

A Mixed Frequency–Time Approach for Distortion Analysis of Switching Filter Circuits

KENNETH S. KUNDERT, STUDENT MEMBER, IEEE, JACOB WHITE, MEMBER, IEEE, AND ALBERTO SANGIOVANNI-VINCENTELLI, FELLOW, IEEE

Abstract—Designers of switching filter circuits are often interested in steady-state distortion due to both static effects, such as nonlinearities in the capacitors, and dynamic effects, such as the charge injection during MOS transistor switching or slow operational-amplifier settling. Steady-state distortion can be computed using the circuit simulation program SPICE, but this approach is computationally very expensive. Specialized programs for switched-capacitor filters can be used to rapidly compute steady-state distortion, but do not consider dynamic effects. In this paper we present a new mixed frequency–time approach for computing both forms of steady-state distortion. The method is computationally efficient and includes both static and dynamic distortion sources. The method has been implemented in a C program, *Nitswit*, and results from several examples are presented.

I. INTRODUCTION

IN GENERAL, analog circuit designers rely heavily on circuit simulation programs like SPICE [1] or ASTAP [2] 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 simulating switching filter circuits, such as the switched-capacitor filters used in integrated circuits or the switching converters used in high-power applications, the classical circuit simulation algorithms become extraordinarily computationally expensive. This is because the period of the clock is usually orders of magnitude smaller than the time interval of interest to a designer. The nature of the calculations used in a circuit simulator im-

plies that an accurate solution must be computed for every cycle of the clock in the interval of interest, and this can involve thousands of cycles.

The most common approach to reducing the computational burden of switching filter simulation is to first break the circuit up into functional blocks such as operational amplifiers and switches. Each functional block is simulated, using a traditional circuit simulator, for some short period. The simulations of the functional blocks are used to construct extremely simple macromodels, which replace the functional blocks in the circuit. The result is a much simplified circuit that can be simulated easily. This simplified circuit is then simulated for the thousands of clocks cycles necessary to construct a solution meaningful enough to verify the design.

In programs specifically for switched-capacitor filters, like *Diana* [3] and *Switcap* [4], the simulation efficiency is enormously increased by the use of the “slow-clock” approximation. After each clock transition, every node in the circuit is assumed to reach its equilibrium point before another transition occurs. This assumption, along with the use of algebraic macromodels, allows the filter to be treated as a discrete-time system with one time point per clock transition. A set of difference equations is then used to describe the filter.

Specialized simulation programs are extremely efficient for determining frequency- and time-domain response of switching filters, but macromodels in general, as well as the “slow clock” approximation, tend to ignore second-order effects that can change distortion characteristics. In particular, as switching filters are being pushed to operate at ever higher frequencies, the assumption that signals reach equilibrium between clock transitions is often violated. Also, since signals between clock transitions are not computed, it is possible to miss events that occur in these intervals that might interfere with proper operation and contribute to distortion (e.g., clock feedthrough spikes causing an operational amplifier to saturate). Lastly, it is not possible to capture the effects of dynamic distortion processes, such as the important effect of the channel

Manuscript received August 31, 1988; revised December 7, 1988. This work was supported by the Defense Advanced Research Projects Agency under Contract N00014-87-K-825, Hewlett-Packard, and the California MICRO program.

K. S. Kundert and A. Sangiovanni-Vincentelli are with the Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA 94720.

J. White is with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139.

IEEE Log Number 8826164.

conductance on charge redistribution when a transistor switch turns off.

In this paper we present another approach to the simulation of switching filter circuits that is efficient for calculating steady-state distortion, but does not depend on macromodels or the slow clock approximation. The method exploits the property of switching filter circuits that node voltage waveforms over a given high-frequency clock cycle are similar, but not exact duplicates, of the node voltages waveforms in preceding 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 begin by describing our assumptions about switching filter circuits and presenting the mixed frequency-time method. In Section III, we discuss some of the computations involved in the method. In Section IV we briefly describe our program, *Nitswit*, and present comparison and application results. Finally, in Section V, we present our conclusions.

II. THE MIXED FREQUENCY-TIME METHOD

Very little can be assumed about the behavior of the node voltage waveforms in a switching filter circuit over a given clock cycle, because the circuits involved are very nonlinear and are usually switching rapidly. However, the node voltage waveforms over a whole clock cycle usually vary slowly from one cycle to the next, as controlled by the input signal. This implies that if the input is periodic, and the switching filter circuit is in steady state, then the sequence formed by sampling the node voltages at the beginning of each clock cycle is periodic (Fig. 1). We derive our method by assuming this to be true, and further assuming that the periodic function that describes the sequence of initial points in each clock cycle can be accurately represented as a truncated Fourier series using few terms.

If the sequence of initial points of each clock cycle can be described by a Fourier series with J terms, then once J initial points are known, all the initial points are known. This implies that given our Fourier assumption, to compute the steady-state behavior of a switching filter circuit we need only find the initial points of J clock cycles (a similar idea in a different context was presented in [5]).

In the next two subsections we describe two relations that can be exploited to construct a nonlinear algebraic system of J equations in J initial points (solving this system is discussed in Section III). The first relation, described in Section II-A, is derived from the Fourier series assumption, and is a linear relationship between the initial points of an evenly distributed set of J cycles and the initial points of the corresponding J cycles that immediately follow (Fig. 2). The second relation is derived from solving the differential equation system that describes the analog circuit, for the time interval of one clock cycle, J times, each time using one of the distributed set of J initial

points as an initial condition. This results in another set of values for the initial points of the following J cycles. Insisting that this set match the set resulting from the Fourier relation yields a nonlinear algebraic system in J unknowns, which can be solved for the J initial points, and this is described in Section II-B.

