Applying Harmonic Balance to Almost-Periodic Circuits

KENNETH S. KUNDERT, STUDENT MEMBER, IEEE, GREGORY B. SORKIN, STUDENT MEMBER, IEEE, AND ALBERTO SANGIOVANNI-VINCENTELLI, FELLOW, IEEE

Abstract—**Harmonic balance is a powerful technique for the simulation of nonlinear microwave circuits. It solves directly for the steady-state response of a circuit in the frequency domain, and so is often considerably more efficient than traditional time-domain methods when circuits exhibit widely separated time constants and mildly nonlinear behavior. With harmonic balance the linear component models are evaluated in the frequency domain, which for distributed devices results in easier model development and reduced computational complexity.**

Harmonic balance has had limited application for simulating circuits, such as mixers, that have a steady-state response that contains almost-periodic signals. The reason is that to model a nonlinear device, whose behavior is more conveniently computed in the time domain, harmonic balance requires the transformation of signals from the frequency domain into the time domain and vice versa. For circuits that have a periodic response, the discrete Fourier transform (DFT) is used. Previously, no satisfactory transform existed for almost-periodic signals. In this article, a new Fourier transform algorithm for almost-periodic functions (the APFT) is developed. It is both efficient and accurate. Unlike previous attempts to solve this problem, the new algorithm does not constrain the input frequencies and uses the theoretical minimum number of time points.

Also presented is a particularly simple derivation of harmonic Newton (the algorithm that results when Newton's method is applied to solve the harmonic balance equations) using the APFT; this derivation uses the same matrix representation used in the derivation of the APFT. Since the APFT includes the DFT as a special case, all results are applicable to both the periodic and almost-periodic forms of harmonic Newton. The simple derivation of harmonic Newton, combined with the rigorous definition of terms and the careful exploration of the error mechanisms of the APFT, makes this article a good base for future research.

NOMENCLATURE

$\mathbb{Z}, \mathbb{R}, \mathbb{C}$	The integer, real, and complex numbers.
$\boldsymbol{C} = \mathbb{R}^2$	Throughout this article, the trigonometric Fourier series is used rather than the exponential. Thus, a Fourier coefficient is described using the coefficients of sine and cosine. The pair of these two coefficients are said to reside in $\boldsymbol{C}$ as opposed to $\mathbb{C}$. $\boldsymbol{C}$ is related to $\mathbb{C}$ in that $[a, b]^T \in \boldsymbol{C}$ corresponds to $a + jb \in \mathbb{C}$.
$\|\cdot\|_\infty$	The l_∞ norm. For $x \in \mathbb{R}^N$, $\|x\|_\infty = \max_i \lvert x_i \rvert$. For $A \in \mathbb{R}^{N \times N}$, $\|A\|_\infty = \max_i \sum_{j=1}^{N} \lvert a_{ij} \rvert$.
$\|\cdot\|_2$	The Euclidean or l_2 norm. For $x \in \mathbb{R}^N$, $\|x\|_2 = (\sum_{i=1}^{N} x_i^2)^{1/2}$. For vectors in $\mathbb{R}^N$, the l_2 and l_∞ norms are equivalent. That is, $\frac{1}{\sqrt{N}}\|x\|_2 \leq \|x\|_\infty \leq \|x\|_2$ for all $x \in \mathbb{R}^N$.
j	Imaginary operator, $j = \sqrt{-1}$.
$\mathbf{0}, \mathbf{1}$	The zero vector or matrix and the identity matrix.
t, ω	Time, radial frequency.
λ	A fundamental frequency.
Λ, Λ_K	An at most countable set of frequencies, and a finite set with K elements.
$P(T)$	The space of all periodic waveforms of bounded variation with period T.
$AP(\Lambda)$	The space of almost-periodic functions constructed as a linear combination of sinusoids at frequencies in the set Λ.
$AP(\lambda_1, \cdots, \lambda_d)$	The set of quasi-periodic functions with fundamental frequencies $\lambda_1, \lambda_2, \cdots, \lambda_d$. Equals $AP(\Lambda)$ where Λ is the module constructed from the basis of fundamental frequencies.
$\mathcal{F}, \mathcal{F}^{-1}$	Abstract forward and inverse Fourier operators.
Γ, Γ^{-1}	Matrix representation of the forward and inverse Fourier operators.
x, X	Arbitrary waveform and its spectrum. $X = \mathcal{F}x$.
$\leftrightarrow$	Laplace transform relation.
f	Function that maps waveforms to waveforms. Sometime f is an arbitrary differentiable function; other times it is used to represent the sum of currents entering a node or nodes.
F	Function that maps spectra to spectra. Related to f in that if $y = f(x)$ then $Y = F(X)$.
H	The maximum number of harmonics considered.
K	The number of frequencies present in the spectra.
N	The number of nodes in a circuit.
S	The number of time points present in the sampled waveforms.
k, l	Frequency indices. Usually, $k, l \in \{0, 1, \cdots, K-1\}$.

Manuscript received April 21, 1987; revised September 3, 1987. This work was supported in part by the Hewlett-Packard Company and in part by a grant from the MICRO program of the state of California.

The authors are with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720. K. Kundert is also with Hewlett-Packard, Santa Rosa, CA 95404, and G. Sorkin is also with the IBM T. J. Watson Research Center, Yorktown Heights, NY 10598.

IEEE Log Number 8717979.

0018-9480/88/0200-0366$01.00

m, n	Node indices. $m, n \in \{1,2,\cdots,N\}$.
r, s	Time indices. $r, s \in \{1,2,\cdots,S\}$.
v, V	Node voltage waveforms, spectra.
u, U	Input current waveforms, spectra.
i, I	Function from voltage to current for nonlinear resistors, and its frequency-domain equivalent.
q, Q	Function from voltage to charge for nonlinear capacitors, and its frequency-domain equivalent.
y	Matrix-valued impulse response of the circuit with all nonlinear devices removed.
Υ	Laplace transform of y.
Y	Phasor equivalent to Υ.
Ω	Matrix used to multiply each particular frequency component in a vector of spectra by the correct ω_k to perform the frequency-domain equivalent of time differentiation.

I. Introduction

QUITE OFTEN analog and microwave circuits exhibit characteristics or behavior that make them difficult to simulate using traditional time-domain simulation techniques. In particular, it is expensive to find the steady state for a circuit that exhibits widely separated time constants because the differential equation solver must continue until any transient behavior has vanished.

Harmonic balance [1] differs from transient analysis 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.

Analog and microwave circuits have other characteristics that are troublesome to time-domain simulators. For example, they often contain distributed devices. All but the most idealized distributed device models are difficult to formulate in the time domain, requiring either a lumped approximation or the impulse response. Once the model is formulated, it is usually expensive to evaluate, either because the lumped approximation is of high order or because the impulse response must be convolved with the terminal voltage waveforms.

