Matlab Based Time–Domain Simulation of Multiconductor Transmission Line Systems

Lubomír Brančík
Brno University of Technology, Faculty of Electrical Engineering and Communication
Purkyňova 118, 612 00 Brno
Czech Republic
email: brancik@feec.vutbr.cz

Abstract-- **The paper deals with a technique of the time–domain simulation of transient phenomena in linear distributed systems using Matlab language. The distributed parts are formed by multiconductor transmission lines (MTL) which can generally be nonuniform, frequency–dependent, and under nonzero initial conditions. In principle the solution is formulated in terms of a modified nodal admittance (MNA) equation method in the frequency domain. Then an improved fast method of numerical inversion of Laplace transforms (NILT) in vector or matrix form is applied to gain a solution in the time domain. This NILT method is based on the FFT in conjuction with the quotient–difference algorithm of Rutishauser to ensure both high speed of computation and sufficient accuracy of results.**

Index Terms-- **Laplace transform inversion, Matlab language, time–domain simulation, transmission line system**

I. INTRODUCTION

A possible method which is general enough to be used for an analysis of lumped–distributed circuits is a modified nodal admittance equation method (MNA) [1,2]. In [3] this technique is applied for the time–domain simulation of multiconductor transmission line systems using Matlab language. It can be illustrated by a block diagram in Fig.1.

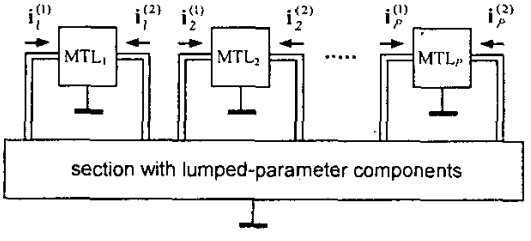

Fig. 1. Linear system with MTL sections

Therein the MNA matrix equation describing MTL systems under nonzero initial conditions is formulated and an effectiveness of the solution in terms of the Matlab language is shown. Unlike [1,2], where modal analysis technique is used, admittance matrices of the MTLs are computed by means of the chain matrices. In this way inhomogeneities of the MTLs can easily be taken into account if necessary. To incorporate nonzero initial conditions on the MTLs matrix convolution integrals must be solved. In Matlab language, this can effectively be made by the FFT when three-dimensional arrays are utilized [4]. From general point of view, the solution is performed in the frequency domain and then a fast NILT method in vector or matrix forms is used to obtain the solution in the time domain. Unlike mentioned work [3] an improved NILT method based on the FFT and quotient-difference algorithm of Rutishauser is here used to ensure both high speed of the computation and the sufficient precision of the results. Moreover, to save a CPU time considerably the Matlab language capabilities to process multidimensional arrays in parallel are utilized with advantages.

II. MNA MATRIX EQUATION FORMULATION

As is shown e. g. in [1,2] the modified nodal admittance matrix equation in the time domain can be written as

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

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 circuit. Applying Laplace transformation the frequency-domain representation has the form

$$[\mathbf{G}_M + s\mathbf{C}_M]\mathbf{V}_M(s) + \sum_{k=1}^{P} \mathbf{D}_k \mathbf{I}_k(s) = \mathbf{I}_M(s) + \mathbf{C}_M \mathbf{v}_M(0). \qquad (2)$$

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) 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 they are stated from the basic MTL matrix equation as follows.

Suppose a generally nonuniform MTL of a length l, with per-unit-length matrices $\mathbf{R}(x)$, $\mathbf{L}(x)$, $\mathbf{G}(x)$, $\mathbf{C}(x)$. In the time domain a basic MTL matrix equation has a form as [5]

$$\frac{\partial}{\partial x}\begin{bmatrix} \mathbf{v}(x,t) \\ \mathbf{i}(x,t) \end{bmatrix} = \begin{bmatrix} \mathbf{0} & -\mathbf{R}(x) \\ -\mathbf{G}(x) & \mathbf{0} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{v}(x,t) \\ \mathbf{i}(x,t) \end{bmatrix} - \begin{bmatrix} \mathbf{0} & \mathbf{L}(x) \\ \mathbf{C}(x) & \mathbf{0} \end{bmatrix} \cdot \frac{\partial}{\partial t}\begin{bmatrix} \mathbf{v}(x,t) \\ \mathbf{i}(x,t) \end{bmatrix}, \qquad (3)$$

