Hierarchical Symbolic Piecewise-Linear Circuit Analysis

Junjie Yang, Sheldon X.-D. Tan, Zhenyu Qi, Martin Gawecki

Department of Electrical Engineering
University of California, Riverside, CA 92521, USA

ABSTRACT

Abstract— *This paper presents a hierarchical transient analysis method for piecewise-linear (PWL) circuits suitable for early stage verification of analog and mixed-signal circuits. The new method is based on a novel parameterized modeling of PWL devices, which results in very compact circuit matrices compared to existing PWL simulation algorithms based on ideal diode models. The new PWL symbolic analysis features exact symbolic solution of linear systems via graph based hierarchical analysis technique and PWL modeling of nonlinear devices. The resulting parameterized PWL circuit equations are solved by a modified Katzenelson event-driven algorithm to obtain transient responses for all subcircuits based on the symbolic solutions. Since PWL components are modeled symbolically in the new method, symbolic solutions are built only once and used repeatedly throughout the entire simulation, which makes the new PWL simulation algorithm more efficient than the Newton-Raphson (NR) based numerical methods which require solving of linear equations (LU decomposition) at every step in NR iterations. Experimental results on a number of analog circuits show that the proposed hierarchical method outperforms the commercial simulator, SIMetrix and the flat PWL simulator.*

1. INTRODUCTION

Piecewise-linear(PWL) modeling has been a preferable choice for the behavioral-level simulation at the early stage of nonlinear analog design because of its easy operability and flexibility. Piecewise-linear simulation uses piecewise-linear approximation to describe general (weakly or hard) nonlinear behavior. Piecewise-linear simulator offers several advantages over traditional Newton-Raphson based nonlinear circuit simulators. It can model the different-level components in a uniform way and is highly suitable for the mixed analog-digital circuits. Besides, PWL algorithms have excellent convergence properties, especially, for hard nonlinear circuits due to the nature of piecewise-linear modeling. Simplified PWL models can be used to obtain considerably faster simulation with the controlled loss of accuracy. [9, 3]

There are two kinds of PWL simulation tools for DC and transient analysis in the past. The first category is pure numerical methods such as PLATO [7], Popcorn [8], constant system matrix method [9], SPECS [3], PLANET [5] etc. PLATO uses multi-rate integration techniques together with efficient sparse matrix methods to solve piecewise linear equations; SPECS formulates equations on the state variable basis using tree/link hybrid analysis, but it assumes all branch currents are piecewise constant in time domain which precludes inductors. Also in PLATO and PLANET tools, the ideal diode model is used to implicitly model each segment of PWL components and the resulting circuit equations lead to Linear Complementary Problems (LCP), which are solved numerically by Katzenelson's algorithm [6].

*This work is funded by NSF CAREER Award CCF-0448534, NSF Grant OISE-0451688 and UC Regent's Faculty Fellowship(04-05).

However, such ideal-diode based PWL component modeling will lead much larger circuit matrix as $\sum k_i$ extra rows and columns are added, where k_i is the number of segments of the ith PWL component. For a circuit with 20 PWL components and 5-segment approximation for each I-V curve, there will be 100 extra rows and columns. Obviously the large circuit matrix, which in turn will increases simulation time, will be generated when the number of the PWL components or the number of segments becomes large. Even for the Popcorn [8] and the constant system matrix method [9] in which no ideal diode model is used, there are a lot of matrix operations and updates whenever the network state switch from one region into another linear region.

The second PWL simulation category is by symbolic analysis which includes Mathematica-based techniques [11], behavioral modeling [1] and recent complementary-decision diagrams method [10, 2]. But for the first two methods intensive human interaction is still required, which makes them unsuitable for analyzing large PWL circuits. The complementary-decision diagrams method is also based on the ideal diode model of PWL components and it can solve the LCP problem symbolically, which leads to explicit symbolic expressions for PWL circuits. But this method still suffers from the large matrix problems as matrix size grows with the product of the number of PWL segments and the number of PWL components. This makes symbolic analysis very difficult as flat symbolic analysis (even with DDD techniques) still can't analyze very large circuits (more than 100 nodes) and symbolic hierarchical approach to the LCP problem is difficult (if possible at all) as there is not identity matrix (thus inverse matrix) for solving the LCP problem [4].