A. The Delay Operator

Consider the sequence of initial points of each clock cycle at some circuit node n , and denote the sequence by $v_n(\tau_1), v_n(\tau_2), \dots, v_n(\tau_S)$ where S is the number of clock cycles in an input period (Fig. 1). If it is assumed that this sequence can be accurately approximated by a truncated Fourier series, then

$$v_n(\tau_s) = V_0 + \sum_{k=1}^K (V_k^C \cos k\omega\tau_s + V_k^S \sin k\omega\tau_s) \quad (1)$$

where ω is the fundamental frequency of the input signal, K is the number of harmonics, and $J = 2K + 1$ is the number of unknown coefficients. Given (1), there is a linear relation between any collection of J initial points and any other collection of J initial points. However, as mentioned above, we are most interested in the linear operator that maps a collection $v_n(\tau_{\eta_1}), \dots, v_n(\tau_{\eta_J})$ into $v_n(\tau_{\eta_1} + T), \dots, v_n(\tau_{\eta_J} + T)$ where T is the clock period and $\{\eta_1, \dots, \eta_J\}$ is a subset of $\{1, \dots, S\}$ (Fig. 2). This linear operator will be referred to as the delay matrix.

Deriving the delay matrix is a two-stage process. First, the J points $v_n(\tau_{\eta_1}), \dots, v_n(\tau_{\eta_J})$ are used to calculate the Fourier coefficients. Then the Fourier series (using these coefficients) is evaluated at the J times, $\tau_{\eta_1} + T, \dots, \tau_{\eta_J} + T$. The Fourier coefficients are then eliminated to yield the desired direct relation. To compute the Fourier coefficients, write (1) as a system of J linear equations in J unknowns [6]:

$$\Gamma^{-1} \begin{bmatrix} V_0 \\ V_1^C \\ V_1^S \\ \vdots \\ V_K^C \\ V_K^S \end{bmatrix} = \begin{bmatrix} v_n(\tau_{\eta_1}) \\ v_n(\tau_{\eta_2}) \\ v_n(\tau_{\eta_3}) \\ \vdots \\ v_n(\tau_{\eta_J}) \end{bmatrix} \quad (2)$$

where $\Gamma^{-1} \in \mathbb{R}^{J \times J}$ is given by

$$\begin{bmatrix} 1 & \cos \omega\tau_{\eta_1} & \sin \omega\tau_{\eta_1} & \cdots & \cos K\omega\tau_{\eta_1} & \sin K\omega\tau_{\eta_1} \\ 1 & \cos \omega\tau_{\eta_2} & \sin \omega\tau_{\eta_2} & \cdots & \cos K\omega\tau_{\eta_2} & \sin K\omega\tau_{\eta_2} \\ 1 & \cos \omega\tau_{\eta_3} & \sin \omega\tau_{\eta_3} & \cdots & \cos K\omega\tau_{\eta_3} & \sin K\omega\tau_{\eta_3} \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 1 & \cos \omega\tau_{\eta_J} & \sin \omega\tau_{\eta_J} & \cdots & \cos K\omega\tau_{\eta_J} & \sin K\omega\tau_{\eta_J} \end{bmatrix} \quad (3)$$

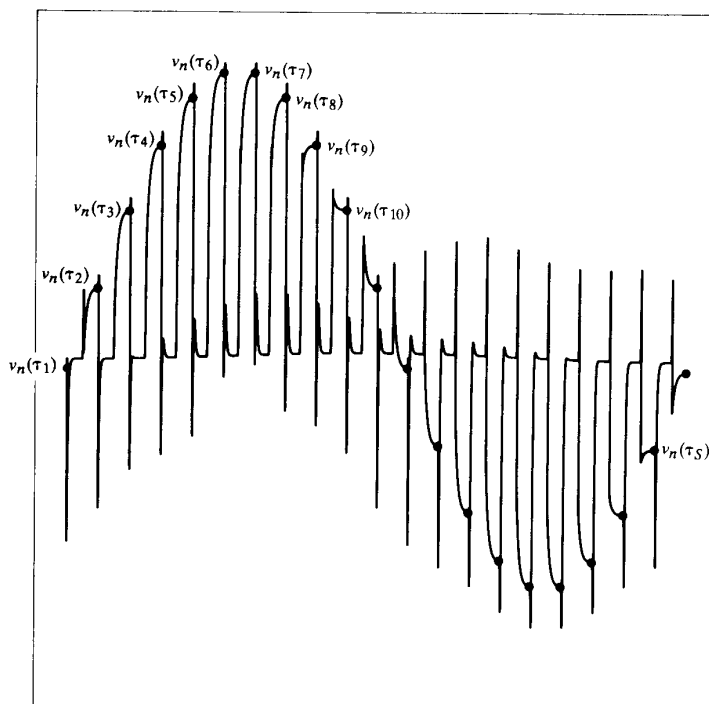


Fig. 1. Response of a switching filter circuit to a periodic function, with the initial point of each cycle denoted.