0-7803-7763-X/03/$17.00 ©2003 IEEE

and, after a Laplace transformation is applied, the (3) leads to

$$\frac{d}{dx}\begin{bmatrix}\mathbf{V}(x,s)\\ \mathbf{I}(x,s)\end{bmatrix}=\begin{bmatrix}0 & -\mathbf{Z}(x,s)\\ -\mathbf{Y}(x,s) & 0\end{bmatrix}\cdot\begin{bmatrix}\mathbf{V}(x,s)\\ \mathbf{I}(x,s)\end{bmatrix}+\begin{bmatrix}\mathbf{0} & \mathbf{L}(x)\\ \mathbf{C}(x) & \mathbf{0}\end{bmatrix}\cdot\begin{bmatrix}\mathbf{v}(x,0)\\ \mathbf{i}(x,0)\end{bmatrix}, \quad (4)$$

Here $\mathbf{V}(x,s)=\mathrm{L}\left[\mathbf{v}(x,t)\right]$ and $\mathbf{I}(x,s)=\mathrm{L}\left[\mathbf{i}(x,t)\right]$ are column vectors of the Laplace transforms of voltages and currents at a distance x from MTL's left end, respectively, $\mathbf{v}(x,0)$ and $\mathbf{i}(x,0)$ are column vectors of initial voltage and current distributions, respectively, and $\mathbf{0}$ means zero matrix. The $\mathbf{Z}(x,s)=\mathbf{R}(x)+s\mathbf{L}(x)$ and $\mathbf{Y}(x,s)=\mathbf{G}(x)+s\mathbf{C}(x)$ are series impedance and shunting admittance matrices, respectively. In a compact matrix form the (4) looks like

$$\frac{d}{dx}\mathbf{W}(x,s)=\mathbf{M}(x,s)\mathbf{W}(x,s)+\mathbf{N}(x)\mathbf{w}(x,0)\,. \quad (5)$$

Then taking $\mathbf{W}(0,s)$ as the solution at $x=0$ (MTL's input) the solution at $x=l$ (MTL's output) can be written as [6]

$$\mathbf{W}(l,s)=\mathbf{\Phi}_0^l(s)\mathbf{W}(0,s)+\int_0^l\mathbf{\Phi}_\xi^l(s)\mathbf{N}(\xi)\mathbf{w}(\xi,0)d\xi\,, \quad (6)$$

where $\mathbf{\Phi}_0^l(s)$ is an integral matrix (matrizant) defined with an infinite series of matrix integrals or with so-called product-integral, see e.g. [6]. In the case of a uniform MTL the matrix exponential function is used for its exact calculation as

$$\mathbf{\Phi}_0^l(s)\Big|_{\mathbf{M}(x,s)=\mathbf{M}(s)}=e^{\mathbf{M}(s)l}\,. \quad (7)$$

In general case, however, only an approximate integral matrix can be calculated. It can be done by dividing the MTL into a sufficiently large number m of sections presupposing $\mathbf{M}(s)$ constant in each of them. Then taking basic property of the matrizant into account a recurrent formula is valid as follows

$$\tilde{\mathbf{\Phi}}_0^{x_j}(s)=e^{\mathbf{M}(\zeta_j,s)\Delta x_j}\cdot\tilde{\mathbf{\Phi}}_0^{x_{j-1}}(s),\ \text{with}\ \tilde{\mathbf{\Phi}}_0^{0}(s)=\mathbf{E} \quad (8)$$

as an identity matrix, $\Delta x_j=x_j-x_{j-1}$, $j=1,2,\ldots,m$, $x_0=0$, $x_m=l$ and $\zeta_j\in\left\langle x_{j-1},x_j\right\rangle$. In the case of a uniform MTL the result is the same as according to (7).