In this paper a general hierarchical piecewise symbolic analysis approach is proposed which features parameterized PWL modeling, hierarchical symbolic analysis and event-driven (Katzenelson's method) PWL analysis scheme. Our parameterized PWL Modified Nodal Analysis(MNA) matrix is very compact compared with ideal-diode based circuit formulation method. Hierarchical scheme allow much larger nonlinear analog circuits (actually without limitation) to be analyzed symbolically.

The paper is organized as follows: Section 2 introduces the parameterized representation of PWL circuits; Section 3 reviews the concept of the DDD graphs for symbolic analysis of linear circuits and introduces the time-domain DDD; Section 4 reviews the hierarchical symbolic analysis of PWL circuits; Section 5 presents the general Katzenelson's algorithm; Following that, Section 6 proposes our hierarchical method for PWL transient analysis; Section 7 gives the experimental results for a number of nonlinear circuits which are compared with SIMetrix [12] and the flat PWL simulator; Section 8 concludes the paper.

2. PARAMETERIZED REPRESENTATION OF PWL DEVICES

For the PWL I-V curve of a two-terminal device shown in Fig. 1, the MNA stamp can be written below for any segment in which the PWL component operates

$$\begin{array}{c} \\ n_1 \\ n_2 \\ n_{pwl} \end{array} \begin{array}{c} \begin{array}{ccc} n_1 & n_2 & n_{pwl} \end{array} \\ \left[\begin{array}{ccc|c} & & & 1 \\ & & & -1 \\ \hline 1 & -1 & -R_i & \end{array}\right] \end{array} \left[\begin{array}{c} V_{n_1} \\ V_{n_2} \\ I_{pwl} \end{array}\right] \left[\begin{array}{c} \\ \\ V_i \end{array}\right]$$

here R_i is the reciprocal of the slope of the ith segment and V_i is its intercept with x-axis. Since this R_i and V_i are symbolic variables for any segment that PWL component operates, this MNA expression is the same for all the segments of one PWL component. For each PWL component there is just one extra row needed no matter how many segments are used to represent the I-V curve of the nonlinear component, which reduces the MNA size drastically compared to ideal diode model for large nonlinear circuits.

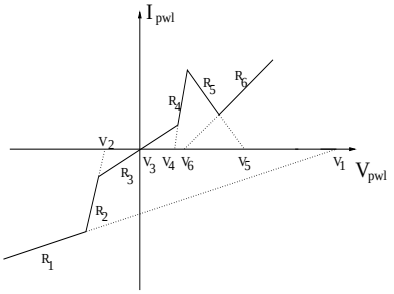

Figure 1: The piecewise-linear I-V curve for a two-terminal nonlinear component.

Similarly, we can also write the stamps of PWL voltage-controlled voltage source as

$$\begin{array}{c} \\ n_1 \\ n_2 \\ n_3 \\ n_4 \\ n_{pwl} \end{array} \begin{array}{c} \begin{array}{ccccc} n_1 & n_2 & n_3 & n_4 & n_{pwl} \end{array} \\ \left[\begin{array}{cccc|c} & & & & 1 \\ & & & & -1 \\ & & & & 0 \\ & & & & 0 \\ \hline -R_i & R_i & 1 & -1 & 0 \end{array}\right] \end{array} \left[\begin{array}{c} V_{n_1} \\ V_{n_2} \\ V_{n_3} \\ V_{n_4} \\ I_{pwl} \end{array}\right] \left[\begin{array}{c} \\ \\ \\ \\ V_i \end{array}\right]$$

and PWL voltage-controlled current source as

$$\begin{array}{c} \\ n_1 \\ n_2 \\ n_3 \\ n_4 \\ n_{pwl} \end{array} \begin{array}{c} \begin{array}{ccccc} n_1 & n_2 & n_3 & n_4 & n_{pwl} \end{array} \\ \left[\begin{array}{cccc|c} & & & & 1 \\ & & & & -1 \\ & & & & 0 \\ & & & & 0 \\ \hline 0 & 0 & 1 & -1 & -R_i \end{array}\right] \end{array} \left[\begin{array}{c} V_{n_1} \\ V_{n_2} \\ V_{n_3} \\ V_{n_4} \\ I_{pwl} \end{array}\right] \left[\begin{array}{c} \\ \\ \\ \\ V_i \end{array}\right]$$

With all those PWL components, general equations for a PWL circuit can be written as

$$Hx = b \tag{1}$$

where H is the PWL circuit matrix and x is the circuit unknown vector. b is the right-side part composed of system excitations like current sources. Notice that the resulting PWL circuit matrix is a standard MNA circuit matrix, which avoids the complementary linear problem introduced by using ideal diode models for PWL components. The main advantage of using the standard MNA formulation is that all the linear symbolic/numerical techniques can be used to solve the resulting PWL matrix easily.

