An Envelope-Following Method for the Efficient Transient Simulation of Switching Power and Filter Circuits

K. Kundert J. White A. Sangiovanni-Vincentelli

Research Laboratory of Electronics
Dept. of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
Cambridge, MA 02139

Abstract

The transient behavior of circuits like switching power converters and switched capacitor filters are expensive to simulate because these circuits are clocked at a frequency whose period is orders of magnitude smaller than the time interval of interest to the designer. It is possible to reduce the simulation time without compromising accuracy by exploiting the property that the behavior of such a circuits in a given high frequency clock cycle is similar, but not identical, to the behavior in the preceeding and following cycles. In particular, the "envelope" of the high-frequency clock can be followed by accurately computing the circuit behavior over occasional cycles. In this paper the implementation of such an envelope-following method that is particularly efficient for switching power and filter circuits is described, and results demonstrating the method's effectiveness are presented.

1 Introduction

In general, analog circuit designers rely heavily on circuit simulation programs like SPICE [nagel75] or ASTAP [weeks73] to insure the correctness and the performance of their designs. These programs simulate a circuit by first constructing a system of differential equations that describes the circuit, and then solving the system numerically with a time discretization method such as backward-Euler. When applied to circuits like switching power converters or switched-capacitor filters, such classical circuit simulation algorithms become extraordinarily computationally expensive. This is because switching power converters and switched-capacitor filters use high frequency clocks whose periods are typically orders of magnitude smaller than the time intervals of interest to a designer. The nature of the calculations used in a circuit simulator implies that to construct the solution over the time interval of interest, an accurate solution must be computed for every cycle of the high frequency clock in the interval, and this can involve hundreds of cycles. The infeasibility of simulating such circuits with classical techniques has led frustrated designers to develop specialized analog computers on which to do their simulation [kassakian79].

Fast approximate techniques have been developed for simulating switching power converters and switched capacitor filters, but the two problems are approached quite differently. The most common approach to simulating switched capacitor filters is first to break the circuit up into functional blocks such as operational amplifiers and switches, and replace each with a simple macromodel. It is then assumed that after each clock transition, every node in the circuit reaches its equilibrium point before another transition occurs. This assumption, along with the use of algebraic macromodels, allows the filter to be treated as a low order discrete-time system whose solution for hunderds of clock cycles can be computed quickly [tsividis79, deman80]. The most common techniques for simulating switching power converters is to treat the switches as ideal, and the remaining circuitry as linear. With this approximation, the solution over hundreds of clock cycles can be computed rapidly [hsiao87].

Although programs based on the above techniques have served designers well, they are based on idealizations of the circuits involved which may eliminate behavior that is important to a designer. In this paper we present an approach for the detailed transient simulation of switching power and filter circuits which does not involve any idealization of the behavior, and is much more efficient than classical direct methods when the clock period is small compared to the simulation interval. This method, referred to as **envelope-following** [petzold81] exploits the property of such circuits that the node voltage waveforms over a given high frequency clock cycle are similar, but not exact duplicates, of the node voltages waveforms in proceeding or following cycles. This suggests that it is possible to construct a solution accurate over many high frequency clock cycles by calculating the solution accurately for a few selected cycles.

In the next section, we present the details of an envelope-following method that is effective for many types of switching circuits. In Section 3 we describe some of the computations involved in the method. Their implementation in the program *Nitswit* along with results from using *Nitswit* to simulate several switching power and filter circuits is described in Section 4.

2 The Envelope-Following Method

Most circuits can be described by a system of differential equations of the form

$$\frac{d}{dt}p(x(t),u(t)) + f(x(t),u(t)) = 0, \qquad (1)$$

where $x(t) \in \Re^N$, the state, is the vector of capacitor voltages and inductor currents, $u(t) \in \Re^M$ is the vector of input sources, $p(x(t),u(t)) \in \Re^N$ is the vector of capacitor charges

and inductor fluxes, and $f(x(t), u(t)) \in \Re^N$ is the vector of resistive currents and inductor voltages. If the state x is known at some time t_0, it is possible to solve (1) and compute the state at some later time t_1. In general, one can write

$$x(t_1) = x(t_0) + \phi(x(t_0), t_0, t_1) \quad (2)$$

where $\phi : \Re^n \times \Re \times \Re \rightarrow \Re^n$ is a state transition function for the differential equation.