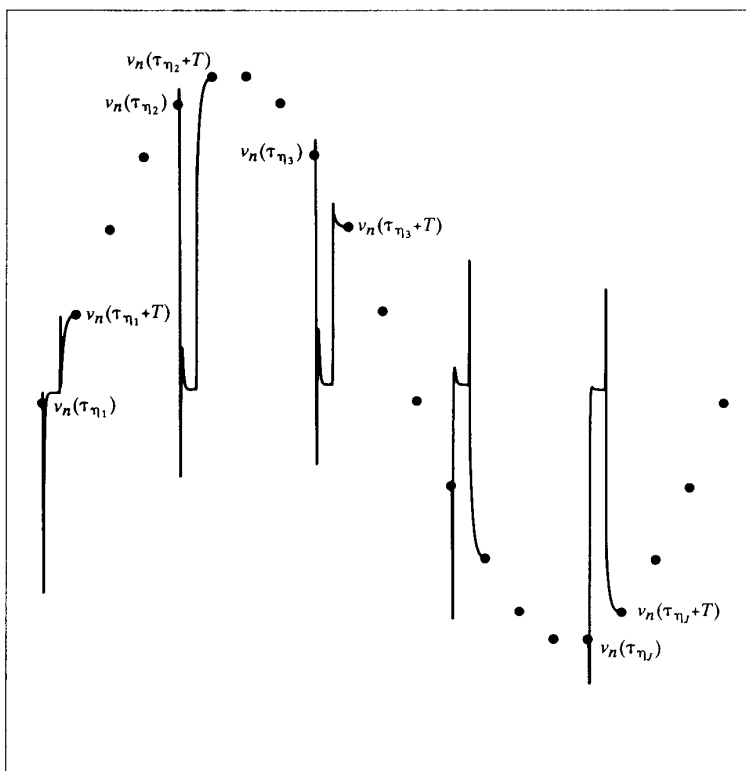


Fig. 2. Discrete waveform that is constructed by sampling the response of a circuit at the initial point of each clock cycle. Also shown are the J cycles that calculated in detail.

The matrix Γ^{-1} maps the Fourier coefficients to a sequence and is referred to as the inverse discrete Fourier transform. If the times $\tau_{\eta_1}, \dots, \tau_{\eta_J}$ are reasonably evenly distributed over one period of the input signal, then Γ^{-1} is invertible (for example, see [5]). Its inverse, the forward discrete Fourier transform, is denoted by Γ . We can also write

$$\Gamma^{-1}(T) \begin{bmatrix} V_0 \\ V_1^C \\ V_1^S \\ \vdots \\ V_K^C \\ V_K^S \end{bmatrix} = \begin{bmatrix} v_n(\tau_{\eta_1} + T) \\ v_n(\tau_{\eta_2} + T) \\ v_n(\tau_{\eta_3} + T) \\ \vdots \\ v_n(\tau_{\eta_J} + T) \end{bmatrix} \quad (4)$$

where $\Gamma^{-1}(T) \in \mathfrak{R}^{J \times J}$ is given by

$$\begin{bmatrix} 1 & \cos \omega(\tau_{\eta_1} + T) & \sin \omega(\tau_{\eta_1} + T) & \cdots & \sin K\omega(\tau_{\eta_1} + T) \\ 1 & \cos \omega(\tau_{\eta_2} + T) & \sin \omega(\tau_{\eta_2} + T) & \cdots & \sin K\omega(\tau_{\eta_2} + T) \\ 1 & \cos \omega(\tau_{\eta_3} + T) & \sin \omega(\tau_{\eta_3} + T) & \cdots & \sin K\omega(\tau_{\eta_3} + T) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \cos \omega(\tau_{\eta_J} + T) & \sin \omega(\tau_{\eta_J} + T) & \cdots & \sin K\omega(\tau_{\eta_J} + T) \end{bmatrix}. \quad (5)$$

Given a sequence, a delayed version is computed by applying Γ to the sequence to compute the Fourier coefficients, and then multiplying the vector of coefficients by $\Gamma^{-1}(T)$:

$$\begin{bmatrix} v_n(\tau_{\eta_1} + T) \\ v_n(\tau_{\eta_2} + T) \\ \vdots \\ v_n(\tau_{\eta_J} + T) \end{bmatrix} = \Gamma^{-1}(T) \Gamma \begin{bmatrix} v_n(\tau_{\eta_1}) \\ v_n(\tau_{\eta_2}) \\ \vdots \\ v_n(\tau_{\eta_J}) \end{bmatrix}. \quad (6)$$

Thus, the delay matrix $\mathcal{D}(T) \in \mathfrak{R}^{J \times J}$ is defined as

$$\mathcal{D}(T) = \Gamma^{-1}(T) \Gamma. \quad (7)$$

As the delay matrix is a function only of ω , K , $\{\tau_{\eta_1}, \dots, \tau_{\eta_J}\}$, and T , it can be computed once and used for every node.

B. The Differential Equation Relation

We assumed that any switching filter circuit to be simulated can be described by a system of differential equations of the form

$$\frac{d}{dt} q(v(t), u(t)) + i(v(t), u(t)) = 0 \quad (8)$$

where $v(t) \in \mathfrak{R}^N$ is the vector of node voltages, $u(t) \in \mathfrak{R}^M$

is the vector of input sources, $q(v(t), u(t)) \in \mathfrak{R}^N$ is the vector of sums of charges at each node, and $i(v(t), u(t)) \in \mathfrak{R}^N$ is the vector of sums of currents entering each node. If the node voltages are known at some time t_0 , it is possible to solve (8) and compute the node voltages at some later time t_1 . In general, one can write

$$v(t_1) = \phi(v(t_0), t_0, t_1) \quad (9)$$

where ϕ is referred to as the state transition function for the differential equation and can be expanded as

$$\phi(v(t_0), t_0, t_1) = \begin{bmatrix} \phi_1(v(t_0), t_0, t_1) \\ \vdots \\ \phi_N(v(t_0), t_0, t_1) \end{bmatrix} \quad (10)$$

where $\phi_n: \mathfrak{R}^{N \times 1 \times 1} \rightarrow \mathfrak{R}$ for all nodes $n \in \{1, \dots, N\}$.