3. TIME-DOMAIN DDD GRAPHS

3.1 The DDD Graph based Method for Deriving Transfer Functions

In this subsection, we briefly review the determinant decision diagrams (DDDs), to derive the symbolic solution of a linear circuit [14].

Determinant Decision Diagrams [14] are compact and canonical graph-based representation of determinants. A DDD graph is similar to binary decision diagrams (BDDs) except that a sign is associated with each node to represent the sign of a product term from the expansion of a determinant. Also like BDDs, DDDs are very capable of representing huge number of symbolic terms from a determinant. The hierarchical approach using DDD graphs can essentially drive symbolic solutions for arbitrary large linear circuits [16, 17]

3.2 The DDD Based Method for Deriving Symbolic Time-domain Solutions

In this paper, we built a time-domain DDD to derive the symbolic expressions for PWL circuits at any active linear region. For symbolic transient analysis, the major difference, compared with symbolic analysis in frequency domain, is that symbolic solutions for all the unknown are required instead of just transfer functions. In addition to the different stamps for reactive components like capacitors and inductors, where backward Euler formula is used to obtain the stamps of capacitors and inductors, the parameterized PWL components have their unique stamps as shown above.

For symbolic solutions, according to Cramer's rule, the kth component x_k of the unknown vector x in Eq.(1) is obtained as follows:

$$x_k = \frac{det(H_k)}{det(H)}. \tag{2}$$

Where H_k is a $n \times n$ matrix defined as

$$H_k = \begin{bmatrix} a_{1,1} & \ldots & a_{1,k-1} & b_1, & a_{1,k+1} & \ldots & a_{1,n} \\ a_{2,1} & \ldots & a_{2,k-1} & b_2, & a_{2,k+1} & \ldots & a_{2,n} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ a_{n,1} & \ldots & a_{n,k-1} & b_n, & a_{n,k+1} & \ldots & a_{n,n} \end{bmatrix}. \tag{3}$$

Notice that both numerator and denominator are determinant, so

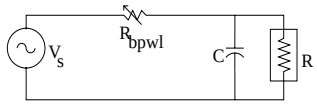

Figure 2: A simple circuit with a PWL component.

we can use two shared DDD graphs to represent the solution for x_k.

We take a simple circuit as an example on how to compute the symbolic solution represented by DDD graphs. For the circuit shown in Fig. 2, there is one two-terminal PWL component R_{bpwl}. Suppose the time step for transient analysis is h, then the MNA equation can be written as

$$\begin{bmatrix} 0 & -1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & -R_i & -1 \\ 0 & 0 & -1 & \frac{1}{R}+\frac{h}{C} \end{bmatrix} \begin{bmatrix} V_1 \\ I_{V_1} \\ I_{R_{bpwl}} \\ V_2 \end{bmatrix} \begin{bmatrix} 0 \\ V_s \\ V_i \\ I_C \end{bmatrix} \tag{4}$$

where R_i and V_i are the reciprocal of the slope and intercept of any segment that the PWL can operate at, R and C are for resistor and capacitor respectively, $\frac{h}{C}$ and I_C are the conductance and current source associated with the capacitor which is generated by backward-Euler integration method. Here we just show how to represent the current I_{bpwl} through the PWL element R_{bpwl} with DDD graphs. In the above equation, I_{bpwl} corresponds to the third row and we can write

$$I_{bpwl} = \frac{det(H_3)}{det(H)} \tag{5}$$

where H is the 4×4 matrix of the left side of Eq.(4) and H_3 is the matrix from H by replacing its third column by the right-side vector $[0\ V_s\ V_i\ I_C]^T$.

Correspondingly, the DDD graph representation for the determinants is shown in Fig.3.