We now consider that the circuit to be simulated has as an input a clock with known period T that is much smaller than the simulation interval. In addition, we assume the sequence formed by sampling the state at the beginning of each clock cycle, $x(0), x(T), x(2T), \ldots, x(mT), \ldots$, changes slowly as a function of m, the clock cycle number. For such a circuit, a designer is usually interested in the transient behavior of an envelope of the solution, which we define as the continuous function derived by interpolating the sequence formed by sampling the state every time interval T. Note that our use of "envelope" is not the common usage. Here, the envelope is not unique given $x(t)$; the envelope generated by interpolating the sequence $x(0+\tau), x(T+\tau), x(2T+\tau), \ldots$ depends on τ.

A "differential-like" equation can be written for the elements of the sequence $x(0), x(T), x(2T), \ldots$ associated with one envelope of the solution to (1). Applying (2), the elements of the sequence can be related by

$$x(mT) - x((m-1)T) = \phi(x((m-1)T), (m-1)T, mT). \quad (3)$$

The relation in (3) indicates how rapidly the initial point of each clock cycle changes from one cycle to the next, and in that sense is like a differential equation. This similarity can be exploited to derive methods for approximately solving for the $x(mT)'s$. For example, the value of $x((m+l)T)$ can be approximated by

$$x((m+l)T) - x((m+1)T) \approx (l-1)\phi(x(mT), mT, (m+1)T), \quad (4)$$

which is loosely analogous to solving a differential equation by forward-Euler.

To compute an envelope for a system with period T using a forward-Euler envelope-following algorithm with a fixed cycle-step l, a simple repetitive two-step process can be used. Given $x(0)$, the first step is to calculate $x(T)$ by solving (1) over the interval $[0, T]$ using a standard discretization technique. The second step is to set $x(lT) = x(T) + (l-1)[x(T) - x(0)]$. This process is repeated to compute $x(2lT), x(3lT), \ldots$. Note that calculating the solution over a long interval only requires solving the differential equation every l^{th} cycle.

Although simple to describe, a forward-Euler based envelope-following method is not very effective for solving switching circuits because maintaining stability severely limits the size of the cycle-step l, just as with the standard forward-Euler algorithm. A more stable algorithm is to approximate the value of $x((m+l)T)$ by

$$x((m+l)T) - x(mT) \approx l\phi(x((m+l-1)T), (m+l-1)T, (m+l)T), \quad (5)$$

which is analogous to backward-Euler for the differential case. This approach allows for larger cycle-steps than the forward-Euler based approach, but leads to more a complicated equation to compute each cycle-step. To see this, consider computing $x(lT)$ given $x(0)$ based on (5). An $x((l-1)T)$ must be found such that when used as an initial condition for (1), the $x(lT)$ computed with standard discretization techniques satisfies $x(lT) - x(0) = l[x(lT) - x((l-1)T)]$ This is a boundary value problem, and is in general difficult to solve. For the case of switching power or filter circuits, the above boundary value problem can be solved efficiently using a Newton method, and this is presented in the next section.

3 Solution by Newton

As mentioned in the previous section, each cycle-step of the backward-Euler envelope-following method requires the simultaneous solution of

$$x((m+l)T) - x(mT) = l[x((m+l)T) - x((m+l-1)T)] \quad (6)$$

and

$$x((m+l)T) - x((m+l-1)T) = \phi(x((m+l-1)T), (m+l-1)T, (m+l)T). \quad (7)$$

for $x((m+l-1)T)$ and $x((m+l)T)$, where where $x(mT)$ is presumed known. Therefore, (6) and (7) represents $2n$ equations in $2n$ unknowns.

