NONLINEAR CIRCUIT SIMULATION IN THE FREQUENCY-DOMAIN*

Kenneth S. Kundert
Alberto Sangiovanni-Vincentelli
University of California, Berkeley, CA. 94720

Abstract Simulation in the frequency-domain avoids many of the severe problems experienced when trying to use traditional time-domain simulators such as Spice [1] to find the steady-state behavior of analog, RF, and microwave circuits. In particular, frequency-domain simulation eliminates problems from distributed components and high-Q circuits by forgoing a nonlinear differential equation representation of the circuit in favor of a complex algebraic representation.

This paper describes the *spectral Newton* technique for performing simulation of nonlinear circuits in the frequency-domain, and its implementation in *Harmonica*. Also described are the techniques used by *Harmonica* to exploit both the structure of the spectral Newton formulation and the characteristics of the circuits that would be typically seen by this type of simulator. These techniques allow *Harmonica* to be used on much larger circuits than were normally attempted by previous nonlinear frequency-domain simulators, making it suitable for use on Monolithic Microwave Integrated Circuits (MMICs).

1. Introduction

It is common for circuits designed to operate at RF and microwave frequencies to be pseudo-linear in nature. By this it is meant that input signals are sinusoidal and small enough so that few harmonics are produced. This does not imply that the nonlinearities in the circuit can be neglected. Indeed, mixers and oscillators fit this description and yet they fundamentally depend on nonlinear effects to operate. It is also common for these circuits to have a large number of distributed components such as transmission lines, whose models often include loss, dispersion, and coupling effects. These distributed components are very difficult and often impractical to simulate in the time-domain because the partial differential equations that describe these structures often do

*ICCAD85

not have closed-form solutions. In addition, time-domain simulators are not able to exploit the pseudo-linear nature of these circuits, and often require an excessive amount of time because the steady-state solution is desired. Using a time-domain simulator to find the steady-state solution requires that the circuit be simulated until the transient solution vanishes, resulting in a very expensive simulation when the circuit is high-Q or narrow-band.

Simulating these circuits in the frequency-domain avoids these problems and eases the problem of formulating the equations for distributed components by transforming the time-domain differential equations into algebraic complex equations. The pseudo-linear nature of these circuits is naturally exploited since the amount of cpu time required for a frequency-domain simulation is proportional to the number of frequencies present. Another method has been proposed to find the steady-state solution [2, 3]. The shooting method, as it is often called, iteratively solves the circuit in the time-domain for one period; on each iteration the initial condition is varied, attempting to make the signals at the end of the period exactly match those at the beginning. The shooting method does work on autonomous circuits, but does not help with distributed components and is not capable of finding almost-periodic solutions.

Previous efforts at nonlinear frequency-domain simulation were based on the use of *harmonic balance* to formulate the frequency-domain equations and an optimizer to solve them [4, 5, 6]. Using an optimizer to solve these equations results in the number of harmonics and nonlinear devices being severely limited. It is possible to remove this limit by instead solving the nonlinear equations with Newton's method [7]. When this is done the circuit equations can be reformulated in a more natural way, and by doing so the name harmonic balance becomes somewhat of a misnomer. So the more appropriate name *spectral Newton* was coined.

2. Spectral Newton

In order to apply the spectral Newton method, two conditions must be satisfied. First, the circuit must be asymptotically stable and must have a steady-state solution for the given excitation; chaotic and subharmonic behavior is specifically excluded. Second, all nonlinear devices must be lumped and their constitutive relationship must be algebraic, differentiable, and expressible in one of the following forms:

$$\begin{array}{llll} i = i(v) & q = q(v) & i = i(\phi) & q = q(\phi) \\ v = v(i) & v = v(q) & \phi = \phi(i) & \phi = \phi(q) \end{array}$$

Though not necessary, we will assume that the circuit has a periodic solution. Extension of these results to almost-periodic solutions is straightforward [7]. For simplicity, we will further assume that a nodal formulation is being used and that only voltage controlled resistive and capacitive nonlinearities are allowed.