Another characteristic that is troublesome to time-domain simulators, and the one that provides the dominant theme of this article, is that many analog and microwave circuits, such as mixers, have inputs at two or more independent frequencies. These frequencies may be such that the ratio of the highest to the lowest frequency generated by the nonlinearities is large. For example, the down-conversion mixer in the HP8505 network analyzer [2] supports a maximum input frequency of 1.3 GHz with the local oscillator frequency always offset from the input by 100 kHz. The ratio of the input to the output frequency can be as high as 13 000 to 1. Furthermore, the output is fed directly into a high-Q low-pass filter that has a long settling time. To simulate this circuit in the time domain requires a sampling rate well over 1.3 GHz and a simulation interval of at least 100 μs—a minimum of 10^6 time points are needed. It is difficult to present meaningful results in the presence of such a large number of data, particularly with the vastly different time scales involved. Normally, this problem is avoided by converting the solution into the frequency domain, but the many unequally spaced time points generated by the simulator, along with the nonperiodic signals make this a difficult task.

Harmonic balance is a promising way to avoid these problems since it operates in the frequency domain. The computational complexity depends only on the size of the circuit and the number of frequencies being used, and not on the actual frequencies or the time constants present in the circuit. Furthermore, the solution is obtained in the frequency domain, so the troublesome conversion needed by a time-domain simulator to present the results is avoided. If it is desirable to view the results in the time domain, conversion from the frequency domain to the time domain is not difficult.

With harmonic balance, the linear device equations are evaluated in the frequency domain and the nonlinear device equations are evaluated in the time domain. When signals in the circuit are periodic, the discrete Fourier transform (DFT) provides the needed conversion between the two domains. To date however, there has been no satisfactory way to analyze nonlinear circuits such as mixers that have two or more input signals with arbitrary input frequency and power, and hence have signals that are nonperiodic. Signals in the steady-state response of mixers are made up of several sinusoids at possibly nonharmonically related frequencies, and so are *almost periodic* [3]. This article introduces an accurate and efficient algorithm, the *almost-periodic Fourier transform*, or *APFT*, for computing the forward and inverse Fourier transforms of almost-periodic functions. Unlike previous methods, the APFT does not constrain the input frequencies and uses the theoretical minimum number of time points.

Harmonic balance converts a system of nonlinear integrodifferential equations into a system of nonlinear algebraic equations whose solution is the coefficients of the sinusoids that make up the steady-state response. There are several ways available to solve the algebraic system [4]; the approach we choose is Newton's method. We refer to the combination of harmonic balance and Newton's method as the *harmonic Newton algorithm*. A new and concise statement of the harmonic Newton algorithm is given that is valid for both the periodic and almost-periodic cases.

Section II contains a brief summary of the notation and definitions used throughout the article and then formulates the problem to be solved. Harmonic balance is introduced in Section III as a way of converting a system of integro-

differential equations into a larger system of algebraic equations that is solved for the steady-state solution. Section IV introduces the APFT as a generalization of the DFT and discusses its error mechanisms. In Section V the new APFT algorithm is presented. Lastly, in Section VI, the harmonic Newton algorithm is derived using the APFT. Several methods are given to increase its efficiency, and the results of applying harmonic Newton to several circuits with almost-periodic steady-state responses are given.

II. Background

A. Overview of Harmonic Balance

When linear circuits are excited by a sinusoid, their steady-state response, if it exists, is sinusoidal and at the same frequency as the input. While nonlinear circuits are capable of a dazzling variety of wonderful and bizarre behavior, the circuits of interest to designers generally have a periodic steady-state response to a sinusoidal input; the period of the response is usually equal to that of the input, though occasionally it will be some rational multiple. Because the response is periodic, it is representable as a Fourier series, that is, as a linear combination of sinusoids whose periods evenly divide the period of the response. If the stimulus to the circuit contains two or more sinusoids that are not harmonically related, the circuit responds in steady state at the sum and difference frequencies of the input sinusoids and their harmonics; such a response is referred to as being almost periodic. Thus, for the circuits we are interested in, a stimulus constructed as a sum of sinusoids results in a steady-state response that is also a sum of sinusoids. The response contains an infinite number of sinusoids; usually all but a few are negligible.

Harmonic balance differs from traditional transient analysis in two fundamental ways. These differences allow harmonic balance to exploit the behavior described above for circuits in steady state 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 [5], 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 [6]. 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. The computation of these forward and inverse Fourier transforms when signals are almost periodic is the kernel of this paper.

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 integrodifferential 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.

B. Definitions

A *signal* is a function that maps either $\mathbb{R}$ (the reals) or $\mathbb{Z}$ (the integers) into $\mathbb{R}$ or $\boldsymbol{C}$ (the space of real pairs).[1] The

[1] Throughout this article, 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

domain and range of the map are physical quantities; the domain is typically time or frequency, and the range is typically voltage or current. 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 $\boldsymbol{C}$-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

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

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

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

A waveform is *almost periodic* if it can be uniformly approximated by the sum of at most a countable number of sinusoids [7]. 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

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

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

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

then Λ is a *module*[2] of dimension d and 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 a module, then $AP(\Lambda)$ is also denoted $AP(\lambda_1, \lambda_2, \cdots, \lambda_d)$. Waveforms belonging to such a set are referred to as *quasi-periodic*. 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 $X_k = [X_k^C, X_k^S]^T \in \boldsymbol{C}$ 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 $\mathcal{F}$, referred to as the *Fourier transform*, that maps x to X. It is a homeomorphism, which allows us to talk of x and X as two different representations of the same signal whenever $X = \mathcal{F}x$.

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 in *steady state* if all signals present in the system are almost periodic and it is in *periodic steady state* if all signals are periodic. 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.

C. 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 [8], all results presented in this article 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 Kirchhoff's current law, the circuit can be described by

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

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: $\mathbb{R}^N \to \mathbb{R}^N$ are differentiable functions representing, respectively, the sum of the currents 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

---

$\boldsymbol{C} = \mathbb{R}^2$ rather than $\mathbb{C}$. Hence, we are using $\boldsymbol{C}$ rather than $\mathbb{C}$ as the scalar field to construct the vector space for spectra. The correspondence between $\boldsymbol{C}$ and $\mathbb{C}$ is established by the invertible function ψ: $\mathbb{C} \to \boldsymbol{C}$ that maps $a + jb$ to $[a, b]^T$.