An iterative Newton's method can be applied to solving the above system. In general, the Newton method applied to the problem of finding an $x \in \Re^n$ such that $F(x) = 0$, $F : \Re^n \rightarrow \Re^n$, yields the iteration equation $J_F(x^k)[x^{k+1} - x^k] = -F(x^k)$, where k is the Newton iteration count and $J_F \in \Re^{n \times n}$ is the Jacobian of F. Reorganizing (6) and (7) into the form to apply Newton's method leads to

$$(1-l)x((m+l)T) + lx((m+l-1)T) - x(mT) \quad (8)$$

$$x((m+l)T) - x((m+l-1)T) - \phi() \quad (9)$$

$$= F(x((m+l)T), x((m+l-1)T)) = 0 \quad (10)$$

In this case, $J_F(x((m+l)T), x((m+l-1)T))$ is given by

$$\begin{bmatrix} (1-l)I_n & lI_n \\ I_n & -I_n - \frac{\partial}{\partial x}\phi() \end{bmatrix} \quad (11)$$

where I_n is the identity matrix of size n.

The most time-consuming computation in this Newton iteration is evaluating J_F and F, which involves computing the state transition function, $\phi(x((m+l-1)T), (m+l-1)T, (m+l)T)$, and its derivative. The state transition function can be evaluated by numerically integrating (1) from $(m+l-1)T$ to $(m+l)T$ given $x((m+l-1)T)$. The derivative of the state transition function, referred to as the sensitivity matrix, represents sensitivity of $x((m+l)T)$ to changes in $x((m+l-1)T)$, and can be computed with a small amount of additional work during the numerical integration.

To show how the computation of the state transition function and its derivative fit together, consider numerically integrating (1) with backward-Euler, which we chose for its simplicity and because it appears to be one of the better formulas for clocked analog circuits. Given some initial time t_0 and some initial condition, $x(t_0)$, applying backward-Euler to (1) results in the following algebraic equation,

$$g(x(t_0+h), x(t_0)) = \frac{1}{h}(p(x(t_0+h)) - p(x(t_0))) + f(x(t_0+h)) = 0 \quad (12)$$

where $h \in \Re$ is the timestep. Note we have dropped explicitly denoting the dependence of p and f on the input u for simplicity.

Equation (12) is usually solved with Newton-Raphson, for which the iteration equation is

$$J_g(x^{(k)}(t_0+h))(x^{(k+1)}(t_0+h) - x^{(k)}(t_0+h)) = \quad (13)$$
$$-g(x^{(k)}(t_0+h), x^{(k)}(t_0)) \quad (14)$$

where $J_g(x(t)) \in \Re^{N\times N}$ is the Frechet derivative of the nonlinear equation in (12) and is given by

$$J_g(x(t)) = \frac{\partial g(x(t),\cdot)}{\partial x(t)} = \frac{1}{h}\frac{\partial p(x(t))}{\partial x(t)} + \frac{\partial f(x(t))}{\partial x(t)}. \quad (15)$$

Solving (12) yields an approximation to $x(t_0+h) = \phi(x(t_0), t_0, t_0+h)$. Implicitly differentiating (12) for $x(t_0+h)$ with respect to $x(t_0)$ yields

$$J_g(x(t_0+h))\frac{\partial x(t_0+h)}{\partial x(t_0)} = \frac{1}{h}\frac{\partial p(x(t_0))}{\partial x(t_0)} \quad (16)$$

Given a $x(t_0)$, (12) can be repeatedly applied to find $x(t_0+T) = \phi(x(t_0), t_0, t_0+T)$, and (16) can be repeatedly applied to find the sensitivity matrix $\partial x(t_0+T)/\partial x(t_0) = \partial\phi(x(t_0), t_0, t_0+T)/\partial x(t_0)$ [aprille72]. Note that J_f is required in both (14) and (16), and thus the sensitivity matrix update can be made more efficient by factoring J_g once and using it for both computations. However, the sensitivity matrix is still expensive to compute, because it is an $N \times N$ full matrix.