In terms of the multiport theory the integral matrix acts as the chain matrix $\mathbf{\Phi}(s)$. Thus after denoting

$$\mathbf{W}(0,s)=\mathbf{W}^{(1)}(s)=[\mathbf{V}^{(1)}(s),\mathbf{I}^{(1)}(s)]^T\,, \quad (9)$$

$$\mathbf{W}(l,s)=\mathbf{W}^{(2)}(s)=[\mathbf{V}^{(2)}(s),-\mathbf{I}^{(2)}(s)]^T\,, \quad (10)$$

$$\int_0^l\mathbf{\Phi}_\xi^l(s)\mathbf{N}(\xi)\mathbf{w}(\xi,0)d\xi=\mathbf{W}^{(0)}(s)=[\mathbf{V}^{(0)}(s),\mathbf{I}^{(0)}(s)]^T\,, \quad (11)$$

the MTL can be described by (6) in a decomposed form as

$$\begin{bmatrix}\mathbf{V}^{(2)}(s)\\ -\mathbf{I}^{(2)}(s)\end{bmatrix}=\begin{bmatrix}\mathbf{\Phi}_{11}(s) & \mathbf{\Phi}_{12}(s)\\ \mathbf{\Phi}_{21}(s) & \mathbf{\Phi}_{22}(s)\end{bmatrix}\cdot\begin{bmatrix}\mathbf{V}^{(1)}(s)\\ \mathbf{I}^{(1)}(s)\end{bmatrix}+\begin{bmatrix}\mathbf{V}^{(0)}(s)\\ \mathbf{I}^{(0)}(s)\end{bmatrix}. \quad (12)$$

After some manipulations the admittance equations taking nonzero initial conditions into account have a form as

$$\begin{bmatrix}\mathbf{I}^{(1)}(s)\\ \mathbf{I}^{(2)}(s)\end{bmatrix}=\begin{bmatrix}\mathbf{Y}_{11}(s) & \mathbf{Y}_{12}(s)\\ \mathbf{Y}_{21}(s) & \mathbf{Y}_{22}(s)\end{bmatrix}\cdot\begin{bmatrix}\mathbf{V}^{(1)}(s)\\ \mathbf{V}^{(2)}(s)\end{bmatrix}-\begin{bmatrix}\mathbf{Y}_{12}(s) & \mathbf{0}\\ \mathbf{Y}_{22}(s) & \mathbf{E}\end{bmatrix}\cdot\begin{bmatrix}\mathbf{V}^{(0)}(s)\\ \mathbf{I}^{(0)}(s)\end{bmatrix}, \quad (13)$$

where submatrices $\mathbf{Y}_{11}(s)=-\mathbf{\Phi}_{12}^{-1}(s)\mathbf{\Phi}_{11}(s)$, $\mathbf{Y}_{12}(s)=\mathbf{\Phi}_{12}^{-1}(s)$, $\mathbf{Y}_{22}(s)=-\mathbf{\Phi}_{22}(s)\mathbf{\Phi}_{12}^{-1}(s)$, and $\mathbf{Y}_{21}(s)=\mathbf{Y}_{12}^T(s)$ because of a reciprocity of the MTL. In the case of a uniform MTL the equality $\mathbf{Y}_{11}(s)=\mathbf{Y}_{22}(s)$ is then valid. Considering now the k–th MTL the (13) can be written in a compact matrix form

$$\mathbf{I}_k(s)=\mathbf{Y}_k(s)\mathbf{V}_k(s)-\mathbf{X}_k(s)\mathbf{\Gamma}_k(s)\,. \quad (14)$$

Finally after substituting (14) into basic MNA equation (2) the resultant MNA equation can be written in the form as

$$\mathbf{V}_M(s)=\left[\mathbf{G}_M+s\mathbf{C}_M+\sum_{k=1}^{P}\mathbf{D}_k\mathbf{Y}_k(s)\mathbf{D}_k^T\right]^{-1}\cdot\left[\mathbf{I}_M(s)+\mathbf{C}_M\mathbf{v}_M(0)+\sum_{k=1}^{P}\mathbf{D}_k\mathbf{X}_k(s)\mathbf{W}_k^{(0)}(s)\right]. \quad (15)$$

