

4-30-1993

# A Robust High Precision Algorithm for Sinewave Parameter Estimation

Kendall Ann Rydell  
*Portland State University*

Follow this and additional works at: [https://pdxscholar.library.pdx.edu/open\\_access\\_etds](https://pdxscholar.library.pdx.edu/open_access_etds)



Part of the [Electrical and Computer Engineering Commons](#)

Let us know how access to this document benefits you.

---

## Recommended Citation

Rydell, Kendall Ann, "A Robust High Precision Algorithm for Sinewave Parameter Estimation" (1993).  
*Dissertations and Theses*. Paper 4685.  
<https://doi.org/10.15760/etd.6569>

This Thesis is brought to you for free and open access. It has been accepted for inclusion in Dissertations and Theses by an authorized administrator of PDXScholar. Please contact us if we can make this document more accessible: [pdxscholar@pdx.edu](mailto:pdxscholar@pdx.edu).

AN ABSTRACT OF THE THESIS OF Kendall Ann Rydell for the Master of Science in Electrical and Computer Engineering presented April 30, 1993.

Title: A Robust High Precision Algorithm for Sinewave Parameter Estimation.

APPROVED BY THE MEMBERS OF THE THESIS COMMITTEE:

[REDACTED]

*Y. C. Jenq*/Chair

[REDACTED]

W. Robert Daasch

[REDACTED]

*Eugene A. Enneking*

The estimation of sinewave parameters has many practical applications in test and data processing systems. Measuring the effective bits of an analog-to-digital converter and linear circuit identification are some typical examples. If a sinewave's frequency is known, there is an established linear method to estimate the other parameters. But when none of the parameters are known (which is usually the case in practical situations), the estimation problem becomes more difficult.

Traditional approaches to this task applied an iterative, sinewave curve-fit algorithm. Two problems with this technique are that convergence is often slow and not always guaranteed and the results of different trials may be inconsistent due to trapping at a local minimum.

Recently, a non-iterative algorithm has been developed which computes all four sinewave parameters directly. The algorithm combines a nonlinear technique and windowing to compute the estimates. Although this method is faster and more consistent than the curve-fit approach, one disadvantage is that the accuracy of some estimates tends to deteriorate rapidly if the sinusoid is corrupted by a high level of noise distortion.

This study presents an improved algorithm to extract the four parameters of an unknown sinusoid from a sampled data record even though the samples may be distorted by a high level of noise. Given this record, the proposed method first computes the FFT (Fast Fourier Transform) of the data. Analysis of the resulting frequency spectrum provides a rough estimate of the sinewave's fundamental frequency. Next, a bandpass filter designed around this frequency is used to eliminate much of the noise from the samples. Applying the existing four-parameter estimation algorithm to the filtered data, yields a more accurate frequency estimate. Finally, this new value, together with the original (noisy) data record is input to the three-parameter estimation algorithm to determine the remaining sinewave parameters.

Simulation results indicate this proposed (new) algorithm not only shows substantial improvement in the accuracy of parameter estimates, but also produces consistent results for higher levels of noise distortion than previous methods have achieved.

A ROBUST HIGH PRECISION ALGORITHM FOR SINEWAVE  
PARAMETER ESTIMATION

by

KENDALL ANN RYDELL

A thesis submitted in partial fulfillment of the  
requirements for the degree of

MASTER OF SCIENCE  
in  
ELECTRICAL AND COMPUTER ENGINEERING

Portland State University  
1993

TO THE OFFICE OF GRADUATE STUDIES:

The members of the committee approve the thesis of Kendall Ann Rydell  
presented April 30, 1993.

