Efficient Simulation of Lossy and Dispersive Transmission Lines

Tuyen V. Nguyen*

IBM Microelectronics, Hopewell Junction, NY 12533

Abstract - This paper presents an efficient method, based on the transfer function simulation method, for the transient analysis of lossy and dispersive transmission lines terminated by nonlinear circuits. This method combines the efficiency provided by rational function approximation with the accuracy of using modal decomposition. For the task of computing the convolution integral, the transfer function simulation method was found to be more straightforward for implementation and more efficient than the recursive convolution method, thus providing further improvement in simulation efficiency. A robust and efficient method for the analysis of frequency dependent (dispersive) transmission lines is also reported. This method has been implemented in AS/X, IBM production circuit simulator, and has been used routinely for the analysis of practical lossy and dispersive coupled transmission lines.

1. Introduction

Today it is well known that accurate and efficient simulation of lossy multiconductor transmission lines with general nonlinear terminations is necessary in the design of high speed integrated circuits and systems. For high speed packaging and interconnect designs, it is also important to take into account the frequency dependence of the line parameters due to skin effect and proximity effect [3][4]. The conventional approach is to use Fast Fourier Transform (FFT) [5] or numerical inversion of Laplace Transform (NILT) [6] to convert the line description from the frequency domain to the time domain. Then numerical convolution is employed to compute the responses at the boundaries of the lines during the transient analysis in the time domain. These approaches are, therefore, very costly in terms of both CPU and memory usage. These problems obviate the need for an efficient method that can still ensure the accuracy for the transient simulation of lossy and dispersive transmission lines.

It has been recognized that rational function approximation can provide the answers to the above problems [1][2]. First, rational function approximation eliminates the need to use FFT or NILT. Secondly, this type of approximation allows an efficient evaluation of the convolution integral in the time domain. In the context of transmission line analysis, there are two different, but related, approaches to the computation of the convolution integral using rational function approximation. The first method was called ‘‘recursive convolution’’, and was first reported in [2]. The other method was called ‘‘transfer function simulation’’, and was first reported in [1]. Although recursive convolution significantly reduce the cost of numerical convolution in the time domain, simulation efficiency can be further improved by using the transfer function simulation method. Recent works [7][8][9][10] seem to concentrate on the recursive convolution method and overlook the transfer function simulation method. It is the goal of this paper to compare the two methods and to point out the advantages of the transfer function simulation method.

The task of computing the rational function approximations for the line parameters is of paramount importance in ensuring the accuracy of simulation results. This problem is especially difficult for frequency dependent transmission lines. For lossy lines with constant parameters, Pade approximation has proven to be an efficient way to compute the rational function approximations. However, Pade approximation can not be directly applied to the case of frequency dependent lines. For these dispersive lines, the rational function approximations are usually obtained by nonlinear least square fit. The next section of the paper will briefly review the modal decomposition model for coupled transmission lines. Modal decomposition is employed for the extraction of the ideal mode delays to ensure simulation accuracy. Section 3 discusses the computation of rational function approximations. Section 4 provides a detailed derivation of the transfer function simulation method in comparison to the recursive convolution method. The proposed method has been used extensively in production for the design of packages and interconnects. Practical examples are presented in Section 5 to substantiate the efficiency and accuracy of the proposed approach.

2. Modal Decomposition for Coupled Transmission Lines

The differential equations for a lossy (n+1) conductor transmission line (n signal lines plus a reference conductor) can be expressed in matrix form as

$$\begin{aligned} \frac{\partial}{\partial x} V(x,s) &= -Z(s)\, I(x,s) \\ \frac{\partial}{\partial x} I(x,s) &= -Y(s)\, V(x,s) \end{aligned} \tag{1}$$

where V is the (nx1) vector of line voltages with respect to the reference conductor, I is the (nx1) vector of the line currents, $Z(s) = R(s) + sL(s)$, $Y(s) = G(s) + sC(s)$; R, L, C, and G are nxn real symmetric matrices of per unit length line resistances, inductances, capacitances, and conductances. R, L, C, and G may be functions of frequency, but independent of x. The solution of the above differential equations can be written as

$$\begin{aligned} I_1(s) &= Y_0(s)\, V_1(s) + J_1(s) \\ I_2(s) &= Y_0(s)\, V_2(s) + J_2(s) \end{aligned} \tag{2}$$

where

$$\begin{aligned} J_1(s) &= -e^{-dQ(s)} \left[Y_0(s)\, V_2(s) + I_2(s) \right] \\ &= -B^{-1}(s)\, e^{-d\Gamma(s)}\, B(s) \left[Y_0(s)\, V_2(s) + I_2(s) \right] \end{aligned} \tag{3}$$

