A DFT-BASED APPROACH TO THE QUASI-PERIODIC STEADY-STATE SIMULATION OF IC's AND NOISE ANALYSIS ALGORITHM IN SPECTRE

by

Charmine S. Tung

Memorandum No. UCB/ERL M91/42

17 May 1991

A DFT-BASED APPROACH TO THE QUASI-PERIODIC STEADY-STATE SIMULATION OF IC's AND NOISE ANALYSIS ALGORITHM IN SPECTRE

by

Charmine S. Tung

Memorandum No. UCB/ERL M91/42

17 May 1991

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Acknowledgements

I would like to thank my advisor, Professor Sangiovanni-Vincentelli, for his valuable guidance, time and patience throughout this project. I wish to express my appreciation to Kenneth Kundert for his guidance and many invaluable inputs. I would also like to thank Professor Brayton for reading and providing comments for this report.

I would like to express my appreciation to Yo-Chien Yuan of EESOF, Jaijeet Roychowdhury, and Boern Johnson for many stimulating discussions on this work. I would like to thank Anantha Chandrakasan, and Narendra Shenoy for reading and providing useful feedback for this report. I also gratefully acknowledge the efficient hardware support from Brad Kreb even during the holiday season.

Thanks to all my friends for their encouragement and support. In particular, thanks to Annie Lai, Anita Chiu, Steven Low and Jenny Kong who helped pick up and deliver the report for me when I am not in the area.

Finally, to my parents, my brother, my youngest uncle and aunt, I am deeply thankful for their constant love and support. I would like to dedicate this report to my parents.

This work was supported by Darpa and Hewlett Packard Company.

Contents

List of Figures

Chapter 1

Introduction

1.1 Overview: SPECTRE and Harmonic Balance

Though nonlinear time-domain circuit simulators such as SPICE are well accepted for use with low frequency analog circuits, there is very little use of such simulators at microwave frequencies. In the microwave region, frequency-domain simulators are more suitable for a number of reasons. First, microwave circuits contain a large number of distributed components that are more easily analyzed in the frequency-domain. Second, microwave circuits tend to be AC-coupled and are often narrow-band. Time-domain simulation is very expensive for these circuits because the simulation must run long enough for the initial transient to decay. Finally, most microwave measurements are made in the frequency-domain.

Currently the most used technique for frequency-domain nonlinear circuit simulation is harmonic balance. Harmonic balance represents signals in the frequency-domain as a collection of Fourier coefficients of sinusoids. Kirchoff's current law is formulated in the frequency-domain and evaluated by considering the linear and the nonlinear devices separately. Linear devices can be evaluated directly in the frequency-domain. For an arbitrary nonlinear device, there is no known way to compute the response directly in the frequency-domain. Therefore, the excitation spectra of a nonlinear device is first transformed to the time-domain sampled waveform. The device response is then evaluated in the time-domain, and transformed back to the frequency-domain. Figure 1.1 shows this procedure. Since the coefficients of the steady-state response form an algebraic function of the coefficients of the stimulus, the nonlinear integro-differential equations that describe a circuit are converted

by harmonic balance into a system of nonlinear algebraic equations whose solution is the steady-state response of the circuit.

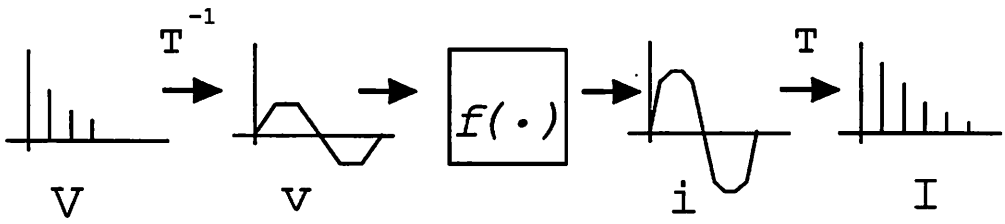


Figure 1.1: Nonlinear Device Evaluation Procedure. T^{-1} and T represent the inverse and forward Fourier transforms.

SPECTRE is a general purpose harmonic balance circuit simulator. It can calculate the DC operating point, perform linearized small-signal analysis and predict the large-signal periodic and quasi-periodic steady-state response of nonautonomous circuits. Currently the periodic and quasi-periodic steady-state response of circuits are analyzed using different algorithms. The main difference is the Fourier transform being used. When signals in circuits are periodic, the Discrete Fourier Transform (DFT) is used to perform the forward and inverse Fourier transforms required by the procedure for evaluating nonlinear devices. DFT can not be used directly for quasi-periodic steady-state analysis because using equally-spaced time points leads to a singularity in the transform matrix. The Almost Periodic Fourier Transform (APFT) was introduced in [3] to overcome these problems. The APFT contains a built-in time-point selection algorithm to avoid ill-condition in the transform matrix.

Recently, a new way of applying the DFT to almost-periodic circuits has been proposed [1]. This method is based on scaling of the fundamental frequencies of the quasi-periodic signals to avoid ill-conditioning. The DFT-based approach to the quasi-periodic steady-state simulation improves the simulation time and provides better control of aliasing.

The effect of electronic noise in circuit performance is very important for the design of analog circuits. Since the noise sources in circuits are small in magnitude, a linearized small-signal analysis can be performed to analyze the effect of noise in a circuit. The incorporation of a small-signal noise analysis extends the capability of SPECTRE to

analyzing electronic noise in circuits.

The main content of this report consists of the description of an approach using harmonic balance with DFT for quasi-periodic signals and the small-signal noise analysis. Chapter 2 presents the motivation, procedure, and results of the implementation of a quasi-periodic steady-state analysis using DFT. The accuracy, computation time, and the resource usage of this DFT based quasi-periodic steady state analysis are compared to those of the APFT based quasi-periodic steady state analysis. Chapter 3 contains descriptions and results of the small signal noise analysis in SPECTRE. Finally, the conclusion of this report is included in Chapter 4.

Chapter 2

Harmonic Balance for Quasi-periodic Signals with DFT

This chapter describes the implementation of a DFT-based quasi-periodic steady-state analysis. Some definitions[1][4] and the motivation of this implementation are presented first, followed by a description of the implementation. Finally, the accuracy, computation time, and resource usage of this implementation are compared with those of the original APFT-based quasi-periodic steady-state analysis.

2.1 Definitions

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

$$x(t) = \sum_{k=0}^{\infty}(X_k^C \cos\omega_k t + X_k^S \sin\omega_k t) \tag{2.1}$$

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

$$\sum_{k=0}^{\infty}[(X_k^C)^2 + (X_k^S)^2] < \infty \tag{2.2}$$

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

- A signal that is almost-periodic over a set of frequencies Ω is of the form

$$x(t) = \sum_{\omega_k \in \Omega} (X_k^C \cos \omega_k t + X_k^S \sin \omega_k t) \tag{2.3}$$

where $\Omega = \{ \omega_0, \omega 1, \ldots \}$, and 2.2 is satisfied.

- If Ω is finite with K elements, it is denoted by Ω_K. If there is a set of d frequencies $\lambda_1, \lambda_2, ..., \lambda_d$ which are linearly independent, and Ω is such that

$$\Omega = \{\omega | \omega = k_1\lambda_1 + k_1\lambda_1 + ... + k_d\lambda_d; k_1, k_2, ..., k_d \in Z\} \tag{2.4}$$

then Ω is a module of dimension d and the frequencies λ_1, λ_2,... , λ_d are referred to as the fundamental frequencies and form a basis (called the fundamental basis) for Ω.

- A quasi-periodic waveform is an almost periodic waveform over a set of frequencies Ω that is a module of finite dimension.

- Sampling Theorem[4]: For any sampling interval Δ, the Nyquist critical frequency is given by $f_c = 1/(2\Delta)$. If a continuous function $h(t)$, sampled at an interval Δ, happens to be bandwidth limited to frequencies smaller in magnitude than f_c, i.e., if $H(f) = 0$ for all $|f| > f_c$, then the function $h(t)$ is completely determined by its samples h_n.

In order to make the process of finding the steady-state response of a circuit tractable, it is necessary to truncate the signal frequencies to a finite set. In the periodic steady-state simulation, only the first H harmonics of a signal are taken into account. In the quasi-periodic case where there are two or more independent driving frequencies in a circuit, two popular truncation schemes[1] are often used.

- Box Truncation: Only the first H harmonics of each fundamental are considered. With the box truncation, we have:

$$\Omega_K = \{\omega | \omega = k_1\lambda_1 + k_2\lambda_2 + ... + k_d\lambda_d;$$
$$|k_j| = 0, 1, ..., H; \text{for } 1 \leq j \leq d; \text{ first nonzero } k_j \text{ positive}\} \tag{2.5}$$

where K = $1/2((2H+1)^d + 1)$ is the total number of frequency points considered. Figure 2.1(a) shows the result of a box truncation of order three for a signal with two fundamentals.

- Diamond Truncation: The absolute sum of the indices k_j is limited to be less than H. With the diamond truncation, we have:

$$\Omega_K = \{\omega | \omega = k_1\lambda_1 + k_2\lambda_2 + ... + k_d\lambda_d; k_j \in Z; \sum_{j=1}^{d} |k_j| \leq H; \text{ first nonzero } k_j \text{ positive}\} \tag{2.6}$$

where K = $\frac{2^{d-1}H^d}{d!}$ is the total number of frequency points considered. Figure 2.1(b) shows the result of a diamond truncation of order three for a signal with two fundamentals.

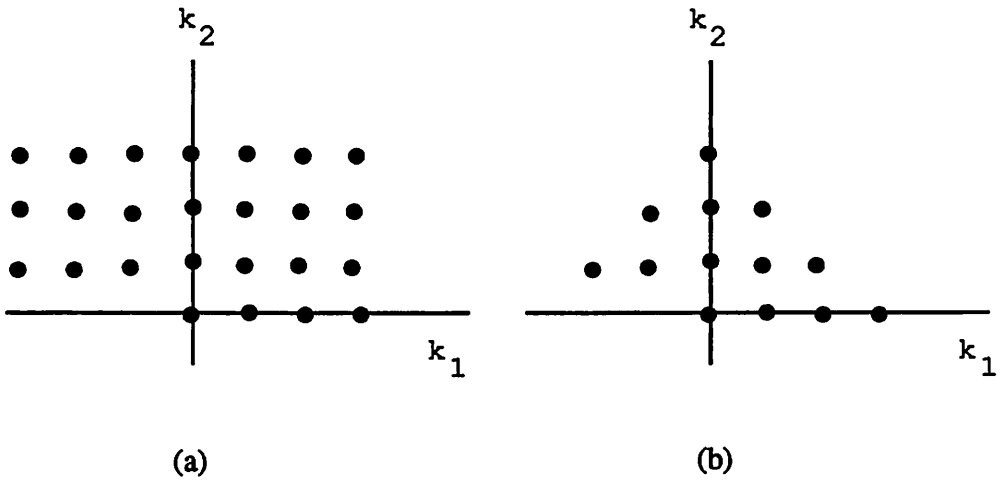


Figure 2.1: Two truncation schemes, the box (a) and the diamond (b) truncations.

2.2 Background and Motivation

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, DFT is used to provide the needed conversion between the two domains. Signals in the steady-state response of circuits like mixers are made up of several sinusoids at possibly non-harmonically related frequencies, and so are almost-periodic. Constructing transform matrices for almost-periodic signals with equally spaced time points, such as those in DFT, cause singularity in the transform matrices. Therefore DFT can not be used directly for almost-periodic steady-state simulation. The development of the almost-periodic Fourier transform (APFT) extends the application of harmonic balance to almost-periodic circuits by providing the Fourier transform of almost-periodic signals.