Now reconsider the J initial points at some circuit node n , $v_n(\tau_{\eta_1}), \dots, v_n(\tau_{\eta_J})$ (Fig. 2). For each $j \in \{1, \dots, J\}$ and each $n \in \{1, \dots, N\}$ we can write

$$v_n(\tau_{\eta_j} + T) = \phi_n(v(\tau_{\eta_j}), \tau_{\eta_j}, \tau_{\eta_j} + T) \quad (11)$$

where T is the clock period. Note that $v_n(\tau_{\eta_j} + T)$ is the initial point of the cycle immediately following the cycle beginning at τ_{η_j} . Also, the node voltages at τ_{η_j} can be related to the node voltages at $\tau_{\eta_j} + T$ by the delay matrix $\mathcal{D}(T)$. That is,

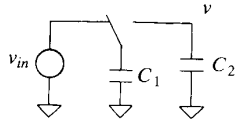
$$\mathcal{D}(T) \begin{bmatrix} v_n(\tau_{\eta_1}) \\ \vdots \\ v_n(\tau_{\eta_J}) \end{bmatrix} = \begin{bmatrix} v_n(\tau_{\eta_1} + T) \\ \vdots \\ v_n(\tau_{\eta_J} + T) \end{bmatrix}. \quad (12)$$

It is possible to use (11) to eliminate the $v_n(\tau_{\eta_j} + T)$ terms from (12), which yields

$$\mathcal{D}(T) \begin{bmatrix} v_n(\tau_{\eta_1}) \\ \vdots \\ v_n(\tau_{\eta_J}) \end{bmatrix} = \begin{bmatrix} \phi_n(v(\tau_{\eta_1}), \tau_{\eta_1}, \tau_{\eta_1} + T) \\ \vdots \\ \phi_n(v(\tau_{\eta_J}), \tau_{\eta_J}, \tau_{\eta_J} + T) \end{bmatrix} \quad (13)$$

for each $n \in \{1, \dots, N\}$.

The collection of equations given in (13) is a system of NJ equations in NJ unknowns, in which the unknowns are the N vectors of node voltages at the beginnings of J

Fig. 3. Simple switched-capacitor RC low-pass filter.

cycles. Note that since this system of equations was constructed assuming the truncated Fourier series in (1) was exact, the computed initial points will only approximate the exact steady-state solution. Finally, note also that once the J initial points are computed, the Fourier coefficients in (1) can be found easily by multiplication by Γ , and therefore distortion harmonics are directly available.

C. A Simple Example

As a simple example of the mixed frequency–time algorithm, consider the simple switched-capacitor RC one-pole filter shown in Fig. 3. It is easy to show that

$$\phi_1(v(\tau_s), \tau_s, \tau_s + T) = \frac{C_1 v_{in}(\tau_s) + C_2 v(\tau_s)}{C_1 + C_2}. \quad (14)$$

Since the circuit is linear, it is only necessary to consider dc and the fundamental in the solution, and so only three samples are needed. Assume the fundamental is 1 Hz and the clock is 6 Hz, and choose the samples to be taken at $t = \{0, \frac{1}{3}, \frac{2}{3}\}$. Then

$$\Gamma^{-1} = \begin{bmatrix} 1 & \cos 0 & \sin 0 \\ 1 & \cos \frac{2\pi}{3} & \sin \frac{2\pi}{3} \\ 1 & \cos \frac{4\pi}{3} & \sin \frac{4\pi}{3} \end{bmatrix} \quad (15)$$

$$\Gamma^{-1}(T) = \begin{bmatrix} 1 & \cos \frac{\pi}{3} & \sin \frac{\pi}{3} \\ 1 & \cos \pi & \sin \pi \\ 1 & \cos \frac{5\pi}{3} & \sin \frac{5\pi}{3} \end{bmatrix} \quad (16)$$

and $D(T) = \Gamma^{-1}(T)\Gamma$. Substituting into (13)

$$D(T) \begin{bmatrix} v(\tau_0) \\ v(\tau_2) \\ v(\tau_4) \end{bmatrix} = \frac{1}{C_1 + C_2} \begin{bmatrix} C_1 v_{in}(\tau_0) + C_2 v(\tau_0) \\ C_1 v_{in}(\tau_2) + C_2 v(\tau_2) \\ C_1 v_{in}(\tau_4) + C_2 v(\tau_4) \end{bmatrix}. \quad (17)$$

By giving values for C_1 , C_2 , and $v_{in}(\tau_s)$ for $s = 0, 2, 4$, this linear equation is easily solved for $v(\tau_s)$.

III. SOLUTION BY NEWTON–RAPHSOON

The collection of equations given in (13) can be reorganized into a system of NJ equations in NJ unknowns as

$$F \begin{bmatrix} v_1(\tau_{\eta_1}) \\ \vdots \\ v_N(\tau_{\eta_1}) \\ \vdots \\ v_1(\tau_{\eta_J}) \\ \vdots \\ v_N(\tau_{\eta_J}) \end{bmatrix} = \mathcal{D}_N(T) \begin{bmatrix} v_1(\tau_{\eta_1}) \\ \vdots \\ v_N(\tau_{\eta_1}) \\ \vdots \\ v_1(\tau_{\eta_J}) \\ \vdots \\ v_N(\tau_{\eta_J}) \end{bmatrix} - \begin{bmatrix} \phi_1(v(\tau_{\eta_1}), \tau_{\eta_1}, \tau_{\eta_1} + T) \\ \vdots \\ \phi_N(v(\tau_{\eta_1}), \tau_{\eta_1}, \tau_{\eta_1} + T) \\ \vdots \\ \phi_1(v(\tau_{\eta_J}), \tau_{\eta_J}, \tau_{\eta_J} + T) \\ \vdots \\ \phi_N(v(\tau_{\eta_J}), \tau_{\eta_J}, \tau_{\eta_J} + T) \end{bmatrix} \quad (18)$$

and

$$F \begin{bmatrix} v(\tau_{\eta_1}) \\ \vdots \\ v(\tau_{\eta_J}) \end{bmatrix} = 0 \quad (19)$$

where $F: \mathfrak{R}^{NJ} \rightarrow \mathfrak{R}^{NJ}$, and $\mathcal{D}_N \in \mathfrak{R}^{NJ \times NJ}$ is given by

$$\mathcal{D}_N(T) = \begin{bmatrix} d_{11}I_N & \cdots & d_{1J}I_N \\ \vdots & & \vdots \\ d_{J1}I_N & \cdots & d_{JJ}I_N \end{bmatrix} \quad (20)$$

where $d_{ij} \in \mathfrak{R}$ is the ij th element of the delay matrix $\mathcal{D}(T)$, and $I_N \in \mathfrak{R}^{N \times N}$ is the identity matrix.