In the time-domain a circuit can be modeled as a system of N nonlinear differential equations, here written in compact form as

$$f(v,t) = i_s(t) \qquad v(0) = v_0 \tag{1}$$

Let $U = \{h | h : \Re \to \Re^N\}$.[1] Then $v \in U$ is the vector of unknown node voltage waveforms; $v_0 \in \Re^N$ is the unknown initial condition that results in the solution being periodic, i.e. $v(t) = v(t + T_0)\ \forall\ t$; $i_s \in U$ is the vector of source current waveforms; and $f : U \times \Re \to \Re^N$. In order to solve this system it is traditional to discretize it in time and apply some numeric integration method. However if only the steady-state response is of interest, it is possible to transform this system into the frequency-domain and solve it without resorting to numeric integration. To solve the system in the frequency-domain, it is necessary to truncate the number of harmonics considered to a finite, and in general small, H. The truncation is analogous to discretization in the time-domain and is theoretically not a limitation because for all realizable circuits there exists a frequency beyond which there is negligible power.

Since the nonlinear devices are lumped, $f(v,t)$ can be rewritten as

$$f(v,t) = i(v(t)) + \frac{d}{dt}q(v(t)) + \int_0^t y(t-\tau)v(\tau)d\tau \tag{2}$$

where $i, q : \Re^N \to \Re^N$ are differentiable functions representing respectively the sum of the currents exiting the nodes due to the nonlinear conductors and the sum of the charge exiting the nodes due to the nonlinear capacitors; and $y(t) \in \Re^N$ is the impulse response of the circuit with the nonlinear devices turned off.[2]

Since y is linear, the Laplace transform may be used to transform it into the frequency-domain, $y(t) \leftrightarrow \Upsilon(s)$. Furthermore, since v is periodic and the circuit is stable

$$\int_0^t y(t-\tau)v(\tau)d\tau \leftrightarrow YV$$

where $v \leftrightarrow V \in C^{HN}$ contains the node voltage phasor for each node and each frequency, and $Y \in C^{HN \times HN}$ is a block node admittance matrix for the linear portion for the circuit.

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

$$Y_{mn} = [Y_{mn}(k\omega_o, l\omega_o)] \qquad k, l \in \{0, 1, \ldots, H-1\}$$

$$Y_{mn}(k\omega_o, l\omega_o) = \begin{cases} \Upsilon_{mn}(\jmath k\omega_o) & k = l \\ 0 & k \neq l \end{cases}$$

where m, n are the node indices; k, l are the frequency indices, and $\jmath = \sqrt{-1}$.

Since v, i and q are periodic, (1) and (2) can be transformed into the frequency domain by applying the Fourier series.

$$F(V) = I(V) + \jmath\Omega Q(V) + YV = I_s \tag{3}$$

where $i_s \leftrightarrow I_s \in C^{HN}$ contains the source current phasor for each node and frequency; $f \leftrightarrow F$, $i \leftrightarrow I$, $q \leftrightarrow Q : C^{HN} \rightarrow C^{HN}$; and $\Omega \in C^{HN \times HN}$

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

$$\Omega_{mn} = \begin{cases} 0 & m \neq n \\ \mathrm{diag}\{0, \omega_o, 2\omega_o, \ldots, (H-1)\omega_o\} & m = n \end{cases}$$

and $\omega_o = 2\pi/T_o$.

The Newton-Raphson method is used to solve (3) for V, which requires that $F(V)$ be differentiated with respect to V. However, since $f(v, t)$ and $v(t)$ are constrained to be real functions, $F(V)$ is non-analytic, which implies that its derivative $J(V)$ cannot be represented using complex numbers. To circumvent this problem each complex number is written as an equivalent vector in $\Re^2$. To perform this conversion, some more notation will be defined. Let $X \in C$. Then define $X^R, X^I \in \Re$, $\bar{X} \in \Re^2$ such that $X^R = \mathrm{Re}\{X\}$, $X^I = \mathrm{Im}\{X\}$, and $\bar{X} = [X^R \quad X^I]^T$. Similar notation is used for vectors and matrices. Using this notation, $\bar{F}(\bar{V}), \bar{V} \in \Re^{2HN}$ and (3) is solved with the iteration

$$\bar{V}^{(k+1)} = \bar{V}^{(k)} - \bar{J}(\bar{V}^{(k)})^{-1}[\bar{F}(\bar{V}^{(k)}) - \bar{I}_s] \tag{4}$$

where $J \in \Re^{2NH \times 2NH}$ is the spectral Jacobian, i.e.

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

If $V^{(0)}$ is chosen close enough to a solution, then given certain mild conditions on (3), the sequence converges to that solution [9].