The APFT can be described as follows. By considering a finite set of frequencies, it is possible to sample a waveform at a finite number of time points and calculate its Fourier coefficients. The Fourier transform F and its inverse F^{-1} can be viewed as matrices acting on the sampled waveform and the Fourier coefficients. That is, a signal

$$x(t) = \sum_{\omega_k \in \Omega_K} (X_k^C \cos \omega_k t + X_k^S \sin \omega_k t) \tag{2.7}$$

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

$$\begin{bmatrix} 1 & \cos\omega_1 t_1 & \sin\omega_1 t_1 & \cdots & \cos\omega_{K-1} t_1 & \sin\omega_{K-1} t_1 \\ 1 & \cos\omega_1 t_2 & \sin\omega_1 t_2 & \cdots & \cos\omega_{K-1} t_2 & \sin\omega_{K-1} t_2 \\ 1 & \cos\omega_1 t_3 & \sin\omega_1 t_3 & \cdots & \cos\omega_{K-1} t_3 & \sin\omega_{K-1} t_3 \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 1 & \cos\omega_1 t_S & \sin\omega_1 t_S & \cdots & \cos\omega_{K-1} t_S & \sin\omega_{K-1} t_S \end{bmatrix} \begin{bmatrix} X_0 \\ X_1^C \\ X_1^S \\ \vdots \\ X_{K-1}^C \\ X_{K-1}^S \end{bmatrix} = \begin{bmatrix} x(t_1) \\ x(t_2) \\ x(t_3) \\ \vdots \\ x(t_S) \end{bmatrix} \tag{2.8}$$

If ω_k's are distict, and if $S = 2K-1$, this system is invertable for almost all choices of time points and can be compactly written as $\Gamma^{-1}X = x$. Inverting Γ^{-1} gives $\Gamma x = X$. Given a finite set Ω_K of distinct frequencies ω_k, and a set of time points, we say that Γ and Γ^{-1} are one implementation of the APFT[3]. For quasi-periodic signals, if the time points are not chosen carefully, the matrices representing Γ and Γ^{-1} can be very ill-conditioned, resulting in large numerical errors. Choosing equally spaced time points in the APFT is a particular bad case because it typically gives condition numbers on the order of $n!$. To

ensure the well-condition of the transform matrices, the APFT is implemented by selecting the time points in such a way that the rows of Γ^{-1} are nearly orthogonal[1].

The truncation of a set of analysis frequencies Ω into a finite set of K frequencies, Ω_K, introduces errors in the computation. The truncation eror is the error caused by the fact that the Fourier coefficients of the frequencies omitted from Ω are not actually zero. Although these frequencies are neglected in the transforms, they still exist in the time-domain samples x. Through the transforms, these omitted frequencies contribute errors in the frequency components under consideration. This kind of error is referred to as aliasing.

The effect of aliasing in APFT is unpredictable. Consider the case where a quasi-periodic signal over a frequency set Ω, is truncated to a finite set of size K, Ω_K. We can view APFT as a collection of K filters[3], each of which is responsible for computing one Fourier coefficient. These filters take a sequence of S samples as input and output one Fourier coefficient. Therefore the calculation of Fourier coefficients of interest depends on all frequency components in Ω. The response of a typical APFT coefficient filter is shown in Figure 2.2 . For this filter, $\omega_0, \omega_1, ..., \omega_5, \in \Omega_K, \omega_6, \omega_7, ..., \omega_9 \notin \Omega_K$, and $\omega_0, \omega_1, ..., \omega_9 \in \Omega$. The actual error due to aliasing is proportional to the response of the filter at frequencies in Ω but not in Ω_K. The error introduced by aliasing at a particular frequency of interest is a weighted sum of the Fourier coefficient of each truncated frequency. The weights are the magnitude of the filter response at each truncated frequencies. Whether or not the aliasing error will affect the accuracy of the APFT depends on 1) the Fourier coefficients of the truncated frequencies, 2) the transfer function (gain) of the filter response at the truncated frequencies, and 3) the Fourier coefficients of the frequencies of interest. When the aliasing error become large enough to overpower the actual Fourier coefficient, the APFT is no longer accurate. This situation occurrs when the Fourier coefficients of the frequency of interest is small, and that of the truncated frequencies are significant. Unfortunately there is no regular pattern in the APFT filter response, therefore, it is very hard to control the effect of aliasing in the APFT-based approach on the quasi-periodic steady-state simulation.

Whereas all frequencies within the range of interest are prone to aliasing errors in the APFT approach, not every frequency of interest is affected by aliasing if DFT is used. For example, suppose there are only two frequencies with significant Fourier coefficients being truncated, then only two frequencies within the range of interest will be affected. In the case of APFT, all of the frequencies of interest will be affected. This advantage of DFT is mainly due to the regular pattern in which aliasing occurrs, shown in Figure 2.3. The

aliasing pattern of DFT[4] can be viewed as the omitted frequencies folding back into the frequency range with respect to the truncation frequency (i.e. the highest harmonic above which the spectrum is truncated). For circuits that behave nearly linearly and are driven by smooth input signals, the Fourier coefficients of the response tend to drop off rapidly in size at upper harmonics. In this case, the aliasing only affects higher order frequencies, and its effect drops as the order of frequency decreases when DFT is used. With the use of DFT, the effect of aliasing can be controlled by choosing the order of the truncation method.

The gain in computation speed is another benefit of using DFT instead of APFT in the quasi-periodic steady-state simulation. The use of APFT requires a costly initialization procedure that is not necessary if DFT is used. APFT initialization requires a total of S^3 operations to perform the time point selection, the formation of Γ^{-1}, and the inversion of Γ^{-1} to find Γ. In addition, the regular pattern of DFT can be exploited to accelerate the Jacobian decomposition procedure by effectively using the harmonic Newton relaxation algorithm. In general, the more linear the devices are behaving, the more the DC component dominates the higher harmonics in the spectrum, and the faster the magnitude of the harmonics drop off as the order of frequency increases. As a result, elements in the conversion matrix far from the diagonal will be small compared to those on the diagonal. To reduce the density of the harmonic Jacobian, these small terms far from the diagonal are ignored. Harmonic Newton relaxation uses the 'guard harmonic' which is the smallest harmonic k such that its magnitude is smaller than a certain threshold, to zero out all the higher harmonics. Though the approximated Jacobian may reduce the rate of convergence, the gain in efficiency that it provides in matrix decomposition can make up for this loss. DFT's regular aliasing pattern preserves the diagonally dominent structure of the Jacobian for which the guard harmonic method can be applied. In the APFT-approach, the unpredictable occurrance of aliasing destroys the regular Jacobian structure that is needed for using the harmonic Newton relaxation algorithm.

2.3 Implementation

In this section, we present mapping techniques to allow the use of DFT with harmonic balance in quasi-periodic steady-state simulation. Consider a nonlinear resistor

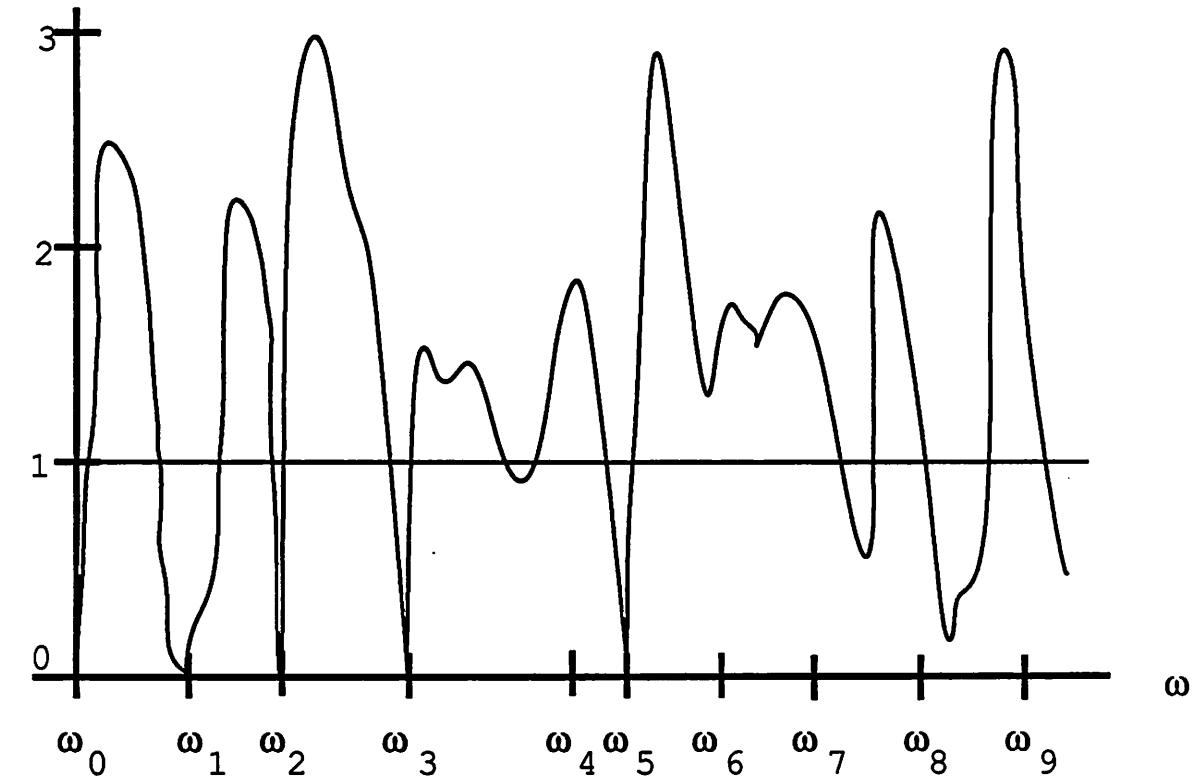


Figure 2.2: Response of a typical APFT coefficient filter.

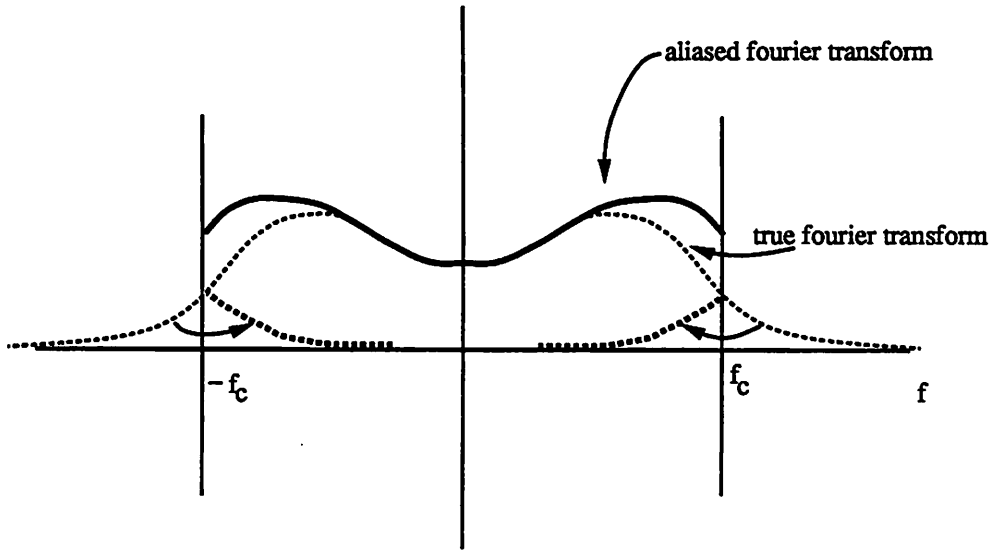


Figure 2.3: The aliasing pattern of DFT.

with the following i-v characteristics:

$$i(v) = v^2 \tag{2.9}$$

Assume it is driven by a voltage source, $v = \cos \alpha t + \cos \beta t$. The response of this nonlinear resistor is

$$i(v(t)) = 1 + 1/2\cos(2\alpha t) + \cos(\alpha t - \beta t) + 1/2\cos(2\beta t) \tag{2.10}$$

Notice that the coefficient of each sinusoid is independent of the actual frequency α and β. This is because the nonlinearity in the circuit is algebraic. When evaluating nonlinear devices, we are only interested in the spectrum of the device response, not the actual frequencies. Therefore, we can choose the fundamental frequencies freely. If the chosen fundamental frequencies are multiples of some common arbitrary frequency, the resulting signals will be periodic, and DFT can be used to perform the transformations. A scheme which maps a quasi-periodic signal into a periodic signal is presented in the following section.