Applying Newton's method to (18) leads to the iteration equation

$$J_F \begin{bmatrix} v^{(l)}(\tau_{\eta_1}) \\ \vdots \\ v^{(l)}(\tau_{\eta_J}) \end{bmatrix} \begin{bmatrix} v^{(l+1)}(\tau_{\eta_1}) - v^{(l)}(\tau_{\eta_1}) \\ \vdots \\ v^{(l+1)}(\tau_{\eta_J}) - v^{(l)}(\tau_{\eta_J}) \end{bmatrix} = -F \begin{bmatrix} v^{(l)}(\tau_{\eta_1}) \\ \vdots \\ v^{(l)}(\tau_{\eta_J}) \end{bmatrix} \quad (21)$$

where l is the iteration number and $J_F \in \mathfrak{R}^{NJ \times NJ}$ is the Fréchet derivative of F given by

$$\mathcal{D}_N(T) - \text{diag} \left(\frac{\partial \phi(v(\tau_{\eta_1}), \tau_{\eta_1}, \tau_{\eta_1} + T)}{\partial v(\tau_{\eta_1})}, \dots, \frac{\partial \phi(v(\tau_{\eta_J}), \tau_{\eta_J}, \tau_{\eta_J} + T)}{\partial v(\tau_{\eta_J})} \right). \quad (22)$$

There are two important pieces to the computation of one Newton iteration: factoring the matrix J_F , which is sparse, and evaluating J_F and F , which involves computing the state transition function $\phi(v(\tau_{\eta_j}), \tau_{\eta_j}, \tau_{\eta_j} + T)$ and

its derivative for each $j \in \{1, \dots, J\}$. The state transition functions can be evaluated by numerically integrating (8) over the J periods. The derivatives of the state transition functions, referred to as the sensitivity matrices, can be computed with a small amount of additional work during the numerical integration [7].

To show how the computation of the state transition function and its derivative fit together, consider numerically integrating (8) with backward Euler, which we chose for simplicity and because it appears to be one of the best formulas for switching filter circuits. Given some initial time t_0 and some initial condition $v(t_0)$, applying backward-Euler to (8) results in the following algebraic equation:

$$f(v(t_0 + h), v(t_0)) = \frac{1}{h} (q(v(t_0 + h)) - q(v(t_0))) + i(v(t_0 + h)) = 0 \quad (23)$$

where $h \in \mathfrak{R}$ is the time step. Note that we have dropped explicitly denoting the dependence of q and i on the input u for simplicity.

Equation (23) is usually solved with Newton-Raphson, for which the iteration equation is

$$J_f(v^{(l)}(t_0 + h))(v^{(l+1)}(t_0 + h) - v^{(l)}(t_0 + h)) = -f(v^{(l)}(t_0 + h), v^{(l)}(t_0)) \quad (24)$$

where l is the iteration index and $J_f(v(t)) \in \mathfrak{R}^{N \times N}$ is the Frechet derivative of the nonlinear equation in (23) and is given by

$$J_f(v(t)) = \frac{\partial f(v(t), v(t_0))}{\partial v(t)} = \frac{1}{h} \frac{\partial q(v(t))}{\partial v(t)} + \frac{\partial i(v(t))}{\partial v(t)}. \quad (25)$$

Solving (23) yields an approximation to $v(t_0 + h) = \phi(v(t_0), t_0, t_0 + h)$. Implicitly differentiating (23) for $v(t_0 + h)$ with respect to $v(t_0)$ yields

$$J_f(v(t_0 + h)) \frac{\partial v(t_0 + h)}{\partial v(t_0)} = \frac{1}{h} \frac{\partial q(v(t_0))}{\partial v(t_0)} \frac{\partial v(t_0)}{\partial v(t_0)}. \quad (26)$$

Note here that

$$\frac{\partial v(t_0)}{\partial v(t_0)} \in \mathfrak{R}^{N \times N}$$

is the identity matrix and in general

$$\frac{\partial v(t_0 + h)}{\partial v(t_0)} \in \mathfrak{R}^{N \times N}$$

is not the identity.

Given $v(t_0)$, (23) can be repeatedly applied to find $v(t_0 + T) = \phi(v(t_0), t_0, t_0 + T)$, and (26) can be repeatedly applied to find the sensitivity matrix $\partial v(t_0 + T)/\partial v(t_0) = \partial \phi(v(t_0), t_0, t_0 + T)/\partial v(t_0)$. Note that J_f is required in

both (24) and (26), and thus the sensitivity matrix update can be made more efficient by factoring J_f 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. We return to this point at the end of Section IV.

IV. IMPLEMENTATION IN *Nitswit*

Both the classical direct methods and the mixed frequency-time methods have been implemented in the simulation program *Nitswit*, which is written in the computer language "C." *Nitswit* takes as input a file with a SPICE-like description of the circuit, that is, a list of elements (MOS transistors, resistors, capacitors, etc.) with their node connections, and a list of options to select among methods. If the mixed frequency-time method is used, a switching clock period and an input frequency, along with a number of harmonics, must be specified. The program produces output waveforms as in Fig. 1 for direct methods, and waveforms as in Fig. 2 and Fourier coefficients for the sampled waveforms with the mixed frequency-time algorithm.