[2] In this module, the vectors are real numbers and the scalars are integers. It is a module because it is closed under vector addition and scalar multiplication.

impulse response of the circuit with the nonlinear devices removed.[3]

III. Harmonic Balance

A. 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; $\mathcal{F}v = V$, $\mathcal{F}i(v) = \mathcal{F}i(\mathcal{F}^{-1}V) = I(V)$, and $\mathcal{F}q(v) = \mathcal{F}q(\mathcal{F}^{-1}V) = Q(V)$. 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

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

where

$$Y = [Y_{mn}], \qquad m, n \in \{1,2,\cdots,N\}$$
$$Y_{mn} = [Y_{mn}(k,l)], \qquad k,l \in \mathbb{Z}$$

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

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

where Υ is the Laplace transform of y [5], and $j = \sqrt{-1}$.

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

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

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

$$\Omega = [\Omega_{mn}], \qquad m, n \in \{1,2,\cdots,N\}$$
$$\Omega_{mn} = \begin{cases} [\Omega_{mn}(k,l)] & \text{if } m = n \\ \mathbf{0} & \text{if } m \neq n \end{cases}$$
$$\Omega_{mn}(k,l) = \begin{cases} \begin{bmatrix} 0 & -\omega_k \\ \omega_k & 0 \end{bmatrix} & \text{if } k = l \\ \mathbf{0} & \text{if } k \neq l. \end{cases}$$

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

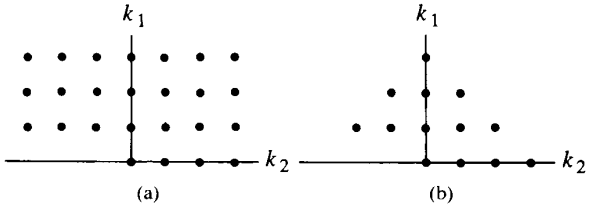


Fig. 1. Two different ways of truncating the set of frequencies to be finite.

That $\mathcal{F}\dot{q}(v) = \Omega Q(V)$ follows from the differentiation rule of the Fourier series. Equation (6) is simply the restatement of Kirchhoff'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 ensure 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 [9]. 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 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.

B. Truncation and Discretization

To make the process of finding the solution to (6) computationally tractable, it is necessary to truncate the frequencies to a finite set. When stimulating a circuit at d fundamental frequencies, the circuit responds in steady state at frequencies equal to sums and differences of the fundamental frequencies and their harmonics. Thus, the set of response frequencies is a module. We propose two ways of truncating this set to a finite number of frequencies.

The first approach limits consideration to the first H harmonics of the fundamental frequencies:

$$\Lambda_K(\lambda_1, \lambda_2, \cdots, \lambda_d) = \{\omega | \omega = k_1\lambda_1 + k_2\lambda_2 + \cdots + k_d\lambda_d;$$
$$k_j \in \mathbb{Z};\ |k_j| \leq H \text{ for } 1 \leq j \leq d;\ \text{first nonzero } k_j \text{ positive}\} \tag{7}$$

where $K = \frac{1}{2}((2H+1)^d + 1)$. The first nonzero k_j must be positive to eliminate frequencies from Λ_K that are negatives of each other. When there are two fundamentals ($d = 2$), this truncation results in a square grid of frequency indices as illustrated in Fig. 1(a).

$$\begin{bmatrix} 1 & \cos\omega_1 t_1 & \sin\omega_1 t_1 \cdots \cos\omega_{K-1} t_1 & \sin\omega_{K-1} t_1 \\ 1 & \cos\omega_1 t_2 & \sin\omega_1 t_2 \cdots \cos\omega_{K-1} t_2 & \sin\omega_{K-1} t_2 \\ 1 & \cos\omega_1 t_3 & \sin\omega_1 t_3 \cdots \cos\omega_{K-1} t_3 & \sin\omega_{K-1} t_3 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & \cos\omega_1 t_S & \sin\omega_1 t_S \cdots \cos\omega_{K-1} t_S & \sin\omega_{K-1} t_S \end{bmatrix} \cdot \begin{bmatrix} X_0 \\ X_1^C \\ X_1^S \\ \vdots \\ X_{K-1}^C \\ X_{K-1}^S \end{bmatrix} = \begin{bmatrix} x(t_1) \\ x(t_2) \\ x(t_3) \\ \vdots \\ x(t_S) \end{bmatrix} \tag{9}$$

The second method of truncation limits the absolute sum of the indices k_j to be less than or equal to H:

$$\Lambda_K(\lambda_1, \lambda_2, \cdots, \lambda_d) = \left\{ \omega | \omega = k_1\lambda_1 + k_2\lambda_2 + \cdots + k_d\lambda_d; \; k_j \in \mathbb{Z}; \; \sum_{j=1}^{d} |k_j| \leq H; \text{ first nonzero } k_j \text{ must be positive} \right\} \tag{8}$$

where $K \approx 2^{d-1}H^d/d!$. For $d = 2$, $K = H^2 + H + 1$. When there are two fundamentals, this truncation produces a diamond grid as shown in Fig. 1(b). Other truncation schemes are certainly possible. The truncation scheme directly affects the efficiency and accuracy of the simulation, and should be chosen to fit the particular problem being solved.

Now that only a finite set of frequencies Λ_K is being used, the requirement that the fundamental frequencies be linearly independent over the rationals may be relaxed as long as each $\omega_k \in \Lambda_K$ still corresponds uniquely to a valid sequence of harmonic indices $\{k_j\}$.

Once Λ has been truncated to some finite subset Λ_K, it is possible to represent the waveforms as sequences of finite length. If we assume that $\omega_0 = 0 \in \Lambda_K$, then the number of samples of each waveform must be $S = 2K - 1$ to uniquely fix the Fourier coefficients. This done, the Fourier transform becomes a finite-dimensional operator that depends both on Λ_K and on the S time points used to sample the waveform. Once the fundamental frequencies and the truncation scheme are specified, Λ_K is fixed, but we are free to choose the time points as we see fit with the one constraint that $\mathcal{F}$ be invertible.

IV. Almost-Periodic Fourier Transform

A. Matrix Formulation

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 $\mathcal{F}$ and its inverse $\mathcal{F}^{-1}$ can be viewed as matrices acting on the vectors of samples and coefficients, respectively. That is,

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

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

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 DFT.

The DFT is a special case of (9) 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. (We will say more about this later.) Unfortunately, the DFT and 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. One contribution of this article is the development of an algorithm for choosing time points that gives a well-conditioned system. We will briefly present previous work and then present our APFT algorithm.

B. Previous Work

Ushida and Chua [10] use equally spaced time points, but avoid the ill-conditioning problem by using extra time

points. In doing so, the matrix Γ^{-1} becomes a tall rectangular matrix. To make the system square again, both sides of (9) are multiplied by $(\Gamma^{-1})^T$, which results in

$$(\Gamma^{-1})^T\Gamma^{-1}X = (\Gamma^{-1})^T x.$$

Thus (9) is converted into a least squares problem that is solved in the traditional manner using the normal equation. Unfortunately, the normal equation is notoriously ill-conditioned and so a new ill-conditioning problem may be introduced.