* The author is now with Motorola ULSI Lab, Austin,TX

$$J_2(s) = -e^{-dQ(s)}\,[\,Y_0(s)\,V_1(s) + I_1(s)\,] = -B^{-1}(s)\,e^{-d\Gamma(s)}\,B(s)\,[\,Y_0(s)\,V_1(s) + I_1(s)\,] \quad (4)$$

$Q^2(s) = Y(s)\,Z(s)$, $Q(s) = B^{-1}(s)\,\Gamma(s)\,B(s)$, and $Y_0(s) = Q^{-1}(s)\,Y(s)$. Γ is a diagonal matrix of the eigenvalues of Q, and B is the matrix of the left eigenvectors of Q. The port currents and voltages are given by

$$V_1(s) = V(0,s) \qquad V_2(s) = V(d,s)$$
$$I_1(s) = I(0,s) \qquad I_2(s) = -I(d,s) \quad (5)$$

with d being the length of the transmission line.

Note that each product in (2), (3), and (4) corresponds to a convolution computation in the time domain. At each time point during the transient simulation, the currents J_1 and J_2 represent past values, which have been computed at previous time points. The admittance matrix stamp of the transmission line can be computed from (2). This matrix stamp is then employed in the formulation of the linearized equation of the global circuit, that contains the nonlinear terminations as well as the transmission line, for the Newton iterations. Consequently, for transient simulation, the time domain description of Y_0, B, B^{-1}, and $e^{-d\Gamma}$ must be computed. These time domain descriptions can be obtained in a straightforward manner from the rational function approximations of these matrices in the frequency domain.

3. Computation of Rational Function Approximations

For lossy lines with constant parameters, Pade approximation has proven to be an efficient method for obtaining the rational function approximations [7][8][9][10][11]. Typically, Pade approximation is used in conjunction with modal decomposition in the frequency domain. Modal decomposition is employed for the extraction of high frequency delays in order to ensure simulation accuracy. Moreover, the extraction of mode delays helps keep low orders of approximation for the remaining terms. Otherwise, generalized moment matching method must be used to obtain very high order approximations of the delay terms [12]. High order rational function approximations will generally degrade simulation efficiency because the order of the rational function approximation directly affects the cost of the convolution computation, which lies at the very heart of a transient analysis. This problem becomes more pronounced as the number of coupled lines increases.

For transmission lines with frequency dependent parameters, the rational function approximations must be able to approximate the line parameters over a range of frequency. It is tempting to simply use high order Pade approximations to compute the rational function approximations. However, it should be noted that Pade approximation is numerically unstable in the sense that it is difficult to compute stable high order approximations [12][13]. In addition, even though Pade approximation has good extrapolation properties, it is still a local approximation based on the series expansion about some frequency. What is required is a “global” approximation over a range of frequency. For lossy lines with constant parameters, with the extraction of mode delays, low order approximations (2 to 6) are usually sufficient to ensure simulation accuracy. For the dispersive case, the orders of the rational approximations are higher, and the approximations are usually obtained by nonlinear optimization.

The work reported in this paper employs modal decomposition in the frequency domain, and fits rational function approximations to the frequency dependent eigenvalues, eigenvectors, and admittance matrices. The simple procedure proposed in [1] was used to match the eigenvalues and eigenvectors at different frequencies. Basically, if the frequency data points are sufficiently close (at least one point per decade), the dot product of the two eigenvectors corresponding to the same eigenvalue at two different frequencies should be close to one. Otherwise, the dot product should be close to zero. It was found that this simple procedure works well in practice.

For the fitting of frequency data, polynomial fit can theoretically be used to obtain the rational function approximations. Although polynomial fit is efficient, it does not impose constraints on the pole locations, thus can not guarantee the stability of the computed models. In order to ensure stability, the following form of the rational function for nonlinear least square fit is used

$$H(s) = \frac{(s+z_1)\ldots(s+z_m)}{\left(s+p_1^2\right)\ldots\left(s+p_n^2\right)}$$

Note that the pole locations are constrained to be in the left half plane during the optimization process. Strictly speaking, this function is complex, and both the phase and the magnitude of the function must be fitted. However, if the function is *minimum phase*, then only the magnitude characteristic need to be approximated; the approximating function will give a good fit for the phase characteristic as well [16]. A minimum phase function is a function all the zeros of which are in the left half of the complex plane. This is true for impedance/admittance functions of physical networks. This assumption is not true for eigenvectors and exponential propagation functions. However, in practice, it was found that fitting the magnitude is adequate for a majority of cases.