The only impediment in evaluating this expression is finding the contribution of the nonlinear elements to $F(V)$ and $J(V)$ because it is extremely difficult to formulate the nonlinear device equations directly in the frequency-domain. To avoid this problem, the node voltages are transformed into the time-domain and applied to the nonlinear devices.

The response current of these devices is then calculated and converted back into the frequency-domain and added to $F(V)$. Calculation of $J(V)$ is similar except that the node voltage waveforms are applied to the devices' derivative equations and the resulting waveforms are converted into the frequency-domain and added to $J(V)$. The calculation of the spectral Jacobian will be covered in more detail in the next section. Since the signals are assumed to be periodic, the Fast Fourier Transform (FFT) may be used to perform the transformations between the frequency- and time-domains. If the periodic signal restriction is loosened to allow almost-periodic signals, then the Discrete Fourier Transform (DFT) should be used.

Spectral Newton Algorithm	
Given:	Initial guess of node voltage spectra taken from DC and small-signal AC analysis of circuit.
Step 1:	Convert node voltage spectra into time-domain.
Step 2:	Evaluate nonlinear devices for output current and derivative waveforms.
Step 3:	Convert the waveforms into the frequency-domain.
Step 4:	Build and solve the spectral Newton update equation (4).
Step 5:	Check $F(V)$ and ΔV for convergence, if not converged, go to step 1.

3. Spectral Jacobian

In our method, the spectral Jacobian is organized as the block matrix

$$\bar{J}(\bar{V}) = \left[\frac{\partial \bar{F}_m(\bar{V})}{\partial \bar{V}_n} \right] \qquad m, n \in \{1, 2, \ldots, N\} \tag{5}$$

where $\bar{F}_m, \bar{V}_n \in \Re^{2H}$ are vectors of phasors, one phasor for each frequency. $\bar{F}_m$ equals the sum of currents exiting node m and $\bar{V}_n$ equals the node voltage of node n. This block matrix is referred to as the block node admittance matrix because its structure is identical to the node admittance matrix. The blocks have the form

$$\frac{\partial \bar{F}_m}{\partial \bar{V}_n} = \left[\frac{\partial \bar{F}_m(\bar{V}, k\omega_o)}{\partial \bar{V}_n(l\omega_o)} \right] \qquad k, l \in \{0, 1, \ldots, H-1\} \tag{6}$$

where $\bar{F}_m(\bar{V}, k\omega_o) \in \Re^2$ is the k^{th} harmonic of $\bar{F}_m$ and $\bar{V}_n(l\omega_o) \in \Re^2$ is the l^{th} harmonic of $\bar{V}_n$.

$$\frac{\partial \bar{F}_m(\bar{V}, k\omega_o)}{\partial \bar{V}_n(l\omega_o)} = \begin{bmatrix} \frac{\partial F_m^R(\bar{V}, k\omega_o)}{\partial V_n^R(l\omega_o)} & \frac{\partial F_m^R(\bar{V}, k\omega_o)}{\partial V_n^I(l\omega_o)} \\ \frac{\partial F_m^I(\bar{V}, k\omega_o)}{\partial V_n^R(l\omega_o)} & \frac{\partial F_m^I(\bar{V}, k\omega_o)}{\partial V_n^I(l\omega_o)} \end{bmatrix}$$

This derivative consists of the sum of terms

$$\frac{\partial \bar{F}_m(\bar{V}, k\omega_o)}{\partial \bar{V}_n(l\omega_o)} = \tag{7}$$

$$\frac{\partial \bar{I}_m(\bar{V}, k\omega_o)}{\partial \bar{V}_n(l\omega_o)} + \begin{bmatrix} 0 & -k\omega_o \\ k\omega_o & 0 \end{bmatrix} \frac{\partial \bar{Q}_m(\bar{V}, k\omega_o)}{\partial \bar{V}_n(l\omega_o)} + \bar{Y}_{mn}(k\omega_o, l\omega_o)$$

$$\bar{Y}_{mn}(k\omega_o, l\omega_o) = \begin{bmatrix} Y^R_{mn}(k\omega_o, l\omega_o) & -Y^I_{mn}(k\omega_o, l\omega_o) \\ Y^I_{mn}(k\omega_o, l\omega_o) & Y^R_{mn}(k\omega_o, l\omega_o) \end{bmatrix} \tag{8}$$