To solve voltages and currents at a coordinate x from the beginning (1) of the MTL the (12) can look like

$$\begin{bmatrix}\mathbf{V}(x,s)\\ \mathbf{I}(x,s)\end{bmatrix}=\begin{bmatrix}\mathbf{\Phi}_{11}(x,s) & \mathbf{\Phi}_{12}(x,s)\\ \mathbf{\Phi}_{21}(x,s) & \mathbf{\Phi}_{22}(x,s)\end{bmatrix}\cdot\begin{bmatrix}\mathbf{V}^{(1)}(s)\\ \mathbf{I}^{(1)}(s)\end{bmatrix}+\begin{bmatrix}\mathbf{V}^{(0)}(x,s)\\ \mathbf{I}^{(0)}(x,s)\end{bmatrix}, \quad (16)$$

where $\mathbf{\Phi}(x,s)$ is a partial chain matrix computed according to (8) and $[\mathbf{V}^{(0)}(x,s),\mathbf{I}^{(0)}(x,s)]^T=\mathbf{W}^{(0)}(x,s)$ is expressed by the matrix integral (11) while replacing indices l by x. As this integral expression is that of a convolution type the method based on the FFT can be used for its calculation. In the work [4] there is proposed to utilize three-dimensional arrays when capabilities of the Matlab language to treat multidimensional arrays in parallel are utilized with advantages. The necessary voltage $\mathbf{V}^{(1)}(s)$ and the current $\mathbf{I}^{(1)}(s)$ appertaining to the k–th MTL can be extracted from the equations

$$\mathbf{V}_k(s)=\mathbf{D}_k^T\mathbf{V}_M(s)\,, \quad (17)$$

and (14), respectively.

III. ADVANCED FFT-BASED NILT METHOD

An original $f(t)$ to a Laplace transform $F(s)$ can be expressed by the Bromwich integral

$$f(t)=\frac{1}{2\pi j}\int_{c-j\infty}^{c+j\infty}F(s)e^{st}ds\,, \quad (18)$$

under the basic assumption $|f(t)|\le Ke^{\alpha t}$, K real positive, α as the exponential order of the real function $f(t)$, $t\ge 0$, and $F(s)$ defined for $\mathrm{Re}[s]>\alpha$. Integrating the integral (18) numerically an approximate formula in the discrete form

$\tilde{f}_k = \tilde{f}(kT)$, $k = 0,\cdots,N-1$, can be derived as [7]

$$\tilde{f}_k = C_k\{2\,\mathrm{Re}[\sum_{n=0}^{N-1} F_n z_k^n + \sum_{n=0}^{\infty} G_n z_k^n] - F_0\}, \qquad (19)$$

where

$$C_k = \frac{\Omega}{2\pi}\cdot e^{ckT}, z_k = e^{-jk\Omega}, F_n = F(c - jn\Omega), G_n = F_{N+n}, \quad (20)$$

with T and $\Omega = 2\pi/(NT)$ as sampling periods in the original and the transform domain, respectively. The error analysis has resulted in an approximate formula for the coefficient c as

$$c \approx \alpha - \Omega/2\pi \cdot \ln E_r\ , \qquad (21)$$

where E_r denotes the desired relative error. The finite sum in (19) is evaluated by the FFT supposing $N = 2^m$, m integer. This enables to get the set of N points in a single calculation step. Consequently the required maximum time is taken as $t_m = (M-1)T$, with $M = N/2$ as the number of resultant computed points. To minimize an error towards its theoretical value E_r the infinite sum in (19) must be evaluated as much accurately as possible. For this purpose just the quotient-difference algorithm is used to accelerate its convergence. Thus taking only $2P+1$ first terms of this sum into account the continued fraction is determined as [8]

$$v(z_k, P) = d_0/(1 + d_1 z_k/(1 + \cdots + d_{2P} z_k))\ , \quad \forall k\ , \qquad (22)$$

that corresponds to Padé rational approximation of the power series. The q-d algorithm can shortly be explained as follows, see Fig.2.

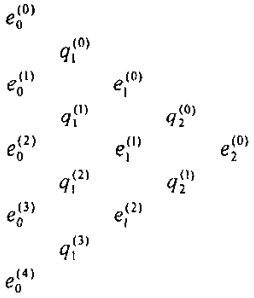

Fig. 2. Quotient-difference algorithm diagram

The first two columns are formed as

$$e_0^{(i)} = 0\ , \qquad i = 0,\cdots,2P\ , \qquad (23)$$

$$q_1^{(i)} = G_{i+1}/G_i\ , \ i = 0,\cdots,2P-1\ , \qquad (24)$$

and successive columns are given by the rules as follows:

for $r = 1,\cdots,P$,

$$e_r^{(i)} = q_r^{(i+1)} - q_r^{(i)} + e_{r-1}^{(i+1)}\ , \ i = 0,\cdots,2P-2r\ , \qquad (25)$$

for $r = 2,\cdots,P$,

$$q_r^{(i)} = q_{r-1}^{(i+1)} e_{r-1}^{(i+1)} / e_{r-1}^{(i)}\ , \quad i = 0,\cdots,2P-2r-1\ . \qquad (26)$$