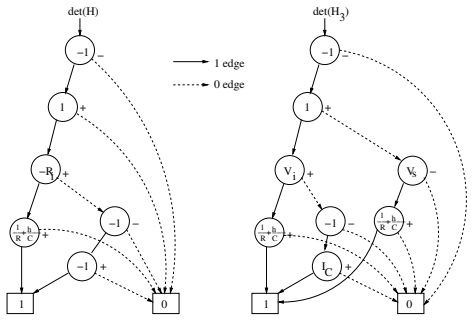

Figure 3: The DDD graph representation for $det(H)$ and $det(H_3)$ for the simple circuit.

and the expression for the solution I_{bpwl} is

$$I_{bpwl} = \frac{V_i(\frac{1}{R}+\frac{h}{C}) + I_C - V_S(\frac{1}{R}+\frac{h}{C})}{-R_i(\frac{1}{R}+\frac{h}{C}) - 1} \tag{6}$$

4. THE HIERARCHICAL SYMBOLIC ANALYSIS FOR PWL CIRCUITS

Hierarchical analysis is to reduce the size of the circuit matrix via subcircuit reduction. Such a reduction process can be performed using Schur decomposition method. A fully symbolic analysis based on determinant decision diagrams was proposed in [16], which can handle arbitrary large linear circuits.

In this paper, we apply this hierarchical symbolic analysis method to obtain the exact symbolic solution of PWL circuit matrices, which actually is the MNA matrix except some circuit parameters are changing with time due to PWL devices. Specifically, suppose the Eq.(1) represents a hierarchical PWL circuit with one subcircuit. We partition the circuit unknowns–the node-voltage variables and branch-current variables into three disjoint groups x^I,x^B and x^R, where the superscripts I, B and R stand for, respectively, internal variables, boundary variables and the rest of variables, then the Eq.(1) can be written as

$$\begin{bmatrix} H^{II} & H^{IB} & 0 \\ H^{BI} & H^{BB} & H^{BR} \\ 0 & H^{RB} & H^{RR} \end{bmatrix} \begin{bmatrix} x^I \\ x^B \\ x^R \end{bmatrix} \begin{bmatrix} b^I \\ b^B \\ b^R \end{bmatrix} \tag{7}$$

where the matrix H^{II} is the internal matrix associated with internal variable vector x^I. Then subcircuit suppression is used to eliminate all the variables in x^I and the equation are simplified as:

$$\begin{bmatrix} H^{BB^*} & H^{BR} \\ H^{RB} & H^{RR} \end{bmatrix} \begin{bmatrix} x^B \\ x^R \end{bmatrix} \begin{bmatrix} b^{B^*} \\ b^R \end{bmatrix} \tag{8}$$

where

$$H^{BB^*} = H^{BB} - H^{BI}(H^{II})^{-1}H^{IB} \tag{9}$$

and

$$b^{B^*} = b^B - H^{BI}(H^{II})^{-1}b^I \tag{10}$$

Once the parent circuit variables x^B are known, we can obtain the internal variables of subcircuit A^{II} by solving

$$H^{II}x^I = b^I - H^{IB}x^B. \tag{11}$$

In this way, we can compute all the unknown variables in the subcircuits.

It was shown in [18] each element in the H^{BB^*} and b^{B^*} can be represented by the ratio of two determinants, which in turn can be represented by DDD graphs. Therefore, the whole solution of circuit unknown can be represented by hierarchical tree of DDD graphs.

5. REVIEW OF GENERAL KATZENELSON'S ALGORITHM

Katzenelson's algorithm is introduced by Katzenelson in 1965 but still extensively used in PWL DC and transient analysis [6]. Consider a flat piecewise-linear circuit for which the equation can be written as in any linear region for some input signals

$$H^m x + w^m = y \tag{12}$$

where H^m is the system matrix for the mth linear region and w^m

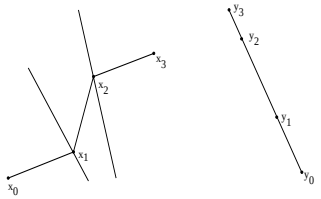

Figure 4: Illustration of the inverse mapping from y to x and state transition between linear regions.

is the corresponding constant vector for the linearization in this region. Variable y typically represents some independent current sources. Compared with Eq.(1), we can find w^m and y are in fact two parts of b and they are separated intentionally. At the beginning the operating regions are unknown so we select the node voltages arbitrarily. Suppose there is an initial solution x_0 which meats

$$H^m x_0 + w^m = y_0 \tag{13}$$

Since $y_0 \neq y$ we need to apply some iterative steps for correction as shown in Fig. 4. Let

$$\Delta y_k = H_k^m x_k + w_k^m - y = H_k^m \Delta x_k \tag{14}$$

represents the error vector of right-hand side at kth step. If the final solution is in this linear region, then we use $x = x_k + \Delta x$ to obtain the results. But if any one component crosses into a new linear region, then we can increase x_k by a small step instead of a full step so that the component goes into the boundary. So we have

$$x_{k+1} = x_k + \lambda \Delta x_k \tag{15}$$

where λ is the minimum possible value and meets the requirement of $0 < \lambda < 1$ to ensure there is only one component crossing into a new linear region at one time. Otherwise a so-called corner problem appears if there are two and more components changing into new regions. There, however, exist iterative procedures developed to solve this problem [13]. This procedure repeats until the final operating region and solution are obtained.