2.3.1 Mapping Scheme

This section presents schemes[1] that map quasi-periodic signals to periodic signals by scaling the fundamental frequencies of the quasi-periodic signals. The main objective is to construct a map that converts the harmonic indices of a two-tone signal into the harmonic number of its corresponding periodic signal. Again, since the values of the actual frequencies do not affect the Fourier coefficients, we only consider the order according to which we place the Fourier coefficients of the frequencies into a spectrum. The mapping scheme described below are specific to two-tone signals (i.e. quasi-periodic signals with only two fundamental frequencies). For signals with three or more fundamental frequencies, specific mapping schemes need to be developed. Examples are used to demonstrate the mapping procedure.

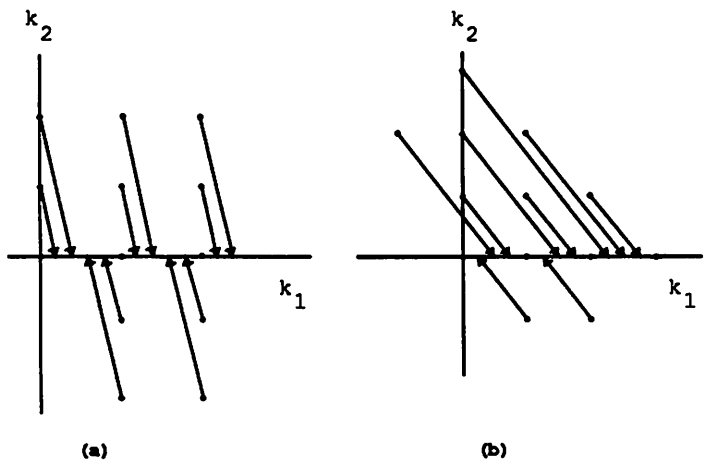


Figure 2.4: Two mapping schemes, the mapping scheme for the box (a) and for the diamond(b) truncations.

Consider a set of frequencies that is truncated using the box truncation scheme. It can be represented as

$$\Omega_K = \{\omega | \omega = k_1\lambda_1 + k_2\lambda_2; 0 \leq k_1 \leq H_1, |k_2| \leq H_2, k_1 \neq 0 \text{ if } k_2 < 0\} \tag{2.11}$$

Let $\alpha_1 = 1$ and $\alpha_2 = \lambda_1/[\lambda_2(2H_2 + 1)]$ be the scaling factors of the two fundamentals. The

correspondence of the original frequencies and artificial frequencies is given by

$$k\lambda_0 = k_1\alpha_1\lambda_1 + k_2\alpha_2\lambda_2 \tag{2.12}$$

where

$$\lambda_0 = \frac{\lambda_1}{2H_2+1} \tag{2.13}$$

and

$$k = (2H_2+1)k_1 + k_2 \tag{2.14}$$

Figure 2.4(a) shows this map. The resulting scaled set of frequencies is equally spaced and densely packed.

A set of frequencies that is truncated by using diamond truncation is of the form

$$\Omega_K = \{\omega|\omega = k_1\lambda_1 + k_2\lambda_2; |k_1| + |k_2| \leq H, k_1 + k_2 \geq 0, k_1 \neq k_2, \text{ if } k_2 > 0\} \tag{2.15}$$

Let $\alpha_1 = 1$ and $\alpha_2 = (\lambda_1 + H)/[\lambda_2(2H+1)]$ be the scaling factors of the two fundamentals. The correspondence of the original frequencies and artificial frequencies is given by

$$k\lambda_0 = k_1\alpha_1\lambda_1 + k_2\alpha_2\lambda_2 \tag{2.16}$$

where

$$\lambda_0 = \frac{\lambda_1}{H+1} \tag{2.17}$$

and

$$k = (H+1)k_1 + Hk_2 \tag{2.18}$$

This mapping scheme results in a densely packed and equally spaced frequency set. Figure 2.4(b) illustrates this scheme.

The complete nonlinear devices evaluation procedure implemented for the use of DFT is illustrated in Figure 2.5. First, the quasi-periodic node voltages are mapped by the mapping scheme, denoted by M, into their corresponding periodic signals. The periodic signals are then transformed into their time-domain sampled waveform by DFT, and the device response is evaluated in the time-domain and transformed back to frequency-domain. The true spectrum of the response is recovered by the inverse map (M^{-1}).

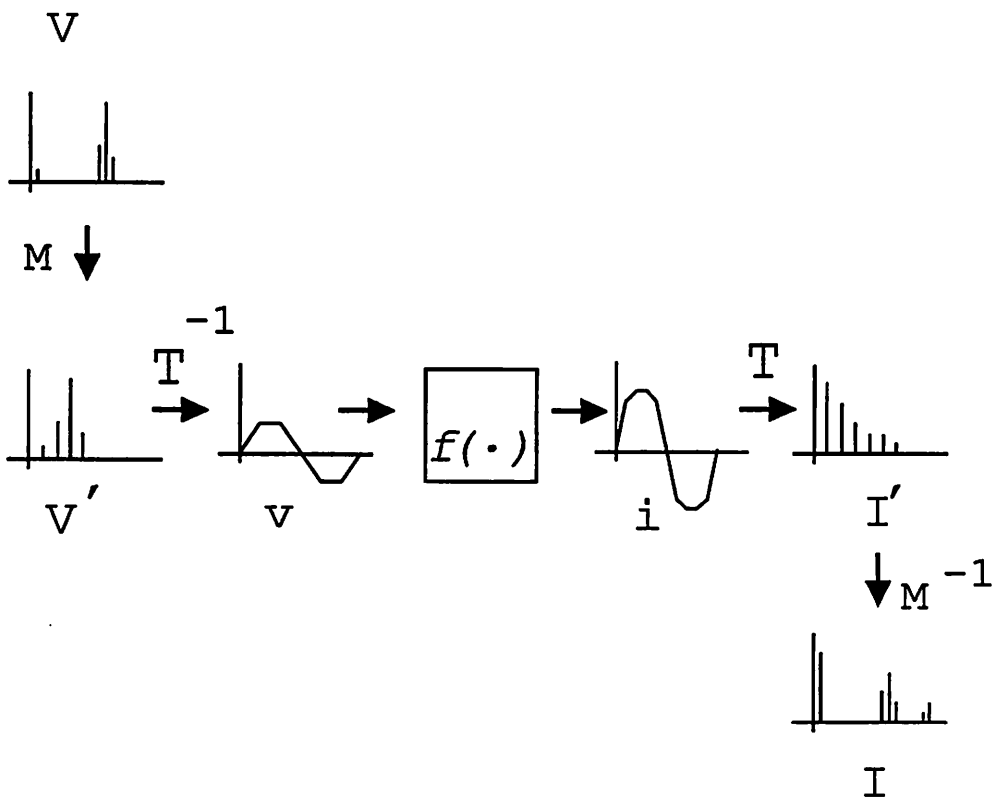


Figure 2.5: Nonlinear devices evaluation procedure.

2.3.2 Dual Frequency Set Analysis Scheme

In the actual implementation of Spectre, the Fast Fourier Transform (FFT) algorithm is used for transforming periodic signals instead of DFT. To use FFT, the size of the spectrum has to be an integer power of two ($2^n, n \in Z$). Therefore, the number of frequencies used for transformation is in general larger than the actual analysis frequencies. Since the number of analysis frequencies determines the size of the conversion matrix which in turn affects memory usage and speed of computation, it is wise to keep the number of analysis frequencies as small as possible. As a result, a dual frequency set analysis scheme[7] is used in our implementation. One frequency set is used for the analysis, and the other is used for the transformation.

2.4 Results and Comparisons

This section presents the results of the DFT-based steady-state analysis. A few test circuits are analyzed using both DFT and APFT, and their results are compared in terms of the accuracy, computation speed, and memory usage. In the following section, we first describe the procedure used to verify the correctness of the analysis results.

2.4.1 Verification Procedure

First, a simple nonlinear circuit is analyzed by hand to verify the correctness of the analysis. Consider a nonlinear resistor with constitutive equation

$$i(v) = v^4 \tag{2.19}$$

This resistor is being driven by a voltage waveform

$$v(t) = V_1 \cos(\omega_1 t) + V_2 \cos(\omega_2 t). \tag{2.20}$$

The resistor responds with a waveform of

$$\begin{aligned} i(v(t)) = {} & (3/8V_1^4 + 3/2V_1^2V_2^2 + 3/8V_2^4) + \\ & (3/2V_1^3V_2 + 3/2V_1V2^3)\cos(\omega_1 + \omega_2)t + \\ & (3/2V_1^3V_2 + 3/2V_1V2^3)\cos(\omega_1 - \omega_2)t + \\ & (1/2V_1^4 + 3/2V_1^2V_2^2)\cos 2\omega_1 t + \end{aligned}$$

$$
\begin{aligned}
&(1/2V_2^4 + 3/2V_1^2V_2^2)\cos 2\omega_2 t + \\
&3/4V_1^2V_2^2\cos(2\omega_1 + 2\omega_2)t + \\
&3/4V_1^2V_2^2\cos(2\omega_1 - 2\omega_2)t + \\
&1/2V_1^3V_2\cos(3\omega_1 + \omega_2)t + \\
&1/2V_1^3V_2\cos(3\omega_1 - \omega_2)t + \\
&1/2V_1V_2^3\cos(\omega_1 + 3\omega_2)t + \\
&1/2V_1V_2^3\cos(-\omega_1 + 3\omega_2)t + \\
&1/8V_1^4\cos 4\omega_1 t + 1/8V_2^4\cos 4\omega_2 t
\end{aligned}
\tag{2.21}
$$

The spectrum consists of the following elements:

$$
\begin{aligned}
I_{DC} &= 3/8V_1^4 + 3/2V_1^2V_2^2 + 3/8V_2^4 \\
I(1,1) &= 3/2V_1^3V_2 + 3/2V_1V_2^3 \\
I(1,-1) &= 3/2V_1^3V_2 + 3/2V_1V_2^3 \\
I(2,0) &= 1/2V_1^4 + 3/2V_1^2V_2^2 \\
I(0,2) &= 1/2V_2^4 + 3/2V_1^2V_2^2 \\
I(2,2) &= 3/4V_1^2V_2^2 \\
I(2,-2) &= 3/4V_1^2V_2^2 \\
I(3,1) &= 1/2V_1^3V_2 \\
I(3,-1) &= 1/2V_1^3V_2 \\
I(1,3) &= 1/2V_1V_2^3 \\
I(-1,3) &= 1/2V_1V_2^3 \\
I(4,0) &= 1/8V_1^4 \\
I(0,4) &= 1/8V_2^4
\end{aligned}
\tag{2.22}
$$

where I_{DC} is the dc term. I(k_1, k_2) is the spectral element at frequency $\omega = k_1\omega_1 + k_2\omega_2$. The result for $V_1 = 1$ and $V_2 = 1$ is computed by hand and compared with the program result to verify its correctness.