In this work, the Levenberg-Marquardt algorithm [17] is employed for the nonlinear least square fit. However, a simple implementation of this algorithm usually does not work robustly, especially for attenuation (the magnitude of the propagation) data, which is dependent on the line length. In this case, a more general procedure, such as the frequency partitioning scheme proposed in [18], is required. In this scheme, the frequency range of interest is divided into a number of smaller intervals. Then the approximation is computed in each interval. Since the range of frequency is reduced, low order models and fewer iterations are required to yield a good approximation. The result is then screened for extraneous roots. An extraneous root (pole or zero) is the one with magnitude much greater than the maximum frequency. Pole zero cancellation occurs when the relative difference between a pole and zero is less than 5% [18]. A root is also considered extraneous if its value is close to another root in a neighboring interval. The resulting poles and zeros for all the intervals are then used as the initial guess for the optimization over the whole frequency range of interest. This frequency partitioning scheme also provides a means for determining the order of the approximation.

In addition, a modified version of the incremental procedure proposed in [1] is also employed to provide better efficiency. In essence, this approach is an automation of the procedure for obtaining the rational function approximation from the Bode plot [19]. Basically, one pole or zero is added incrementally as the frequency data are scanned. The location of the poles and zeros are then adjusted locally using first

order sensitivity information to minimize the fitting errors. This method was found to work well for a good number of cases in practice. When this method fails to produce a satisfactory approximation, its results can be used as the initial guess for the Levenberg-Marquardt method described above.

Finally, it should be noted that for long lines, the attenuation may vary significantly over the frequency range of interest. It is, therefore, necessary to break up the lines into a number of shorter sections. In practice, it was found that high order poles are usually required to obtain a good fit for propagation data, especially at high frequencies where the skin effect comes into play.

4. Transfer function simulation and numerical convolution

This section discusses the scalar time domain convolution,

$$y(t) = \int_0^t h(u)\,x(t-u)\,du = \int_0^t h(t-u)\,x(u)\,du \tag{6}$$

or, equivalently,

$$Y(s) = H(s)\,X(s) \tag{7}$$

where $Y(s)$, $H(s)$, and $X(s)$ are scalar functions of frequency; $y(t)$, $h(t)$, and $x(t)$ are the equivalent time domain functions. In general, the computation of $y(t)$ using the convolution integral (6) is a computationally expensive operation both in terms of CPU and memory usage. The direct evaluation of (6) at each time point during a transient simulation would require the storage of the entire past history of the simulation and the integral must be summed over the entire simulation interval.

However, if $H(s)$ can be represented by a rational function in the s domain as

$$H(s) = \left(\sum_{i=0}^{M} a_i s^i\right) \Big/ \left(\sum_{i=1}^{N} b_i s^i\right) = \frac{U(s)}{D(s)} \tag{8}$$

where $b_N = 1$, the computation of $y(t)$ can be performed efficiently without storing the entire history of $x(t)$. In the recursive convolution method, the rational function is expanded in partial fraction form as

$$H(s) = d_0 + \sum_{j=1}^{N} \sum_{k=1}^{\Omega_j} \frac{d_{jk}}{(s+p_j)^k} \tag{9}$$

where p_j is the j-th pole, Ω_j is the order of the j-th pole, and d_{jk}'s are the associated residues. The time domain equivalent of this rational function can be expressed as a sum of polynomial weighed exponential terms. Hence, the transformation results can be obtained directly without using Fast Fourier Transform (FFT) or numerical inversion of Laplace Transform (NILT). In addition, this form of approximation allows the computation of the convolution integral given by (6) to be performed in a recursive and efficient manner. Note that complex conjugate pole pairs usually occur in the computation of the Pade approximation for constant RLGC coupled lines. The case of repeated real poles can also occur in dealing with the problem of positive poles for Pade approximation [14], but usually arises in fitting the attenuation data of frequency-dependent lines.