6. NEW HIERARCHICAL TRANSIENT ANALYSIS OF PWL CIRCUITS

In our new method, the Katzenelson's algorithm is combined with hierarchical symbolic method to perform the transient analysis for the hierarchical piecewise linear circuits. The new algorithm is described as follows:

1. Build symbolic expressions for all the subcircuit and top level circuit in a bottom up way. For each subcircuits, we need to build symbolic expressions in terms of DDD graphs for its parents due to Eq.(9) and Eq.(10) and expressions for solving its internal variables due to Eq.(11).

2. At the top level (root) circuit, formulate the equations for the root circuit

$$H_k^m \Delta x_k = \Delta y_k \tag{16}$$

The Δy on right-hand side are symbolic expression, which is composed of independent sources and the sources produced by reactive elements' companion models.

3. Initially, we assume both the voltages at all nodes and the currents through all branches are zero since for most devices the current will not exist if no voltage exist. That means for Eq.(12) at $t = t_0$ we can take $x_0 = 0$ and $y_0 = 0$ as the initial solution.

4. At each time point $t = t_n$, all input sources and the current and voltage sources associated with reactive components are updated; We use the solution at time t_{n-1} as our initial solution and correspondingly the linear region variables are used to update the matrix H_k^m for the root circuit which is the matrix for our initial solution of Katzenelson's algorithm; Also update the matrix for all the subcircuits. Then we search to the answer of PWL circuit at current time step:

   (a) Calculate the right hand side Δy for all subcircuits with

$$\Delta y = y - y_0 \tag{17}$$

   If $t = 0$, then $\Delta y = y$; otherwise Δy equals to the difference of right-hand side between current time $t = t_n$ and previous time step $t = t_{n-1}$; Then use Eq.(10) to calculate the right hand side for all middle circuits and root circuit in a bottom-up way.

   (b) Substitute H_k^m and Δy into the symbolic solution of Eq.(16) to obtain numerical solution Δx; And then substitute the solutions into its subcircuit equation to obtain solution x^{II} for all subcircuits; Repeat this in a top-down way until all the solutions of leaf circuits are obtained; In this procedure all the DDD graphs for the determinant and cofactors are evaluated.

   (c) For root circuit and all subcircuits, compute λ, where $\lambda = min(\lambda_i)$ and λ_i is the value which makes the ith PWL component to reach the boundary x_i^b:

$$\lambda_i = \frac{(x_i^b - x_i)}{\Delta x_i} \tag{18}$$

   and λ meets the condition $0 < \lambda \leq 1$.

   (d) Update the solution for root circuit and all subcircuits at time $t = t_n$ as

$$\begin{aligned} x_0 &= x_0 + \lambda \Delta x \\ y_0 &= y_0 + \lambda \Delta y \end{aligned} \tag{19}$$

   (e) If $\lambda = 1$, x_0 is the final solution at time $t = t_n$. Otherwise switch to the new linear region of the PLW component, which makes the minimum λ. Update the matrix H_k^m of the corresponding root circuit or subcircuit with corresponding linear variables, go to step (a) to repeat the steps (a)-(e) until λ = 1, which means the PWL solution is found.

5. Go to next time point $t = t_{n+1}$.

Notice that circuit solutions need to be solved repeatedly to find the valid solutions at each time step in Katzenelson's algorithm. This makes our symbolic based approach more attractive as we need to build the symbolic expressions only once and evaluate them for many times, which in contrast with numerical method where circuit equations has to be solved (LU decomposed) at every iteration in Katzenelson's algorithm. In some sense, the improved Katzenelson iteration is similar to the Newton-Raphson iteration to reach a solution for a nonlinear equation. Katzenelson's algorithm, however, is guaranteed to converge as all the linear region will be searched after sufficient events.

7. EXPERIMENTAL RESULTS

In this section, we take some analog circuit examples to show the effectiveness of our new method for PWL circuits. All of them are finished on a Linux PC with 2.4Ghz CPU and 484M RAM.