Only the calculation of $\frac{\partial I^R_m(\bar{V}, k\omega_o)}{\partial V^R_n(l\omega_o)}$ will be performed, the calculation of the other terms in $\frac{\partial \bar{I}_m(\bar{V}, k\omega_o)}{\partial \bar{V}_n(l\omega_o)}$ and $\frac{\partial \bar{Q}_m(\bar{V}, k\omega_o)}{\partial \bar{V}_n(l\omega_o)}$ is similar.

$$I_m(V, k\omega_o) = \frac{1}{T_o} \int_0^{T_o} i_m(v(t)) e^{-\jmath k\omega_o t} dt$$

$$I^R_m(\bar{V}, k\omega_o) = \frac{1}{T_o} \int_0^{T_o} i_m(v(t)) \cos(k\omega_o t) dt$$

The function v is considered implicitly to be a function of its frequency-domain equivalent, V; so the chain rule can be employed to calculate the derivative.

$$\frac{\partial I^R_m(\bar{V}, k\omega_o)}{\partial V^R_n(l\omega_o)} = \frac{1}{T_o} \int_0^{T_o} \frac{\partial i_m(v(t))}{\partial v_n(t)} \frac{\partial v_n(t)}{\partial V^R_n(l\omega_o)} \cos(k\omega_o t) dt$$

Now the derivative of $v_n(t)$ is calculated.

$$v_n(t) = \sum_{k=-\infty}^{\infty} V_n(k\omega_o) e^{\jmath k\omega_o t}$$

$$v_n(t) = V^R_n(0) + 2\sum_{k=1}^{\infty} V^R_n(k\omega_o)\cos(k\omega_o t) - V^I_n(k\omega_o)\sin(k\omega_o t)$$

For $l = 0$, the derivative is trivial; for $l \neq 0$

$$\frac{\partial v_n(t)}{\partial \bar{V}_n(l\omega_o)} = \begin{bmatrix} \frac{\partial v_n(t)}{\partial V^R_n(l\omega_o)} \\ \frac{\partial v_n(t)}{\partial V^I_n(l\omega_o)} \end{bmatrix} = \begin{bmatrix} 2\cos(l\omega_o t) \\ -2\sin(l\omega_o t) \end{bmatrix}$$

So if $l \neq 0$

$$\begin{aligned} \frac{\partial I^R_m(\bar{V}, k\omega_o)}{\partial V^R_n(l\omega_o)} &= \frac{2}{T_o} \int_0^{T_o} \frac{\partial i_m(v(t))}{\partial v_n(t)} \cos(l\omega_o t)\cos(k\omega_o t) dt \\ &= \frac{1}{T_o} \int_0^{T_o} \frac{\partial i_m(v(t))}{\partial v_n(t)} [\cos((k+l)\omega_o t) + \cos((k-l)\omega_o t)] dt \end{aligned}$$

Now let $G_{mn}(k\omega_o) \in C$ be the k^{th} harmonic of $\frac{\partial i_m(v(t))}{\partial v_n(t)}$, i.e., let

$$G_{mn}(k\omega_o) = \frac{1}{T_o} \int_0^{T_o} \frac{\partial i_m(v(t))}{\partial v_n(t)} e^{\jmath k\omega_o t} dt \tag{9}$$

Then for $l = 0$

$$\frac{\partial \bar{I}_m(\bar{V}, k\omega_o)}{\partial \bar{V}_n(0)} = \begin{bmatrix} G^R_{mn}(k\omega_o) & 0 \\ G^I_{mn}(k\omega_o) & 0 \end{bmatrix} \tag{10}$$

and for $l \neq 0$

$$\frac{\partial \bar{I}_m(\bar{V}, k\omega_o)}{\partial \bar{V}_n(l\omega_o)} = \tag{11}$$
$$\begin{bmatrix} G^R_{mn}((k+l)\omega_o) + G^R_{mn}((k-l)\omega_o) & G^I_{mn}((k+l)\omega_o) + G^I_{mn}((k-l)\omega_o) \\ G^I_{mn}((k+l)\omega_o) - G^I_{mn}((k-l)\omega_o) & G^R_{mn}((k-l)\omega_o) - G^R_{mn}((k+l)\omega_o) \end{bmatrix}$$

This completes the calculation of the spectral Jacobian. It may now be synthesized from (5), (6), (7), (8), (9), (10) and (11).