The transfer function simulation method, on the other hand, directly employs the polynomial form of (8). In this method, the frequency domain representation given by (8) is converted to a time domain representation using the state space method. The resulting state equation can then be solved by any standard numerical integration technique. The advantages of the transfer function simulation method as compared to the recursive convolution method are twofold. First, it does not require the computation of poles and residues. Hence, it is not necessary to distinguish between the various cases of (repeated) real poles or complex conjugates. The implementation is, therefore, simpler and more straightforward. Secondly, in many cases, the transfer function method is significantly more efficient than the recursive convolution method, especially for the case of complex conjugate or repeated real poles. The reason for this improvement is that the transfer function method does not require the evaluation of sine and cosine functions for complex conjugate pole pairs and it does not require some extra computation for repeated real poles. However, the numerical integration of the state equation will introduce an extra error term compared to the recursive convolution method and, therefore, would require more time steps. Overall, numerical experiments have shown that the transfer function method is more efficient and should be preferred. The reason is that the convolution computation is at the heart of a transient analysis of transmission lines and is usually performed in the innermost loop of the analysis. Therefore, the overall simulation efficiently depends greatly on the efficiency of the convolution computation.

The derivation of the transfer function simulation method is started by rewriting (8) as

$$\begin{aligned} Y(s) &= U(s)\,E(s) \\ X(s) &= D(s)\,E(s) \end{aligned} \tag{10}$$

where the intermediate variable $E(s) = X(s)/D(s)$ has been introduced. Taking the inverse Laplace transform of (10) yields

$$y(t) = \sum_{i=0}^{M} a_i e^{(i)}(t), \quad x(t) = \sum_{i=0}^{N} b_i e^{(i)}(t) \tag{11}$$

where $e^{(i)}(t)$ denotes the i^{th} derivative of $e(t)$. Introduce the following intermediate state variables

$$z_1(t) = e^{(0)}(t) \quad \ldots \quad z_N(t) = e^{(N-1)}(t)$$

Then

$$\dot{z}_1(t) = z_2(t) \quad \ldots \quad \dot{z}_{N-1}(t) = z_N(t)$$

$$\dot{z}_N(t) = e^{(N)}(t) = x(t) - \sum_{i=1}^{N} b_{i-1} z_i$$

and the state space representation of the system can be written as

$$\begin{aligned} \dot{\underline{z}}(t) &= \underline{A}\underline{z}(t) + \underline{b}x(t) \\ y(t) &= \underline{c}^T \underline{z}(t) + dx(t) \end{aligned} \tag{12}$$

where $\underline{z}(t) = \left[z_1(t) \ \ldots \ z_N(t)\right]^T$ and

$$\underline{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ & & & \dots & & \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 \\ a_0 & a_1 & a_2 & a_3 & \dots & a_{N-2} & a_{N-1} \end{bmatrix} \quad \underline{b} = \begin{bmatrix} 0 & \dots & 0 & 1 \end{bmatrix}^T \quad d = 1$$

$$\underline{c} = \begin{bmatrix} (a_0 - b_0) & \dots & (a_{N-2} - b_{N-2}) & (a_{N-1} - b_{N-1}) \end{bmatrix}^T$$

The solution of the state equation in (12) can be written as

$$\underline{z}(t) = e^{\underline{A}(t-t_0)}\underline{z}(t_0) + \int_{t_0}^{t} e^{\underline{A}(t-\tau)}\underline{b}x(\tau)\,d\tau \tag{13}$$

Note that in a transient simulation, some form of interpolation will be inevitably employed to compute x(t). This interpolation will introduce errors into the computation of $\underline{z}(t)$ and $y(t)$. Secondly, an exact evaluation of the exponential matrix $e^{\underline{A}(t-t_0)}$ would require the computation of the eigenvalues of $\underline{A}$, or equivalently the poles given in (9). In this sense, the computation of $\underline{z}(t)$ using an exact evaluation of the exponential matrix is equivalent to the recursive convolution discussed above. However, the state equation can be solved numerically using standard integration methods such as the Backward-Difference Formula (BDF) or Gear method without having to evaluate the exponential matrix. The numerical integration will be more efficient but will also introduce extra error terms. In order to simplify the following discussion, consider the well known trapezoidal rule for numerical integration. In other words, approximate the "excitation" $x(t)$ as

$$x(t) = x(t_{n-1}) + \frac{\Delta x_n}{\Delta t_n}(t - t_{n-1}) + R_2(t) \tag{14}$$

where $\Delta x_n = x(t_n) - x(t_{n-1})$, $\Delta t_n = t_n - t_{n-1}$, and $R_2(t)$ is the remainder term. Substitution of (14) into (13) and performing the integration yields

$$\underline{\xi}(t_n) = e^{\underline{A}\Delta t_n}\underline{\xi}(t_{n-1}) + \underline{\varepsilon}_2 \tag{15}$$

where

$$\underline{\xi}(t) = \underline{z}(t) + \underline{A}^{-1}\left[\underline{b}x(t) + \underline{A}^{-1}\underline{b}\left((\Delta x_n)/(\Delta t_n)\right)\right]$$

and

$$\underline{\varepsilon}_2 = \int_{t_0}^{t} e^{\underline{A}(t-\tau)}\underline{b}R_2(\tau)\,d\tau$$