The details of the steps used in *Nitswit* program are described above, and a summary of the steps is given in the following algorithm:

Nitswit Algorithm for Steady State

The superscript l is the iteration count and ϵ is a small positive number.

$l \leftarrow 0$.

Select J clock cycles.

Guess $v^0(\tau_{\eta_1}), \dots, v^0(\tau_{\eta_J})$.

repeat {

$l \leftarrow l + 1$.

for each ($j \in \{1 \dots J\}$) {

Numerically integrate cycle j with

the initial condition $v^{l-1}(\tau_{\eta_j})$ to

compute $v^{l-1}(\tau_{\eta_j} + T)$ and the sensitivity matrix.

}

If $\max_{j \in \{1, \dots, J\}} \|v^{l-1}(\tau_{\eta_j} + T) - D(T)v^{l-1}(\tau_{\eta_j})\| < \epsilon$,

leave.

Compute

$v^l(\tau_{\eta_1}), \dots, v^l(\tau_{\eta_J})$ from $v^{l-1}(\tau_{\eta_1}), \dots, v^{l-1}(\tau_{\eta_J})$ as in (17).

}

A. Application Examples

Nitswit is particularly efficient when simulating switched-capacitor filters. One reason is that switched-capacitor filter are usually followed by a sampler, and so only the initial point of each cycle is needed. Also, the circuits are generally designed to exhibit little distortion, so if driven by a sinusoid, only a few harmonics of the sequence of initial points are significant and only a few full clock cycles need to be computed. Finally, the state

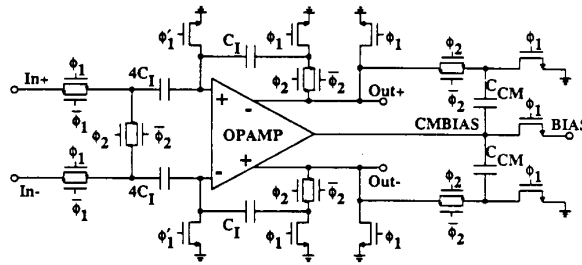


Fig. 4. Fully differential switched-capacitor sample-and-hold amplifier [8].

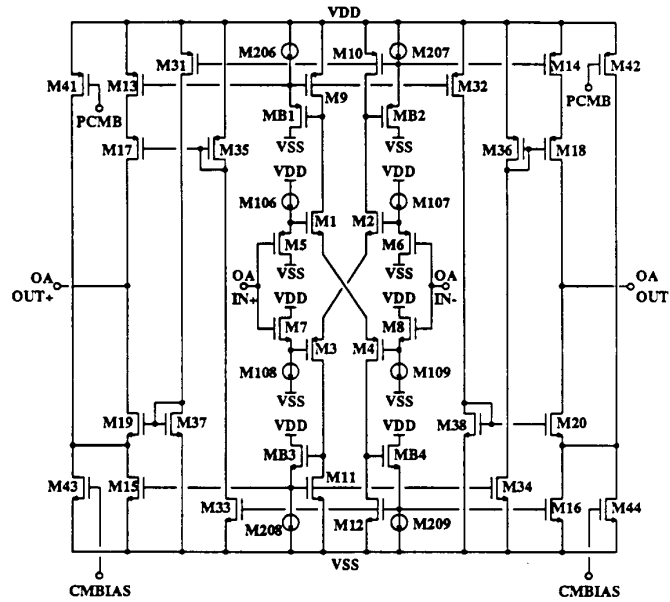


Fig. 5. Operational amplifier used in the sample-and-hold amplifier of Fig. 4 [8].

transition function for a switched-capacitor filter over a clock cycle is nearly affine (linear plus a constant), and therefore the Newton method in (21) converges in just a few iterations.

To demonstrate the effectiveness and versatility of the algorithms used in *Nitswit*, we consider analyzing the distortion of an unusual switched-capacitor circuit. Fig. 4 shows a high-speed fully differential switched-capacitor sample-and-hold amplifier [8]. This circuit precedes an analog-to-digital converter and has all three characteristics mentioned above. To measure the distortion of this circuit a sine wave is applied to the input and a periodic clock is applied to the sample/hold input. The output signal is then sampled at the end of each hold interval and a Fourier series is constructed from the sampled signal. Sampling of the output at the end of the hold interval is appropriate and eliminates settling effects resulting from transitions of the sample/hold signal that are ignored by the analog-to-digital converter.

Fig. 5 shows the operational amplifier used in the sample and hold of Fig. 4. The combined circuit contains 65 nodes. The distortion of this circuit was measured with *Nitswit* versus the amplitude of the input signal (Fig. 6)

and versus the sample/hold clock frequency (Fig. 7). Of particular interest to the designer is the distortion versus clock rate. This is a quantity that cannot be determined except with a circuit-level simulator such as *Nitswit* or by measuring the actual circuit.

The Fourier series (1) for the sampled signal was truncated after three harmonics. For the case where the input was a 2-V differential 500-kHz sine wave and the sample/hold clock rate was 10 MHz, this gave results that were identical to direct methods to within the truncation error of the integration method.