Gilmore [1] samples the waveform using several small sets of equally spaced time points. The DFT is applied to each set individually. The sets are too small to prevent aliasing in the computed spectra. The aliasing is eliminated by taking an appropriate linear combination of the computed spectra. Since the DFT is used, the method is constrained to periodic signals, though it can be much more efficient than the standard DFT on sparse spectra. The total number of time points used is normally greater than the theoretical minimum by about 50 percent. The numerical stability of this approach is unknown.

C. Condition Number and Orthonormality

It is now necessary to discuss the conditioning of a system of equations, a concept alluded to earlier. Formally, the condition number of a matrix A is defined as $\kappa(A) = \|A\|\,\|A^{-1}\|$ [12]. The condition number of a matrix is important because it is a measure of how much errors can be amplified during the course of solving a matrix equation. For example, consider solving $Ax = b$ for x when both A and b are contaminated with error. Write the contaminated system as

$$(A + \delta A)(x + \delta x) = b + \delta b.$$

If $\|\delta A\|$ and $\|\delta b\|$ are small, then $\|\delta x\|$ can be bounded [12] with

$$\frac{\|\delta x\|}{\|x\|} \leqslant \kappa(A)\left(\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|b\|}\right) + \text{higher order terms}.$$

The problem of ill-conditioning in (9) can be visualized by considering each equation as defining a hyperplane in the Euclidean space $\mathbb{R}^{2K-1}$. Let $\rho_s \in \mathbb{R}^{2K-1}$ be such that ρ_s^T is the sth row in Γ^{-1}; then the sth hyperplane is defined as the set of all points X such that $\rho_s^T X = x(t_s)$. Thus, ρ_s is a vector orthogonal to the hyperplane. The solution to (9) is the intersection of all the hyperplanes. If the system is degenerate because two or more planes are coincident, then the intersection is not a single point and the system of equations has an infinite number of solutions. If there are no coincident hyperplanes, but two or more of the planes are nearly parallel, then a unique solution exists; however, high-precision arithmetic is needed to find it accurately.

A matrix is degenerate if and only if there is a linear dependence among its row vectors, and it is natural to suppose that a matrix has a small (good) condition number if its rows are nearly orthonormal (and thus "far" from being linearly dependent). We now prove this to be true.

Consider an invertible $N \times N$ matrix A. Suppose that the rows a_n of A, regarded as vectors, are nearly orthonormal. In particular, suppose that each vector has unit Euclidean length and that the orthogonal component of each vector a_n with respect to the space S_n spanned by the others is at least $\alpha \leqslant 1$ (it would be exactly 1 if the vectors were precisely orthonormal).

When forming the product $A^{-1}A = \mathbf{1}$, each row of A^{-1} can be thought of as the coefficients of a linear combination of the rows of A. This linear combination yields a row in the identity matrix—a vector of length 1. Suppose that the nth element in a row of A^{-1} has absolute value $r > 1/\alpha$. Then the component of the resulting linear combination that is in the direction orthogonal to S_n is determined solely by ra_n, and will have magnitude greater than $r\alpha > 1$. Since the linear combination is a vector of unit length, this is a contradiction. Therefore, no element of any row of A^{-1}, and thus no element of A^{-1}, has absolute value greater than $1/\alpha$.

Since $A \in \mathbb{R}^{N\times N}$, it follows that $\|A^{-1}\|_\infty$ (the l_∞ norm of A^{-1}) is no more than N/α. And since, by assumption, the Euclidean norm of the rows of A equals one, $\|A\|_\infty \leqslant N$ (employing the equivalence of the l_2 and l_∞ norms in $\mathbb{R}^N$), and therefore, $\kappa(A) \leqslant N^2/\alpha$. In short, the near orthonormality of a matrix places an upper bound on its condition number.

Note that multiplying a matrix by a scalar β does not affect its condition number, since the norms of the matrix and its inverse are multiplied by, respectively, β and $1/\beta$. Thus, if all rows of a matrix have equal Euclidean length (not necessarily one) and, when scaled to one, satisfy the orthonormality property, the matrix is still well conditioned. If the rows of a matrix are nearly orthonormal after they have been scaled to unit length, we say that they are (or the original matrix is) nearly orthogonal.

D. Condition Number and Time Point Selection

Given a finite set of frequencies Λ_K, any set of $S = 2K - 1$ time points yields a Γ^{-1} whose row vectors (consisting of a single 1 and a set of sine–cosine pairs) have Euclidean norms $\sqrt{K}$. Thus, if we could find a set of time points so that these rows were nearly orthogonal, it would follow from the discussion above that Γ^{-1}, and therefore Γ, would be well conditioned.

However the relation between the time points and the orthogonality of the resultant row vectors is clearly rather involved; finding a set of times which define nearly orthogonal row vectors seems to be quite difficult. One approach is to write down *a priori* a set of orthogonal vectors and then look for time points that generate vectors close to these prespecified ones; this is equivalent to defining the approximate phases of each sine wave and looking for a time where every wave is in the appropriate phase. This in turn can be thought of as a set of approximate equalities modulo 2π, but it is far from clear under what circumstances a solution exists or how to go about finding it.

Another approach is to choose time points equally spaced within a time interval larger than the period corresponding to the smallest nonzero frequency in Λ_K. As we discuss later, however, experience shows that this method of time-point selection gives the worst results of any method we tried.

E. Condition Number and Truncation Error

As mentioned previously, the condition number provides a measure of how much the error is amplified during a calculation. Roundoff is one source of error in the transform, but there is another that is normally much larger—the error due to truncating Λ to Λ_K (this error is referred to as aliasing when using the DFT). The Fourier coefficients of the frequencies omitted from Λ are presumably small but may not be exactly zero, and thus these frequencies contribute to the vector x of samples; this contribution is unaccounted for in the calculation of Γ^{-1} and Γ. Because of this, the computation of X will be in error.

Fortunately, this error can be bounded. Suppose that the overlooked sinusoids contribute an error δx to the observed sample vector $x + \delta x$. From this we calculate the Fourier coefficients $X + \delta X$ using

$$X + \delta X = \Gamma(x + \delta x).$$

By construction we know that $X = \Gamma x$. Thus, $\delta X = \Gamma \delta x$, and $\|\delta X\| \leq \|\Gamma\| \|\delta x\|$. By definition, $\kappa = \|\Gamma\| \|\Gamma^{-1}\|$. It is easily shown that $K \leq \|\Gamma^{-1}\|_\infty < \sqrt{2}K$, so $\|\Gamma\|_\infty \leq \kappa_\infty / K$ and

$$\|\delta X\|_\infty \leq \frac{\kappa_\infty}{K} \|\delta x\|_\infty.$$

That is, κ_∞ / K is the upper bound on how much the error due to coefficients of truncated frequencies is amplified in the process of transforming a waveform to the frequency domain. In practice, error amplification factors often approach this bound, so it is very important to select a set of time points such that κ is small.