Note that $\underline{\varepsilon}_2$ is the error due to the interpolation for $x(t)$. An estimate for this error term can be derived in a similar manner to the error derivation for the recursive convolution [15]. The error due to the numerical integration can be derived as follows. Rewrite (15) as

$$e^{-(\underline{A}\Delta t_n)/2}\underline{\xi}(t_n) = e^{(\underline{A}\Delta t_n)/2}\underline{\xi}(t_{n-1})$$

The trapezoidal approximation is then obtained by using the first two terms of the Taylor expansions for the two exponential matrices. In other words,

$$\tilde{\underline{\xi}}(t_n) \approx \left(I - \frac{1}{2}\underline{A}\Delta t_n\right)^{-1}\left(I + \frac{1}{2}\underline{A}\Delta t_n\right)\underline{\xi}(t_{n-1})$$

Hence, the error term due to the numerical integration can be written as

$$\begin{aligned} \varepsilon_1 &= \left|\tilde{\underline{x}}(t_n) - \underline{x}(t_n)\right| = \left|\tilde{\underline{\xi}}(t_n) - \underline{\xi}(t_n)\right| \\ &= \left|e^{\underline{A}\Delta t_n} - \left(I - \frac{1}{2}\underline{A}\Delta t_n\right)^{-1}\left(I + \frac{1}{2}\underline{A}\Delta t_n\right)\right|\left|\underline{\xi}(t_{n-1})\right| \\ &\approx \frac{1}{12}\underline{A}^2\Delta t_n^{\,2}\left\{\Delta t_n\left[\underline{A}\underline{z}(t_{n-1}) + \underline{b}x(t_{n-1})\right] + \underline{A}^{-1}\underline{b}\Delta x_n\right\} \end{aligned} \tag{16}$$

The total error in the computation of (13) is then given by

$$\varepsilon = \varepsilon_1 + \varepsilon_2$$

Note that $\varepsilon_1 = 0$ for the recursive convolution method. This error is due to the numerical integration. The above derivation can be extended to general BDF formula of order r using the techniques discussed above. In practice, numerical stability limits the maximum interpolation order to $r = 2$. Finally, one important point that should be noted is that the matrix A has very simple structure. Consequently, the matrix inversion $(I - (\underline{A}\Delta t_n)/2)^{-1}$ can be computed efficiently. In fact, the LU factorization and backward substitution for this method requires the same storage as the recursive convolution method. However, no sine or cosine computations are required.

5. Examples

All the examples in this section are simulated using the production circuit simulator AS/X on an IBM RS6000 Model 520 workstation. For verification purpose, the simulation results are compared with measurements for the dispersive case, and with the brute force approach of treating lossy coupled lines as cascaded series of 50 sections for the constant lossy case. Each section contains a lossless coupled line connected to a lumped network. The lumped network is used to model the line loss as represented by the R and G matrices

Example 1:This example is taken from [20]. The circuit schematic is shown in Fig. 1. All the transmission lines are lossy but independent

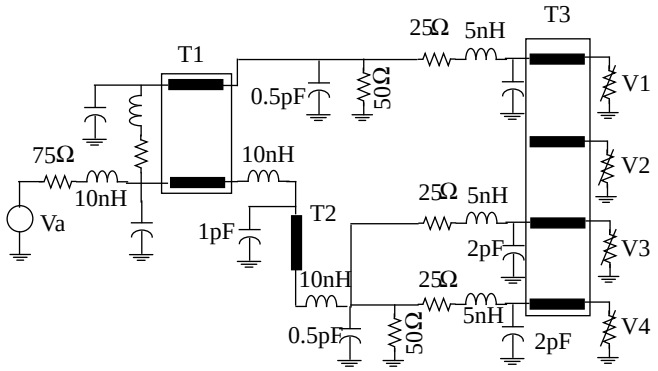


Fig. 1: An interconnection circuit with lossy transmission lines and nonlinear terminations.