4. Harmonica

We are currently developing a simulator based on the spectral Newton algorithm. Unlike previous efforts [4, 7], which were aimed at circuits containing only one or two nonlinear devices, *Harmonica* is designed to quickly analyze large circuits with many nonlinear devices. This advance is made possible by using spectral Newton, by exploiting the structure and characteristics of the spectral Jacobian, and by exploiting the linear and almost-linear behavior of the devices.

The spectral Jacobian is quite large and moderately dense, having about $4H$ elements per row or column. Naively applying sparse matrix techniques is not enough to solve the Newton update equation (4) efficiently. It is necessary to make some judicious approximations when constructing and decomposing the Jacobian to reduce the density of the matrix. The Jacobian is only used to generate new iterates, and is not used when confirming convergence, so errors from approximations in the Jacobian only affect the rate and region of convergence, not the accuracy of the final solution. An approximate spectral Jacobian results in the loss of quadratic convergence, but the gain in efficiency more than makes up for this loss.

In a node admittance matrix, any particular element is the sum of contributions from zero or more devices. This is also true for the block node admittance matrix generated by the spectral Newton algorithm. In the block node admittance matrix, contributions from linear devices come as diagonal blocks, i.e. only the diagonal 2×2 sub-blocks are nonzero. Nonlinear devices contribute full blocks, however if the device is behaving almost-linearly the elements on the diagonal of the block are the largest and as the distance from the diagonal increases their magnitude decreases rapidly. This results from (10) and (11), and from the bandwidth of the derivative spectrum (9) being small if the device is behaving almost-linearly.

The effort required to LU decompose the spectral Jacobian can be significantly reduced if two approximations are made. First, in those blocks that have contributions only from elements behaving linearly or almost-linearly, the small elements far from the diagonal should be set to zero and the operations that would normally be performed on these elements should be avoided. The decision of which elements are small enough to ignore can be made by comparing the magnitude of the upper harmonics of the derivative spectrum to some small fraction of the DC component. The value $10^{-4}G_{mn}(0)$ seems to work well. Of those harmonics smaller than the cutoff criterion, only the first should be kept: all others should be set to zero. This last nonzero harmonic is called the *guard harmonic*. Second, all nonzero fill-ins that result during LU decomposition from operations involving the guard harmonic should be ignored. This prevents the bandwidth of the blocks from growing unnecessarily during the decomposition. These two approximations allow *Harmonica* to exploit linear and almost-linear behavior in the circuit. To get the most from them, pivoting of the block node admittance matrix should be done with the additional goal of exploiting the reduced bandwidth of the blocks.

The last technique used to accelerate the spectral Newton iteration is to only occasionally reevaluate the spectral Jacobian [9]. This works well if the Jacobian is not changing much between iterations. It can greatly reduce the time required for an iteration because device evaluations and forward- and backward-elimination of the LU decomposed Jacobian are much faster than the decomposition of the spectral Jacobian.

Harmonica is written in the C programming language.

5. Results

Execution times for *Harmonica* are a strong function of the number of harmonics simulated, the strength of the nonlinear behavior, and the number of devices behaving nonlinearly. Before applying the techniques given in the previous section each iteration requires $O(N^{1.5}H^3)$ operations. After applying those techniques, and measuring the execution times of only a few circuits, each iteration seems to require $O(N^{1.5}H)$ operations. The iteration count remains relatively constant as the number of harmonics changes.

The times for three circuits are presented in Table 1. The first two circuits are well-suited to simulation in the frequency-domain and poorly suited to time-domain simulation. With the last circuit, the roles are reversed. The first is a traveling wave amplifier (TWA) [10] that contains four bipolar transistors and ten transmission lines of noncommensurate length. Note that the transmission lines are constrained to be ideal by SPICE, *Harmonica* easily handles lossy and dispersive lines. The second circuit contains a differential pair and a crystal lattice filter. This circuit demonstrates the ease with which *Harmonica* handles high-Q circuits.

The last circuit, a simple noninverting amplifier containing a $\mu A741$, is troublesome to *Harmonica* because the op amp is internally acting strongly nonlinear: the large load causing the output stage to operate class B. This example demonstrates that *Harmonica* is able to handle strongly nonlinear circuits, though it may run longer than traditional simulators.