In general, the circuits being analyzed are so complicated that it is impossible to use hand calculation to verify the results of the analysis. Some alternative method is needed. Using the same example as above, let's fix V_1 at a constant value, and set V_2 to a value that is small compared to V_1. We denote V_1 and V_2 as the large-signal and small-signal input voltage respectively. ω_1 and ω_2 are then the corresponding large-signal and small-signal fundamental frequencies. Then

$$\begin{aligned}
I(1,1) &\propto V_2 \\
I(1,-1) &\propto V_2 \\
I(2,0) &\propto V_1^4 \\
I(0,2) &\propto V_2^2 \\
I(2,2) &\propto V_2^2 \\
I(2,-2) &\propto V_2^2 \\
I(3,1) &\propto V_2 \\
I(3,-1) &\propto V_2 \\
I(1,3) &\propto V_2^3 \\
I(4,0) &\propto V_1^4 \\
I(0,4) &\propto V_2^4
\end{aligned} \tag{2.23}$$

By observation of equation 2.23, we arrive at the following relation: Given $V_2 << V_1$

$$\begin{aligned}
I(k_1,k_2) &\propto V_2^{k_2} && \text{if } k_2 \neq 0 \\
I(k_1,k_2) &\propto V_1^n, n \in Z, && \text{if } k_2 = 0
\end{aligned} \tag{2.24}$$

Equation 2.24 shows the relation between the Fourier coefficients and input signal voltages through the harmonic index. This relation, resulting from the algebraic nonlinearity, provides an alternative method for verifying the analysis results: In a quasi-periodic circuit

driven by an input voltage signal $V_1 \cos\omega_1 t + V_2 \cos\omega_2 t$, if we fix V_1 at a large constant value, and sweep V_2 logarithmically, the log-log plot of $I(k_1, k_2)$ vs. V_2 should be linear as long as $V_2 << V_1$. In the case where $k_2 = 0$, the plot should be a flat line because the Fourier coefficients are independent of V_2. When $k_2 \neq 0$, the plot should be a line of slope k_2. Therefore, from the log-log plot of $I(k_1, k_2)$ vs. V_2, we can see whether the analysis result is correct. The range of V_2 over which the log-log plot is linear defines the dynamic range of the analysis. There are a number of factors that affects the dynamic range. First, when V_2 becomes comparable to V_1, Equation 2.23 is no longer true. Therefore the plot will deviate from a straight line at large V_2. Second, when aliasing errors become larger than the actual Fourier coefficient, the plot will no longer be linear. Among all the frequencies that are truncated, those frequency components that are pure harmonics of the large-signal fundamental frequency are larger in magnitude compared to others. This is because they are proportional to the power of large-signal voltage V_1. Since these components have Fourier coefficients that are independent of V_2, they display a flat line on the log-log plot. Finally, a computer's numerical resolution presents another lower limit to the size of V_2, and therefore introduces nonlinearities at small V_2.

2.4.2 Test Circuits and Their Results

This section presents four test circuits and their results from both DFT-based and APFT-based quasi-periodic steady-state analyses. Since it is not feasible to perform hand calculation on these circuits, the dynamic ranges of these circuits are observed and used to verify the accuracy of the analyses. The dynamic ranges of the DFT-based periodic steady-state analysis are compared to those of the APFT-based approach.

Test Circuit 1., shown in Figure 2.6 is an ideal polynomial conductor circuit. Test Circuit 2 is a single stage bipolar junction transistor mixer circuit, shown in Figure 2.8. Figure 2.10 shows Test Circuit 3 which is a simple diode mixer. Finally, Test Circuit 4, the largest circuit among the four, is a GaAsFET traveling-wave amplifier (shown in Figure 2.16). The purpose of test circuit 4 is to check the performance of the DFT-based analysis on large circuits. The dynamic range of this circuit shown in Figure 2.16 verifies that the result is correct.

Figure 2.6 through Figure 2.16 shows the four test circuits and the plot of the dynamic ranges of these circuits for the DFT and APFT-based approaches. From these

plots, three characteristics are observed. First, regardless which Fourier transform is used, the dynamic ranges of frequency components that are low in magnitude are smaller than those with large magnitude. Second, the DFT-based analysis generally has a larger dynamic range than the APFT-based approach at the same frequency. Finally, to study the relation between the dynamic range and the order of truncation method, results with different order of diamond truncation method were generated for the diode mixer circuit. These results show that when the order of truncation is low, the dynamic range of the DFT-based approach is larger than that of the APFT-based approach. As the order of truncation method increases, the dynamic range of APFT-based approach improves and approaches that of the DFT-based analysis.

The above observation shows that both APFT-based and DFT-based approaches generate correct results. However, the DFT-based approach has a larger dynamic range than the APFT-approach when aliasing is present. With a fixed order of truncation, the number of frequencies truncated is fixed, so is the size of aliasing error. The smaller the Fourier coefficient of a particular frequency of interest, the more it is overpowered by the aliasing error. Therefore, frequencies with small Fourier coefficients have smaller dynamic ranges regardless which type of Fourier transform is used. The reason why DFT-based analysis generally provides larger dynamic range than the APFT-based approach can be understood by comparing the aliasing pattern of these two Fourier transforms. When DFT is used, the effect of aliasing can be viewed as those truncated frequencies folding back into the frequency range with a unity gain and being added to the frequencies within the range considered. Therefore only a certain number of higher order frequencies of interest are affected by aliasing. When APFT is used, the aliasing may affect any frequency element, and the error is magnified according to the response of a particular filter which can be greater than unity (see Figure 2.3). As a result, for a given frequency of interest, the Fourier coefficient is more likely to be affected by aliasing if APFT is used, and hence the dynamic range of APFT-based approach is smaller than that of the DFT-based approach. As we increase the order of truncation, more frequency components are considered, and those frequencies truncated are smaller in magnitude. The effect of aliasing is lessened accordingly. As a result, the dynamic range of APFT-based analysis increases as the order of truncation increases.

2.4.3 Computation Time and Memory Usage

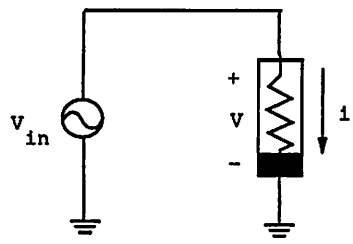
In addition to better control of aliasing, gain in computation speed is another goal for implementing the DFT-based quasi-periodic steady-state analysis. Tables 2.1 and 2.2 show the computation time and resource usage for the diode test circuit and the bjt mixer circuit with different order of diamond truncation. From these data, the DFT-based analysis requires less computation time and less memory usage compared with the APFT-based analysis. These data were generated on a VAX 8650. In Spectre, FFT is used. As mentioned previously, using FFT requires the number of frequencies considered to be a power of 2. In order to satisfy this requirement, the number of frequencies used in the FFT-based approach are generally larger than used in the APFT-based approach. This also increases the number of time points that needs to be evaluated per device. Therefore, when the number of nonlinear devices in a circuit becomes large, the speedup may not be as obvious.

H	K	time		physical memory		virtual memory	
		(FFT)	(APFT)	(FFT)	(APFT)	(FFT)	(APFT)
3	13	7.65s	14.48s	475KB	483KB	759KB	741KB
4	21	23.48s	56.55s	628KB	610KB	978KB	938KB
5	31	61.9 s	178.05s	630KB	656KB	980KB	1MB
6	43	152.27s	471.08s	1.01MB	1.2MB	1.83MB	1.92MB
7	57	372.25s	1111 s	1.05MB	1.33MB	1.83MB	1.99MB

Table 2.1: Computation time and memory usage for diode test circuit

H	K	time		physical memory		virtual memory	
		(FFT)	(APFT)	(FFT)	(APFT)	(FFT)	(APFT)
3	13	2.77s	4.25s	554KB	576KB	779KB	741KB
5	31	7.65s	38.27s	694KB	773KB	1.04KB	1MB
7	57	19.12s	205.05s	1.16MB	1.43MB	1.92MB	2.05MB

Table 2.2: Computation time and memory usage for BJT mixer test circuit



$$i = v^8 + v^7 + v^5 + v^4 + v^2$$

Figure 2.6: Test circuit 1. polynomial conductor circuit.

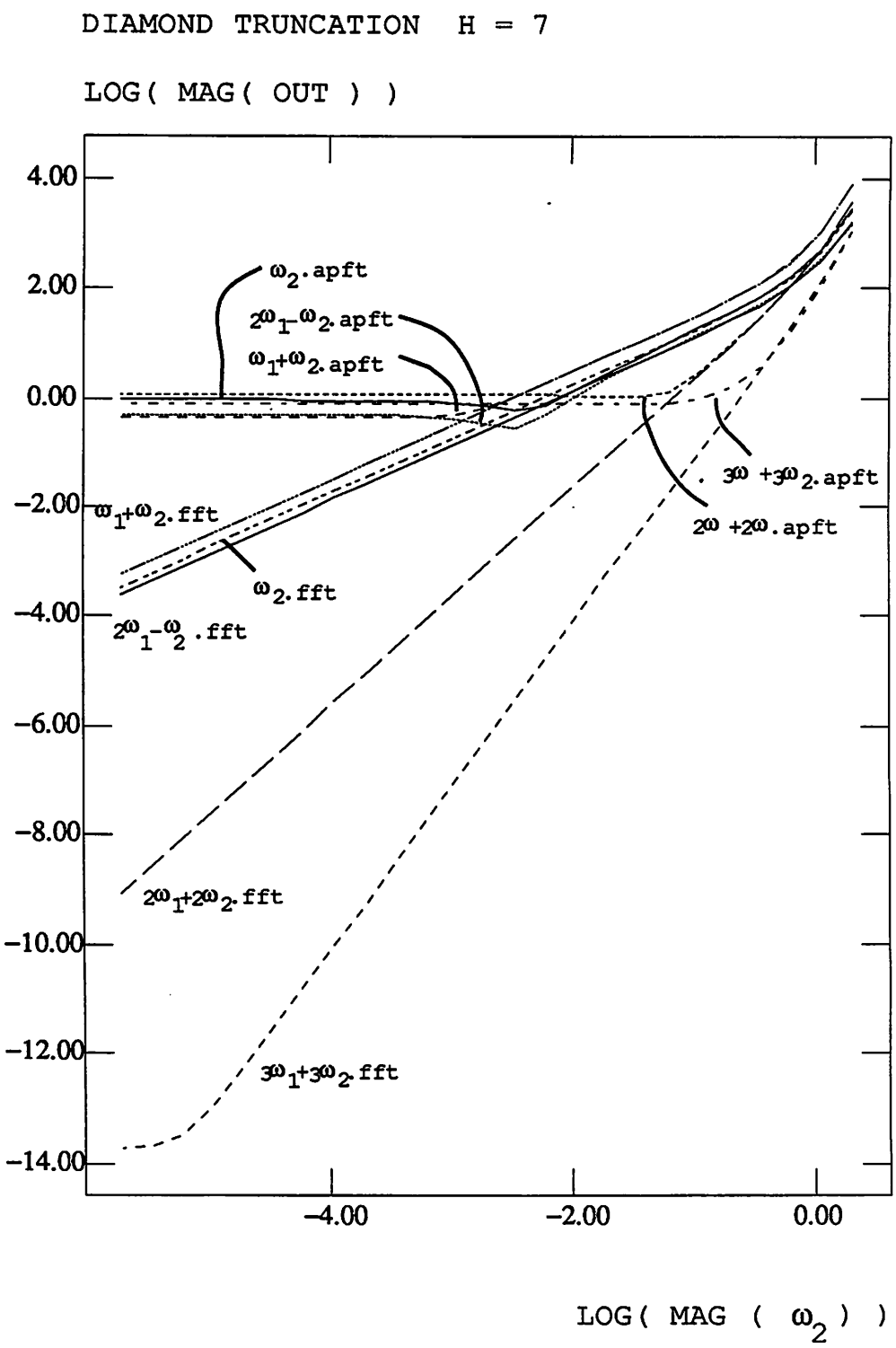


Figure 2.7: Dynamic range plots of the polynomial conductor circuit.