of frequency. All the nonlinear elements are described by the equation $I = V^3$. The parameters of the single line are $R = 1\Omega/\text{m}$, $L = 600\text{nH/m}$, $C = 1\text{nF/m}$, $G = 5\text{mS/m}$. The parameters of the two coupled lines are

$$L = \begin{bmatrix} 600 & 50 \\ 50 & 600 \end{bmatrix} \text{nH/m3} \qquad C = \begin{bmatrix} 1.2 & -0.11 \\ -011 & 1.2 \end{bmatrix} \text{nF/m}$$

$$R = \begin{bmatrix} 2.25 & 0.225 \\ 0.225 & 2.25 \end{bmatrix} \Omega\text{/m} \qquad G = \begin{bmatrix} 7.5 & 0 \\ 0 & 7.5 \end{bmatrix} \text{mS/m}$$

The parameters of the four coupled lines are

$$L = \begin{bmatrix} 1 & .11 & .03 & 0 \\ .11 & 1 & .11 & .03 \\ .03 & .11 & 1 & .11 \\ 0 & .03 & .11 & 1 \end{bmatrix} \mu\text{H/m} \quad C = \begin{bmatrix} 1.5 & -.17 & -.03 & 0 \\ -.17 & 1.5 & -.17 & -.03 \\ -.03 & -.17 & 1.5 & -.17 \\ 0 & -.03 & -.17 & 1.5 \end{bmatrix} \text{nF/m}$$

$$R = \begin{bmatrix} 3.5 & .35 & .035 & 0 \\ .35 & 3.5 & .35 & .035 \\ .035 & .35 & 3.5 & .35 \\ 0 & .035 & .35 & 3.5 \end{bmatrix} \Omega\text{/m} \quad G = \begin{bmatrix} 10 & 1 & 0.1 & 0 \\ 1 & 10 & 1 & 0.1 \\ 0.1 & 1 & 10 & 1 \\ 0 & 0.1 & 1 & 10 \end{bmatrix} \text{mS/m}$$

The length of the single line is 0.2m. The length of the two conductor line is 0.3m. The length of four conductor line is 0.4m. The excitation is a pulse of 2 volt amplitude and 10ns pulse width with 1ns rise and fall times. The output waveforms at node V3 of the longest path are shown in Fig. 2 for both the proposed method and the brute force

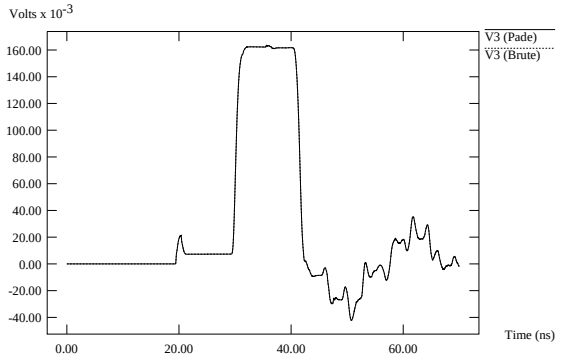


Fig. 2: Voltage responses at node V3 of the circuit in Fig. 1 for both the proposed and the brute force methods

method. The results are almost identical. The proposed method, using the transfer function simulation method, took only 132s of CPU time while the brute force method required 2134s. For a comparison, the recursive convolution method required 214s, about 62% more than the transfer function simulation method.

Example 2: This is a practical example used to demonstrate the capability of AS/X to simulate dispersive coupled lines. The physical structure of the transmission line is shown in Fig. 3. The length of the

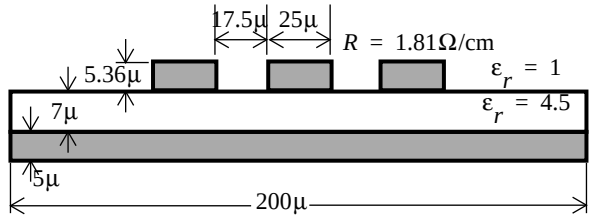


Fig. 3: The physical structure of a three conductor line in Example 2

line is 7.62cm. The equivalent circuits of the measurement setup and the test structure include lumped elements to model the output inductance and resistance of the pulse generator, the scope connection, the probe inductance, and the test line pad capacitance. Lossless transmission lines are used to model the rigid as well as the flexible coaxial cables. The complete equivalent circuit consists of 52 lump elements and 3 ideal transmission lines in addition to the three conductor line under test. In this experiment, one outer conductor was excited by the pulse generator and the coupled noises on the other two quiet conductors are measured.

