FINDING THE STEADY-STATE RESPONSE OF ANALOG AND MICROWAVE CIRCUITS

Kenneth S. Kundert and Alberto Sangiovanni-Vincentelli
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley, CA. 94704

ABSTRACT

Being able to compute the steady-state response of a circuit is of prime importance when working with analog and microwave circuits. It is possible to compute the steady-state response using conventional transient analysis methods, however, doing so is very inefficient when the circuit exhibits time constants that are long compared to the time interval of interest. This paper presents two methods that are capable of computing the steady-state response of a circuit directly — harmonic balance, a frequency-domain method, and the finite-difference method, which is based in the time domain. Both of these methods are suitable for use on nonlinear circuits that contain distributed devices and that exhibit either periodic or quasiperiodic steady-state solutions.

1 INTRODUCTION

Often when simulating analog and microwave circuits, the steady-state behavior of the circuit is of primary interest. For example, the distortion, power level, noise, and transfer characteristics such as gain and impedance of a circuit can only be determined from its steady-state response. Thus, it is desirable for a circuit simulator used on analog and microwave circuits to be able to compute efficiently and accurately the steady-state solution of such circuits.

One approach to calculating the steady-state response is to integrate the differential equations that describe the circuit from some chosen initial state until any transient behavior dies out; an approach that suffers from several fundamental drawbacks.

First, it can be quite difficult to determine when the transient has died out. If the time constants involved in the transient are large, the response must be observed over a long interval to be able to conclude that the circuit is in steady-state. If not observed for a period of time which is long enough, the circuit may be erroneously declared as being in steady state when in reality is quite far from it. This can be a serious problem. For example, to determine the linearity of an amplifier, the steady-state response of the circuit to a sinusoidal input is computed and its deviation from a pure sinusoid evaluated. Any transient that is present when the output is computed is erroneously considered to be distortion.

Second, even if it is possible to detect correctly that steady state has been achieved, it may take a long time for the transients to decay, and thus, this approach would involve an expensive calculation. It is sometimes possible to avoid this problem by using a customized integration method to accelerate the process of reaching steady state. For example, if the circuit exhibits lightly damped oscillations, it is possible to have the integration method follow the envelope of the solution rather than the solution itself [petzold81].

Finally, many analog and microwave circuits, such as mixers, have inputs at two or more independent frequencies. These frequencies are often such that the ratio of the highest to lowest frequency generated by the nonlinearities in the circuit is large. In a transient simulation, the size of the time step is proportional to the highest frequency and the length of the simulation interval proportional to the lowest. As a result, these circuits often require a vast number of time points.

This paper presents a number of methods for directly computing the steady-state response of nonlinear circuits. However, before introducing these methods, we will clarify what is meant by steady state. In the most general terms, the steady-state solution of a differential equation is the one that is asymptotically approached as the effect of the initial condition dies out. An important aspect of this definition is that the effect of the initial condition must decay, but it is not necessary for it to disappear completely. For example, a circuit such as a flip-flop with more than one stable state has distinct initial conditions that eventually result in different steady-state solutions. However, each steady-state solution is reached regardless of small changes in the initial condition. This is equivalent to saying that if the differential equation is at a steady-state solution and is perturbed slightly and temporarily, it will return to the same solution. Such a solution is referred to as being asymptotically stable. Notice that this definition excludes lossless linear LC oscillators.

There are several different kinds of steady-state behavior that are of interest. The first is DC steady state. Here the solution is an equilibrium point of the circuit and is independent of time. Stable linear circuits driven by sinusoidal sources eventually exhibit sinusoidal steady state solutions, which are characterized as being purely sinusoidal except possibly for some DC offset. Periodic steady state is the steady-state response of a nonlinear circuit driven by a periodic source. A periodic steady-state solution consists solely of a DC offset and harmonically related sinusoids. The period of the solution is usually equal to that of the input, though occasionally the periods of the two will be multiples of some common period. If a nonlinear circuit is driven with several periodic sources at unrelated frequencies, the steady-state response is called quasiperiodic. A quasiperiodic steady-state response consist of sinusoids at the sum and difference frequencies of two or more fundamental frequencies and their harmonics. The frequencies of the input signals usually equal that of the fundamentals, thought sometimes they are even multiples. Other forms of steady-state behavior exist, but are beyond the scope of this paper.

In this paper we discuss the computation of periodic and quasiperiodic solutions of ordinary differential equations. We do not guarantee that these solutions represent steady-state solutions because the methods presented are unable to distinguish between stable and unstable solutions.

The traditional approach to finding the periodic steady-state response of a circuit is to use shooting methods [aprille72, skelboe80]. These methods exploit the possibility of finding an initial condition for the differential equations that describe the circuit such that no transient is excited and the result-

IEEE 1988 CUSTOM INTEGRATED CIRCUITS CONFERENCE

ing response is the steady-state solution. Shooting methods attempt to find such initial conditions using an iterative algorithm. The algorithm for a periodically driven circuit begins by guessing the initial conditions and simulating the circuit for one period using these conditions. The response is checked to see if it is periodic and if not, how far is it from being periodic. A new initial guess is then generated that presumably results in a response that is closer to being periodic. The method used to generate the new initial conditions is what differentiates the shooting method variants [kundert88a].