[Redacted Signature]

Y.C. Jenq, Chair

[Redacted Signature]

W. Robert Daasch

[Redacted Signature]

Eugene A. Enneking

APPROVED:

[Redacted Signature]

Rolf Schaumann, Chair, Department of Electrical Engineering

[Redacted Signature]

Roy W. Koch, Vice Provost for Graduate Studies and Research

## TABLE OF CONTENTS

	PAGE
LIST OF TABLES.....	vii
LIST OF FIGURES.....	viii
CHAPTER	
I INTRODUCTION.....	1
Sinewave Parameter Estimation.....	1
Why Sinusoids.....	2
Degree of Precision.....	3
Degree of Robustness.....	3
Scope of Study.....	3
II THE PARAMETER ESTIMATION PROBLEM.....	5
Ideal ADC Operation.....	6
Non-Ideal Characteristics.....	7
Quantization Error.....	8
Noise Sources.....	10
III TRADITIONAL APPROACHES TO THE PROBLEM -- ITERATIVE ALGORITHMS.....	12
Curve Fitting.....	12
Advantages.....	14

	Disadvantages.....	14
	A Recent Solution.....	14
IV	A METHODOLOGY TO COMPARE THE ALGORITHMS.....	15
	The Test Sinewave.....	15
	Sampling .....	17
	Noise Models.....	18
	Linear Noise	
	Nonlinear Noise	
	Average Power	
	Noise Levels	
	Experimental Procedure.....	23
	Software	
	Trials	
	Error Analysis	
	Summary.....	25
V	THE FOUR-PARAMETER ALGORITHM.....	26
	Estimating Amplitude and DC Offset.....	26
	Intuitive	
	Derivation	
	Estimating Frequency and Phase Angle.....	33
	Intuitive	
	Derivation	
	Simulations.....	40
	Linear Simulations.....	41
	Linear Results.....	43
	Nonlinear Simulations.....	44
	Nonlinear Results.....	46
	Conclusions.....	47

VI	THE THREE-PARAMETER ALGORITHM.....	48
	The Process.....	48
	Algorithm Performance.....	51
	The Experiments	
	Sensitivity to Linear Distortion	
	Linear Results	
	Sensitivity to Input Frequency Variation	
	Results	
	Conclusions .....	54
VII	THE PROPOSED ALGORITHM.....	56
	Filter Design and Application.....	56
	The Process.....	65
	Simulations.....	68
	Linear Simulations	
	Linear Results	
	Nonlinear Simulations	
	Nonlinear Results	
	Conclusions.....	72
VIII	COMPARISON OF ALGORITHMS.....	73
	Linear Distortion .....	73
	Comparisons .....	74
	Nonlinear Distortion.....	77
	Comparisons.....	77
	Wrapping Up.....	78
IX	CONCLUSIONS.....	82
	Summary of Results.....	83
	Topics for Further Study.....	85



REFERENCES.....	87
-----------------	----

## APPENDICES

A FOUR-PARAMETER ALGORITHM PERFORMANCE.....	88
B THREE-PARAMETER ALGORITHM PERFORMANCE.....	102
C PROPOSED (NEW) ALGORITHM PERFORMANCE.....	109

## LIST OF TABLES

TABLE	PAGE
I Comparing Signal Strengths: Sine and White Noise.....	22
II Comparing Signal Strengths: Sine and Harmonics.....	22
III Phase Unwrapping.....	37
IV Incremental Frequency Variations.....	53

## LIST OF FIGURES

FIGURE	PAGE
1. Block diagram of a basic ADC (analog-to-digital converter).....	6
2. Transfer characteristic of an ideal 3-bit ADC.....	7
3. Transfer characteristics of practical ADCs.....	9
4. Modeling distorting effects of a practical ADC.....	10
5. Analog sinewave.....	16
6. PDF for a uniform random variable.....	20
7. Frequency spectrum of typical input sinewave.....	28
8. Frequency domain characteristics of the rectangular window.....	30
9. Time domain characteristics of several common windows.....	30
10. Frequency-domain characteristics of several common windows.....	31
11. Magnitude spectrum of the Blackman-Harris window.....	32
12. Windowing the sinewave spectrum.....	33
13. Estimating frequency and phase.....	34
14. Phase Unwrapping.....	35
15. Regions of a sinewave cycle.....	37
16. Plot of unwrapped phase angles.....	38
17. Modeling linear distortion in the test sinewave.....	42
18. Modeling nonlinear (harmonic) distortion in the test sinewave.....	45
19. Frequency spectrum of the basic test sinewave.....	57
20. Characteristics of some common frequency selective filters.....	58
21. Frequency spectrum of a practical test sinewave (including distortion).....	59