Then the coefficients d_n, $n = 0,\cdots,2P$, are given by

$$d_0 = G_0\ ,\ d_{2m-1} = -q_m^{(0)}\ ,\ d_{2m} = -e_m^{(0)}\ , \qquad (27)$$

$m = 1,\cdots,P$. A practical evaluation of the continued fraction (22) can also be based on recurrent formulae [8]. Finally the $v(z_k, P)$ is used in (19) instead of the original infinite sum.

To invert the equation (15) very fast the vector version of the NILT method is used as follows [9]. If a transform to be inverted is a vector $\mathbf{F}^J(s) = [F_1(s), F_2(s), \cdots F_J(s)]^T$ then a NILT formula in a matrix form can be written as

$$\tilde{\mathbf{f}}^{J\times M} = \mathbf{C}^{J\times M} \circ \{2\,\mathrm{Re}[\mathbf{R}^{J\times M}\{\underset{<2>}{FFT}(\mathbf{F}^{J\times N})\} + \mathbf{V}_P^{J\times M}] - \mathbf{F}_0^{J\times M}\}, \quad (28)$$

where all terms are matrices of upper indexed sizes computed according to (20), but formed for all the vector components. The subscript $< 2 >$ means the FFT operation runs along the 2nd dimension (columns) but in parallel for all the rows. The $\mathbf{V}_P^{J\times M}$ is the matrix resulted from (22), the $\mathbf{R}^{J\times M}\{.\}$ denotes the operator of $N \to M$ matrix dimension reduction and the symbol $\circ$ means Hadamard product of matrices. Similarly to find the time-domain solution (16) the matrix version of the NILT method is the most effective to use [10]. The formula is

$$\tilde{\mathbf{f}}^{J\times M\times L} = \mathbf{C}^{J\times M\times L} \circ \{2\mathrm{Re}\mathbf{R}^{J\times M\times L}\{\underset{<2>}{FFT}(\mathbf{F}^{J\times N\times L})\} + \mathbf{V}_P^{J\times M\times L}] - \mathbf{F}_0^{J\times M\times L}\}, \quad (29)$$

with terms as 3–dimensional arrays of upper indexed sizes.