When trying to apply shooting methods to analog and microwave circuits, two fundamental problems are encountered. Shooting methods find the periodic steady-state response of a circuit by exploiting the fact that, under certain circumstances, it is possible to formulate the periodicity constraint (that $x(t) = x(t+T)$ for all t) as a two-point boundary-value constraint (that $x(t) = x(t+T)$ for some *fixed* t) [keller76, stoer80]. Unfortunately, in general, the steady-state response of a mixer, a very important communication circuit, is not periodic, but rather quasiperiodic. The quasiperiodic steady-state constraint cannot be formulated as a boundary-value constraint. Shooting methods are inappropriate for this type of circuit.

On each iteration, shooting methods compute a new initial condition. For lumped circuits, under non degeneracy conditions [desoer69], the initial conditions are a vector of voltages across the capacitors and currents through the inductors. However, if the circuit contains distributed devices, its initial condition can no longer be given using a finite set of numbers. To specify completely the initial conditions for a distributed device requires that the voltage and current be given everywhere along its length. Thus, the initial conditions are specified not with a number, but with a function. Functional initial conditions complicate hopelessly shooting methods and so they are not practical for circuits containing distributed devices. For these reasons, the shooting methods will not be discussed further in this paper.

Harmonic balance [kundert88a] is another method for computing the steady-state response of a circuit that is popular for microwave circuits. It differs from shooting methods in that it assumes that the circuit's steady-state response consists of a sum of sinusoids, and proceeds to find the coefficients of the sinusoids that satisfy the differential equation. Thus, the steady-state solution is calculated directly and any transient is avoided. Harmonic balance is efficient if only a few sinusoids are needed to approximate the solution to the desired accuracy. It is attractive, therefore, when the circuit is driven by sinusoidal sources and when the nonlinearities are driven mildly.

A promising method for finding steady-state solutions that has yet to find application in circuit simulation is the finite-difference method. When finding periodic solutions, finite-difference methods replace the differential equations with finite-difference equations on a mesh of points that cover one period. Trial solutions consist of discrete values, one for each point on the mesh, that do not necessarily satisfy the difference equations but do satisfy the periodicity constraint. The trial solution at each point is adjusted until it also satisfies the difference equations. Finite-difference methods can handle distributed devices and under certain circumstances can compute quasiperiodic steady-state solutions.

This paper is organized as follows. In Section 2, background material is presented and the problem to be solved is introduced. Harmonic balance is covered in depth in Section 3. Section 4 discusses finite-difference methods and their relationship with harmonic balance.

2 BACKGROUND

2.1 Definitions

A *signal* is a function that maps either $\mathbb{R}$ (the reals) or Z (the integers) into $\mathbb{R}$ or C (the space of real pairs)†. The domain and range of the map are physical quantities; the domain is typically time or frequency, and the range is typically voltage, current or charge. A signal whose domain is time is called a *waveform*; one whose domain is frequency is called a *spectrum*. All waveforms are assumed $\mathbb{R}$-valued whereas all spectra are assumed C-valued.

A waveform x is *periodic* with *period* T if $x(t) = x(t+T)$ for all t. $P(T)$ denotes the set of all periodic functions with period T that can be uniformly approximated by the sum of at most a countable number of T-periodic sinusoids. Thus, $P(T)$ consists of waveforms of the form

$$x(t) = \sum_{k=0}^{\infty} (X_k^C \cos\omega_k t + X_k^S \sin\omega_k t), \tag{1}$$

where $\omega_k = 2\pi k/T$, $X_k^C, X_k^S \in \mathbb{R}$, and

$$\sum_{k=0}^{\infty} [(X_k^C)^2 + (X_k^S)^2] < \infty. \tag{2}$$

A waveform is *almost periodic* if it can be uniformly approximated by the sum of at most a countable number of sinusoids [hale80]. We use $AP(\Lambda)$ to denote the set of all almost periodic waveforms over the set of frequencies Λ. Thus, $AP(\Lambda)$ consists of waveforms of the form

$$x(t) = \sum_{\omega_k \in \Lambda} (X_k^C \cos\omega_k t + X_k^S \sin\omega_k t), \tag{3}$$

where $\Lambda = \{\omega_0, \omega_1, \omega_2, \cdots\}$ and (2) is satisfied. If Λ is finite with K elements, it is denoted Λ_K. If there is a set of d frequencies $\{\lambda_1, \lambda_2, \cdots, \lambda_d\}$ and Λ is such that

$$\Lambda = \{\omega \mid \omega = k_1\lambda_1 + k_2\lambda_2 + \cdots + k_d\lambda_d;\ k_1, k_2, \cdots, k_d \in Z\} \tag{4}$$

then the frequencies $\{\lambda_1, \lambda_2, \cdots, \lambda_d\}$ are referred to as the *fundamental frequencies* and form a basis (called the fundamental basis) for Λ. The sequence of fundamental frequencies $\{\lambda_j\}$ should be linearly independent over the rationals (that is $\sum_{j=1}^{d} k_j\lambda_j = 0$ implies $k_1 = k_2 = \cdots = k_d = 0$) so that each $\omega \in \Lambda$ corresponds uniquely to a sequence of harmonic indices $\{k_j\}$. If Λ is constructed from such a basis, then $AP(\Lambda)$ is also denoted $AP(\lambda_1, \lambda_2, \cdots, \lambda_d)$. Waveforms belonging to this set are referred to as *quasiperiodic*. Note that $P(T) = AP(\lambda_1)$ if $\lambda_1 = 2\pi/T$, and $P(T) \subset AP(\lambda_1, \lambda_2, \cdots, \lambda_d)$ if for some j, $\lambda_j = 2\pi/T$.