22. Magnitude (upper) and impulse (lower) responses for (a) 16th order and (b) 8th order Butterworth bandpass filters.....	60
23. Magnitude (upper) and impulse (lower) responses for (a) 4th order and (b) 2nd order Butterworth bandpass filters.....	61
24. Magnitude (upper) and impulse (lower) responses for a bandpass filter with a bandwidth of (a) $\pm 0.01$ and (b) $\pm 0.1$ .....	63
25. Magnitude (upper) and impulse (lower) responses for a bandpass filter with a bandwidth of (a) $\pm 1.0$ and (b) $\pm 10$ .....	64
26. Magnitude (upper) and impulse (lower) responses for a bandpass filter with a bandwidth of $\pm 100$ .....	65
27. Magnitude (a) and impulse (b) responses of a fourth order Butterworth bandpass filter with bandwidth = 20.....	66
28. Frequency spectrum of (a) the original noisy sinewave, (b) the filtered sinewave and (c) the result after truncating transients.....	67
29. Process flowchart for the proposed algorithm.....	69
30. Comparison of amplitude estimation errors -- linear distortion.....	75
31. Comparison of dc offset estimation errors -- linear distortion.....	75
32. Comparison of frequency estimation errors -- linear distortion.....	76
33. Comparison of phase estimation errors -- linear distortion.....	76
34. Comparison of amplitude estimation errors -- nonlinear distortion.....	79
35. Comparison of dc offset estimation errors -- nonlinear distortion.....	79
36. Comparison of frequency estimation errors -- nonlinear distortion.....	80
37. Comparison of phase estimation errors -- nonlinear distortion.....	81

## CHAPTER I

### INTRODUCTION

#### SINEWAVE PARAMETER ESTIMATION

Estimating the parameters of a sinewave is a basic function in many test and data processing systems. From an accurate knowledge of the values of these four parameters: amplitude, dc offset, frequency, and phase angle, the original sinewave can be completely reconstructed. Because many signal processing tasks are carried out in the digital domain to take advantage of the power and speed of digital computers, most of the "waveforms" dealt with in practice are merely records of digitized data values. Therefore, parameter estimation is often required to extract the actual waveform characteristics from the available data record.

Circuit identification is one important application of parameter estimation. The goal of this process is to identify the unique transfer function which correctly describes a circuit's performance. For "linear" circuits, this task is somewhat simplified because of a special property they possess. A linear circuit can be completely characterized by inputting a sinewave and measuring the parameters of the output signal. However, if only a record of samples is available for the output, the parameters must be "estimated." Any differences between the known input sinewave and the measured output reflect the circuit's impact on the data and determine its unique transfer function.

Parameter estimation is also involved in measuring the effective bits of a waveform digitizer or ADC (analog to digital converter). The number of effective bits is a figure of merit which indicates how much an ADC's nonlinearity has impaired its usefulness at a given frequency [1]. This value measures the average noise power introduced during the

digitizing process and is used to determine the actual resolution of the ADC. A commonly used definition of effective bits is [4, 5]:

$$B = N - \log_2 \left( \frac{\text{RMSE}}{\text{Ideal Noise}} \right) = \log_2 \left( \frac{\text{FullScale}}{\sqrt{12} \cdot \text{RMSE}} \right) \quad (1.1)$$

where  $B$  = the number of effective bits,  $N$  = bits of the digitizer,  $\text{FullScale}$  = the digitizer's full scale value and  $\text{RMSE}$  = root mean square error of the digitized signal. In practice, one has only the digitized sinewave record (and not the original input signal) to work with. Therefore, some type of parameter estimation algorithm is required. Once this process is completed, the *estimated* sinewave is used (as if it were the *actual* sinewave) in (1.1), to compute the effective bits of the digitizer. The accuracy of this calculation depends on how closely the sinewave can be reconstructed from estimates of its four parameters.

