A New Efficient Waveform Simulation Method for RLC Interconnect *via* Amplitude and Phase Approximation

Xiaodong Yang, Walter H. Ku,
Dept. of Electrical and Computer Engineering,
University of California, San Diego.
La Jolla, CA 92093-0407

Chung-Kuan Cheng
Dept. of Computer Science and Engineering,
University of California, San Diego
La Jolla, CA 92093-0114

Abstract

In this paper we use a segmented Chebyshev orthongonal polynomial expansion approach to approximate amplitude and phase response for RLC interconnect simulation. This method has been shown to be efficient in reducing frequency sampling point number or expansion order. Experiments also show that the number of sampling points is not necessarily dependent on the interconnect circuit size, which makes possible large network simulation. Additionally, because of the smoothing-effect of the *atan* function in evaluation, the phase response, which has major impact on the transient edge of output response, can be easily approximated. Experiments show the proposed simulation approach achieves 30-50 times speed-up over spice3f4.

1. Introduction

With deep sub-micron technology and clock frequencies in the GHz range, signal integrity problems have a significant adverse effect on the proper functioning and performance of VLSI systems. Besides conventional timing analysis such as delay estimation, more detailed signal waveform simulation becomes increasingly important to verify a circuit's or system's performance. However, due to the overwhelming size of interconnect circuits, approximations have to be purposely introduced with relaxed accuracy to achieve simulation speed.

In recent years, moment matching [1-3] and Krylov space projection based model order reduction techniques [4-8] were introduced for interconnect circuit analysis or simulation. These techniques use a reduced set of poles to approximate an original system while preserving the system's controllability and observability[9]. To achieve satisfactory speed-up, the reduced order model should be small enough to avoid the overhead in pole extraction and solving a reduced order dense system. If a system has a few well-separated dominant poles, the above methods would be quite efficient. However, for the system with continuously distributed or clustered poles the pole extraction procedures may converge very slowly or converge to spurious poles[10]. For multipoint moment matching or model order reduction approach, the error is strongly relied on the selection of expansion points. Moreover, it is still an unsolved problem on how to determine a suitable reduce order without a prior knowledge.

Recently, amplitude and phase response was shown to be efficient in RLC interconnect delay estimation[11]. Here, we extend it into waveform simulation. In a complex frequency domain s, transfer function is a singular function when s approaches poles. Therefore, poles have to be extracted to serve as the bridge between time and frequency domain. However, in Fourier transform domain, Amplitude and Phase response (AP response) are continuous functions since no pole exists in the imaginary axis for a lossy stable system. Consequently, it is possible to approximate AP response directly without pole extraction. Although SPICE can provide accurate information about AP response, it can not be used for large network simulation because of its complexity.

In this paper we present an efficient AP response approximation method for RLC interconnect simulation using a Chebyshev polynomial expansion method, which converges quickly and can achieve near optimal approximation under the MinMax criteria (minimize the maximal error). The proposed approach can automatically select the expansion points and also the expansion order to meet a predefined frequency domain error tolerance.

The paper is organized as follows: in section 2 the Chebyshev orthongonal polynomial expansion based AP response approximation method will be discussed. In section 3 experimental results are presented followed by the contribution and conclusions of this paper.

2. Amplitude and Phase response approximation using segmented Chebyshev orthogonal polynomial expansion

2.1 Background of AP Response

First we give some basic formulas to make the further discussion clear. The transfer function $H(jw)$ is defined as[12]:

$$H(jw) = H(s)\big|_{s = jw} = \int_{-\infty}^{\infty} h(t)e^{-jwt}dt \qquad (2.1)$$

For analytical convenience, $H(jw)$ is usually separated into amplitude response $A(w)$ and phase response $\Phi(w)$, which are actually the superimposed result of poles and zeros.

$$H(jw) = R(w) + jI(w) = A(w)e^{j\Phi(w)} \qquad (2.2)$$

$$A(w) = \sqrt{R(w)^2 + I(w)^2} = |H(jw)| \qquad (2.3)$$

$$\Phi(w) = \operatorname{atan}\left(\frac{I(w)}{R(w)}\right) = \angle H(jw) \qquad (2.4)$$