The pair $X_k = [X_k^C \ X_k^S]^T \in C$ is the *Fourier coefficient* of the *Fourier exponent* ω_k and $X = [X_0, X_1, X_2, \cdots]^T$ is called the frequency-domain representation, or spectrum, of x. Conversely, x is the time-domain representation, or waveform, of X. If all the frequencies $\omega_k \in \Lambda$ are distinct, (i.e., $\omega_i \neq \omega_j$ for all $i \neq j$) then there exists a linear invertible operator $\mathbb{F}$, referred to as the *Fourier transform*, that maps x to X. This is a more general definition of the Fourier transform than is used elsewhere in that we define the transform for almost-periodic signals as well as periodic signals.

† Throughout this paper, the trigonometric Fourier series is used rather than the exponential to avoid problems with complex numbers and nonanalytic functions when deriving the harmonic Newton algorithm. Thus, a signal at one frequency in a spectrum is described using the coefficients of sine and cosine. The pair of these are said to reside in $C = \mathbb{R}^2$ rather than $\mathbb{C}$.

A collection of devices is called a *system* if the devices are arranged to operate on input signals (the *stimulus*) to produce output signals (the *response*). A system is *autonomous* if both it and its stimulus are time invariant, otherwise it is *forced*. An oscillator is an example of an autonomous system while an amplifier, a filter, and a mixer are all examples of forced systems. Lastly, an *algebraic* or *memoryless* device or system is one whose response is only a function of the present value of its stimulus, not past or future values.

2.2 Problem Formulation

In the interest of keeping notation simple, we consider only nonlinear time-invariant circuits consisting of independent current sources and voltage controlled resistors, capacitors and distributed devices. These restrictions are mostly cosmetic; they allow the use of simple nodal analysis to formulate the circuit equations. If a more general equation formulation method such as modified nodal analysis is used [sangiovanni81], all results presented in this paper can be applied to circuits containing inductors, voltage sources, and current-controlled components. We further assume that the distributed devices are linear, that the circuit is nonautonomous (or forced), and that it has a steady-state solution.

Let N be the number of nodes in the circuit, and assume it has an isolated asymptotically stable almost-periodic solution $v \in AP^N(\Lambda)$; that is, v is a vector of node voltage waveforms, each of which is almost periodic on the set of frequencies Λ. Further assume that the source current waveforms belong to $AP^N(\Lambda)$, and that all device constitutive equations are differentiable when written as functions of voltage. Now, using Kirchoff's current law, the circuit can be described by

$$f(v,t) = i(v(t)) + \dot{q}(v(t)) + \int_{-\infty}^{t} y(t-\tau)v(\tau)d\tau + u(t) = \mathbf{0} \quad (5)$$

where f is the function that maps the node voltage waveforms into the sum of the currents entering each node; $t \in \mathbb{R}$ is time; $\mathbf{0} \in \mathbb{R}^N$ is the zero vector; $u \in AP^N(\Lambda)$ is the vector of source current waveforms; $i, q : \mathbb{R}^N \rightarrow \mathbb{R}^N$ are differentiable functions representing, respectively, the sum of the current entering the nodes from the nonlinear conductors, and the sum of the charge entering the nodes from the nonlinear capacitors; and y is the matrix-valued impulse response of the circuit with the nonlinear devices removed‡.

2.3 Discretization

Closed form solutions to general nonlinear differential equations are not known. Thus, numerical techniques have to be used to solve equations of the form described by (5). Numerical techniques *discretize* (i.e. approximate) the differential equations to yield a finite system of algebraic equations.

A discretization method approximates the solution using a finite linear combination of prespecified basis functions. This accomplishes two goals. First, it allows the solution to be approximated using a finite collection of numbers, such as the coefficients of the linear combination. Second, since the basis functions are known and are being combined linearly, it is possible to precompute algebraic formulas for dynamic operations such as time differentiation, integration, delay and convolution, thereby converting the differential equation into a system of algebraic equations.

‡ To remove a nonlinear device, simply replace its constitutive equation $y = f(x)$ with $y = 0$.

One example of discretization methods are the backward difference methods [gear71]. These methods break the interval for which the solution is desired into a finite number of subintervals using a mesh. The solution is approximated on each subinterval by a low order polynomial. The choice of polynomials is constrained to assure that the resulting function and perhaps its first few derivatives are continuous at the mesh points. In this example, the basis functions are nonzero for only a short interval, and so affect the solution only locally. An example for which the basis functions are used globally is harmonic balance, which uses sinusoids as the basis functions. Local methods can rapidly and accurately respond to abrupt changes in the solution whereas global methods can accurately represent smooth solutions over long time intervals with few basis functions.

Once the basis functions are specified, it is still necessary to chose how the solution is to be represented. The conventional approach is to use sampling, which specifies the approximate solution by giving its value and perhaps the value of its derivatives at the mesh points. The basis functions are used when computing the discrete approximation to the time derivative and when interpolating to find the solution between the given time points. Methods that use a sampled-data representation for the solution are considered finite-difference methods. Another approach is to recognize that the solutions are being approximated by a linear combination of basis functions and use the coefficients of the basis functions to represent the solution. Methods that represent the solution using these coefficients are called expansion methods [kundert88a], of which harmonic balance is an example. The issue of whether to use samples or coefficients to represent the solution is primarily a question of efficiency. Once the basis functions and the formula for computing the time derivative are chosen, the essential characteristics of the discretization method are fixed.