## WHY SINUSOIDS

This study focuses on the process of estimating the four parameters of a sinusoid: amplitude, dc offset, frequency and phase angle. Sinewaves are emphasized because of their importance in test and measurement systems. Signal processing applications commonly use sinusoids as their stimulus. This is due to several unique qualities these waveforms possess. One primary factor is that sinewaves are relatively easy to generate in practice at the frequencies of interest with adequate fidelity. Also, sinewaves have a simple mathematical model which simplifies the algorithms used for data analysis. Another advantage, is that many other signals can be broken into a sum of sinusoids and worked with in terms of these simpler components. Finally, sinusoids are mathematically related to complex exponentials (phasors) which makes them important in frequency domain analysis. For these reasons, parameter estimation is targeted toward sinusoids for most practical applications.

## DEGREE OF PRECISION

With the continual advances in electronic technology, the precision of digital processing systems is steadily increasing. To be of any use in the previous applications mentioned, the accuracy of the sine parameter estimation algorithm must also keep pace. For example, in calculating the number of effective bits of an ADC, the goal is to determine the average noise power introduced by the converter. For this application, any error resulting from the parameter estimation process must be considerably smaller than that introduced by the converter in order not to distort the results. Considering that 12-bit ADCs (with corresponding resolution down to 0.0002) are not uncommon, estimation algorithms must have a high degree of precision.

## DEGREE OF ROBUSTNESS

In addition to being very accurate, modern applications demand an estimation algorithm that is able to perform reliably in the presence of distortion as well. Most of the digital data available for analysis is acquired through some form of sampling. This process inevitably introduces a degree of quantization error which distorts the original data. Given this distorted version of the data as input, a parameter estimation algorithm must be able to extract, as accurately as possible, the true parameters of the original analog signal.

## SCOPE OF STUDY

The primary purpose of this study is to develop an improved algorithm for sinewave parameter estimation. In the course of this development, two previously derived algorithms will be examined in some depth, based on the success of their approaches, to discover potential areas for improvement. Numerous simulations, to represent a range of input conditions, will be used to characterize their performance. Any weakness revealed

by the results of these tests will assist in forming a hypothesis for improvement. After the proposed algorithm has been described, it will be tested with simulations identical to those applied to the previous methods. The results will provide an empirical comparison of estimation accuracy. The following chapter presents a more detailed description of the actual parameter estimation problem.



## CHAPTER II

### THE PARAMETER ESTIMATION PROBLEM

A sinewave can be completely characterized by its four parameters: amplitude (A), dc offset (D), frequency (f), and phase angle ( $\theta$ ). Mathematically, it is usually represented as a continuous function of time by:

$$s(t) = A \cdot \sin(2 \pi f t + \theta) + D \quad (2.1)$$

However, in most *practical* parameter estimation problems, the original analog waveform is neither accessible nor convenient for analysis. Instead, a record of digital values, obtained by sampling and quantizing the continuous sinewave, is more commonly dealt with.

$$S_k = s(k/f_s) = A \cdot \sin(k2\pi f/f_s + \theta) + D, \quad k = 0, 1, \dots, N-1 \quad (2.2)$$

where  $k$  = sequence index

$f$  = fundamental frequency of the sinewave

$f_s$  = system sampling rate

$N$  = number of data points in the record

This type of record represents the typical input to a parameter estimation algorithm. The data it contains may be collected from a variety of signal processing sources -- one of the most likely being some type of waveform digitizer, or ADC. Therefore, to gain a better understanding of the data that the estimation algorithm will have to utilize, it is useful to explore the basic characteristics of ADC operation.

## IDEAL ADC OPERATION

A block diagram of the circuit for an ideal analog to digital converter (ADC) is shown in Figure 1.