For simulation, the line capacitance matrix was computed using C2D [21], a two dimensional capacitance calculation tool. The frequency dependent resistance and inductance matrices were obtained using Z2D [4], a two dimensional skin effect tool. The values of these matrices are shown below,

$$C = \begin{bmatrix} 2.017 & -0.0564 & -0.0069 \\ -0.0564 & 2.022 & -0.0564 \\ -0.0069 & -0.0564 & 2.017 \end{bmatrix} \text{pF/cm}$$

$$R = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{12} & R_{22} & R_{12} \\ R_{13} & R_{12} & R_{11} \end{bmatrix} \text{'k}\Omega\text{/cm} \quad L = \begin{bmatrix} L_{11} & L_{12} & L_{13} \\ L_{12} & L_{22} & L_{12} \\ L_{13} & L_{12} & L_{11} \end{bmatrix} \text{nH/cm}$$

where the frequency dependence of the inductances and resistances are given in Table 1 and Table 2,

Table 1: The frequency dependent inductance matrix

Freq (GHz)	L11	L22	L12	L13
0.01	3.915	3.640	0.2379	-0.8566
0.03	3.399	3.382	0.2201	-0.3792
0.10	2.819	2.823	0.2357	0.0218
0.30	2.582	2.572	0.2836	0.0806
1.00	2.404	2.390	0.2773	0.0804
3.00	2.214	2.198	0.2436	0.0678
10.0	2.051	2.036	0.2175	0.0588
30.0	1.960	1.945	0.2047	0.0546
100.	1.912	1.897	0.985	0.0526
300.	1.891	1.876	0.1960	0.0518

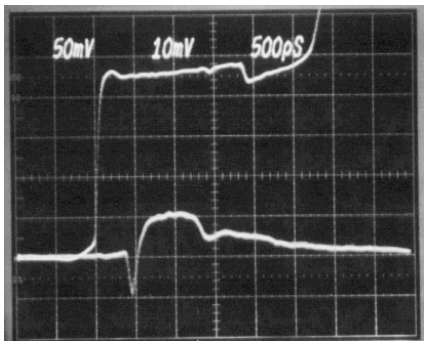
Table 2: The frequency dependent resistance matrix

Freq (GHz)	R11	R22	R12	R13
0.01	2.076	2.064	0.2405	0.2170
0.03	2.176	2.142	0.2411	0.1392
0.10	2.361	2.350	0.2153	0.0734
0.30	2.627	2.622	0.2029	0.0741
1.00	3.664	3.678	0.3731	0.1416
3.00	6.524	6.551	0.8709	0.3193

Table 2: The frequency dependent resistance matrix

Freq (GHz)	R11	R22	R12	R13
10.0	13.82	13.84	1.981	0.6917
30.0	26.25	26.30	3.747	1.280
100.	50.44	50.64	7.022	2.392
300.	91.35	91.96	12.256	4.209

The waveforms for the coupled noise on the passive conductors are obtained from both measurement and simulation. For example, the measurement and simulation results for the input waveforms (at the output connector of the pulse generator) and the waveforms at the far end of the adjacent quiet line are shown in Fig. 4. Note that the mea-



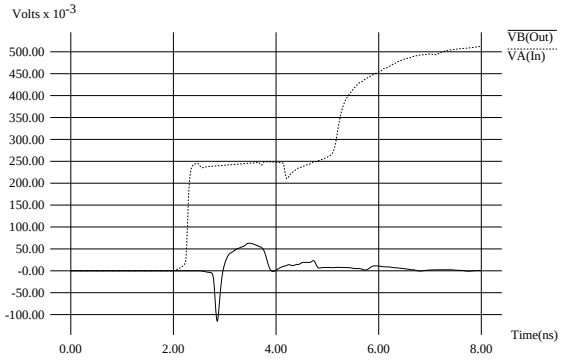


Fig. 4: The input waveforms and voltage responses at the far end of the adjacent quiet conductor by measurement and simulation for Example 2

surement input waveform (VA) and the output waveform (VB) are plotted using different voltage scale. As shown in this figure, the measurement and the simulation results are in good agreement. The simulation of this circuit took only 23s of CPU time.

6. Summary

An efficient method for the transient analysis of lossy and dispersive transmission lines has been reported in this paper. Practical examples have been presented to demonstrate the efficiency and accuracy of the proposed method. This capability has been incorporated into AS/X, the IBM production circuit simulator, and has been used routinely in the design of packages and interconnects.