3 HARMONIC BALANCE

Harmonic balance differs from traditional transient analysis in two fundamental ways. These differences allow harmonic balance to compute periodic and quasiperiodic solutions directly and give the method significant advantages in terms of accuracy and efficiency. Transient analysis, which uses standard numeric integration, constructs a solution as a collection of time samples with an implied interpolating function. Typically the interpolating function is a low order polynomial. However, polynomials fit sinusoids poorly, and so many points are needed to approximate the sinusoidal solutions accurately.

The first difference between harmonic balance and transient analysis is that harmonic balance uses a linear combination of sinusoids to build the solution. Thus, it naturally approximates the periodic and almost-periodic signals found in a steady-state response. If the steady-state response consists of just a few dominant sinusoids, which is common, then harmonic balance needs only a small data set to represent the response accurately. The advantage of using sinusoids to approximate an almost-periodic steady-state response becomes particularly important when the response contains dominant sinusoids at widely separated frequencies.

Harmonic balance also differs from traditional time-domain methods in that time domain simulators represent waveforms as a collection of samples whereas harmonic balance represents them using the coefficients of the sinusoids. (Just as in traditional time-domain methods where it is presumed that a polynomial is used to interpolate between samples, we can use samples to represent the combination of sinusoids, with the understanding that a sum-of-sinusoids interpolation is to be done between samples.) Working with the coefficients and exploiting superposition makes it possible to calculate symbolically the response from linear dynamic operations such as time integration, differentiation, convolution, and delay. Because linear devices respond at the same frequency as the stimulus, it is only necessary to determine the magnitude and phase of the response. Using phasor analysis [desoer69], this is easily done for lumped components such as resistors, capacitors and inductors; while it is not trivial for the more esoteric distributed devices, it is generally much easier to find their response using phasor analysis than to try to determine their response to sampled waveforms in the time domain.

Determining the response of the nonlinear devices is more difficult. There is no known way to compute the coefficients of the response directly from the coefficients of the stimulus for an arbitrary nonlinearity, though it is possible if the nonlinearity is described by a polynomial or a power series [steer83]. We do not wish to restrict ourselves to these special cases, nor to accept the error of using them to approximate arbitrary nonlinearities. Instead, we convert the coefficient representation of the stimulus into a sampled data representation; this is a conversion from the frequency domain to the time domain and is accomplished with the inverse Fourier transform. With this representation the nonlinear devices are easily evaluated. The results are converted back into coefficient form using the forward Fourier transform.

Because the coefficients of the steady-state response are an algebraic function of the coefficients of the stimulus, the dynamic aspect of the problem is eliminated. Thus, the nonlinear integro-differential equations that describe a circuit are converted by harmonic balance into a system of algebraic nonlinear equations whose solution is the steady-state response of the circuit. These equations are solved iteratively using Newton's method.

3.1 Derivation

When applying harmonic balance to (5), both v and $f(v)$ are transformed into the frequency domain. Since v is almost periodic, both $i(v)$ and $q(v)$ are almost periodic; therefore all three waveforms can be written in terms of their Fourier coefficients: $\mathbf{F}v = V$, $\mathbf{F}i(v) = \mathbf{F}i(\mathbf{F}^{-1}V) = I(V)$ and $\mathbf{F}q(v) = \mathbf{F}q(\mathbf{F}^{-1}V) = Q(V)$. Since v, $i(v)$ and $q(v)$ are vectors of waveforms — one waveform for each node in the circuit — V, $I(V)$ and $Q(V)$ are vectors of spectra. The Fourier coefficients of the convolution integral are computed by exploiting its linearity. Assume y satisfies

$$\int_{-\infty}^{\infty} y(t)^T y(t)dt < \infty,$$

and $y(t) = 0$ for all $t < 0$; that is, assume y is causal and has finite energy (or equivalently, that the circuit with all nonlinear devices removed is causal and asymptotically stable); then

$$\mathbf{F}\int_{-\infty}^{t} y(t-\tau)v(\tau)d\tau = YV$$

where

$$Y = [Y_{mn}] \qquad m,n \in \{1,2,\cdots,N\}$$

$$Y_{mn} = [Y_{mn}(k,l)] \qquad k,l \in \mathbf{Z}$$

where m,n are the node indices; k,l are the frequency indices, and

$$Y_{mn}(k,l) = \begin{cases} \begin{bmatrix} \mathrm{Re}\{\mathrm{Y}_{mn}(\mathrm{j}\omega_k)\} & -\mathrm{Im}\{\mathrm{Y}_{mn}(\mathrm{j}\omega_k)\} \\ \mathrm{Im}\{\mathrm{Y}_{mn}(\mathrm{j}\omega_k)\} & \mathrm{Re}\{\mathrm{Y}_{mn}(\mathrm{j}\omega_k)\} \end{bmatrix} & \text{if } k = l \\ \mathbf{0} & \text{if } k \neq l \end{cases}$$

where Y is the Laplace transform of y [desoer69] and $\mathbf{j} = \sqrt{-1}$.

Now (5) can be rewritten in in the frequency domain as

$$F(V) = I(V) + \Omega Q(V) + YV + U = \mathbf{0} \qquad (6)$$

where $U = \mathbf{F}u$ contains the Fourier coefficients for the source currents over all nodes and harmonics, and

$$\Omega = [\Omega_{mn}] \qquad m,n \in \{1,2,\cdots,N\}$$

$$\Omega_{mn} = \begin{cases} [\Omega_{mn}(k,l)] & \text{if } m = n \\ \mathbf{0} & \text{if } m \neq n \end{cases}$$

$$\Omega_{mn}(k,l) = \begin{cases} \begin{bmatrix} 0 & -\omega_k \\ \omega_k & 0 \end{bmatrix} & \text{if } k = l \\ \mathbf{0} & \text{if } k \neq l \end{cases}$$

That $\mathbf{F}\dot{q}(v) = \Omega Q(V)$ follows from the differentiation rule of the Fourier series. Eqn. (6) is simply the restatement of Kirchoff's current law in the frequency domain.