As another example of the capability of *Nitswit*, a single-pole switched-capacitor low-pass active filter with a clock of 500 kHz, a bandwidth of 30 kHz, and an input frequency of 20 kHz was simulated. Tables I and II show the distortion that results as a function of nonlinearities in the capacitor and the finite rise and fall time of the clock waveforms. To simulate the effect of nonlinear filter capacitance on distortion, the filter capacitors are assumed to be first order voltage-controlled nonlinear capacitors with capacitance $c = c_0(1 + \alpha v_c)$. In Table I, the distortion, specifically the magnitude of the second and third harmonics, is given for several different values of α . The distortion for

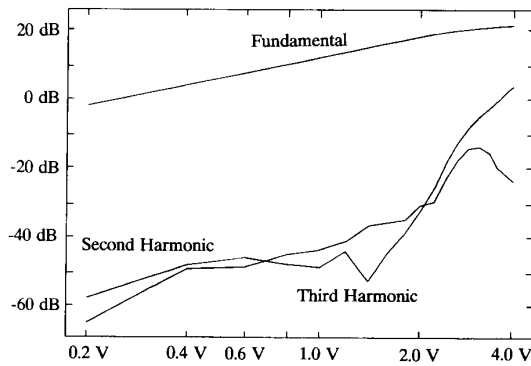


Fig. 6. Distortion of the sample-and-hold amplifier as a function of input signal amplitude. The input signal frequency is 500 kHz and the sample/hold clock rate was 10 MHz.

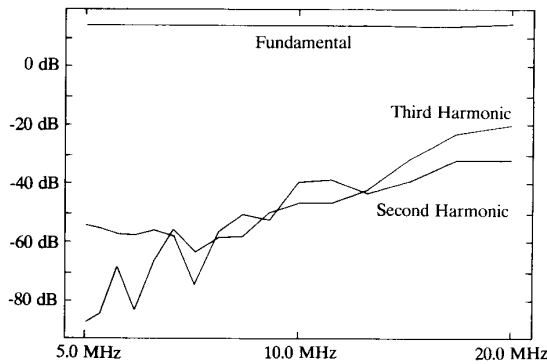


Fig. 7. Distortion of the sample-and-hold amplifier as a function of sample/hold clock frequency. The amplitude of the input signal is 1.25-V differential and its frequency is 500 kHz.

TABLE I
Nitswit RELATIVE DISTORTION RESULTS FOR A SWITCHED-CAPACITOR LOW-PASS FILTER AS A FUNCTION OF INCREASING FILTER CAPACITOR NONLINEARITY

α	— Second Harmonic —	— Third Harmonic —
0.001	0.00057	0.00010
0.01	0.0014	0.00007
0.1	0.0101	0.00024

TABLE II
Nitswit DISTORTION RESULTS FOR A SWITCHED-CAPACITOR LOW-PASS FILTER AS A FUNCTION OF INCREASING CLOCK RISE AND FALL TIME

Rise/Fall Time	— Second Harmonic —	— Third Harmonic —
1ns	0.000020	5.99e-7
10ns	0.00020	0.000037
100ns	0.0018	0.000054

the same low-pass filter circuit with linear filter capacitors is given as a function of the clock rise and fall time in Table II.

B. Comparison to Direct Methods

The program *Nitswit* contains two algorithms capable of finding the steady-state response of a circuit. The first is simply a transient analysis that is continued until a steady state is achieved. The second, of course, is the mixed

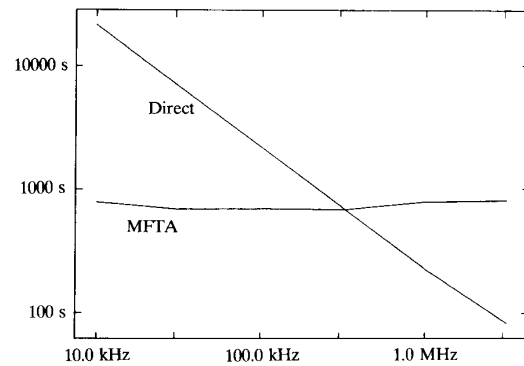


Fig. 8. Time required for simulation of sample-and-hold amplifier to steady state versus input frequency using direct methods and the mixed frequency-time algorithm (MFTA). These times were measured on a HP9000/370 with a floating-point accelerator.

TABLE III
Nitswit RESULTS FROM A VAX 8650 RUNNING ULTRIX 2.0

circuit name	nodes	cycles/period	direct time (sec)	mixed frequency-time		
				harmonics	Newton iterations	time (sec)
<i>sc1pf</i>	2	33	24.5	3	3	4.3
<i>scop</i>	13	100	522	3	6	90
<i>mixer</i>	34	1000	7132	3	4	161
<i>frog</i>	77	1000	12,987	3	6	1228

frequency-time algorithm. Coding both algorithms into the same simulator provides a fair evaluation of the mixed-frequency approach.

Fig. 8 shows the time required to compute the steady state of the sample and hold of Fig. 4 as a function of the input signal frequency for both direct methods and for the mixed frequency-time algorithm. The amplitude of the input signal is 1.25-V differential. The sample/hold clock is fixed at 5 MHz and three harmonics of the sampled signal are computed. This figure shows that the time required for the mixed frequency-time algorithm is roughly independent of the frequency of the input signal whereas the time required for direct methods is proportional to the ratio of the clock frequency to the input frequency.

Results for four circuits are given in Table III. The first, *sc1pf*, is an RC one-pole SC filter. The second, *scop*, is a one-pole active CMOS low-pass filter. The circuit *mixer* is a double-balanced switching mixer with a 1.001-MHz RF input signal and a 1-MHz LO signal. This circuit shows that *Nitswit* is not limited to switched-capacitor circuits. The last circuit, *frog*, is a five-pole Chebyshev active CMOS leapfrog filter with 0.1-dB ripple. This circuit is driven with a 1-MHz clock, has a 20-kHz bandwidth, and is being driven with a 1-kHz test signal to measure its distortion.

Examination of the results above indicates as much as an order of magnitude speed increase over traditional methods, but this is not as much as one would expect. One of the reasons is that these simulations were performed with a SPICE level 1 MOS model modified to conserve charge. This model was chosen simply because it was easy to implement. However, it is generally considered to be too inaccurate to be suitable for analog circuits. Since the

mixed frequency-time algorithm has extra overhead that tends to hide the time required for model evaluation, it is expected that the algorithm should do better with respect to direct methods when a more accurate (and therefore a more complicated) model is used. Also much of the CPU time for large circuits, such as *frog*, is spent calculating the dense sensitivity matrix and factoring the Jacobian in (22). It does turn out, however, that almost all the entries of the sensitivity matrix are near zero, and this suggests significant speed improvements can be achieved by ignoring those terms.