References

[1] A. J. Gruodis and C. S. Chang, "Coupled Lossy Transmission Line Characterization and Simulation," *IBM J. Res. Develop.*, vol. 25, pp. 25-41, Jan. 1981

[2] A. Semlyen and A. Dabuleanu, "Fast and accurate switching transient calculations on transmission lines with ground return using recursive convolution," *IEEE Trans. Power Apparatus Systems*, vol. PAS-94, pp.561-571, 1975

[3] R. E. Matick, *Transmission Lines for Digital and Communication Networks*, McGraw-Hill, New York, 1969, Chapters 7 and 8.

[4] W. T. Weeks, L. L. Wu, M. F. McAllister, and A. Singh, "Resistive and Inductive Skin Effect in REctangular Conductors", *IBM J. Res. Develop.*, vol. 23, pp. 652-660, 1979.

[5] A. R. Djordjevic, T. K. Sarkar and R. F. Harrington, "Time domain response of multiconductor transmission lines," *Proc. of the IEEE*, vol.75, pp.743-764, June 1987.

[6] R. Griffith and M. Nakhla, "Time domain analysis of lossy coupled transmission lines," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-38, pp. 1480-1487, Oct. 1990.

[7] F. Y. Chang, "Transient simulation of VLSI interconnects (A tutorial)," *Proc. IEEE Multi-Chip Module Conf.*, March 1992.

[8] J. E. Bracken, V. Raghavan and R. A. Rohrer, "Interconnect Simulation with Asymptotic Waveform Evaluation," *IEEE Trans. Cirsuits Syst.*, vol. 39, pp. 869-878, Nov. 1992.

[9] S. Lin and E. S. Kuh, "Transient Simulation of Lossy Interconnects Based on the Recursive Convolution Formulation," *IEEE Trans. Cirsuits Syst.*, vol. 39, pp. 879-892, Nov. 1992.

[10] C. Gordon, T. Blazeck, and R. Mittra, "Time-domain simulation of multiconductor transmission lines with frequency dependent losses," *IEEE Trans. Computer-Aided Design*, vol. 11, pp. 1372-1387, Nov. 1992.

[11] T. V. Nguyen, "Transient Analysis of Lossy Coupled Transmission Lines Using Rational Function Approximation," *2nd Topical Meeting on Electrical Performance of Electronic Packaging*, pp. 172-174, Oct. 1993.

[12] E. Chiprout and M. Nakhla, "Addressing High-spped Interconnect Issues in Aymptotic Waveform Evaluation", *Proc. 30th DAC*, pp. 732-736, June 1993.

[13] X. Huang, V. Raghavan and R. A. Rohrer, "AWEsim: A program for the efficient analysis of linear(ized) circuits," *Tech. Digest of the IEEE Intl. Conf. on Computer-Aided Design*, pp. 534-537, Nov. 1990.

[14] H. Liao, W. W. Dai, R. Wang, and F. Y. Chang, "S-Parameter Based macro Model of Distributed-Lumped Networks Using Exponentially Decayed Polynomial Function," *Proceedings of the 30th ACM/IEEE Design Automation Conference*, pp. 726-731, 1993.

[15] T. V. Nguyen, "Recursive Convolution and Discrete Time Domain Simulation of Lossy Coupled Transmission Lines," submitted to *IEEE Trans. on CAD* for publication.

[16] N. Balabanian and T. A. Bickart, *Eletrical Network Theory*, John Wiley & Sons, 1969.

[17] D. W. Marquardt, "An Algorithm for Least Square Estimation of Nonlinear Parameters," *SIAM J.*, vol. 11, pp. 431-441, 1963.

[18] R. J. Bowman and C. C. Brewster, "Determining the Zeros and Poles of Linear Circuit Networks Using Function Approximation," *IEEE Trans. Computer-Aided Design*, vol. CAD-6, pp. 678-689, July 1987.

[19] H. W. Bode, *Network Analysis and Feedback Amplifier Design*, New York:Academic, 1945.

[20] R. Griffith and M. S. Nakhla, "Mixed Frequency/Time Domain Analysis of Nonlinear Circuits," *IEEE Trans. Computer-Aided Design*, vol. 11, pp. 1032-1043, Aug. 1992.

[21] W. T. Weeks, "Calculation of Coefficients of Capacitance of Multiconductor Transmission Lines in the Presence of a Dielectric Interface," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-18, pp. 35-43, 1970.