It is important to realize that the frequency-domain functions for the nonlinear devices (I and Q) are evaluated by transforming the node voltage spectrum V into the time domain, calculating the response waveforms i and q, and then transforming these waveforms back into the frequency domain. To assure that the nonlinear device response waveforms are almost periodic, we require that the nonlinear devices be algebraic. If not (that is, if the device has memory), then the response waveform has a transient component, is not almost periodic, and cannot be accurately transformed into the frequency domain. The restriction that nonlinear devices be algebraic clearly allows nonlinear resistors. Fortunately, it also allows nonlinear capacitors and inductors (actually, any lumped nonlinear component) because their constitutive relations are algebraic when written in terms of the proper variables; v and q for capacitors, and i and ϕ for inductors [chua80]. The conversion between i and q ($i = \dot{q}$) and v and ϕ ($v = \dot{\phi}$) is done in the frequency domain where it is an algebraic operation and does not disturb the steady-state nature of the solution. Nonlinear distributed devices, however, are not algebraic, and the trick of evaluating their response in the time domain and transforming it into the frequency domain cannot be used. Instead, it is necessary to remain in the frequency domain and model the nonlinear device using a Volterra series representation. We will not consider nonlinear distributed devices further.

3.2 The Almost-Periodic Fourier Transform

To make the process of finding the solution to (6) computationally tractable, it is necessary to truncate the frequencies to a finite set. By considering only a finite number of frequencies, it is possible to sample a waveform at a finite number of time points and calculate its Fourier coefficients. Since the spaces involved are now finite dimensional, the first representation theorem of linear algebra shows that the Fourier transform $\mathbf{F}$ and its inverse $\mathbf{F}^{-1}$ can be viewed as matrices acting on the vectors of samples and coefficients, respectively. That is,

$$\sum_{\omega_k \in \Lambda_K} (X_k^C \cos\omega_k t + X_k^S \sin\omega_k t) = x(t)$$

can be sampled at S time points, resulting in the set of S equations and $2K - 1$ unknowns

$$\begin{bmatrix} 1 & \cos\omega_1 t_1 & \sin\omega_1 t_1 & \cdots & \cos\omega_{K-1} t_1 & \sin\omega_{K-1} t_1 \\ 1 & \cos\omega_1 t_2 & \sin\omega_1 t_2 & \cdots & \cos\omega_{K-1} t_2 & \sin\omega_{K-1} t_2 \\ 1 & \cos\omega_1 t_3 & \sin\omega_1 t_3 & \cdots & \cos\omega_{K-1} t_3 & \sin\omega_{K-1} t_3 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 1 & \cos\omega_1 t_S & \sin\omega_1 t_S & \cdots & \cos\omega_{K-1} t_S & \sin\omega_{K-1} t_S \end{bmatrix} \begin{bmatrix} X_0 \\ X_1^C \\ X_1^S \\ \vdots \\ X_{K-1}^C \\ X_{K-1}^S \end{bmatrix} = \begin{bmatrix} x(t_1) \\ x(t_2) \\ x(t_3) \\ \vdots \\ x(t_S) \end{bmatrix} \quad (7)$$

If the frequencies ω_k are distinct, and if $S = 2K - 1$, this system is invertible for almost all choices of time points, and can be compactly written as $\Gamma^{-1}X = x$. Inverting Γ^{-1} gives $\Gamma x = X$. Γ and Γ^{-1} are a discrete Fourier transform pair.

Given a finite set Λ_K of distinct frequencies ω_k, and a set of time points, we say that Γ and Γ^{-1} are one implementation of the almost-periodic Fourier transform for $AP(\Lambda_K)$. Once Γ and Γ^{-1} are known, performing either the forward (using Γ) or inverse (using Γ^{-1}) transform just requires a matrix multiply, or $(2K-1)^2$ operations; this is the same number of operations required by the discrete Fourier transform (DFT).

The DFT is a special case of (7) with $\omega_k = k\omega$ and $t_s = sT/S$, i.e. when the frequencies are all multiples of a single fundamental and the time points are chosen equally spaced within the period. The DFT and its inverse, the IDFT, have the desirable property of being well conditioned, which is to say that very little error is generated when transforming between x and X. From the matrix viewpoint, the high accuracy of the DFT corresponds to the fact that the rows of Γ^{-1} are orthogonal. Unfortunately, the DFT and the IDFT are defined only for periodic signals.

For almost-periodic signals, if the time points are not chosen carefully, Γ^{-1} can be very ill-conditioned. A particularly bad strategy for choosing time points when signals are not periodic seems to be that of making them equally spaced. Unlike the periodic case, it is in general impossible to choose a set of time points over which the sampled sinusoids at frequencies in Λ_K are orthogonal. In fact, it is common for evenly sampled sinusoids at two or more frequencies to be nearly linearly dependent, which causes the severe ill-conditioning problems encountered in practice. Developing an way to choose a set of time points that result in well conditioned transform matrices is beyond the scope of this paper, but an effective and practical algorithm is given in [kundert88c].