IV. MATLAB LANGUAGE IMPLEMENTATIONS

Below the Matlab listings are presented both for the vector and for the matrix version of the NILT method:

```
%******* NILTV – FUNCTION DEFINITION (vector version) *******%
function [ft,t]=niltv(F,tm,pl);
global ft t;
alfa=0; M=256; P=3; Er=1e-10;                    % adjustable
N=2*M; qd=2*P+1; t=linspace(0,tm,M);
NT=2*tm*N/(N-2); omega=2*pi/NT; c=alfa-log(Er)/NT;
s=c-i*omega*(0:N+qd-1); Fsc=feval(F,s); ft=fft(Fsc,N,2);
ft=ft(:,1:M); delv=size(Fsc,1); d=zeros(delv,qd); e=d;
q=Fsc(:,N+2:N+qd)./Fsc(:,N+1:N+qd-1);
d(:,1)=Fsc(:,N+1); d(:,2)=-q(:,1);
for r=2:2:qd-1
 w=qd-r; e(:,1:w)=q(:,2:w+1)-q(:,1:w)+e(:,2:w+1); d(:,r+1)=-e(:,1);
 if r>2
 q(:,1:w-1)=q(:,2:w).*e(:,2:w)./e(:,1:w-1); d(:,r)=-q(:,1);
 end
end
A2=zeros(delv,M); B2=ones(delv,M); A1=repmat(d(:,1),[1,M]); B1=B2;
z=repmat(exp(-i*omega*t),[delv,1]);
for n=2:qd
 Dn=repmat(d(:,n),[1,M]); A=A1+Dn.*z.*A2; B=B1+Dn.*z.*B2;
 A2=A1; B2=B1; A1=A; B1=B;
end
ft=2*real(ft+A./B)-repmat(Fsc(:,1),[1,M]);
ft=repmat(exp(c*t)/NT,[delv,1]).*ft; ft(:,1)=2*ft(:,1);
feval(pl);
%*****************************************************************%

%*************** PLOT1 – FUNCTION DEFINITION ***************%
function pl1                    % multiple plotting into a single figure
global ft t;
plot(t,ft); xlabel('t'); ylabel('f(t)'); grid on;
%*****************************************************************%
```

```
%*************** PLOT2 – FUNCTION DEFINITION ***************%
function pl2                    % plotting into separate figures
global ft t;
for k=1:size(ft,1)
 figure;
 plot(t,ft(k,:)); xlabel('t'); ylabel('f(t)'); grid on;
end
%************************************************************%

%******* NILTM – FUNCTION DEFINITION (matrix version) ******%
function [ft,t,x]=niltm(F,tm,pl);
global ft t x;
alfa=0; M=256; P=3; Er=1e-10;                    % adjustable
N=2*M; qd=2*P+1; t=linspace(0,tm,M);
NT=2*tm*N/(N-2); omega=2*pi/NT; c=alfa-log(Er)/NT;
s=c-i*omega*(0:N+qd-1); Fsc=feval(F,s); ft=fft(Fsc,N,2);
ft=ft(:,1:M,:); dim1=size(Fsc,1); dim3=size(Fsc,3);
d=zeros(dim1,qd,dim3); e=d;
q=Fsc(:,N+2:N+qd,:)./Fsc(:,N+1:N+qd-1,:);
d(:,1,:)=Fsc(:,N+1,:); d(:,2,:)=-q(:,1,:);
for r=2:2:qd-1
 w=qd-r; e(:,1:w,:)=q(:,2:w+1,:)-q(:,1:w,:)+e(:,2:w+1,:);
 d(:,r+1,:)=-e(:,1,:);
 if r>2
 q(:,1:w-1,:)=q(:,2:w,:).*e(:,2:w,:)./e(:,1:w-1,:); d(:,r,:)=-q(:,1,:);
 end
end
A2=zeros(dim1,M,dim3); B2=ones(dim1,M,dim3);
A1=repmat(d(:,1,:),[1,M]); B1=B2;
z=repmat(exp(-i*omega*t),[dim1,1,dim3]);
for n=2:qd
 Dn=repmat(d(:,n,:),[1,M]); A=A1+Dn.*z.*A2; B=B1+Dn.*z.*B2;
 A2=A1; B2=B1; A1=A; B1=B;
end
ft=ft+A./B; ft=2*real(ft)-repmat(real(Fsc(:,1,:)),[1,M]);
ft=repmat(exp(c*t)/NT,[dim1,1,dim3]).*ft; ft(:,1,:)=2*ft(:,1,:);
feval(pl);
%************************************************************%

%*************** PLOT3 – FUNCTION DEFINITION ***************%
function pl3
global ft t x;
m=length(t); tgr=[1:m/64:m,m];                   % 65 time points
for k=1:size(ft,3)
 figure; mesh(t(tgr),x(k,:),ft(:,tgr,k));
 xlabel('t'); ylabel('x'); zlabel(strcat('f_{',num2str(k),'}'));
end
%************************************************************%
```

The niltv and niltm functions are called with 3 parameters: 'F' - the name of another function in which Laplace transform in the vector or matrix form is defined, tm - the maximum time, 'pl' - the name of the plotting function.

V. EXAMPLE

Consider the circuit with three uniform (2+1)-conductor transmission lines in Fig. 3, with the per-unit-length matrices stated in [1]. The MTLs differ only by their lengths, namely $l_1 = 0.05m$, $l_2 = 0.04m$, $l_3 = 0.03m$. A 1 V pulse with 1.5 ns rise/fall times and 7.5 ns width is applied at the input.

To gain waveforms of nodal voltages or branch currents a Matlab function describing the solution (15) is called by the niltv function. In Fig. 4 the input and output voltages and the current i_2 are done as the examples. However, to get voltage or current distributions along MTLs' wires a Matlab function describing the solution (16) is called by the niltm function. In Fig. 5 the voltage distributions are done as the examples.