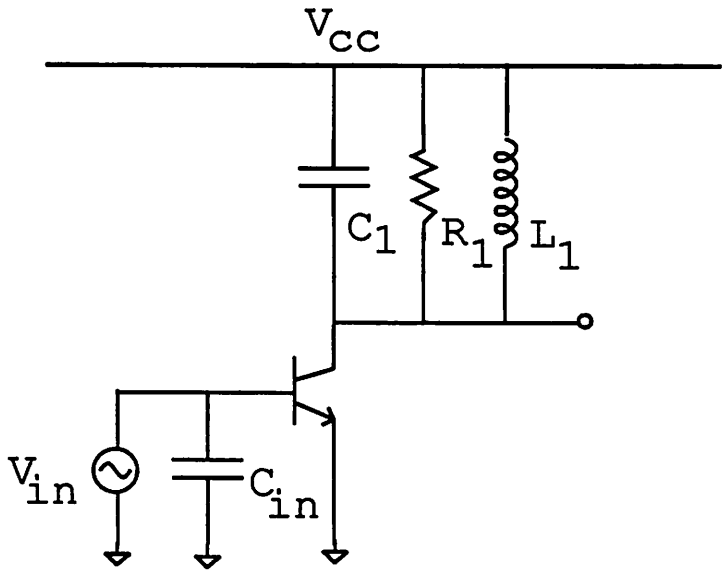


Figure 2.8: Test circuit 2. BJT mixer circuit.

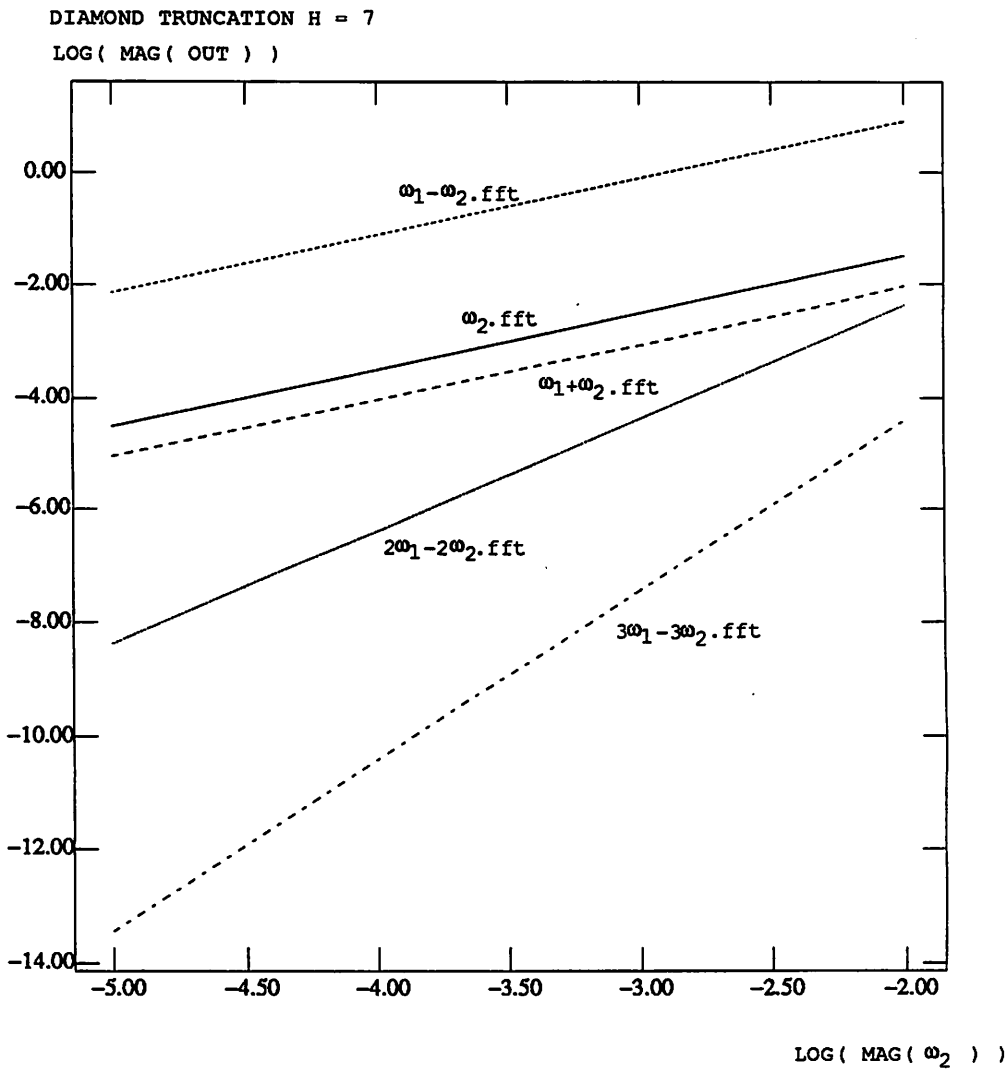


Figure 2.9: Dynamic range plots of the bjt mixer circuit.

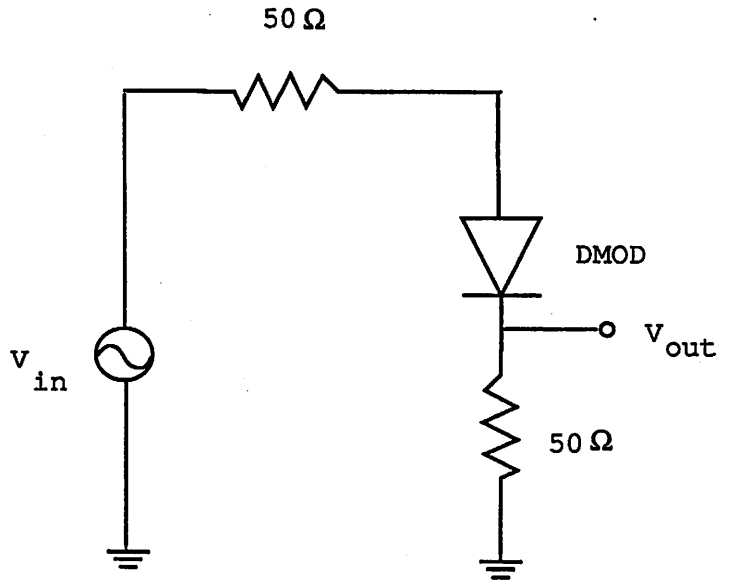


Figure 2.10: Test Circuit 3. diode mixer circuit.

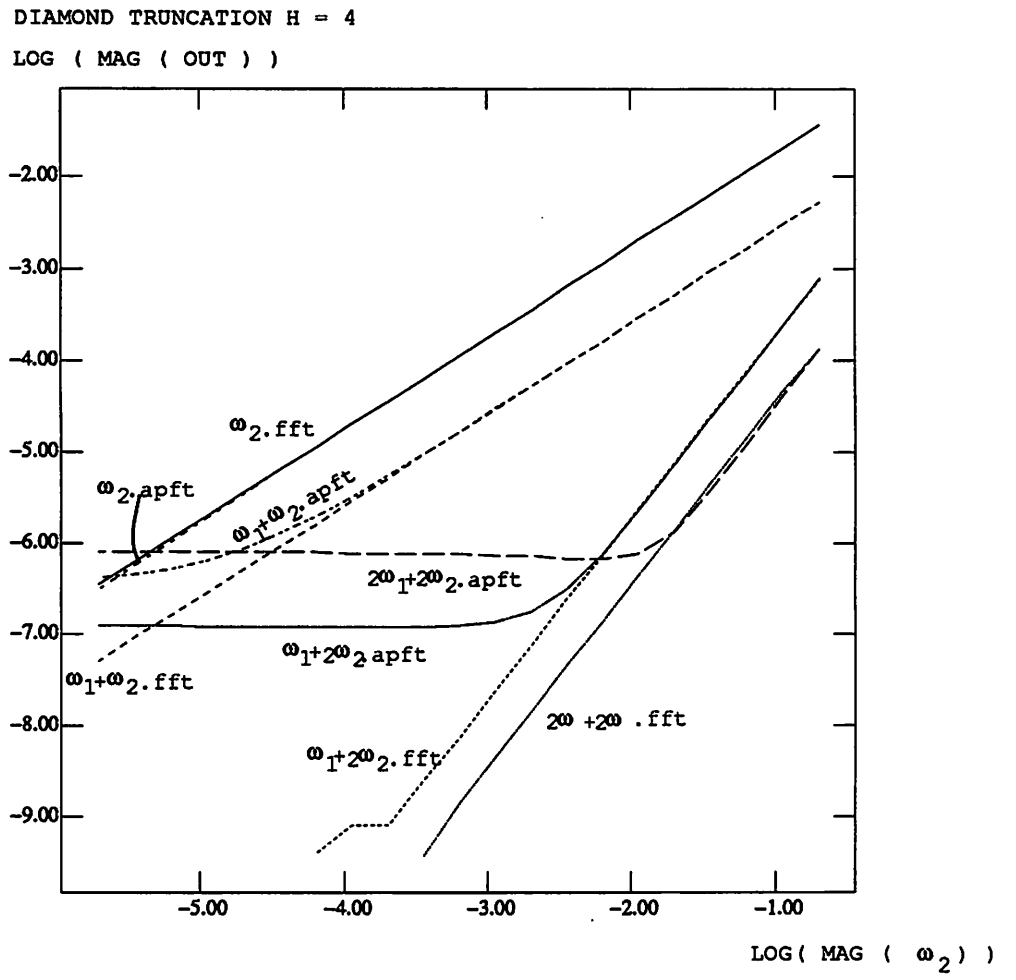


Figure 2.11: Dynamic range plots of the diode circuit with order 4 diamond truncation.