Given $A(w)$, $\Phi(w)$ and a normalized step function as an input excitation, the output response of an aggressor net is evaluated by:

$$V_o(t) = \frac{1}{2} + \frac{1}{\pi}\int_0^{\infty} \frac{A(w)}{w}\sin(wt + \Phi(w))dw \qquad (2.5)$$

For a victim net:

This work was supported in part by grants from the NSF Project MIP-9529077 and the California MICRO program.

$$V_o(t) = \frac{1}{\pi}\int_0^{\infty}\frac{A(w)}{w}\sin(wt + \Phi(w))dw \qquad (2.6)$$

If an input signal is not a step function, it can be easily counted into $A(w)$ and $\Phi(w)$ by

$$A^{\dagger}(w) = A(w) \cdot A_I(w), \Phi^{\dagger}(w) = \Phi(w) + \Phi_I(w) \qquad (2.7)$$

where $A_I(w), \Phi_I(w)$ are the amplitude and phase of an input signal respectively.

2.2 Segmented Chebyshev Orthongonal Polynomial Expansion

Lemma 1 *Among the same degree of polynomials, minmax approximation achieves the smallest maximum deviation from the original function f(x).*[13]

With minmax polynomial approximation the maximum error is minimized and the error is uniformly distributed throughout the approximation interval. However, in general, it is very difficult to find a minmax polynomial. It is shown[13] that the Chebyshev orthogonal polynomial expansion is almost identical to the minmax approximation while it is easy to compute. The *nth* order Chebyshev orthogonal polynomial is defined as:

$$Tn(x) = \cos(n \cdot cos^{-1}(x)), \quad x \in [-1, 1] \qquad (2.8)$$

It can be expressed in a recursive form:

$$T_0(x) = 1 \qquad (2.9)$$

$$T_1(x) = x \qquad (2.10)$$

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x) \qquad n \geq 1 \qquad (2.11)$$

The *nth* order Chebyshev polynomial $T_n(x)$ has *n* zeroes and *n+1* extrema (maxima and minima) in the interval [-1,1]:

$$x_{zero} = \cos\left(\frac{(k-0.5)}{n}\pi\right), \quad k=1,2....n \qquad (2.12)$$

$$x_{extrema} = \cos\left(\frac{k}{n}\pi\right), \quad k=0,1,2...n \qquad (2.13)$$

$$T_n(x_{extrema}) = \pm 1 \qquad (2.14)$$

Using the orthogonal property of Chebyshev polynomials, a continuous function *f(x)* can be approximated as[13]:

$$f(x) \approx \frac{1}{2}C_0 + \sum_{j=1}^{N-1} C_j T_j(x) \qquad (2.15)$$

where N is the expansion order or the number of sampling points. The Chebyshev polynomial coefficient C_j is calculated using zeroes as the sampling points:

$$Cj = \frac{2}{N}\sum_{k=1}^{N} f(x_k)T_j(x_k) \qquad j=0,1,....N-1$$

$$= \frac{2}{N}\sum_{k=1}^{N}\left\{f\left(\cos\left(\frac{k-0.5}{N}\pi\right)\right) \cdot \cos\left(\frac{j(k-0.5)}{N}\pi\right)\right\} \qquad (2.16)$$

Under the minmax criteria, this expansion can achieve near optimal approximation among all the polynomials with a order of N. However, the drawback of (2.16) is that all the sampling points will be totally different if the order is changed. Consequently, all the sampling points used in a previous order can not be reused anymore. Since we can not determine an appropriate expansion order in advance, the order should better be determined incrementally. We found the following formula is more suitable without detrimenting any approximation quality, where extrema are used as sampling points instead of zeroes:

$$Cj = \frac{f(1) - (-1)^j f(-1)}{N} + \frac{2}{N}\sum_{k=1}^{N-1}\left\{f\left(\cos\left(\frac{k}{N}\pi\right)\right) \cdot \cos\left(\frac{k(j-1)}{N}\pi\right)\right\}$$

$$j=1,2....N-1. \qquad (2.17)$$

This formula is known as Clenshaw-Curtis Quadrature[13]. It is easy to see that if the order N is doubled, all the present sampling points can be reused in the next step.