4 Implementation and Test Results

An envelope-following method has been implemented in the *Nitswit* [kundert88] simulation program. The program is written in "C", and runs under the UNIX operating system. The program uses a trapazoidal-rule based envelope-following algorithm in which the cyclesteps are selected based on local truncation error. The boundary value problems generated at each cyclestep are solved with the Newton method described above.

Two techniques are also used to improve the efficiency of the basic algorithm. Although switching power supplies and filters are not linear circuits, the state transition function over one cycle is a nearly affine (linear plus a constant). This property can be exploited to reduce the computation by only computing J_F for the boundary value Newton method on the first iteration at each timestep. This is a significant savings, as then the sensitivity matrix need only be computed once per timestep. The second technique that reduces the computation time is based on the fact that the explicit forward-Euler based envelope-following method is stable for cyclesteps of size two or less. This implies that if truncation error considerations determine that a cyclestep of two or less is appropriate, the boundary value problem can be avoided, and the step can be computed easily.

Exactly how the envelope-following method behaves can be seen by examining figures 1 and 2 below. Figure 1 is the solution to a buck-derived circuit(from [hsiao87]) computed with classical direct method, where the exact solution is only shown between 1 and 2 milliseconds because of point limitations in the plotting program, and Figure 2 is the result produced by the envelope-following method. As can be seen, the envelope-following method computes many fewer cycles, but the ones computed match with the direct method.

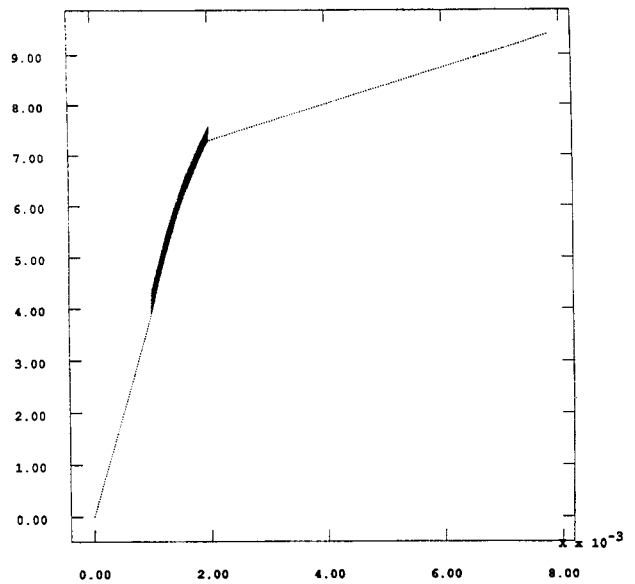

Figure 1: Buck Converter Solution Computed with the Classical Method

In the table below we present a comparison between the cpu time used by classical and envelope-following methods in simulating the start-up transient from three types of switching power supplies, a push-pull flyback converter, *fly*, a resonant converter (from [casey87]), *res*, and a buck-derived circuit, *buck*; and the step response from a switched-capacitor low-pass filter. In each case, the clocking is provided by a user-defined source. As can be seen from the table, the envelope-following method can be very efficient, particularly when the simulation interval is long compared to the clock cylce.

Circuit	Nodes	Inteval/Clock	Classical	Env. Follower.
scop	13	200	601	153
buck	8	1000	940	97
quasi	7	200	144	38
fly	32	40	274	167

Table 1: CPU Time (in seconds) Comparisons for Classical vs Envelope-Following simulation, based on a SUN4