V. CONCLUSION

A new efficient mixed frequency-time approach to computing steady-state and intermodulation distortion of switching filters without resorting to macromodeling or the slow-clock approach has been presented. The method works by computing the solution to the differential equation system associated with a circuit for only J clock cycles, where J is the number of coefficients needed in the Fourier series to represent accurately the sequence of initial points in each clock cycle. Thus, this method is particularly efficient when the number of coefficients in the Fourier series is many fewer than the number of clock cycles in one input signal period.

Since our approach finds the steady-state solution directly and performs a circuit-level simulation, it is capable of accurately predicting distortion performance. This mixed frequency-time approach can also be used when the input consists of the sum of two periodic signals at unrelated frequencies. Thus, the intermodulation distortion can be directly computed, which is particularly useful for band-pass filters. Also, the fact that steady state is computed directly implies an additional advantage over transient methods when high-Q filters are simulated. One final point: the mixed frequency-time method can also be adapted to the macromodeling approach used in other switching filter simulators, accelerating those methods as well when the steady-state solution is desired.

Future work on this method will be to adapt it to other traditionally hard-to-simulate circuits like switching power supplies and phase-locked loops. Another important aspect of this algorithm is that, upon examination of (18), it is clear that the J integrations of the differential equation to compute the J ϕ 's and their derivatives are independent. Therefore, the mixed frequency-time algorithm is extremely well suited to implementation on a parallel processor.

ACKNOWLEDGMENT

The authors would like to acknowledge the discussions with Prof. A. R. Newton; the help of R. Armstrong, A. Lumsdaine, K. Nabors, M. Reichelt, H. Yaghtiel, and S. Lewis; and the support of B. Bratzel and M. Glanville. In addition, we would especially like to thank R. Saleh, for

his many contributions to this project, G. Sorkin, who helped with Fourier transforms for nonuniformly spaced time sequences, and C. Conroy, for his help with the sample-and-hold amplifier.

REFERENCES

- [1] L. W. Nagel, "SPICE2: A computer program to simulate semiconductor circuits," Electron. Res. Lab., Univ. of Calif., Berkeley, Rep. ERL M520, May 1975.
- [2] W. T. Weeks *et al.*, "Algorithms for ASTAP—A network analysis program," *IEEE Trans. Circuit Theory*, pp. 628–634, Nov. 1973.
- [3] H. De Man, J. Rabaey, G. Arnout, and J. Vandewalle, "Practical implementation of a general computer aided design technique for switched capacitor circuits," *IEEE J. Solid-State Circuits*, vol. SC-15, pp. 190–200, Apr. 1980.
- [4] S. C. Fang *et al.*, "SWITCAP: A switched-capacitor network analysis program—Part 1: Basic features," *IEEE Circuits Syst. Mag.*, vol. 5, no. 3, pp. 4–10, Sept. 1983; also "SWITCAP: A switched-capacitor network analysis program—Part 2: Advanced applications," *ibid.*, vol. 5, no. 3, pp. 41–46, Sept. 1983.
- [5] L. Chua and A. Usida, "Algorithms for computing almost-periodic steady-state response of nonlinear systems to multiple input frequencies," *IEEE Trans. Circuits Syst.*, vol. CAS-28, pp. 953–971, Oct. 1981.
- [6] K. S. Kundert, G. B. Sorkin, and A. Sangiovanni-Vincentelli, "Applying harmonic balance to almost-periodic circuits," *IEEE Trans. Microwave Theory Tech.*, vol. 36, pp. 366–378, Feb. 1988.
- [7] T. N. Trick, F. R. Colon, and R. P. Fan, "Computation of capacitor voltage and inductor current sensitivities with respect to initial conditions for the steady state analysis of nonlinear periodic circuits," *IEEE Trans. Circuits Syst.*, vol. CAS-22, pp. 391–396, May 1975.
- [8] S. H. Lewis and P. R. Gray, "A pipelined 5-Msample/s 9-bit analog-to-digital converter," *IEEE J. Solid-State Circuits*, vol. SC-22, pp. 954–961, Dec. 1987.



Kenneth S. Kundert (S'86) received the M.Eng. and B.S. degrees in electrical engineering and computer sciences from the University of California, Berkeley, in 1983 and 1979, respectively. His area of specialization at the time was analog circuit design. He is currently working towards the Ph.D. degree in electrical engineering and computer science at Berkeley and is supported by a Hewlett-Packard doctoral fellowship. His research topic is circuit simulation with particular emphasis on analog and microwave circuits.

From 1979 to 1981 he worked at Hewlett-Packard designing portions of a high-performance microwave network analyzer. Since 1981 he has been at Berkeley, working first on a low-distortion switched-capacitor synchronous detector, then on a sparse matrix package for circuit simulators, and now on frequency-domain circuit simulation. His current interests include the design and simulation of analog circuits and numeric methods.



Jacob White (S'80–M'83) received the B.S. degree in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, and the M.S. and Ph.D. degrees in electrical engineering and computer science from the University of California, Berkeley.

He worked at the IBM T. J. Watson Research Center from 1985 to 1987, and is currently the Analog Devices Career Development Assistant Professor at the Massachusetts Institute of Technology. His current interests are serial and parallel numerical methods for problems in circuit and device design.

Dr. White is a 1988 Presidential Young Investigator.

Alberto Sangiovanni-Vincentelli (M'74–SM'81–F'83), for photograph and biography please see this issue, p. 408.