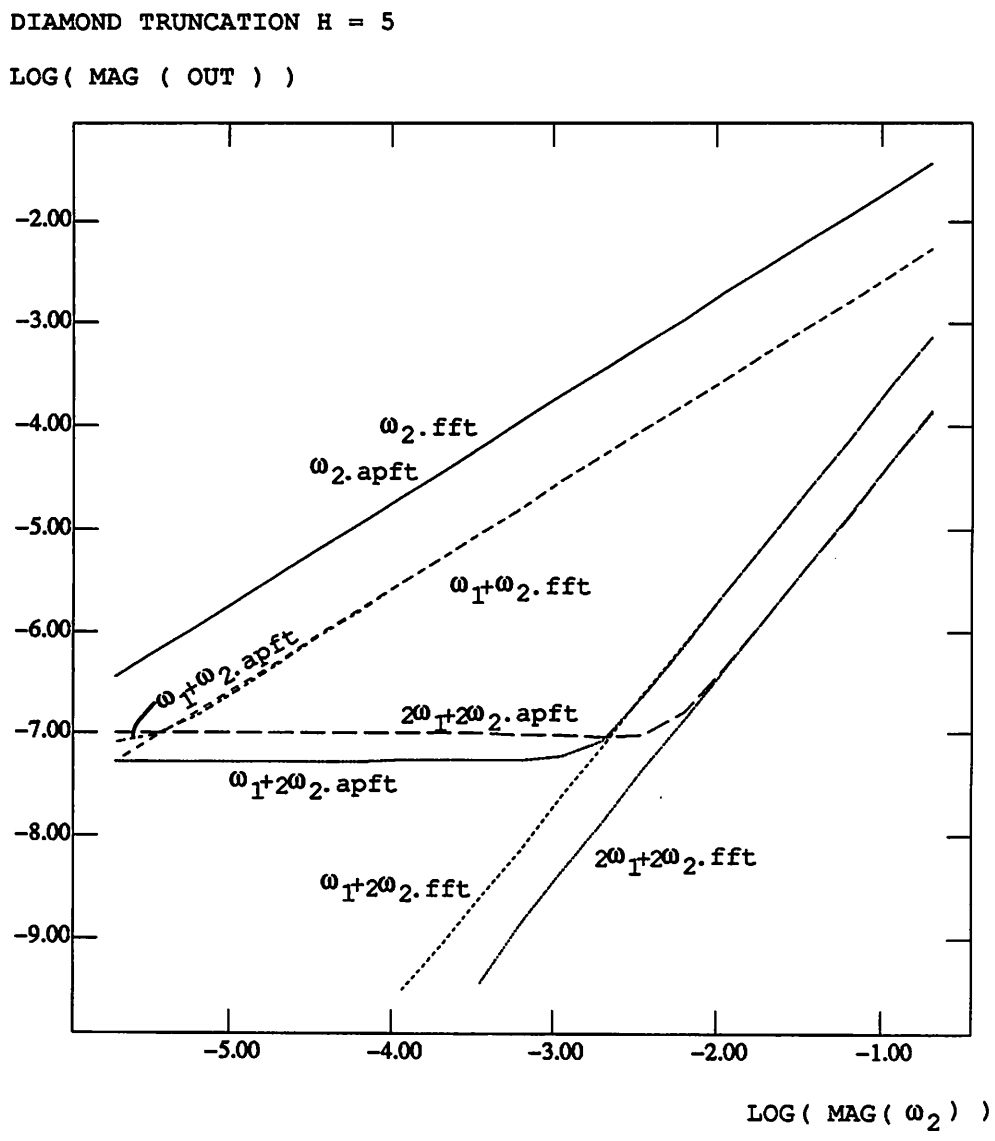


Figure 2.12: Dynamic range plots of the diode circuit with order 5 diamond truncation.

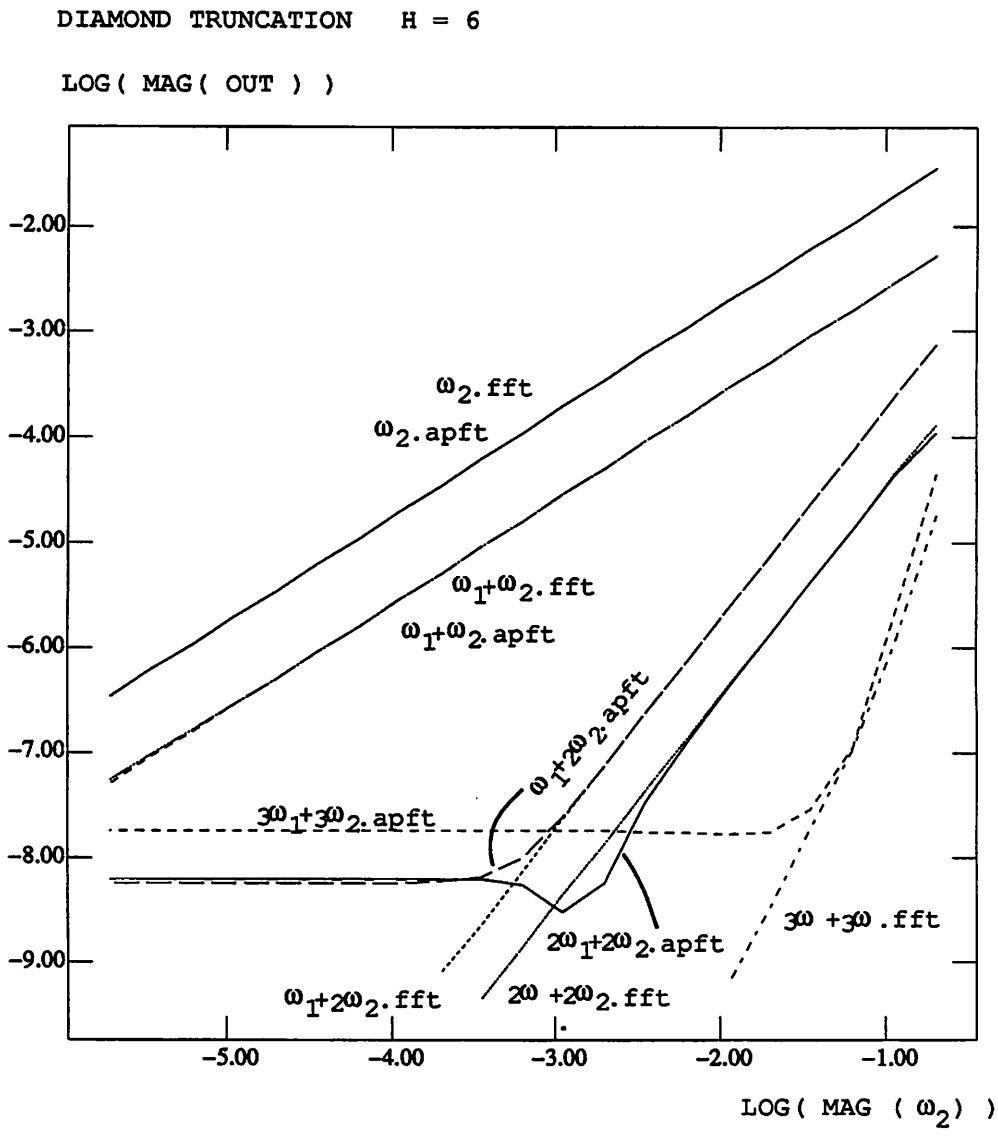


Figure 2.13: Dynamic range plots of the diode circuit with order 6 diamond truncation.

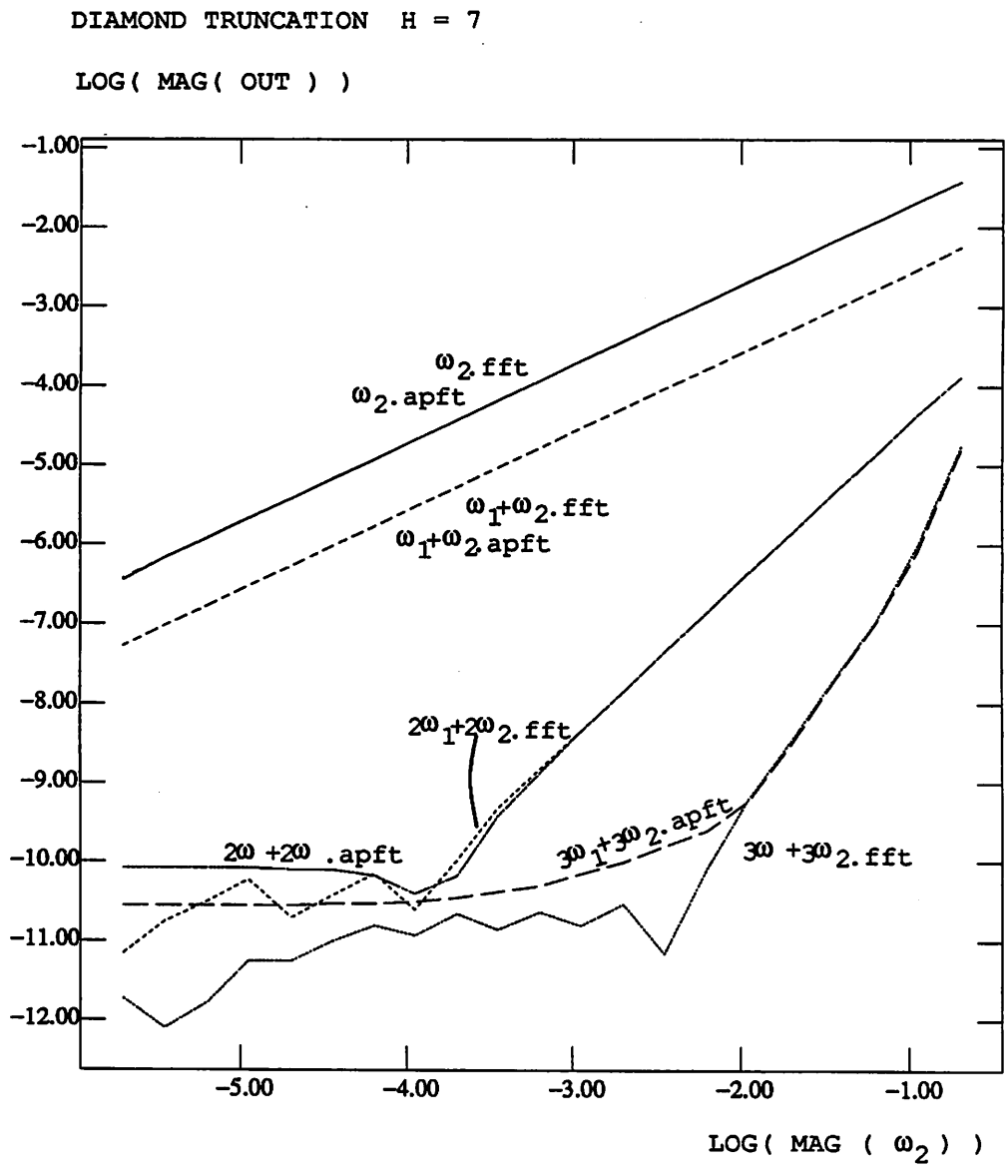


Figure 2.14: Dynamic range plots of the diode circuit with order 7 diamond truncation.