3.3 Harmonic Newton

As shown earlier, the circuit equation

$$f(v,t) = i(v(t)) + \dot{q}(v(t)) + \int_{-\infty}^{t} y(t-\tau)v(t)dt + u(t) = 0 \quad (8)$$

can be written in the frequency domain as

$$F(V) = I(V) + \Omega Q(V) + YV + U = 0. \quad (9)$$

To evaluate the nonlinear devices in (9) it is necessary to convert the node voltage spectrum V into the waveform v and evaluate the nonlinear devices in the time domain. The response is then converted back into the frequency domain. Now that we have developed the almost-periodic Fourier transform, it can be used with (9) to allow harmonic balance to be applied to almost-periodic systems. Assume that the number of frequencies has been truncated to K; $v, u \in AP^N(\Lambda_K)$; and that a set of time points $\{t_0, t_1, \cdots, t_{2K-1}\}$ has been chosen so that Γ^{-1} is nonsingular. Then $V_n = \Gamma v_n$, $I_n(V) = \Gamma i_n(v)$ and $Q_n(V) = \Gamma q_n(v)$.

Applying Newton-Raphson to solve (9) results in the iteration

$$J(V^{(j)})(V^{(j+1)} - V^{(j)}) = -F(V^{(j)}) \quad (10)$$

where

$$J(V) = \frac{\partial F}{\partial V} = \frac{\partial I(V)}{\partial V} + \Omega\frac{\partial Q(V)}{\partial V} + Y.$$

Or

$$J(V) = \left[J_{mn}(V)\right] = \left[\frac{\partial F_m(V)}{\partial V_n}\right] \qquad m,n \in \{1,2, \cdots ,N\}$$

where

$$\frac{\partial F_m(V)}{\partial V_n} = \frac{\partial I_m(V)}{\partial V_n} + \Omega_{mn}\frac{\partial Q_m(V)}{\partial V_n} + Y_{mn}.$$

The derivation of $\partial I_m/\partial V_n$ follows with help from the chain rule.

$$I_m(V) = \Gamma i_m(v)$$

$$\frac{\partial I_m(V)}{\partial V_n} = \Gamma\frac{\partial i_m(v)}{\partial v_n}\frac{\partial v_n}{\partial V_n}$$

Since $i(v)$ is algebraic, $\partial i_m/\partial v_n$ is a diagonal matrix. Using the fact that $\Gamma^{-1}V_n = v_n$,

$$\frac{\partial I_m(V)}{\partial V_n} = \Gamma\frac{\partial i_m(v)}{\partial v_n}\Gamma^{-1}$$

The derivation of $\partial Q_m/\partial V_n$ is identical. Now everything needed to evaluate (10) is available. If the sequence generated by (10) converges, its limit point is the desired solution to (9).

3.4 Acceleration of Harmonic Newton

Of the time spent performing harmonic Newton, most is spent constructing and factoring the Jacobian $J(V)$. There are two things that can be done to reduce this time. First is to employ Šamanskii's method [ortega70]; simply reuse the factored Jacobian from the previous iteration. This eliminates the construction and LU decomposition of the Jacobian, and so only the forward and backward substitution steps are needed. If the circuit is behaving nearly linear, then a Jacobian may be used many times. If, however, the Jacobian is varying appreciably at each step, then Šamanskii's method might take a bad step and slow or preclude convergence. To decide how many times to use an old Jacobian, $\|F(V)\|$ should be monitored, and a new Jacobian computed if the norm is not sufficiently reduced at each step.

The second way to improve the harmonic Newton algorithm is to exploit the sparsity of the Jacobian. The Jacobian is organized as a block node admittance matrix that is sparse. Conventional sparse matrix techniques can be used to exploit its sparsity [kundert86a]. Each block is a conversion matrix that is itself a block matrix, consisting of 2×2 blocks that result from Fourier coefficients being members of C. Conversion matrices are full if they are associated with a node that has a nonlinear device attached, otherwise they are diagonal. In an integrated circuit, nonlinear devices attach to most nodes, so the conversion matrices will in general be full. It often happens, though, that nonlinear devices are either not active or are behaving very linearly. For example, the base-collector junction of a bipolar transistor that is in the forward-active region is reverse biased, and so the junction contributes nothing to its conversion matrices. If there are no other contributions to those conversion matrices, they may be ignored. If there are only contributions from linear components, they are diagonal. During the decomposition, it is desirable to keep track of which conversion matrices are full, which are diagonal, and which are zero, and avoid unnecessary operations on known zero conversion matrix elements.

Experimentally, the computational complexity of the LU decomposition of the block Jacobian matrix is $O(N^{\alpha}K^3)$, where typically $1.1 < \alpha < 1.5$. The amount of memory required is $O(N^{\alpha}K^2)$. Clearly, the cost of harmonic Newton increases very rapidly as K grows. There are other algorithms, such as harmonic relaxation [kundert86b], that do not suffer from such a dramatic increase in resource needs, but these methods will have convergence problems with circuits that behave strongly nonlinear.