The first example comes from an actual circuit design which is a Pulse-Width-Modulated(PWM) system shown in Fig. 5 for power amplification. It uses upper and down triangle waves to sample input signal as shown in Fig. 6. The compared output after comparators CMP1 and CMP2 is fed into the LC filters. The advantage of the double triangle waves for sampling is that output signal is zero when the input is zero (eg. audio signal is in this situation for most of time) in order to save power.

The feedback network, which is composed of R3, R3, R5, R6, R7, R8, C4, C5 and one operational amplifier(OPAMP) OPA2, can reduce distortion of the system. In our experiment, we first use a Spice-like simulator to do the transient analysis of the specific PWM system which is completely implemented with MOS circuits, and then use our PWL tools to simulate it again with the corresponding behavioral piecewise-linear models. The results are compared with each other. The circuit implementation of OPAMP is shown in Fig. 7 which is a three-stage amplifier. For extracting its behavioral model, we conduct DC sweeping and AC analysis to obtain its gain characteristics, finite output swing and slew rate limiting etc., which are the most important factors for deciding its transient behavior. Similar to the procedure recommended in paper [19], the derived OPAMP piecewise-linear model illustrated in Fig. 8, which includes one piecewise-linear voltage-controlled current source GPWL1 for slew rate limiting and one piecewise-linear resistor BPWL1 for output clamping. For the comparator, its circuit implementation is shown in Fig. 9 and it is replaced by a piecewise-linear voltage-controlled voltage source as its behavioral model.

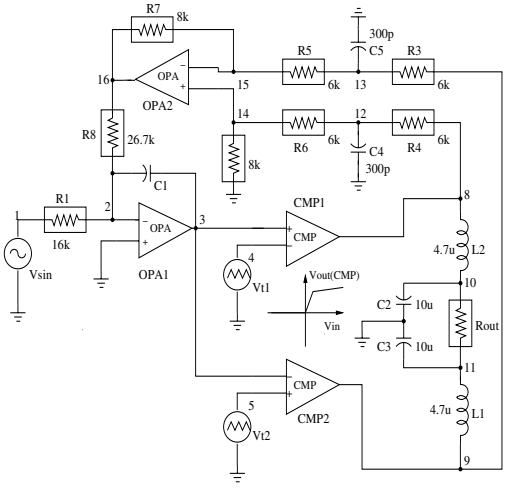

Figure 5: A PWM system with behavioral models of OPAMP.

We use SIMetrix [12], which is a commercial Spice-like simulator and can support BSIM4 MOSFET models to perform the full transistor-level simulation. For the transient simulation of the corresponding piecewise-linear circuits we use SPWL-flat and SPWL-hier tools, which are the flat and hierarchical PWL simulator tools respectively developed by authors. Specifically, The SPWL-flat simulator treats the circuit as a flat circuit while SPWL-hier simulate circuit hierarchically as described in the previous section. The

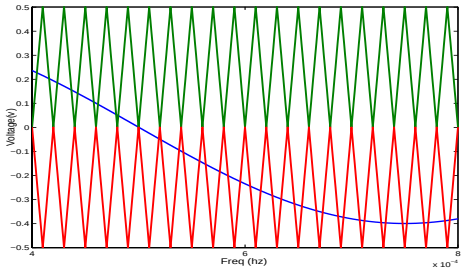

Figure 6: The sampling of signal with the upper and down triangle-wave signal in PWM system.

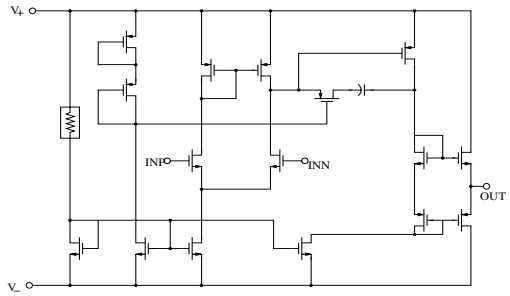

Figure 7: The MOS circuit implementation for the OPAMP.

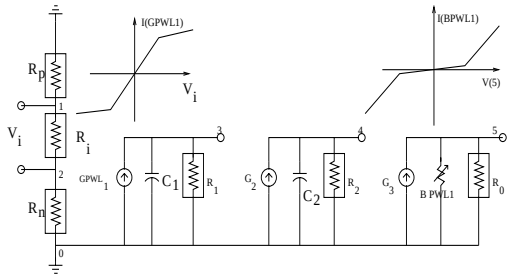

Figure 8: The behavioral model for the OPAMP amplifier.