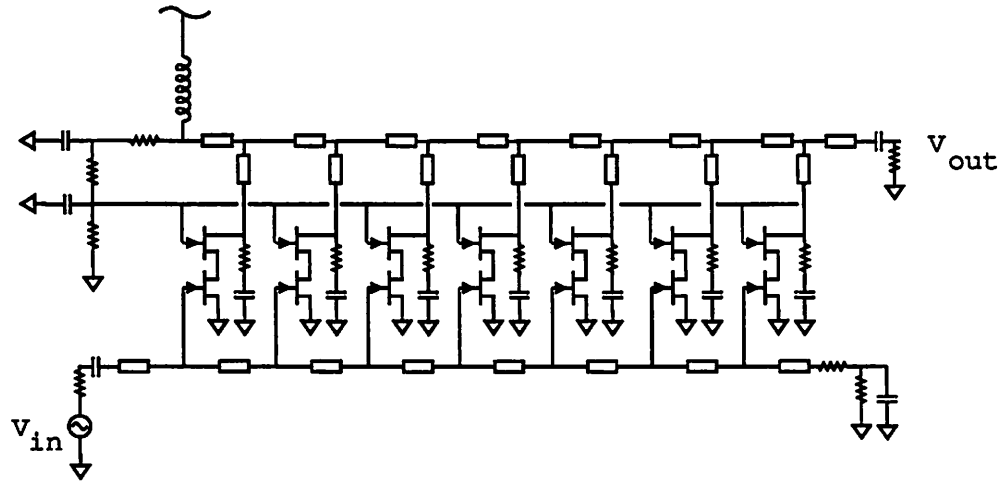


Figure 2.15: Test circuit 4. GaAsFET traveling wave amplifier.

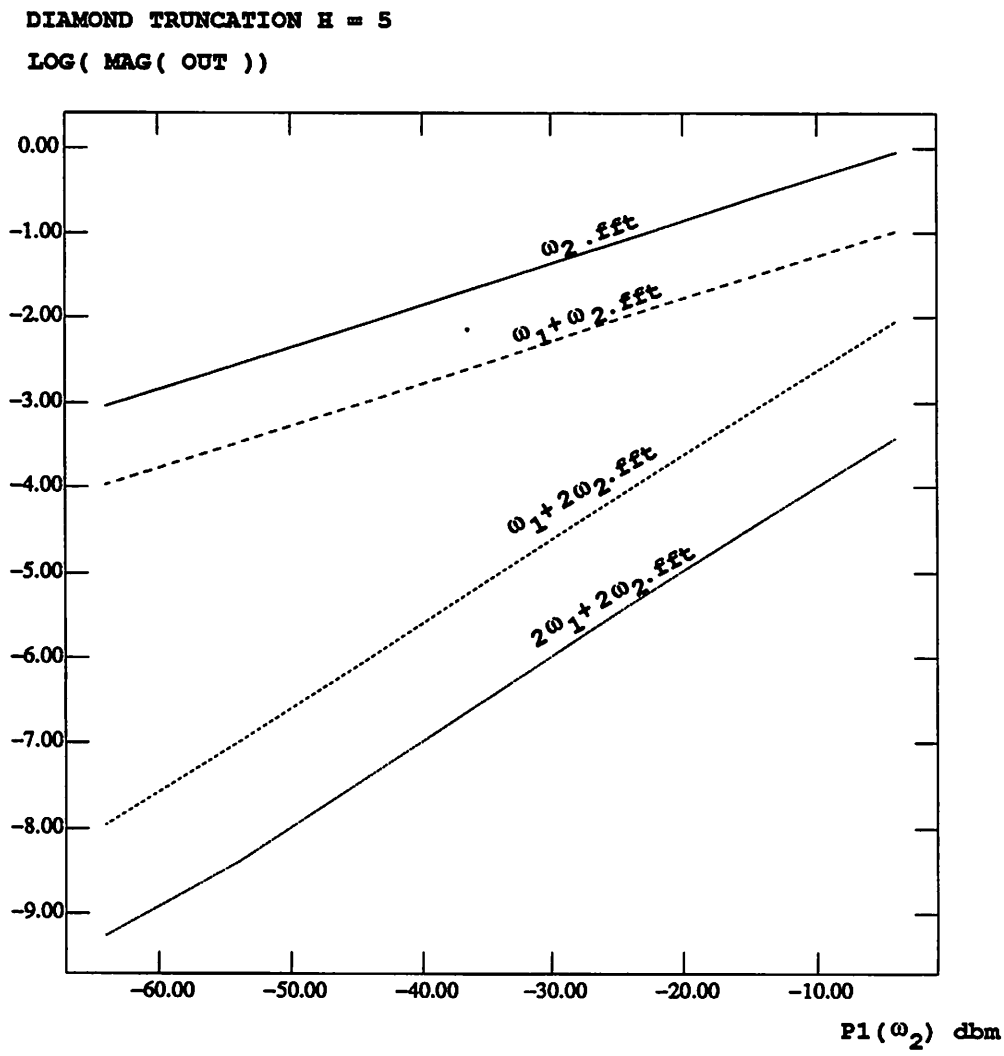


Figure 2.16: Dynamic range plots of the GaAsFET traveling wave amplifier circuit.

Chapter 3

Noise Analysis in Spectre

3.1 Introduction

The effect of electrical noise in circuits is important because it represents a lower limit to the size of an electrical signal that can be amplified by a circuit without significant degradation in signal quality. Therefore small-signal noise analysis is a useful capability for SPECTRE.

Various noise sources and device noise models[8][9] are described in section 3.2. Noise calculations and implementation details are presented in section 3.3. Finally, section 3.4 includes the results of a few test circuits.

3.2 Noise Sources and Device Noise Models

The noise phenomena considered here are caused by the small-current and voltage fluctuations that are generated within the devices themselves. In general the current and voltage fluctuations are specified in terms of their mean-square variations about the average values. They are written as $\overline{i^2}$ or $\overline{v^2}$. There are a number of sources of electrical noise. The noise sources that are implemented in this small-signal noise analysis are:

- Shot Noise: Shot noise is associated with a direct-current flow and is present in diodes and bipolar transistors. Shot noise has a mean- square value

$$\overline{i^2} = 2qI_D\Delta f \ [A^2] \tag{3.1}$$

where q is the electronic charge ($1.6 \times 10^{-19}C$), I_D is the average value of the direct current, and Δf is the bandwidth of the measurement (Hz).

- Thermal Noise: Thermal noise is due to the random thermal motion of electrons. In a resistor R, the thermal noise can be represented as a series voltage generator or as a shunt current generator. These representations are equivalent and they are

$$\overline{v^2} = 4kTR\Delta f \tag{3.2}$$

$$\overline{i^2} = 4kT\frac{1}{R}\Delta f \tag{3.3}$$

- Flicker Noise: Flicker noise is a type of noise that is found in all active devices as well as some discrete passive elements such as carbon resistors. Flicker noise is always associated with a flow of direct current and is represented as

$$\overline{i^2} = K_1\frac{I^a}{f^b}\Delta f \tag{3.4}$$

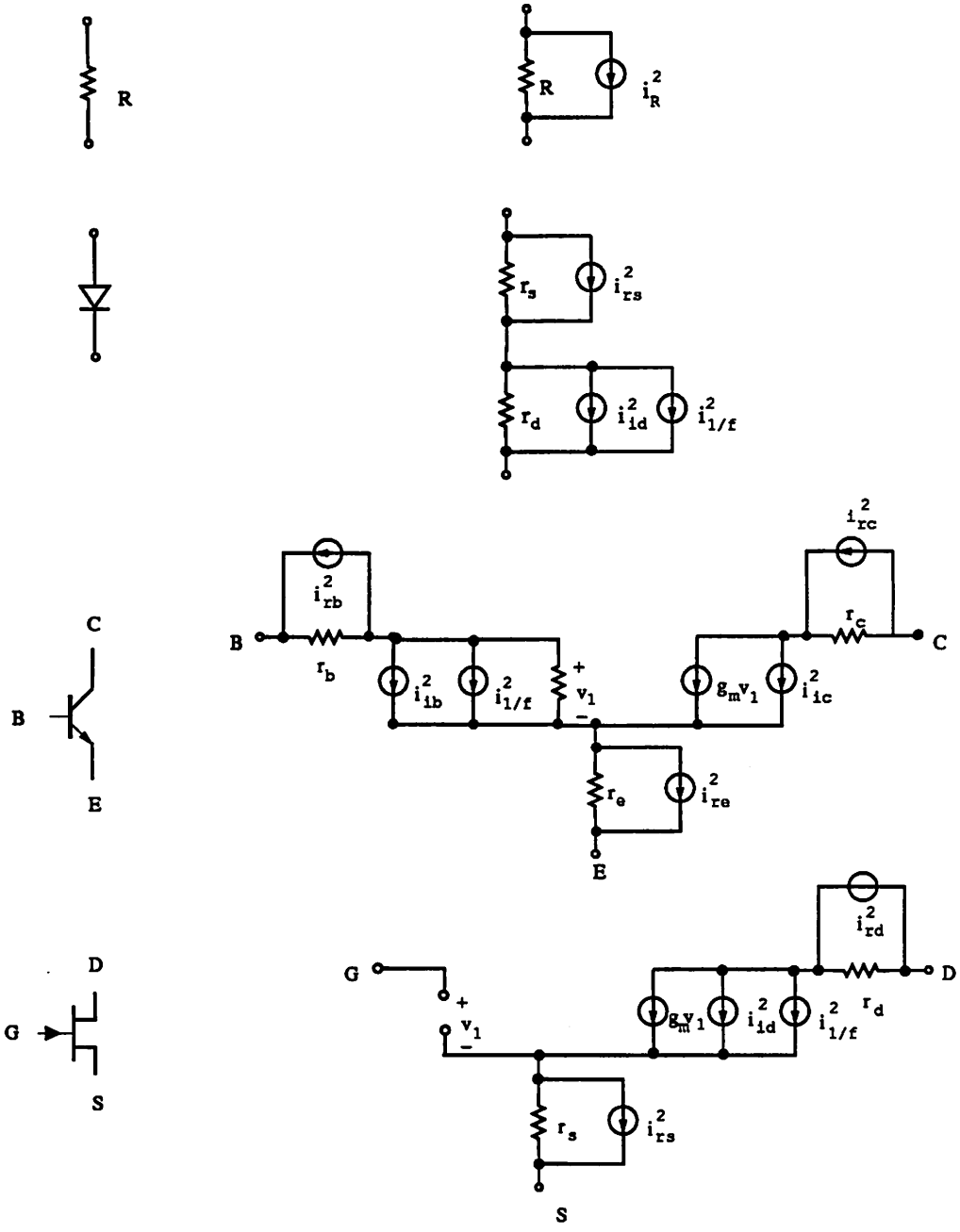
where Δf is a small bandwidth at frequency f, I is a direct current, K_1 is a constant for a particular device, a is a constant in the range 0.5 to 2, and b is a constant close to 1.

The implemented small-signal noise analysis supports noise models of diode, resistor, bipolar junction transistors and junction field effect transistors. Figure 3.1 shows the noise models for these devices.

3.3 Noise Calculations

For most applications, total output noise and input referred noise are more important than the individual device noise. The input referred noise can be calculated once the output noise is computed. Since the noise sources are uncorrelated, they can be calculated independently. The total output noise is then found by summing the noise contributions from all the noise sources. If we convert each noise source to its equivalent mean-square current noise, then the mean-square output noise voltage is given by

$$\overline{V_{out}^2} = \sum_{\text{all noise sources}} Z^2 \quad \overline{i^2} \tag{3.5}$$



Device Noise Models

where $\overline{i^2}$ is the individual mean-square current noise, and Z is the transimpedance from noise source to output. To compute $\overline{V_{out}^2}$, we need to calculate Z for each individual noise source.

Since the noise sources present in a circuit are small in value, a linearized small-signal analysis can be performed. the linearized current equations can be written as

$$Ye = I \tag{3.6}$$

where Y is the node admittance matrix, e is the vector consisting of all node voltages and possibly some branch currents, and I is the vector that contains current excitations at each node. We can also rewrite Equation 3.6 as

$$e = ZI \qquad \text{and} \tag{3.7}$$

$$Z = Y^{-1} \tag{3.8}$$

Z is called the node impedance matrix.