4 FINITE-DIFFERENCE METHODS

4.1 Derivation

Like most approaches to solving differential equations numerically, the finite-difference methods approximate the original system with a set of difference equations. Unlike transient analysis methods, however, finite-difference methods attempt to find the solution at every time point simultaneously. To find a T-periodic solution using a finite-difference method, a mesh $t_0 < t_1 < t_2 < \cdots < t_S$ is chosen where $t_0 = 0$ and $t_S = T$. A finite sequence $\{v_s\}$ is computed as an approximation to $v(t)$ on the mesh, where $v_s \approx v(t_s)$. The difference equations are formed by using a discrete time approximation to the time derivative and the convolution integral. For example, consider the slightly modified version of (5)

$$f(v,t) = i(v) + \dot{q}(v) + \int_{-\infty}^{t} y(t-\tau)v(\tau)d\tau + u(t) = 0. \qquad (11)$$

All symbols have their previous definitions except that i is a function representing the sum of the current entering the nodes from *all* conductors, q represents the sum of the charge entering the nodes from *all* capacitors; and y is the matrix-valued impulse response of the circuits with all *lumped* elements removed.

There are a large number of possible discrete approximations to $\dot{q}$ that can be used [gear71]. Implicit Euler [white86], the simplest approximation that is suitable for circuit simulation, employs linear interpolation between mesh points and is given by

$$\dot{q}_s = \frac{1}{h_s}(q_s - q_{s-1}) \qquad (12)$$

where $h_s = t_s - t_{s-1}$ is the time step. A discrete approximation to the convolution integral would be given by

$$w_s = \sum_{r=0}^{S} y_{sr} v_r \qquad (13)$$

where w_s represents the current entering the nodes from distributed devices at time t_s and y_{sr} is the discrete approximation to the impulse response y. Computation of y_{sr} is done by applying first an interpolation function to the solution between the mesh points and then employing an integration method such as Simpson's rule [stoer80]. The details are beyond the scope of this paper — more information can be found in [hall76].

Discretizing (11) using the approximations of (12) and (13) yields

$$i(v_s) + \frac{1}{h_s}(q(v_s) - q(v_{s-1})) + \sum_{r=0}^{S} y_{sr} v_r + u_s = 0 \qquad (14)$$

where $s = 1, \cdots, S$. As v is T-periodic, $v_0 = v_S$, which results in the following system of nonlinear algebraic equations

$$\begin{aligned}
i(v_1) + \frac{q(v_1) - q(v_S)}{h_1} + \sum_{r=0}^{S} y_{1r} v_r + u_1 &= 0 \\
i(v_2) + \frac{q(v_2) - q(v_1)}{h_2} + \sum_{r=0}^{S} y_{2r} v_r + u_2 &= 0 \\
\vdots \qquad\qquad \vdots \qquad\qquad \vdots \qquad & \quad \vdots \\
i(v_S) + \frac{q(v_S) - q(v_{S-1})}{h_S} + \sum_{r=0}^{S} y_{Sr} v_r + u_S &= 0.
\end{aligned}$$

The system is solved using the Newton-Raphson method.

Finite-difference methods can generate large systems of equations, especially if either the number of unknown waveforms or the number of time points is large. The systems are sparse, and hence, not overly expensive to solve. The sparsity is increased further if there are no distributed devices present in the circuit.

It is possible to reduce the time required for the finite-difference methods by choosing carefully the time steps to achieve a desired accuracy, clustering them in troublesome spots to reduce error where the basis functions accurately approximate the solution only over short intervals while spreading them out in quiescent areas to reduce computer resource usage. If the solution is expected to be smooth, it is possible to use high-order integration methods to achieve low truncation error using widely separated time points. Lastly, it is possible to use Šamanskii's method [ortega70] to reduce execution time further.

4.2 Time-Domain Harmonic Balance

Finite-difference methods are elegant and simple. They also offer great flexibility in the choice of the basis functions. For example, by starting from (9), harmonic balance is written as a finite-difference method by simply multiplying through by the inverse almost-periodic Fourier transform matrix Γ^{-1}. For node n,

$$\Gamma^{-1} I_n(V) + \Gamma^{-1}\Omega_{nn} Q_n(V) + \Gamma^{-1} \sum_{m=1}^{N} Y_{mn} V_n + \Gamma^{-1} U_n = 0.$$

Knowing that $V_n = \Gamma v_n$, $I_n(V) = \Gamma i_n(v)$, $Q_n(V) = \Gamma q_n(v)$ and $U_n = \Gamma u_n$ allows us to simplify this to

$$i_n(v) + \Gamma^{-1}\Omega_{nn}\Gamma q_n(v) + \Gamma^{-1} \sum_{m=1}^{N} Y_{mn} \Gamma v_n + u_n = 0. \qquad (15)$$

$\Gamma^{-1}\Omega_{nn}\Gamma$ and $\Gamma^{-1} Y_{mn}\Gamma$ are constants and so can be precomputed for efficiency. Eqn. (15) shows that harmonic balance can be formulated in the time domain as a finite-difference method. The basic difference between the two approaches is that the frequency-domain version represents the solution using the coefficients of the sinusoids and the finite-difference method represents them in sampled-data form. Though both methods give the same answer, the matrices in the finite-difference method are denser and so that approach will be less efficient.

In changing the basis functions from piecewise linear to sinusoidal, the character of the finite-difference method was completely changed. The rather conventional implicit Euler was replaced with a discrete differentiation operator ($\Gamma^{-1}\Omega_{nn}\Gamma$) that is exact for sinusoids and suitable for almost-periodic signals. However, using sinusoidal basis functions also results in the computed solutions being noncausal. This fact is normally hidden when the solution is smooth, but becomes apparent if the solution exhibits sharp transitions where is manifests itself as preshoot. This "feature" can be disconcerting to the uninitiated.