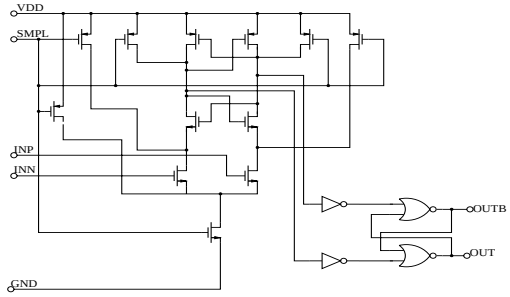

Figure 9: The MOS circuit implementation for the comparator.

transient waveforms are shown in Fig. 10, which are close to actual results from SIMetrix as it forecasts correct amplitudes of the output voltage. The noticeable discrepancy comes from the fact that we only model the key factors like clamping voltage and slew rate. With more accurate PWL models, our simulation results will be more accurate. In Table 1, we list the simulation time for the PWM circuit using SIMetrix and the new methods. It can be seen that about 5 times speedup is achieved. Hence, PWL simulation can easily make the tradeoff between simulation speed and accuracy.

Second, we take two other large filter circuits from [19] as our examples. Due to the page limitation, we only show the band-pass circuit and its simulation results. The schematic of the band-pass

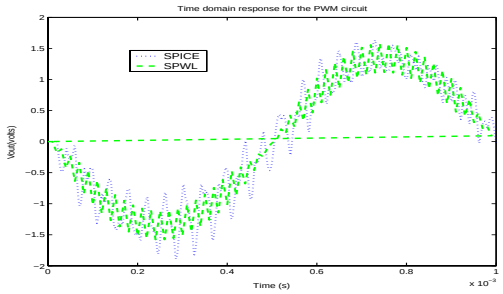

Figure 10: The transient response for the PWM circuit.

Circuits	SIMetrix	SPWL-flat	SPWL-hier
PWM	342s	290.74s	67.61
LowPass	306s	125.02s	44.04s
BandPass	317s	N/A	36.75s

Table 1: Simulation time comparison.

circuit is shown in Fig. 11. Similarly, we use Fig. 8 as its behavioral model for PWL simulation and Fig. 7 as actual circuit implementation of OPAMP. The transient responses from SIMetrix and PWL tools are shown Fig. 12.

The CPU times are also shown in the Table 1. It can be seen that hierarchical PWL simulator gives very accurate results compared with SIMetrix, which use actual circuit implementation, while delivers about 10X speedup over SIMetrix. Also hierarchical PWL simulation is about at least 3X to 5X faster than the flat one. More importantly, it can deal with much larger circuits than the flat one. Practically, this is not limitation to the simulation capacity of the new hierarchical PWL analysis. So the proposed PWL simulator can be effectively used for early stage design and verification of analog design and mixed-signal circuits.

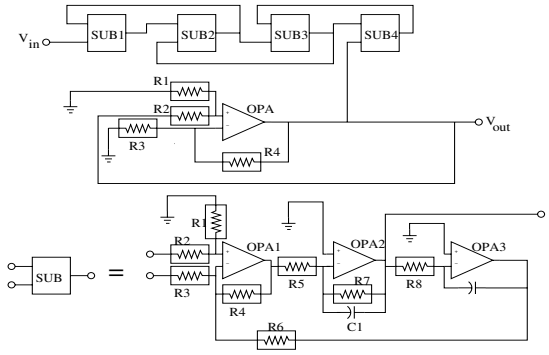

Figure 11: A hierarchical band-pass filter circuit.

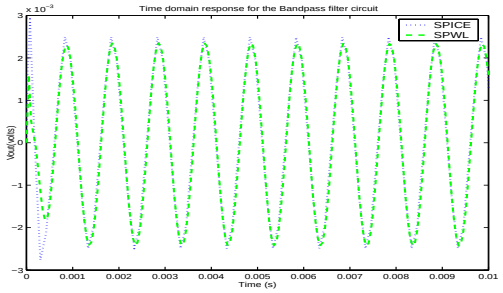

Figure 12: The transient response for the band-pass filter circuit.

8. CONCLUSION

In this paper, we have proposed a new approach for hierarchical transient analyses for nonlinear circuits modeled as piecewise-linear circuits. The new method combines hierarchical symbolic analysis approach with iterative Katzenelson's algorithm to obtain numerical result at each time point. We proposed parameterized PWL modeling of nonlinear devices, which allows standard modified nodal analysis formulation of parameterized PWL circuits and lead to more compact circuit matrices than that of PWL circuits modeled using ideal diodes. The resulting parameterized PWL circuits are solved only once symbolically using hierarchical determinant decision diagrams and are evaluated many times at every time step in the Katzenelson's algorithm. Experimental results on a number of nonlinear analog circuits show that the proposed method outperforms the commercial circuit simulator SIMetrix in CPU times with reasonable accuracy. The easy tradeoff of speed versus accuracy makes PWL simulators ideal simulation tools for early stage design and verification of analog design and mixed-signal circuits.

