostem efektivního využití univerzálního matematického jazyka Matlab.

V habilitační práci jsou rozpracovány dvě v principu odlišné metody NILT. První metoda je založena na aproximaci exponenciální funkce v definičním integrálu Laplaceovy transformace výrazy obsahujícími hyperbolické funkce. Výsledkem jsou nekonečné řady, jejichž konvergenci lze urychlit pomocí Eulerovy transformace. Druhá metoda je založena na přímé numerické integraci Bromwichova integrálu, kdy lze vhodně využít FFT pro částečné vyčíslení vzniklé komplexní Fourierovy řady a ε -algoritmu pro urychlení její konvergence. Obě metody jsou vhodné pro inverzi nejen racionálních, ale i široké třídy iracionálních a transcendentních funkcí, včetně obrazů vedoucích na funkce zpožděné v originální oblasti. V tezi habilitační práce je podrobněji rozvedena pouze druhá z obou metod. Jak ukázaly výsledky mnoha experimentů, tato metoda je obzvláště vhodná pro rychlý a přitom dostatečně přesný výpočet celého intervalu (jedno- i dvojrozměrného) v originální oblasti. Metodu lze efektivně algoritmizovat v jazyku Matlab, především při využití schopností paralelního zpracování dat na vícerozměrných polích. Důsledkem je možnost rychlé inverze obrazů nejen ve skalárním, ale také ve vektorovém i maticovém tvaru. Toho lze výhodně využít právě při simulaci soustav s přenosovými vedeními, včetně vedení vícevodičových.

Ve druhé oblasti habilitační práce jsou nejdříve rozpracovány metody simulace založené na jednorozměrné Laplaceově transformaci. Vyvinuté algoritmy umožňují simulovat přechodné jevy na vícevodičových vedeních ztrátových, frekvenčně závislých, nehomogenních i s nenulovým počátečním rozložením napětí a proudů. Jádrem řešení je přímá maticová metoda výpočtu soustav obyčejných diferenciálních rovnic ve frekvenční oblasti. Pro jednoduché soustavy je využito teorie vícebranů a aplikace zobecněného Nortonova nebo Théveninova teorému pro snadné zahrnutí hraničních podmínek do řešení. U složitějších soustav s vícevodičovými vedeními jako jejich částmi je užito modifikované metody uzlových napětí, která byla rozšířena o možnost zahrnutí počátečních podmínek na vedeních. K tomuto účelu byla v jazyku Matlab vypracována originální metoda výpočtu konvolučního integrálu maticového argumentu založená na aplikaci FFT na trojrozměrných polích. Konečnou fází výpočtu je aplikace 1D-NILT algoritmu pro získání řešení v časové oblasti. Speciálně pro případ homogenních vícevodičových vedení byla vypracována metoda založená na dvojrozměrné Laplaceově transformaci. Zde původně parciální diferenciální rovnice vedení v (x,t) -oblasti jsou transformovány na rovnice algebraické v (q,s) -oblasti. Po zahrnutí hraničních příp. i nenulových počátečních podmínek je aplikací 2D-NILT algoritmu získáno kompletní řešení v (x,t) -oblasti. Pro případ nenulových počátečních podmínek na vedení byl dále teoreticky rozpracován nový postup řešení založený na oddělené inverzi jistých 2D obrazů a následné časové konvoluci výsledků. Metodu 2D Laplaceovy transformace lze s výhodou využít pro účely rychlé demonstrace šíření vln na vedeních. Zvláště ve spojení s jazykem Matlab je možné výsledky snadno zobrazovat v 3D grafech nebo použít pro animaci.

Protože byla habilitační práce předložena jako soubor publikovaných prací, je partie týkající se simulace obvodů obsahujících části s vícevodičovými vedeními (kap. 2.3) do předkládaných tezí aktuálně doplněna.