V. The APFT Algorithm

A. Time Point Selection

Our time point selection algorithm, referred to as *near-orthogonal selection*, was conceived using some of the ideas discussed above.

First, we thought that if selecting evenly spaced time points was likely to yield row vectors particularly close to being linearly dependent, we might be better off selecting time points randomly from a time interval larger than the period corresponding to the smallest nonzero frequency in Λ_K. (We chose an interval equal to three times this period.) Such a choice is particularly attractive given the complexity of the relationship between the time points and the orthogonality of the row vectors; making any more intelligent choice of time points seems quite difficult.

Second, we realized that in essence the problem in recovering X from x is that the linear system may be close to being underdetermined, in a numerical sense. So adding additional equations should increase the accuracy of the calculation of X. In fact, if more than S time points are chosen, Γ^{-1} becomes a tall rectangular matrix, and its pseudoinverse Γ is a wide rectangular matrix satisfying $X = \Gamma x$.

Oversampling with twice as many randomly selected time points as theoretically necessary proves to be successful: it yields a very well conditioned system. However, when using the transform in the context of harmonic balance, all the nonlinear devices must be evaluated at each time point. This is an expensive operation because of the complexity of the nonlinear device models. Thus, oversampling is a costly remedy. It is clear, however, that the rows of the tall Γ^{-1} matrix span the space well (in a numerical sense). Perhaps some carefully chosen subset of these rows might also suffice.

The near-orthogonal selection algorithm takes just this approach; from a Γ^{-1} whose dimension is M rows by S columns, where $S = 2K - 1$ and $M > S$, it selects a set of just S rows, thus requiring no extra time samples. In other words, from a pool of more row candidates then necessary (we chose $M = 2S$, which seems to give good results in practice) and their corresponding time points, a "good" minimal set is selected during the initialization of the algorithm. When actually performing the transform, only the minimal set of time points is used. With harmonic balance, all nonlinear devices are evaluated at each time point. That only the minimum number of time points is used, and not 1.5 to 2 times the minimum as required by the other methods, is one of the significant advantages of the APFT algorithm.

The near-orthogonal selection algorithm is a variation of the Gram–Schmidt orthogonalization procedure [13]. Its input is the matrix formed by randomly choosing twice as many time points as necessary and forming the corresponding row vectors, ρ_s. Initially, these vectors all have the same Euclidean length (i.e., l_2 norm). One of these vectors, say ρ_1, is chosen arbitrarily. Any component in the direction of ρ_1 is removed from the remaining vectors using

$$\rho_s \leftarrow \rho_s - \frac{\rho_1^T \rho_s}{\rho_1^T \rho_1} \rho_1, \qquad s = 2, \cdots, M. \tag{10}$$

The vectors that remain are now orthogonal to ρ_1. Since the vectors initially had the same length, the largest remaining vector was originally most orthogonal to ρ_1. It is chosen to play the role of ρ_1 for the next iteration of the algorithm. This process repeats until the required S vectors have been chosen. The time points that correspond to these vectors are the time points used to form Γ^{-1}. This algorithm is detailed below.

APFT Near-Orthogonal Selection Algorithm

Given:

$\Lambda_K = \{0, \omega_1, \omega_2, \cdots, \omega_{K-1}\}$, the set of frequencies.

Task:

To find a set of $S = 2K - 1$ time points that results in a well-conditioned Γ^{-1}.

Algorithm:

$\omega_{\min} \leftarrow \min(\{|\omega_k|: 1 \leqslant k < K\})$
for $(s \leftarrow 1, \cdots, M)$
{ **random** () *returns numbers uniformly distributed between* 0 *and* 1.
$t_s \leftarrow \dfrac{6\pi}{\omega_{\min}}$ **random** ()
$\rho_s^{(1)} \leftarrow$
$[1, \cos\omega_1 t_s, \sin\omega_1 t_s, \cdots, \cos\omega_{K-1} t_s, \sin\omega_{K-1} t_s]^T$
}
for $(r \leftarrow 1, \cdots, S)$
{ **argmax** () *returns the index of the largest member of a set.*
$k = \textbf{argmax}(\{\|\rho_s^{(r)}\|: r \leqslant s \leqslant M\})$
swap $(\rho_r^{(1)}, \rho_k^{(1)})$
swap$(\rho_r^{(r)}, \rho_k^{(r)})$
swap(t_r, t_k)
for $(s \leftarrow r+1, \cdots, M)$
$$\rho_s^{(r+1)} \leftarrow \rho_s^{(r)} - \frac{\rho_r^{(r)T}\rho_s^{(r)}}{\rho_r^{(r)T}\rho_r^{(r)}}\rho_r^{(r)}$$
}

Results:

The set $\{t_s: 1 \leqslant s \leqslant S\}$ contains the desired time points.

Once the time points are selected, Γ^{-1} is constructed with the rows $\rho_s^{(1)}$ for $s = 1, \cdots, S$. It is easy to verify that the time points are well chosen either by calculating the condition number $\kappa = \|\Gamma\|\,\|\Gamma^{-1}\|$ or by computing the numerical error $\epsilon = \|\Gamma^{-1}\Gamma - \mathbf{1}\|$; both are excellent measures of the numerical stability of the transform.

B. Constructing the Transform Matrix

There is another problem that up to now we have ignored. The arguments to the sine and cosine functions in (9) are potentially very large, which results in excessive roundoff error. For example, assume $\lambda_1 = 2\pi 10^9$ and $\lambda_2 = 2\pi(10^9 + \sqrt{2})$. Then $\omega_{\min} = 2\pi\sqrt{2}$ and so the time points fall between 0 and $3/\sqrt{2}$ seconds. Thus, $\omega_l t_s$ can be as large as 10^{11}, causing two problems. First, on most computer systems, the trigonometry routines are not designed to handle such large arguments and often return meaningless results. This problem is easily avoided by subtracting from the argument as many multiples of 2π as possible without making it negative. The second problem is more troublesome. The approximately 10^{10} multiples of 2π in the argument have no effect on the result except to reduce its accuracy by about 10 digits. Since the $\omega_l t_s$ product must be formed (and so truncated to a finite number of digits by the computer) before the multiples of 2π can be removed, the digits are lost and cannot be reclaimed. While this error cannot be eliminated, it can be controlled by assuming Λ_K is a truncated module (note that up to now we have placed no restrictions on the frequencies in Λ_K except that they be distinct and that $\omega_0 = 0$). From (4), the product $\omega_l t_s$ can be written

$$\omega_l t_s = \sum_{j=1}^{d} k_j \lambda_j t_s.$$

Let

$$\psi_{js} = \textbf{fract}\left(\frac{\lambda_j t_s}{2\pi}\right), \qquad 1 \leqslant j \leqslant d;\ 1 \leqslant s \leqslant S \tag{11}$$

and

$$\phi_{ls} = 2\pi \sum_{j=1}^{d} k_j \psi_{js}. \tag{12}$$

Now $\phi_{ls} = \omega_l t_s - 2\pi m$, where m is some integer and $|\phi_{ls}| \leqslant 2\pi\sum_{j=1}^{d}|k_j|$. Since the k_j are small integers, ϕ_{ls} is an appropriate argument to trigonometry routines on all computers. Because the product $t_s\lambda_j/2\pi$ is formed before the **fract** operator (which removes any integer portion and leaves only the fractional part) is applied, it is the dominant source of roundoff error. By using (11) and (12), the roundoff error can be viewed as resulting from roundoff error in the λ_j and t_s. Since the t_s are chosen randomly, their roundoff errors are of no concern.