9. REFERENCES

[1] F. V. Fernandez, B. Perez-Verdu and A. Rodriguez-Vazquez, "A behavioral modeling of PWL analog circuits using symbolic analysis", *Proc. of the 1998 IEEE International Symposium on Circuits and Systems*, pp.17-20, vol.6, 1998

[2] A. Manthe,L. Zhao, C.-J. R. Shi and K. Mayaram, "Symbolic analysis of nonlinear analog circuits", *Proc. Design, Automation and Test in Europe Conference and Exhibition*, pp.1108-9, 2003.

[3] C. Visweswariah and R.A. Rohrer, "Piecewise approximate circuit simulation", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp.861-70, July 1991.

[4] D.M.W. Leenaerts and W.M.G. van Bokhoven, *Piecewise Linear Modeling and Analysis*, Kluwer Academic Publishers, 1998.

[5] T.A.M. Kevenaar and D.M.W. Leenaerts, "A flexible hierarchical piecewise linear simulator", *Integration-The VLSI Journal*, vol.12, pp.211-35, Dec. 1991.

[6] J. Katzenelson, "An algorithm for solving nonlinear resistor networks", *Bell Syst. J.* vol. 44, pp.1605-1620, 1965.

[7] M.T. van Stiphout, J.T.J. van Eijndhoven and H.W. Buurman, "PLATO: a new piecewise linear simulation tool", *Proc. of the European Design Automation Conference*, IEEE Comput. Soc. Press, pp.235-9, 1990.

[8] S. Topcu, O. Ocah, A. Atalar and M.A. Tan, "A novel algorithm for DC analysis of piecewise-linear circuits: popcorn", *IEEE Transactions on Circuits and Systems I-Fundamental Theory and Applications*, vol. 41, pp.553-6, Aug. 1994.

[9] P. Pejovic and D. Maksimovic, "A new algorithm for simulation of power electronic systems using piecewise-linear device models", *IEEE Transactions on Power Electronics*, vol.10, pp.340-8, May 1995.

[10] A. Manthe, L. Zhao and C.-J. R. Shi, "Symbolic analysis of analog circuits with hard nonlinearity", *Proc. Design Automation Conference*, pp.542-5, 2003.

[11] F. Filippetti and M. Artioli, "Symbolic linear Mathematica based technique for piecewise linear circuits",*IEEE International Conference on Electronics, Circuits and Systems*, pp.625-8, vol. 2, 2000

[12] "SIMetrix–SPICE and Mixed Mode Simulation", *Catena Software Ltd.*, 2003

[13] T. Fujisawa and E.S. Kuh, "Piecewise-linear theory of nonlinear networks", *SIAM Journal on Applied Mathematics*, vol. 22, pp. 307-28, March 1972

[14] C.-J. Shi and X.-D. Tan, "Canonical symbolic analysis of large analog circuits with determinant decision diagrams", *IEEE Trans. Computer-Aided Design*, vol. 19, no. 1, pp. 1–18, Jan. 2000.

[15] C.-J. Shi and X.-D. Tan, "Compact representation and efficient generation of s-expanded symbolic network functions for computer-aided analog circuit design", *IEEE Trans. Computer-Aided Design*, vol. 20, No. 7, pp. 813-827, July 2001.

[16] X.-D. Tan and C.-J. Shi. "Hierarchical symbolic analysis of analog integrated circuits via determinant decision diagrams." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, no. 4, pp. 401-412, 2000.

[17] S. X.-D. Tan, "A general s-domain hierarchical network reduction algorithm", *Proc. IEEE/ACM International Conf. on Computer-Aided Design (ICCAD)*, San Jose, CA, Nov. 2003.

[18] S. X.-D. Tan, Z. Qi and H. Li, "Hierarchical modeling and simulation of large analog circuits", *Proc. Design, Automation and Test in Europe (DATE'04)*, pp. 740–741, Mar., 2004.

[19] J. Vlach and K. Singhal, *Computer Methods for Circuit Analysis and Design*, Van Nostrand Reinhold, New York, 1994.