5 CONCLUSION

In this paper, we presented two methods for finding the steady-state response of a circuit, harmonic balance and the finite-difference method. We showed that the unique characteristics of harmonic balance result from the use of sinusoidal basis functions. These basis functions were installed into a finite-difference method, demonstrating that it is not the frequency domain that makes harmonic balance unique, but rather that it constructs a solution as a sum of sinusoids. It also shows the flexibility of the finite-difference methods. It is also possible to formulate a harmonic-balance-like method (an expansion method) based on implicit Euler (where the piecewise linear functions are specified in terms of the coefficients of the line segments rather than their end points), which establishes that finite-difference methods and harmonic balance are equivalent.

The determination of the steady-state response of a circuit is still an open research area, with many promising avenues to be explored. For example, more work is needed on finding the steady-state response of autonomous circuits such as oscillators. Also, an efficient way to determine whether a computed solution is stable is needed for all the methods presented.

There is still room for considerable innovation also in developing new discretization methods. It is possible to customize a discretization method to particular problem by designing suitable basis functions. The first simulator to use "designer" basis functions is *Harmonica* [kundert], which uses sinusoidal basis functions with the harmonic Newton algorithm. Developing unique basis functions for other individual problems, such as the recursive basis methods [kundert88b], is only now just beginning.

ACKNOWLEDGEMENTS

We would like to acknowledge and thank Greg Sorkin for his help developing and describing the almost-periodic Fourier transform, and Jacob White and Richard Newton for helpful discussions on both conventional and exotic discretization methods. This work was supported by Hewlett-Packard and the California MICRO program.

REFERENCES

[aprille72] Thomas J. Aprille and Timothy N. Trick. Steady-state analysis of nonlinear circuits with periodic inputs. *Proceedings of the IEEE*, vol. 60, no. 1, January 1972, pp. 108-114.

[chua80] L. O. Chua. Device modeling via basic nonlinear circuit elements. *IEEE Transactions on Circuits and Systems*, vol. CAS-27, no. 11, November 1980, pp. 1014-1044.

[desoer69] Charles A. Desoer and Ernest S. Kuh. *Basic Circuit Theory*. McGraw-Hill, 1969.

[gear71] C. William Gear. *Numerical Initial Value Problems in Ordinary Differential Equations*. Prentice-Hall, 1971.

[hale80] Jack K. Hale. *Ordinary Differential Equations*. Krieger, 1980.

[hall76] G. Hall and J. M. Watt (editors). *Modern Numerical Methods for Ordinary Differential Equations*. Oxford University Press, 1976.

[keller76] Herbert B. Keller. *Numerical Solution of Two Point Boundary Value Problems*. Society for Industrial and Applied Mathematics, 1976.

[kundert86a] Kenneth S. Kundert. Sparse matrix techniques. In *Circuit Analysis, Simulation and Design*, part 1, Albert E. Ruehli (editor). North-Holland, 1986.

[kundert86b] Kenneth S. Kundert and Alberto Sangiovanni-Vincentelli. Simulation of nonlinear circuits in the frequency domain. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. CAD-5, no. 4, October 1986, pp. 521-535.

[kundert88a] Kenneth S. Kundert, Alberto Sangiovanni-Vincentelli and Tsutomu Sugawara. Techniques for finding the periodic steady-state response of circuits. In *Analog Methods for Circuit Analysis and Diagnosis*, Takao Ozawa (editor). Marcel Dekker, 1988. To be published.

[kundert88b] Kenneth S. Kundert, Jacob K. White and Alberto Sangiovanni-Vincentelli. Efficient detailed simulation of switched-capacitor filters using a recursive-basis method. In *Proceedings of the 1988 IEEE Custom Integrated Circuits Conference*, May 1988. To be published.

[kundert88c] Kenneth S. Kundert, Gregory B. Sorkin and Alberto Sangiovanni-Vincentelli. Applying harmonic balance to almost-periodic circuits. *IEEE Transactions on Microwave Theory and Techniques*, vol. MTT-36, no. 2, February 1988.

[kundert] Kenneth S. Kundert. *Harmonica User's Guide: A Frequency Domain Simulator for Nonlinear Circuits*. The program will be available in mid 1988. For information write to the EECS Industrial Liaison Program, University of California, Berkeley California, 94720.

[ortega70] J. M. Ortega and W. C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables*. Academic Press, 1970.

[petzold81] Linda R. Petzold. An efficient numerical method for highly oscillatory ordinary differential equations. *SIAM Journal on Numerical Analysis*, vol. 18, no. 3, June 1981, pp. 455-479.

[sangiovanni81] Alberto L. Sangiovanni-Vincentelli. Circuit simulation. In *Computer Design Aids for VLSI Circuits*, P. Antognetti, D. O. Pederson and H. DeMan (editors). Sijthoff & Noordhoff, 1981, pp. 19-112.

[skelboe80] Stig Skelboe. Computation of the periodic steady-state response of nonlinear networks by extrapolation methods. *IEEE Transactions on Circuits and Systems*, vol. CAS-27, no. 3, March 1980, pp. 161-175.

[steer83] Michael B. Steer and Peter J. Khan. An algebraic formula for the output of a system with large-signal, multifrequency excitation. *Proceedings of the IEEE*, vol. 71, no. 1, January 1983, pp. 177-179.

[stoer80] J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis*. Springer-Verlag, 1980.

[white86] Jacob K. White and Alberto Sangiovanni-Vincentelli. *Relaxation Techniques for the Simulation of VLSI Circuits*. Kluwer, 1986.