C. APFT Algorithm Results

The APFT near-orthogonal selection algorithm requires on the order of M^2S operations, where M is the number of time point candidates used, and $S = 2K - 1$, where K is the number of Fourier coefficients. Since we have used $M = 2S$, the asymptotic complexity of the algorithm is the same as that of the matrix inversion needed to compute Γ.

We note that while the initialization of the APFT (that is, the time point selection, the formation of Γ^{-1}, and the inversion of Γ^{-1} to find Γ) requires on the order of S^3 operations, the actual forward and inverse transform requires S^2 operations, the same as the DFT. Thus, the expensive part of the APFT is performed only once per set of frequencies; after this initial overhead has been paid, the APFT is as efficient as the DFT.

To show the numerical stability of our method, we compare the condition number of Γ^{-1} when time points are 1) evenly spaced, 2) randomly spaced, and 3) determined by the near-orthogonal selection algorithm. The condition number κ is roughly proportional to the errors in computing the inverse. On our computer, $\epsilon \approx 10^{-16}\kappa$. Bear in mind that even the DFT, which is theoretically the best conditioned algorithm for the simpler periodic case, has a condition number $\kappa \approx N$, so the best we can hope for is linear growth of the condition number with the number of Fourier coefficients. Observe that, as shown by the results given in Fig. 2, the condition number from near-orthogonal selection is experimentally observed to grow linearly with K. That of random selection appears to grow quadratically, and that of evenly spaced grows exponentially.

The example chosen for comparison was with two fundamentals $\lambda_1 = 2\pi 10^9$ and $\lambda_2 = 2\pi(10^9 + \sqrt{2})$. Thus, the fundamentals differ by only 1 part in 10^9; also, because the fundamentals are incommensurable, the signal is not periodic. Truncation was performed using (8). Comparisons of the condition numbers are shown in Fig. 2 with the order H varying between 1 and 10. To smooth the wide variation seen in the results for the case of randomly

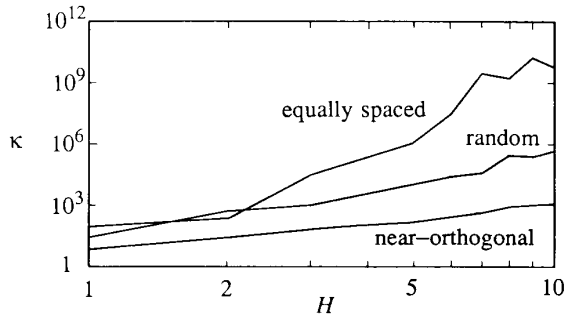


Fig. 2. Condition number of Γ^{-1} versus order H for the two fundamental APFT with truncation performed using (8) and time points chosen evenly spaced, randomly, and using the near-orthogonal selection algorithm.

TABLE I
ERROR ESTIMATES AND EXECUTION TIMES FOR THE APFT ALGORITHM USING DOUBLE PRECISION ARITHMETIC ON A VAX 8650

APFT Summary $\lambda_1 = 10^9$, $\lambda_2 = 10^9 + \sqrt{2}$, truncation performed using (8)						
H	K	S	κ	ϵ	t_{init}	$t_{transform}$
1	3	5	6	2.8×10^{-17}	17 ms	0
2	7	13	24	8.3×10^{-17}	67 ms	0.3 ms
3	13	25	64	1.1×10^{-16}	280 ms	1.7 ms
4	21	41	113	1.6×10^{-16}	1.1 s	3.6 ms
5	31	61	143	1.1×10^{-16}	3.3 s	8.5 ms
6	43	85	270	2.3×10^{-16}	8.6 s	17 ms
7	57	113	420	2.9×10^{-16}	20 s	30 ms
8	73	145	790	3.3×10^{-16}	41 s	49 ms
9	91	181	950	4.8×10^{-16}	79 s	77 ms
10	111	221	1200	4.6×10^{-16}	142 s	116 ms

H is the number of harmonics of each fundamental. K is the total number of frequencies, and S is the number of time samples. κ is the condition number of Γ^{-1} and $\epsilon = \|\Gamma^{-1}\Gamma - 1\|$. t_{init} is the time required to choose the time points and form and invert Γ^{-1}. $t_{transform}$ is the time required to multiply either Γ^{-1} or Γ by a vector.

selected time points, each condition number plotted is the geometric mean of 10 trials. Similarly, because different intervals give widely varying results for evenly spaced points, those condition numbers are geometrically averaged over 10 intervals ranging from 1.5 to 4.5 times $2\pi/\omega_{\min}$. Results obtained from near-orthogonal selection are so consistent that no averaging was needed, as evidenced by the smoothness of that curve. Graphing the condition number clearly shows that both randomly chosen and equally spaced samples have accuracy problems when the number of frequencies is large. Near-orthogonal selection from $2S$ randomly selected time points always results in a reasonable condition number. Table I gives a summary of information on the APFT with the near-orthogonal selection algorithm. Execution times were measured using the C programming language on a VAX 8650 running ULTRIX 2.0.

Recall that coefficients of frequencies not in Λ_K can be amplified by up to κ/K. For order $H = 10$, this amplification factor equals approximately 10^8 for evenly spaced points, 2000 for randomly spaced points, and 10 for points chosen using near-orthogonal selection. Thus, even if the coefficients of neglected frequencies are small, for evenly and randomly spaced points, the error δX due to truncation may be so large as to dominate over the desired coefficients X.