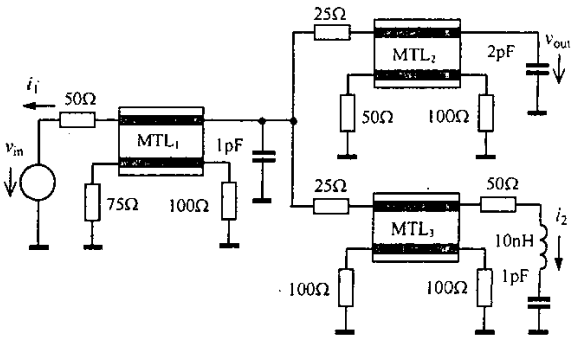

Fig. 3. Linear circuit containing three MTLs

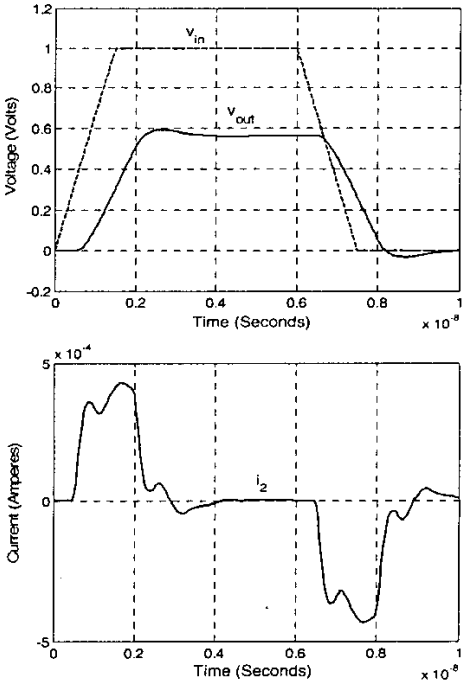

Fig. 4. Voltage and current waveforms

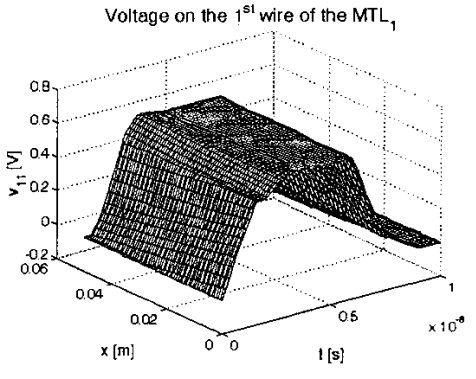

Fig. 5. Voltage distributions along MTLs' wires

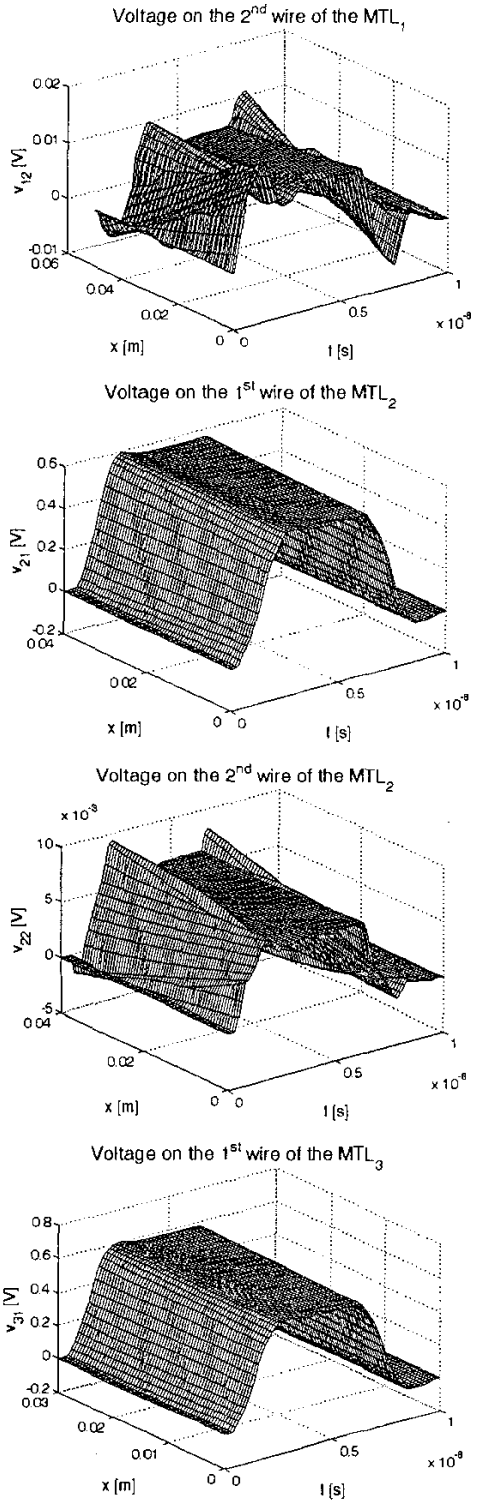

Fig. 5. Voltage distributions along MTLs' wires – cont.

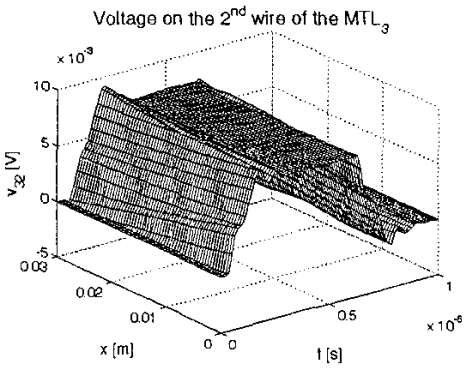

Fig. 5. Voltage distributions along MTLs' wires – cont.

VI. CONCLUSION

All the computations were done on a PC 2GHz/256MB. The CPU times were about 0.5 second and 6 seconds when the niltv and niltm functions were called, respectively. The used novel NILT method based on the FFT and quotient–difference algorithm of Rutishauser leads to approximatelly 20% saving CPU time compared to the previous one using ε-algorithm [3]. Moreover, the method seems to be more numerically stable. The results obtained in the example under consideration correspond very well with those gained by the NILT method in [1] and the AWE technique in [2].

REFERENCES

[1] S. Lum, M. S. Nakhla, and Q. J. Zhang, "Sensitivity Analysis of Lossy Coupled Transmission Lines," *IEEE Transactions on Microwave Theory and Techniques*, vol. 39, no. 12, pp. 2089-2099, Dec. 1991.

[2] E. Chiprout, M. S. Nakhla, *Asymptotic Waveform Evaluation and Moment Matching for Interconnect Analysis*, Carleton University, Kluwer Academic Publisher, Boston, 1994.