The side-effect of (2.17) is that during the iteration the order may increase quickly and the sampling points will aggregate at the two ends of an approximation interval. Consequently, the error may be large in the middle part. To overcome this problem and reduce the expansion order, we divide the whole frequency range into several segments. Since the variation in each sub-interval becomes smaller, the expansion orders can be reduced to small numbers. The sampling point aggregation problem is also avoided. Another advantage of a segmented Chebyshev expansion is that it becomes more flexible to choose an expansion order for different frequency range. For example, the sampling order and error tolerance can be relaxed in high frequency range to reduce the computation cost since the higher frequency response is usually less important than lower frequency.

Before using Chebyshev expansion to approximate $A(w)$ and $\Phi(w)$, the approximation frequency range [*wa,wb*] is mapped into [-1,1] using a linear transformation. The sampling frequency is determined by

$$w_k = \frac{1}{2} \cdot \left(\cos\left(\frac{k}{N}\pi\right) \cdot (wb - wa) + (wa + wb)\right), \quad k=0,1...N\text{-}1 \qquad (2.18)$$

Given an input signal and a sampling frequency w_k, the node voltage $\boldsymbol{V(jw_k)}$ can be evaluated efficiently with the method to be introduced in section 2.3. Amplitude and phase response are computed by (2.3) and (2.4). Since in (2.4) the phase response is wrapped into $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$, it should be unwrapped first before an expansion. Once the Chebyshev polynomial coefficients are computed using (2.17) the approximated AP response can be evaluated easily by (2.15). Usually, the variation of phase response over the frequency range is much smaller than amplitude response because of the smoothing-effect of the ***atan*** function in (2.4). This property makes it easier to approximate the transient edge accurately. The reason is that a phase response has a major impact on transient edges, while an amplitude response has a major impact on the waveform around origin and final state.

2.3 Frequency Sampling Point Evaluation

In Chebyshev polynomial expansion, the complexity in evaluating the node voltage at Chebyshev frequency sampling points determines the efficiency of the whole procedure. In this section we present an efficient numerical method for solving a linear system. In this paper, the circuit structure is restricted to RLC tree structure with capacitive coupling inside, in which resistance and inductance are connected in series and capacitances are shunted to ground or connected as the coupling links.

Assume no mutual inductance existing inside the circuit, the conventional nodal analysis can be used directly for analysis. In this formulation, the matrix size is one half of the MNA formulation.

$$Y(jw) \cdot V(jw) = Is(jw) \qquad (2.19)$$

where the admittance matrix *Y(jw)* is:

$$Y(jw) = Arl \cdot (R + jwL)^{-1} \cdot Arl^T + jwAc \cdot C \cdot Ac^T \qquad (2.20)$$

R,L,C are all diagonal matrices. *Arl* is the incidence matrix of resistance or inductance. *Ac* is the incidence matrix for capacitance. The matrix inverse in (2.20) is trivial since *R* and *L* are all diagonal matrices.

The basic method for solving a linear equation is Gaussian elimination or LU decomposition. For large sparse system, this procedure will cause a lot of fill-in elements and consequently increase computation cost and storage. In general, the complexity is O(N^3). In this section we present two efficient numerical methods which will not cause any fill-in elements in the whole computation process. It has the advantage in both computation and storage.

i) Single net RLC tree with grounded capacitance

Theorem 1: *For a single-net RLC tree with grounded capacitance, the nodal analysis equation of (2.19) can be solved in O(N) operations, where N is the number of nodes. The LU decomposition can be processed without any fill-in elements.*

Proof: Without loss of generality, we take the circuit of Fig. 2.1 as an example to visualize the proof.

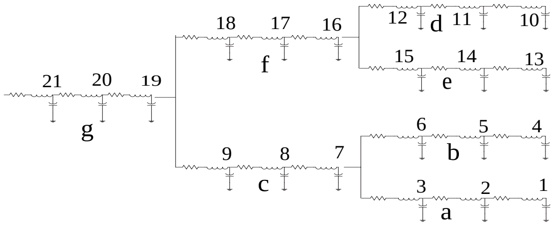

Fig. 2.1 An RLC tree numbered in reversed DFS order

1) Number the node of an RLC tree in a reversed Depth-First-Search (DFS) or Breadth-First-Search (BFS) order.