VI. HARMONIC NEWTON

A. Derivation

As shown earlier, the circuit equation

$$f(v,t) = i(v(t)) + \dot{q}(v(t)) + \int_{-\infty}^{t} y(t-\tau)v(\tau)\,d\tau + u(t) = \mathbf{0} \quad (13)$$

can be written in the frequency domain as

$$F(V) = I(V) + \Omega Q(V) + YV + U = \mathbf{0}. \quad (14)$$

To evaluate the nonlinear devices in (14) 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 APFT, it can be used with (14) to allow harmonic balance to be applied to almost-periodic systems. Assume that $v, u \in AP^N(\Lambda_K)$ 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 (14) results in the iteration

$$J(V^{(j)})(V^{(j+1)} - V^{(j)}) = -F(V^{(j)}) \quad (15)$$

where

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

Or

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

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:

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

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 (15) is available. If the sequence generated by (15) converges, its limit point is the desired solution to (14).

B. 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 Samanskii's method [14]; 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 Samanskii'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 [15]. 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 $\boldsymbol{C}$. 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 $1.1 < \alpha < 1.5$ and K increases as $O(H^d)$, and so the computational complexity of the harmonic Newton algorithm is $O(N^{\alpha}H^{3d})$. The amount of memory required is $O(N^{\alpha}H^{2d})$. Clearly, the cost of harmonic Newton increases very rapidly as either H, the number of harmonics considered, or d, the number of fundamentals, grows. There are other algorithms, such as harmonic relaxation [4], that do not suffer from such a dramatic increase in resource needs, but these methods will have convergence problems with circuits that behave in a strongly nonlinear way.

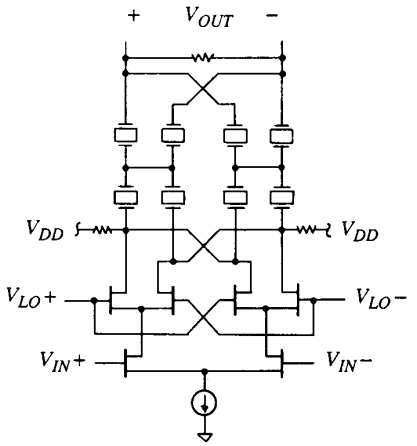


Fig. 3. GaAs double balanced mixer.

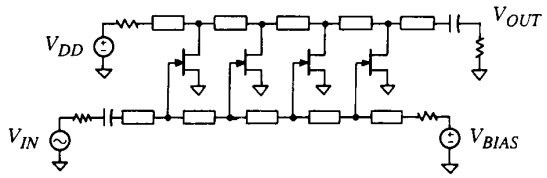


Fig. 4. GaAs traveling wave amplifier.

C. Harmonic Newton Results

The APFT algorithm has been integrated into *Harmonica*,[4] our harmonic balance circuit simulator. *Harmonica* was then used to simulate two GaAs FET [16] circuits. The first is the double-balanced mixer shown in Fig. 3. It is driven with a 50 mV, 5 GHz RF input signal and a 500 mV, 5.001 GHz LO input signal. The output is at 1 MHz and passes through a high-Q 1 MHz bandpass lattice filter. The circuit consists of six GaAs FET's and 27 nodes and was simulated with order $H = 5$, which corresponds to 31 frequencies. *Harmonica* required 4.5 megabytes of physical memory, 7.9 megabytes of virtual memory, and 230 seconds on a VAX 8650 to complete the simulation. The circuit, with the center frequency of the output filter adjusted accordingly, was also simulated with the LO frequency set as close as 1 Hz away from the 5 GHz RF with no apparent change in accuracy. Note that the combination of the widely separated frequencies and the high-Q output filter make it prohibitively expensive to find the steady-state response of this circuit with a time-domain simulator.

The second circuit is the GaAs FET traveling wave amplifier shown in Fig. 4 [17], which is being tested for intermodulation distortion. This circuit is driven by a two-tone input signal; one tone was 200 mV at 10 GHz and the other was 200 mV at 10.4 GHz. The response is shown, both in the time and the frequency domain, in Fig.

[4] *Harmonica*, which is a general-purpose circuit simulator, is expected to be released into the public domain in source code form in early 1988. It should not be confused with a program of the same name being advertised by Compact Software.

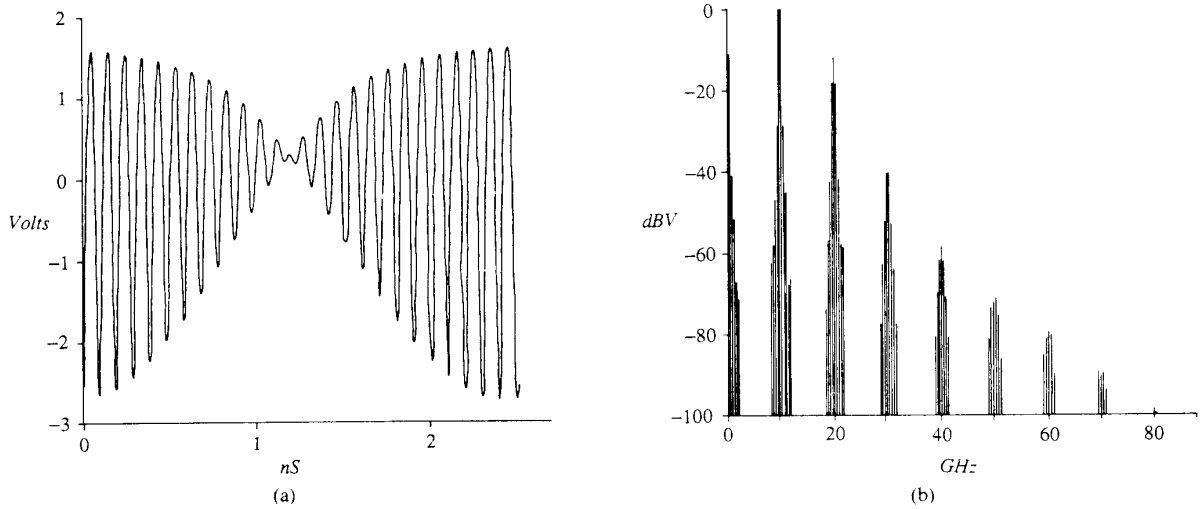


Fig. 5. GaAs traveling wave amplifier response to two-tone input.

TABLE II
EXECUTION TIMES AND MEMORY REQUIREMENTS FOR *Harmonica* RUNNING AN INTERMODULATION DISTORTION TEST ON THE GaAs TRAVELING WAVE AMPLIFIER SHOWN IN FIG. 4

GaAs Traveling Wave Amplifier				
H	K	time	physical memory	virtual memory
Using Λ_K generated by (7).				
1	5	0.63 s	0.55 MB	0.78 MB
2	13	4.2 s	0.87 MB	1.5 MB
3	25	24 s	2.2 MB	3.9 MB
4	41	98 s	7.5 MB	14 MB
5	61	320 s	7.6 MB	14 MB
Using Λ_K generated by (8).				
1	3	0.35 s	0.50 MB	0.80 MB
2	7	1.3 s	0.50 MB	0.80 MB
3	13	4.4 s	0.87 MB	1.5 MB
4	21	15.6 s	2.2 MB	3.9 MB
5	31	43 s	2.3 MB	4.0 MB
6	43	110 s	7.5 MB	14 MB
7	57	245 s	7.6 MB	14 MB

H is the number of harmonics of each fundamental and K is the total number of frequencies.