[3] L. Brančík, "Time-Domain Simulation of Multiconductor Transmission Line Systems under Nonzero Initial Conditions," in *Proc. 15th ECCTD'01*, Espoo, Finland, Aug. 2001, vol. 1, pp. 73-76.

[4] L. Brančík, "Time-Domain Simulation of Nonuniform Multiconductor Transmission Lines under Nonzero Initial Conditions Using Matlab Language," in *Proc. 14th ECCTD'99*, Stresa, Italy, Aug.–Sept. 1999, vol. 2, pp. 1135-1138.

[5] C. R. Paul, *Analysis of Multiconductor Transmission Lines*, John Wiley & Sons, New York, 1994.

[6] F. R. Gantmacher, *The Theory of Matrice*, Chelsea, New York, 1977.

[7] L. Brančík, "Utilization of Quotient-Difference Algorithm in FFT-based Numerical ILT Method," in *Proc. 11th Radioelektronika'2001*, Brno, Czech Republic, May 2001, pp. 352-355.

[8] P. Henrici, *Quotient-difference Algorithms*, Mathematical Methods for Digital Computers, vol. 2, A. Ralston and H. S. Wilf, eds., John Wiley & Sons, New York, 1967, pp. 37-62.

[9] L. Brančík, "Improved Numerical Inversion of Laplace Transforms Applied to Simulation of Distributed Circuits," in *Proc. XI. ISTET'01*, Linz, Austria, August 2001, pp. 51-54.

[10] L. Brančík, "Matlab Oriented Matrix Laplace Transforms Inversion for Distributed Systems Simulation," in *Proc. 12th Radioelektronika'2002*, Bratislava, Slovakia, May 2002, paper No 114.

ACKNOWLEDGEMENTS

This work was supported by the grant GAČR No. 102/03/0241 and realized within the scope of Research programmes MSM 262200011 and MSM 262200022.