By numbering the nodes in a reversed DFS order, the structure of the admittance matrix *Y* is shown in Fig. 2.2.

2) Partition matrix Y recursively as shown in Fig. 2.2.

$$Y = \begin{bmatrix} A & B \\ C & D \end{bmatrix} = LU = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \cdot \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix} \qquad (2.21)$$

$$B = C^T \qquad (2.22)$$

In each partition, *A* is an upleft-most tridiagonal matrix. The sub-blocks of *A* correspond to disjointed branches in an RLC tree with consecutive node numbers. A parent branch and child branches can not co-exist in a same block matrix *A*. In Fig. 2.2, the elements inside a circle are called connection-elements, which connect the child-branches with their parent branch. The elements inside a square are called junction elements, which correspond to the junction node such as node 7,16, 19.

3) $$A = L_{11} \cdot U_{11} \qquad (2.23)$$

Only 3N$_A$ operations are needed for this LU decomposition[14]. N$_A$ is the size of A. No extra storage is needed since L_{11} and U_{11} are all two-diagonal matrices and all the diagonal elements of L_{11} are equal to 1.

4) $$B = L_{11} \cdot U_{12}, C = L_{21} \cdot U_{11} \qquad (2.24)$$

Lemma 2: $U_{12}=B$ (2.25)

Lemma 3: *Only N_c operations are needed to compute L_{21}, where N_c is the number of connection elements in C. L_{21} has the same structure as C.*

5) $$Y^* = L_{22} \cdot U_{22} = D - L_{21} \cdot U_{12} \qquad (2.26)$$

D has the same structural characteristics as *Y* because of the node number ordering and partition method.

Lemma 4: *Y^* has the same structure as D since $L_{21} \cdot U_{12}$ is a diagonal matrix and only has N_j non-zero diagonal elements, where N_j is the number of junction elements in D related to the coupling elements in block matrix C or B.*

6) Repeat the procedures of 2)-5) for Y^* until it is empty.

Therefore, the LU decomposition of the whole matrix can be processed recursively without any fill-in elements. The total operations is *$3N + N_C + 2N_J - 1$*. *N* is the number of nodes in the circuit. N_C is the number of branches in a RLC tree. N_J is the number of junction nodes. The whole procedure is a in-place operation. Therefore, for single net RLC tree the evaluation at each frequency point can be processed in *O(N)* operations and without any extra storage. ❐

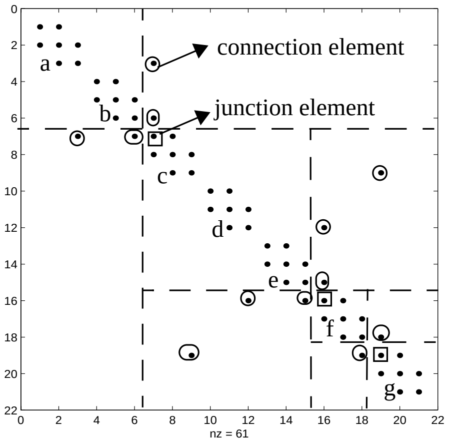

Fig. 2.2 Single net RLC tree admittance matrix structure of the circuit in Fig. 2.1

ii) Multi-net RLC trees with capacitive coupling

Because of the capacitive coupling between trees or branches, a lot of new elements would appear in the admittance matrix structure, which are named as coupling-elements. Their number is usually far larger than other elements. With conventional LU decomposition method these coupling-elements will cause huge number of fill-in elements during the process and damage the sparsity.

Theorem 2. *The nodal analysis equation of (2.19) for multi-RLC-tree with capacitive coupling can be processed in O(N_{nz}) operations. N_{nz} is the number of non-zero elements of an admittance matrix.*

Proof: Similar to a single net tree, we first number the node of each RLC-tree in a reversed BFS or DFS order. The admittance matrix *Y* is split into real and imaginary parts.

$$Y = Y_R + Y_I = \frac{(Y + Y^H)}{2} + \frac{(Y - Y^H)}{2} \qquad (2.27)$$

Y^H is the conjugate matrix of Y. Y_R is a symmetric positive definite and diagonal dominant matrix. It has the same matrix structure as several trees without coupling components inside. From **Theorem 1,** we have the following lemma:

Lemma 5 *The LU decomposition of Y_R can be processed in O(N) operations, N is the size of Y_R.*

To preserve the matrix sparsity in solving the linear equation, a Conjugate Gradient method[14] is applied by using Y_R as a precondition matrix. It is a good preconditioner since in some sense Y_R is a good approximation of *Y*. Besides, Y_R is very easy to solve because of the property presented in **Lemma 5**. The procedure of Precondition Conjugate Gradient method is shown below:

```
V=PCG(Y,Is)
1. Set initial solution $V_0$;  Set Error_tolerance;  Error= $\infty$ ;
2. $r = Is - Y \cdot V0$ ;  k=0;
3. while Error > Error_tolerance
   {  $z0 = z$ ;  $Y_R \cdot z = r$ ;
      if k=0    $\beta = 0$ ;
      else      $\beta = \frac{(r, z)}{(r0, z0)}$ ;
      $P = z + \beta \cdot P0$ ;  $P0 = P$ ;
      $\alpha = \frac{(r, z)}{(Y \cdot P, P)}$ ;  $V = V0 + \alpha \cdot P$ ;  $V0 = V$
      $r0 = r$ ;  $r = r0 - \alpha \cdot Y \cdot P$ ;  $Error = \frac{\|V1 - V0\|_\infty}{\|V0\|_\infty}$ ,
   }
```

It can be proven that the above procedure can find the exact solution in N steps at most. Particularly, if all the eigenvalues of $Y_R^{-1}Y$ are equal, it converges in one step. Experiments show that the convergence speed is very fast due to the precondition. By using previous frequency points as initial solution, the speed can be accelerated further. In the low frequency range, usually 1-2 iterations can achieve a fairly accurate result. On average 4-5 iterations are needed to meet the error requirement. Thus we empirically assume that the converge speed does not rely on N. In each iteration only one matrix-vector multiplication is needed, which dominates the total complexity. By using sparse matrix techniques, the complexity is O(N_{nz}). N_{nz} is the number of nonzero elements of *Y*. Therefore, for multi-net RLC trees with capacitive coupling the nodal analysis equation (2.19) can be solved with O(N_{nz}) operations without destroying any sparsity.❒

2.4 Session summary

The amplitude and phase response is approximated in the range of [*wa*,*wb*]. *wa* is set as 0. *wb* is determined by $wb = 2\pi \cdot f_{knee} = \frac{\pi}{tr}$ [15]. As a summary of section 2.2 and 2.3, the whole procedure is shown below:

Input: Interconnect Circuit Netlist

Output: Chebyshev Coefficients of each node

```
1. Parse circuit; Renumber the node of each tree in a reversed BFS
   order or DFS order; Sparse matrix storage of R, L, C, Arl, Ac;
2. $Yc = Ac \cdot C \cdot Ac^T$ ;
3. Partition frequency range: [wa[1:Ns],wb[1:Ns]];
   Set error_tolerance and initial sampling order N[1:Ns];
4. for k=1:Ns
   { V=0; C[k,:]=0; End_Flag=False; I=1;
     w[1:N[k]]=SamplingFrequencyMapping(wa[k],wb[k],N[k]);
     while End_Flag=False
     {  for i=1:N[k]
        { Initial_Solution(V);
            $Yrl = Arl(R + jw[i]L)^{-1}Arl^T$ ; $Y = Yrl + jw[i]Yc$ ;
            $YR = OptimizedLU(YR)$ ;
            V=OptimizedComplexPCG(Y,YR,Is);
            [A,P]= $AP(V)$ ;
        }
          [C[k,:],error]=ChebEvaluation( A, P ,N[k],I,C[k,:]);
       if error >error_tolerance
        [A,P,w[1:N[k]]]=IncrementalUpdating(wa[k],wb[k],N[k],A,P );
          $N[k] = 2^{I-1}N[k]$ , I++;
      else
          End_Flag=True
     } }
```

OptimizedLU(): LU decomposition without fill-in elements.

OptimizedComplexPCG()*:* a modified PCG procedure, in which some computation strategies are used to reduce the number of complex operations. For example, since most of the elements in Y only have imaginary parts, the operation of complex multiplications can be reduced greatly.

AP()**:** amplitude and phase response computation from the solution of node voltage.