Since *Harmonica* is solving an algebraic system of equations, if sufficient harmonics are computed, it can be much more accurate than a time-domain simulator. This is demonstrated in all the test circuits: when *Harmonica* was able to converge with only eight harmonics computed, the maximum error in any harmonic was less than 1ppm. Furthermore, the worst case error resulting from harmonics not computed was less than 20ppm. These numbers were greatly reduced when more than eight harmonics are computed. SPICE2 computes with a 1000ppm error tolerance.

6. Conclusions

The spectral Newton method for frequency-domain simulation of nonlinear circuits was described along with techniques used by *Harmonica* to increase the efficiency of the method. This method allows circuits that are behaving quasi-linearly to be quickly simulated, even though they may be very high-Q or contain many transmission lines.

Work is being done to at least double the speed of the simulator by further exploiting the structure of the spectral Jacobian.

7. Acknowledgements

We would like to thank Peter Moore for his contributions to *Harmonica*; nd Jacob White, Howard Ko, and the rest of the Berkeley CAD research group for many helpful discussions. This work was supported by Hewlett-Packard and MICRO.

Notes

1. In function space, no one can hear you scream [8].

Circuit	Conditions	SPICE2	*Harmonica* harmonics		
			8	16	32
TWA	$V_{out} = 1$ V	62,500[3]	7	22	56
TWA	$V_{out} = 0.5$ V	NA[4]	6	16	40
Filter		2350	7	20	94
$\mu A741$	$V_{out} = 1$ V $R_L = \infty\Omega$	9	6	13	29
$\mu A741$	$V_{out} = 1$ V $R_L = 10$ KΩ	13	10	28	63
$\mu A741$	$V_{out} = 1$ V $R_L = 10$ KΩ	14	NA[5]	365	575

Table 1. Simulation times for SPICE2 and *Harmonica* for various circuits. Times are given in seconds and were measured on a VAX 11/785 running UNIX 4.3BSD.

2. To turn a nonlinear device off, simply replace its constitutive equation $y = f(x)$ with $y = 0$.
3. This number is an extrapolation made from measurements of times required for smaller simulation intervals. The desired time interval (two periods) causes memory usage to exceed UNIX's 16 MByte limit.
4. This time was not measured.
5. Circuit was behaving too nonlinearly for *Harmonica* to converge with so few harmonics.

References

[1] Laurence W. Nagel. *SPICE2: A Computer Program to Simulate Semiconductor Circuits.* Electronics Research Laboratory Publications, U.C.B., 94720, Memorandum No. UCB/ERL M520, May 1975.

[2] T.J. Aprille, T.N. Trick. Steady-state analysis of nonlinear circuits with periodic inputs. *Proceedings of the IEEE*, vol. 60, pp. 108-114, January 1972.

[3] 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, pp. 161-175, March 1980.

[4] M. Nakhla, J. Vlach. A piecewise harmonic balance technique for determination of the periodic response of nonlinear systems. *IEEE Transactions on Circuits and Systems*, vol. CAS-23, no. 2, pp. 85-91, February 1976.

[5] K. Gopal, M.S. Nakhla, K. Singhal, J. Vlach. Distortion analysis of transistor networks. *IEEE Transactions on Circuits and Systems*, vol. CAS-25, no. 2, pp. 99-106, February 1978.

[6] F. Filicori, O. Monaco, C. Naldi. Simulation and design of microwave class-C amplifiers through harmonic analysis. *IEEE Transactions on Microwave Theory and Techniques*, vol. MTT-27, no. 12, pp. 1042-1051, December 1979.

[7] A. Ushida, L.O. Chua. Frequency-domain analysis of nonlinear circuits driven by multi-tone signals. *IEEE Transactions on Circuits and Systems*, vol. CAS-31, no. 9, pp. 766-778, September 1984.

[8] Richard J. Anobile. *Alien*, Avon, 1979. This modification of the Alien's teaser was found on a mens' bathroom wall in Cory Hall, U.C. Berkeley.

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

[10] J.B. Beyer, S.N. Prasad, R.C. Becker, J.E. Nordman, G.K. Hohenwarter. MESFET distributed amplifier design guidelines. *IEEE Transactions on Microwave Theory and Techniques*, vol. MTT-32, no. 3, pp. 268-275, March 1984.