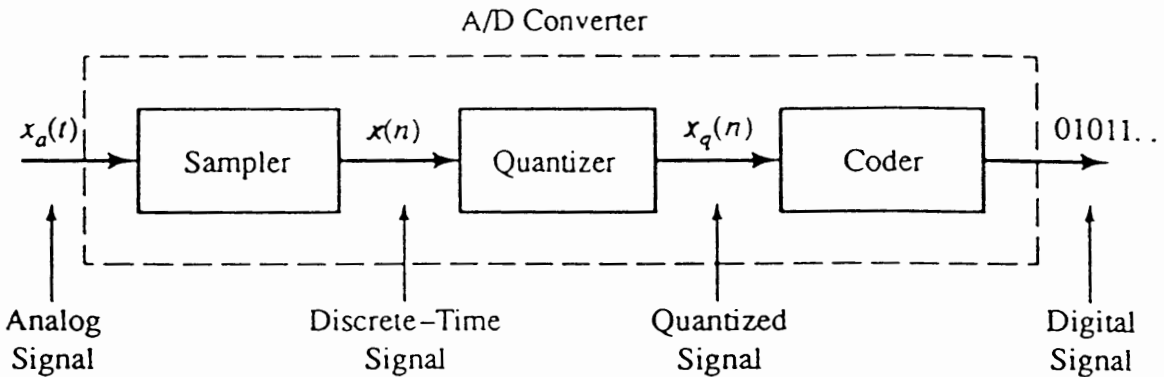


Figure 1. Block diagram of a basic ADC (analog-to-digital converter).

By standard definition, an analog signal is defined at every instant in time and can assume an infinite number of values within any continuous range specified. Mathematically, such signals can be described by functions of a continuous variable (typically "time"). A digital signal, by contrast, is defined only at discrete instants in time and can assume only one of a finite set of discrete values. Digital signals are usually represented by a sequence of binary values of finite word length. The purpose of analog to digital conversion is to generate a sequence of binary values which adequately represents the original analog input.

To accomplish this task, the analog input signal is first sampled at (ideally) uniform time intervals. Sampling consists of monitoring the analog waveform at specified intervals ( $t_{\text{sample}}$ ) and recording the value of the waveform at each instant. Then, the sampled data is quantized. The "quantizer" takes the sampled values and maps them onto one of a finite set of digital values to generate a sequence of output data points. The transfer function of a 3-bit ADC, shown in Figure 2, illustrates the ideal staircase-shaped performance

characteristic. For an ideal ADC, connecting the mid-points of the staircase, should produce a straight line passing through the origin with a 45 degree angle. This requires both a linear transfer characteristic and uniform transition levels ( $\Delta$ ).