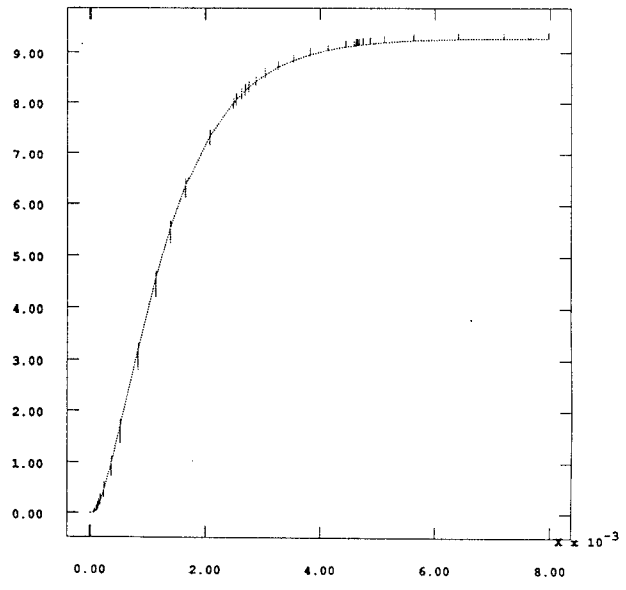

Figure 2: Buck Converter Solution Computed with Envelope-Following

5 Conclusions and Acknowledgements

In this paper it is shown that an envelope-following approach to the simulation of switching power and filter circuits can provide substantial speed improvements over classical simulation methods. Several aspects of the method are still under investigation. In particular, it has been observed that most of the entries in the sensitivity matrix remain close to zero, and how to exploit this is being considered. In addition, the effectiveness of the envelope-following is somewhat dependent on where the cycle boundaries are placed, and an automatic selection method is desirable.

The authors would like to thank Professor Marija Ilic, Professor John Kassakian, and Steven Leeb at MIT's Laboratory for Electromagnetic and Electronic Systems for their assitance in understanding the switching power simulation problem, and Prof. A. R. Newton for his suggestions. This work was supported by the Defense Advanced Research Projects Agency contract N00014-87-K-825, and Analog Devices.

References

[aprille72] T. Aprille, T. Trick. "Steady-state analysis of nonlinear circuits with periodic inputs." *Proc. IEEE*, Jan 72.

[deman80] H. De Man, J. Rabaey, G. Arnout, J. Vandewalle. "Practical implementation of a general computer aided design technique for switched capacitor circuits." *IEEE Journal of Solid-State Circuits*, vol. SC-15, pp. 190-200, April 1980.

[hsiao87] C. J. Hsiao, R. B. Ridley, H. Naitoh, F. C. Lee. "Circuit-Oriented Discrete-Time Modeling and Simulation for Switching Converters." *IEEE Power Electronics Specialists' Conf. Rec.*, 1987

[casey87] L. F. Casey and M. F. Schlect, "A High Frequency, Low Volume, Point-of-Load Power Supply for Distributed Power Systems" *IEEE Power Electronics Specialists' Conf. Rec.*, 1987

[kassakian79] J. Kassakian. "Simulating Power Electronic Systems – A New Approach." *Proc. IEEE*, vol. 67, Oct. 1979.

[kundert88] K. Kundert, J. White, A. Sangiovanni-Vincentelli. "A Mixed Frequency-Time Approach for Finding the Steady-State Solution of Clocked Analog Circuits." *Cust. Int. Circ. Conf.*, May 1988.

[nagel75] L. W. Nagel. *SPICE2: A Computer Program to Simulate Semiconductor Circuits.* Electronics Research Lab Report, ERL M520, Univ. of Calif., Berkeley, May 1975.

[petzold81] L. Petzold. "An Efficient Numerical Method for Highly Oscillatory Ordinary Differential Equations." *SIAM J. Numer. Anal.*, Vol. 18, No. 3, June 1981.

[tsividis79] Y. P. Tsividis. "Analysis of switched capacitor networks." *IEEE Transactions on Circuits and Systems*, vol. CAS-26, pp. 935-946, November 1979.

[verghese86] G. C. Verghese, M. E. Elbuluk, J. G. Kassakian. "A General Approach to Sampled-Data Modeling for Power Electronic Circuits." *IEEE Trans. on Power Electronics*, Vol. PE-1, No. 2, April 1986.

[weeks73] W. T. Weeks, A. J. Jimenez, G. W. Mahoney, D. Mehta, H. Quasemzadeh, T. R. Scott. "Algorithms for ASTAP — A Network Analysis Program." *IEEE Transactions on Circuit Theory*, pp. 628-634, Nov. 1973.