5. The circuit was simulated with $H = 5$ using the truncation scheme given in (7). The computation time and memory requirements for several values of H and for both truncation schemes are shown in Table II. There are a few comments that must be made to clarify some of the results in the table. The memory allocator expands array sizes in factors of two, which is why memory requirements sometimes do not change even though H changes. Each doubling of the array size quadruples the amount of memory required. Most of the approximate factor of two differences between physical and virtual memory requirements can be eliminated by better implementation. Any simulation that needed over 64 frequencies required more memory than the 44 megabytes available from the operating system.

VII. CONCLUSIONS

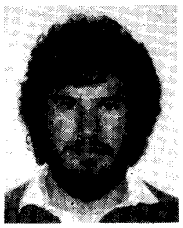
A new almost-periodic Fourier transform that is both efficient and accurate was presented. This transform was combined with harmonic balance to allow circuits with widely separated frequencies to be accurately simulated. Work is continuing on the APFT and its application in harmonic balance to better understand the algorithm, to further increase its efficiency, and to explore its error mechanisms.

REFERENCES

[1] K. S. Kundert, A. Sangiovanni-Vincentelli and T. Sugawara, "Techniques for finding the periodic steady-state response of circuits," in *Analog Methods for Circuit Analysis and Diagnosis*, T. Ozawa, Ed. New York: Marcel Dekker, to be published.
[2] R. Dalichow and D. Harkins, "A precision RF source and down-converter for the model 8505a," *Hewlett-Packard J.*, July 1976.
[3] H. Bohr, *Almost Periodic Funtions*. New York: Chelsea, 1947.
[4] K. S. Kundert and A. Sangiovanni-Vincentelli, "Simulation of nonlinear circuits in the frequency domain," *IEEE Trans. Computer-Aided Design*, vol. CAD-5, pp. 521–535, Oct. 1986.
[5] L. O. Chua, C. A. Desoer and E. S. Kuh, *Linear and Nonlinear Circuits*. New York: McGraw-Hill, 1987.
[6] M. B. Steer and P. J. Khan, "An algebraic formula for the output of a system with large-signal, multifrequency excitation," *Proc. IEEE*, vol. 71, pp. 177–179, Jan. 1983.
[7] J. K. Hale, *Ordinary Differential Equations*. Melbourne, FL: Krieger, 1980.
[8] A. L. Sangiovanni-Vincentelli, "Circuit simulation," in *Computer Design Aids for VLSI Circuits*, P. Antognetti, D. O. Pederson, and H. DeMan, Eds. Norwell, MA: Sijthoff & Noordhoff, 1981, pp. 19–112.
[9] L. O. Chua, "Device modeling via basic nonlinear circuit elements," *IEEE Trans. Circuits Syst.*, vol. CAS-27, pp. 1014–1044, Nov. 1980.
[10] A. Ushida and L. O. Chua, "Frequency-domain analysis of nonlinear circuits driven by multi-tone signals," *IEEE Trans. Circuits Syst.*, vol. CAS-31, pp. 766–778, Sept. 1984.
[11] R. Gilmore, "Nonlinear circuit design using the modified harmonic balance algorithm," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-34, pp. 1294–1307, Dec. 1986.
[12] G. H. Golub and C. F. V. Loan, *Matrix Computations*. Baltimore: The Johns Hopkins University Press, 1983.

[13] G. Dahlquist and A. Björck, *Numerical Methods*. Englewood Cliffs, NJ: Prentice-Hall, 1974.
[14] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*. New York: Academic Press, 1970.
[15] K. S. Kundert, "Sparse matrix techniques," in *Circuit Analysis, Simulation and Design*, vol. 3, pt. 1, A. E. Ruehli, Ed. New York: North-Holland, 1986.
[16] H. Statz, P. Newman, I. W. Smith, R. A. Pucel, and H. A. Haus, "GaAS FET device and circuit simulation in SPICE," *IEEE Trans. Electron Devices*, vol. ED-34, pp. 160–169, Feb. 1987.
[17] J. B. Beyer, S. N. Prasad, R. C. Becker, J. E. Nordman and G. K. Hohenwarter, "MESFET distributed amplifier design guidelines," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-32, pp. 268–275, Mar. 1984.

✠



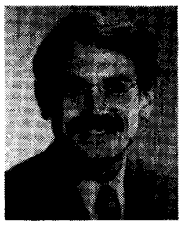
Kenneth S. Kundert (S'86) received the M.Eng. and B.S. degrees in electrical engineering and computer sciences from the University of California, Berkeley, in 1983 and 1979, respectively. His area of specialization at the time was analog circuit design. He is currently working towards the Ph.D. degree in EECS 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.

Gregory Sorkin (S'87) received the A.B. degree in mathematics from Harvard College, Cambridge, MA, in 1983. He is an employee of the IBM T. J. Watson Research Center, and is currently pursuing the Ph.D. degree in electrical engineering at the University of California, Berkeley.

He has done work in the areas of algorithms for VLSI physical design, the theory of simulated annealing, and circuit simulation, as well as graph theory and cellular automata. In addition to these, he is also currently interested in neural networks.

✠



Alberto Sangiovanni-Vincentelli (M'74–SM'81–F'83) received the Dr.Eng. degree (summa cum laude) from the Politecnico di Milano, Italy, in 1971.

From 1971 to 1977, he was with the Istituto di Elettrotecnica ed Elettronica, Politecnico di Milano, Italy, where he held the position of Research Associate, Assistant and Associate Professor. In 1976, he joined the Department of Electrical Engineering and Computer Sciences of the University of California at Berkeley, where he is presently Professor. He is a consultant in the area of computer-aided design to several industries. His research interests are in various aspects of computer-aided design of integrated circuits, with particular emphasis on VLSI simulation and optimization. He was Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, and is Associate Editor of the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS and a member of the Large-Scale Systems Committee of the IEEE Circuits and Systems Society and of the Computer-Aided Network Design (CANDE) Committee. He was the Guest Editor of a special issue of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS on CAD for VLSI. He was Executive Vice-President of the IEEE Circuits and Systems Society in 1983. In 1981, he received the Distinguished Teaching Award of the University of California. At the 1982 IEEE-ACM Design Automation Conference, he was given a Best Paper and a Best Presentation Award. In 1983, he received the Guillemin-Cauer Award for the best paper published in the IEEE Transactions on CAS and CAD in 1981–1982. At the 1983 Design Automation Conference, he received a Best Paper Award.

Dr. Sangiovanni-Vincentelli is a member of ACM and Eta Kappa Nu.