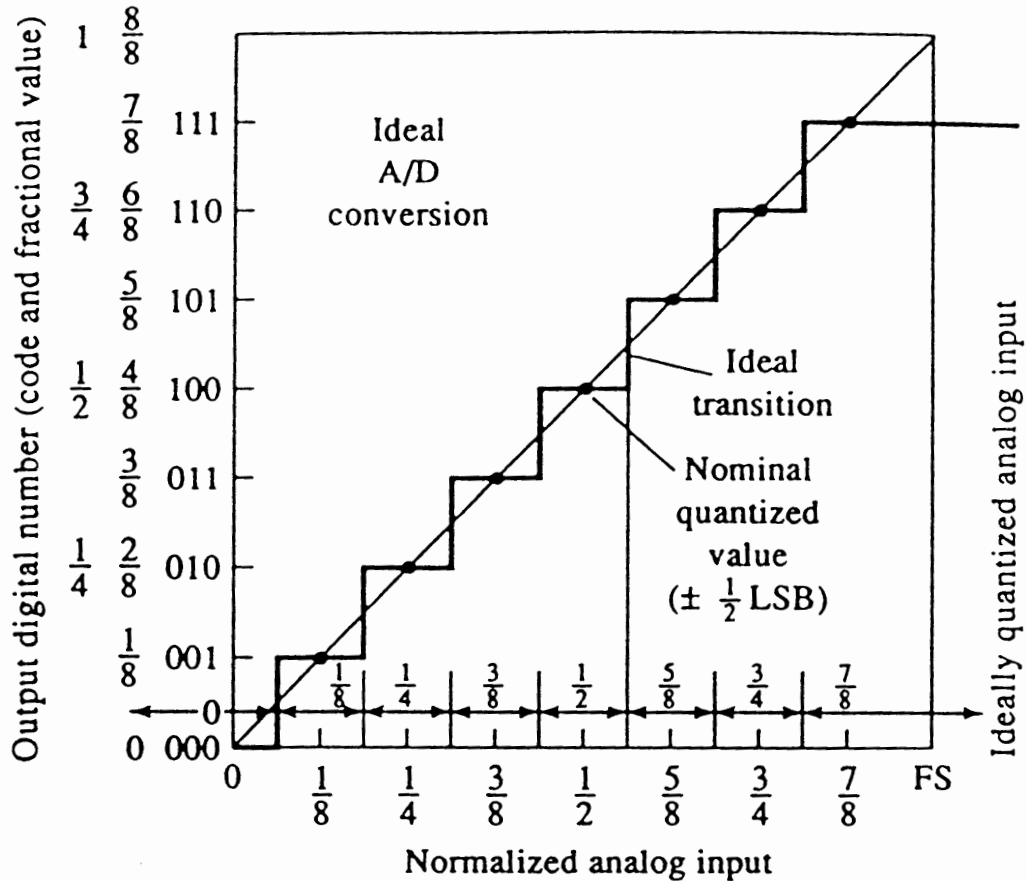


Figure 2. Transfer characteristic of an ideal 3-bit ADC.

After the conversion process is completed, assuming a sampling rate of  $f_s$  (samples/sec) =  $1/t_{\text{sample}}$  and a record length of  $N$  data points, the resulting output signal will have a form similar to the sequence described by (2.2).

### NON-IDEAL CHARACTERISTICS

Up to this point, the discussion has focused on the "ideal" analog-to-digital conversion process (i.e. assuming a perfectly linear transfer characteristic and uniform digital

transition levels or steps). Therefore, the output sequence generated is free of any distortion except that which is inherent in the quantization process itself. However "practical" ADCs are subject to several types of errors that can distort the original input signal even further. For a more realistic example of the type of data input to a parameter estimation algorithm, it is necessary to look at some characteristics of practical ADCs.

Transfer characteristics of several practical ADCs are shown Figure 3. Each exhibits a different type of error which causes distortion to the original data. As mentioned previously, a straight line with a 45 degree angle passing through the origin should intersect the mid-points of the staircase transfer function of a *perfect* ADC. Any deviation from this ideal indicates the presence of some sort of error in the ADC. For example, if the line does not pass through the origin, we say that the ADC has an "offset" error. Offset error occurs when the digital signal does not match the analog signal by a fixed offset value. If the angle of the line is not 45 degrees, then we say the ADC has a "gain" error. Gain error occurs when the output reaches saturation either sooner or later than it should. If the transfer function is not a straight line at all, then we say that the ADC has a nonlinearity error. This type of error can occur if the quantizer levels are not uniformly spaced, which means that all of the  $\Delta$ 's are not equal.

## QUANTIZATION ERROR

Even if none of the specific error types described above are present, the digital output signal produced by the ADC is not a perfect representation of the analog input. When an analog sample is converted to a digital word, a small amount of error is introduced as the continuous input value is "quantized" or mapped onto one of a discrete set of digital values. The distortion introduced by this process is called "quantization error" or "quantization noise." Assuming *FS* refers to the full scale (or maximum expected)