IncrementalUpdating(): increasing the expansion order, saving the solved AP response.

ChebEvaluation()*:* Chebyshev polynomial coefficient computation, AP response approximation by Chebyshev polynomial expansion and error evaluation.

With the AP response represented by Chebyshev coefficients, the time domain waveform can be evaluated using inverse Fourier transform integration or inverse FFT.

3. Experimental Results

The above proposed simulation method is implemented in C language on unix. Several industrial interconnect circuits are used for the experimental test. The runtime is tested on a Sun Sparc5 workstation.

Table 1: Experimental Result

Circuit	R#	L#	C#	CPU Time(s) (AP)	CPU Time(s) (spice3f4)
RLC1	100	90	413	0.62	19.98
RLC2	180	162	1394	1.40	59.75
RC1	100	0	413	0.15	3.87
RC2	180	0	1394	0.27	12.18

RC1 and RC2 are generated by setting inductance value in RLC1 and RLC2 as zero. In testing runtime, all RLC circuits and RC circuits are tested with expansion order N=16 and N=8 respectively. The input signal's rise time is 65ps.

Fig.3.1 and Fig.3.2 show the calculated amplitude and phase response for RLC2 and RC2, which are excited with the same input pattern. In the RLC2 case, there are some spikes existing around the frequency junction point for the amplitude response. It is because the expansion in different segments may not be continuous at the junction points. These spikes may cause small errors in time domain evlaution. An extra procedure could be used to smooth the curve around these points. The phase response, as we mentioned before, is easier to approximate. Therefore the transient edge can be approximated more accurately than origin and final state. Fig. 3.3 and Fig. 3.4 show the node voltage comparison between SPICE and proposed simulation method for RLC2 and RC2 respectively. The errors are very small especially for the transient edges. In RLC2 case, the 50% delay is about 109.8ps, while it is 63.8ps for RC2 case. Clearly, if inductance is ignored, large erros may occur.

Fig. 3.5 and Fig. 3.6 are the crosstalk voltage comparison for the victim net. From the simulation results, we can clearly see the significant effect of inductance on the peak value and the pulse width of crosstalk voltage.