Assume all elements in the circuit are voltage controlled and there are n nodes in the circuit, excluding the ground node. Then

$$\begin{bmatrix} e_1 \\ \vdots \\ e_N \end{bmatrix} = \begin{bmatrix} Z_{11} & \cdots & Z_{1N} \\ \vdots & & \vdots \\ Z_{N1} & \cdots & Z_{NN} \end{bmatrix} \begin{bmatrix} i_1 \\ \vdots \\ i_N \end{bmatrix} \tag{3.9}$$

Suppose the output voltage is taken from node m and n. Then

$$\overline{V_{out}^2} = \overline{(e_m - e_n)^2} \tag{3.10}$$

$$= \overline{e_m^2} - \overline{e_n^2} \tag{3.11}$$

$$= \begin{bmatrix} (Z_{m1} - Z_{n1})^2 & \cdots & (Z_{mN} - Z_{nN})^2 \end{bmatrix} \begin{bmatrix} \overline{i_1^2} \\ \vdots \\ \overline{i_N^2} \end{bmatrix} \tag{3.12}$$

The only parameters in equation 3.12 that need to be solved for are the transimpedances, represented by the vector x. Taking the transpose of Z and multiplying by u, the vector of

all zeros but a 1 in position m and a -1 in position n, will provide us with what we need. That is,

$$x = Z^T u \tag{3.13}$$

$$\begin{bmatrix} (Z_{m1} - Z_{n1}) \\ \\ \vdots \\ \\ (Z_{mN} - Z_{nN}) \end{bmatrix} = \begin{bmatrix} Z_{11} & \cdots & Z_{m1} & \cdots & Z_{n1} & \cdots & Z_{N1} \\ \vdots & & \vdots & & \vdots & . & \vdots \\ Z_{1m} & \cdots & Z_{mm} & \cdots & Z_{nm} & \cdots & Z_{Nm} \\ \vdots & & \vdots & & \vdots & & \vdots \\ Z_{1n} & \cdots & Z_{mn} & \cdots & Z_{nn} & \cdots & Z_{Nn} \\ \vdots & & \vdots & & \vdots & & \vdots \\ Z_{1N} & \cdots & Z_{mN} & \cdots & Z_{nM} & \cdots & Z_{NN} \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ -1 \\ \vdots \\ 0 \end{bmatrix} \tag{3.14}$$

In general, we rarely solve for Y^{-1}, therefore Z, directly. However, we can rearrange equation 3.14 as

$$x = Z^T u \tag{3.15}$$

$$x = (Y^{-1})^T u \tag{3.16}$$

$$Y^T x = u \tag{3.17}$$

equation 3.17 is easy to solve using LU decomposition.

Once the mean-square output voltage noise is computed, the input referred noise can be calculated by dividing the output mean-square noise voltage with the square of the small-signal circuit gain.

While computing the total output noise of the circuit, the contribution from each individual device is stored and ranked. Therefore, the user can monitor the devices which contribute the most noise at the output.

3.4 Results

This section presents the results of a series of test circuits. Test circuits 1 through 3 are designed to test out individual device models. Test circuit 1, shown in Figure 3.1 is a circuit for testing diode noise model. Test circuit 2, shown in Figure 3.2 is a bipolar junction transistor test circuit, Figure 3.3 shows test circuit 3, which is a test circuit for junction

field effect transistor noise model. Test circuit 4 shown in Figure 3.4, is a current feed back pair circuit which is used as an input stage for a low noise transimpedance amplifier. The results of the above circuits agree with those generated by SPICE2.

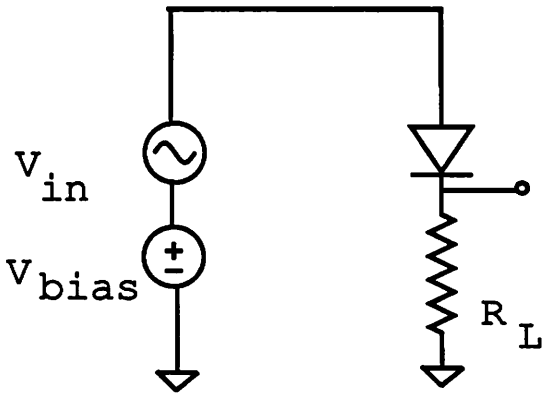


Figure 3.1: Test circuit 1. diode noise model test circuit.

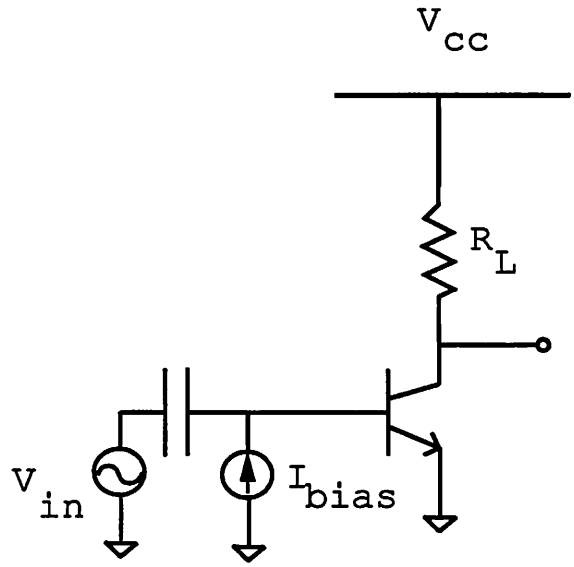


Figure 3.2: Test circuit 2. BJT noise model test circuit.

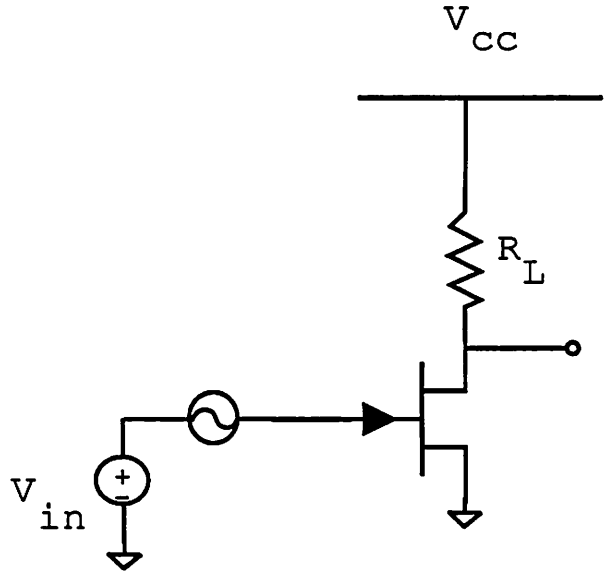


Figure 3.3: Test circuit 3. JFET noise test circuit.

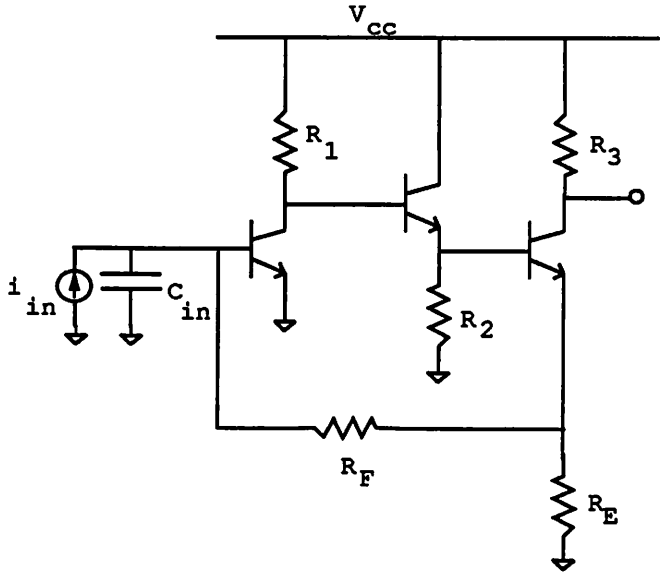


Figure 3.4: Test circuit 4. current feed back pair circuit.

Chapter 4

Conclusions

The DFT-based periodic steady-state analysis and the small-signal noise analysis have been successfully implemented in Spectre. A number of circuits have been tested to verify the correctness of the above implementations.

The results of DFT-based periodic steady-state analysis are compared with those of the APFT-based approach which has been used for Spectre. The DFT-based approach provides accurate results, faster computation time, more efficient memory usage, and better control of aliasing. The ability to exploit the regular aliasing pattern of DFT is the main reason why the above benefits can be achieved. In the implementation of the DFT-based steady-state analysis, a dual frequency set approach is used. That is, the set of analysis frequencies and the frequency set used by the Fourier transform are separated. The use of the dual frequency set analysis is first motivated for the use of the FFT algorithm because it generally requires a larger set of frequencies than the actual number of analysis frequencies needed. Keeping the number of analysis frequencies small saves the time spent in constructing and decomposing the Jacobian, and hence improves computation speed.

Since the noise sources in circuits are small in magnitude, a linearized small-signal analysis can be performed to analyze the effect of noise in a circuit. Noise models for resistors, BJT's, JFET's, and diodes are implemented. Results from the noise analysis are in close agreement with those from SPICE2 noise analysis . The results of Spectre's noise analysis can be interpreted by NUTMEG, a postprocessor. Given a circuit, the users can also isolate devices that contribute the most noise.

Bibliography

[1] Kenneth S. Kundert, "Steady-State Methods for Simulating Analog Circuits", Ph.D. Dissertation, Electronics Research Laboratory, University of California, Berkeley, April 28, 1989.

[2] Kenneth S. Kundert, and Alberto Sangiovanni-Vincentelli, "Simulation of Nonlinear Circuits in the Frequency Domain", *IEEE Transactions on Computer-Aided Design*, Vol. CAD-5, No. 4, October 1986, pp. 521-535.

[3] Kenneth S. Kundert, Gregory B. Sorkin, and Alberto Sangiovanni-Vincentelli, "Applying Harmonic Balance to Almost-Periodic Circuits", *IEEE Transactions on Microwave Theory and Techniques*, Vol. 36, No. 2, Febuary 1988, pp. 366-378.

[4] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes, The Art of Scientific Computing*, Cambridge University Press, 1986.

[5] D. Hente, and R.H. Jansen, "Frequency Domain Continuation Method for the Analysis and Stability Investigation of Nonlinear Microwave Circuits ", *IEE Proceedings*, Vol. 133. Pt. H, No. 5. October 1986.

[6] Chao-Ren Chang, and Michael Steer, "Frequency-Domain Nonlinear Microwave Circuit Simulation using the Arithmetic Operator Method ", *IEEE Transactions on Microwave Theory and Techniques*, Vol. 38, No. 8, August 1990, pp. 1139-1143.

[7] Patrick L. Heron, Chao-Ren Chang, and Michael B. Steer, "Control of Aliasing in the Harmonic Balance Simulation of Nonlinear Microwave Circuits", *1989 IEEE MTT-S Digest*. pp. 355-358.

[8] Paul. R. Gray, Robert. G. Meyer, *Analysis and Design of Analog Intergrated Circuits*, John Wiley & Sons, Inc., 1984.

[9] Gary W. Ng, "Noise Analysis with SPICE III", EECS 244 Project Report, University of California, Berkeley, December, 1989.