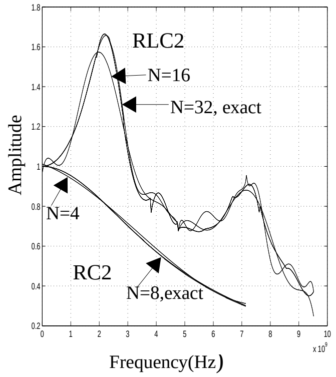

Fig. 3.1 Amplitude approximation for RLC2 and RC2

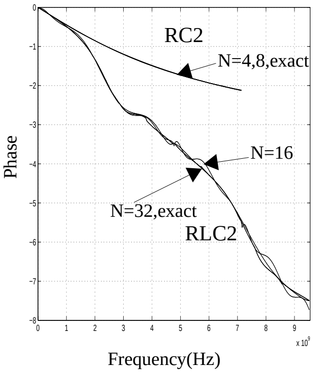

Fig. 3.2 Phase approximation for RLC2 and RC2

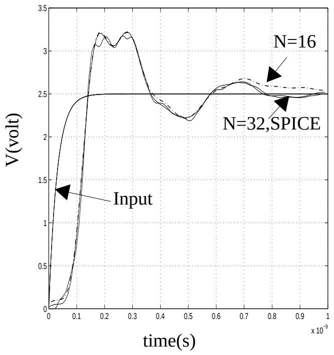

Fig. 3.3 RLC2 output response N=16, N=32

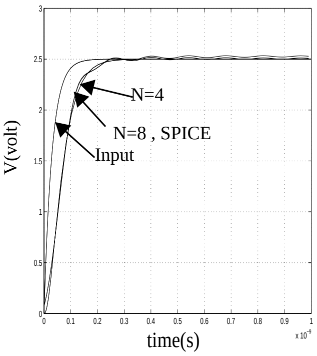

Fig. 3.4 RC2 output response N=4, N=8

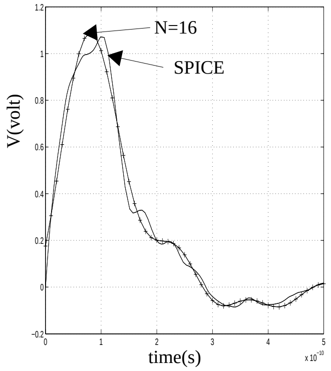

Fig. 3.5 RLC2 Crosstalk Voltage

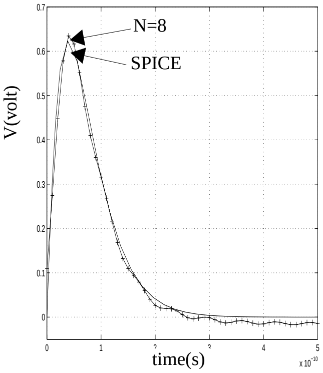

Fig. 3.6 RC2 Crosstalk voltage

4. Conclusions and future work

A new efficient time domain waveform evaluation approach for RLC interconnect circuits simulation was proposed by using amplitude and phase response approximation. Using a segmented Chebyshev orthogonal polynomial expansion over a predefined frequency range, AP response can be approximated efficiently, especially for the phase response which has more impact on transient edges. Efficient numerical methods for solving a linear system are also developed to implement above procedures. In experiments the proposed method can achieve 30~50 times speed-up over SPICE3f4 with negligible errors. The number of sampling points is not necessarily dependent on the circuit size, which makes possible large circuit simulation. As our future work, we will try to extend our current algorithm to more general circuit topology and also consider the effect of mutual inductance.

5. References

[1] L.T.Pillage, R.A. Rohrer,”*Asympotic Waveform Evaluation for Timing Analysis*”, IEEE, Tran. on CAD, Apr. 1990.

[2]C.L. Ratzlaff, L.T.Pillage,”*RICE: Rapid Interconnect Circuit Evaluation Using AWE*”. IEEE, Trans. CAD of Integrated Circuit and System. Vol.13. No.6. June.1994.

[3]E. Chiprout and M.S. Nakhla. “*Transient analysis of high-speed mcm networks using complex frequency hopping”*. Proc. IEEE Multi-Chip Module Conference, Mar,1993.

[4]R.W.Freund, P.Feldmann, “ *Reduced-order model-ling of large passive linear circuits by means of the SyPVL algorithm*”, Proc. ACM/IEEE DAC, June, 1996

[5] L.M.Elfadel, D.D.Ling, “*Block rational Arnoldi algorithm for multipoint passive model-order reduction of multiport RLC networks*”, Proc. ACM/IEEE ICCAD Nov.1997.

[6] K.J. Kerns, I.L. Wemple, A.T.Yang, “*Stable and efficient reduction of substrate model networks using congruence transforms”*. ACM/IEEE Proc. ICCAD Nov. 1995

[7] A. Odabasioglu, M.Celik, L.T. Pileggi, “ *PRIMA: Passive reduced-order interconnect macromodeling algorithm*”,Proc. ACM/IEEE ICCAD Nov. 1997

[8] QingJian Yu, Janet M.Wang, Ernest S. Kuh. “*Multipoint Moments Matching Model for Multiport Distributed Interconnect networks*”. ACM/IEEE Proc. ICCAD. Nov. 1998

[9]Chi-Tsong Chen, Holt, *Linear System Theory and Design*, CBS College Publishing

[10]Karl Meerbergen and Dirk Roose,”*Matrix transformation for computing rightmost eigenvalues of large sparse non-symmetric eigenvalues problems*”. IMA Journal of Numerical Analysis. 1996. P297-346.

[11] Xiaodong Yang, Walter H. Ku, Chung-Kuan Cheng, “ *RLC Interconnect Delay Estimation via Moments of Amplitude and Phase Response*”, ACM/IEEE Proc. ICCAD, Nov. 1999.

[12] C.D.Mcgillem, G.R. Cooper, *Continuous and Discrete Signal and systems*, Holt.Rinehart & WInston Inc.

[13] Kendall E. Atkinson, *An Introduction to Numerical Analysis*, 2nd Edition, John Wiley and Sons

[14] Gene H. Golub. C.F.V.Loan, *Matrix Computation*, Johns Hopkins.

[15] Howard W. Johnson, Martin Graham. *High-Speed digital Design*. Prentice Hall.