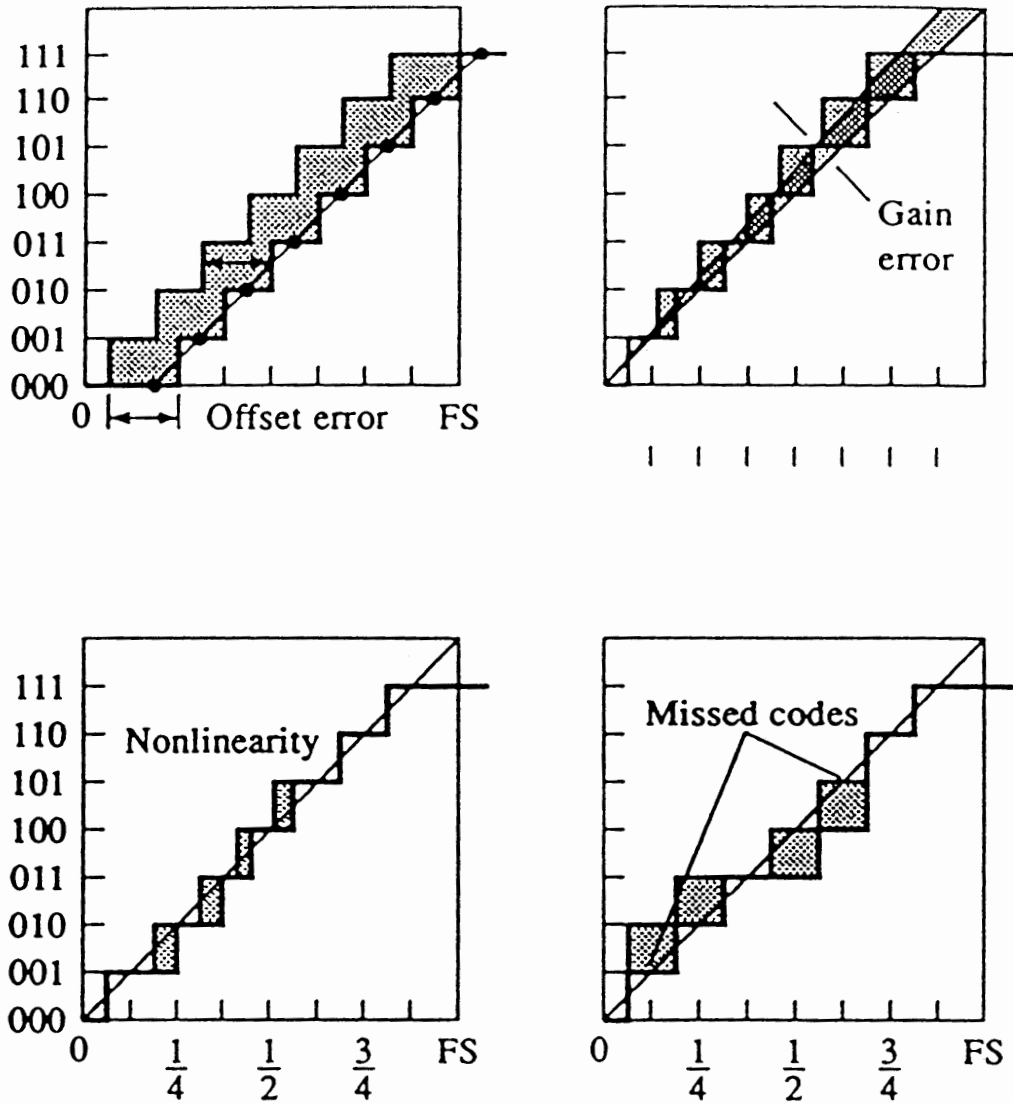


Figure 3. Transfer characteristics of practical ADCs.

value of the input signal and  $n$  to the number of bits of the ADC, the amount of quantization error ( $e_q$ ) will be in the range of  $+\frac{\Delta}{2} \leq e_q \leq -\frac{\Delta}{2}$ , where  $\Delta = \frac{FS}{2^n}$ . If it is assumed that the quantizer is both ideal and uniform (all of the  $\Delta$ 's are equal) and that the input signals are bounded by the valid input range of the quantizer, then all of the noise sources can be eliminated except quantization noise.

## NOISE SOURCES

The most common method used in the analysis of distorting effects of an ADC is to model these effects (such as quantization error) with an additive noise source. Figure 4 illustrates the basic model.

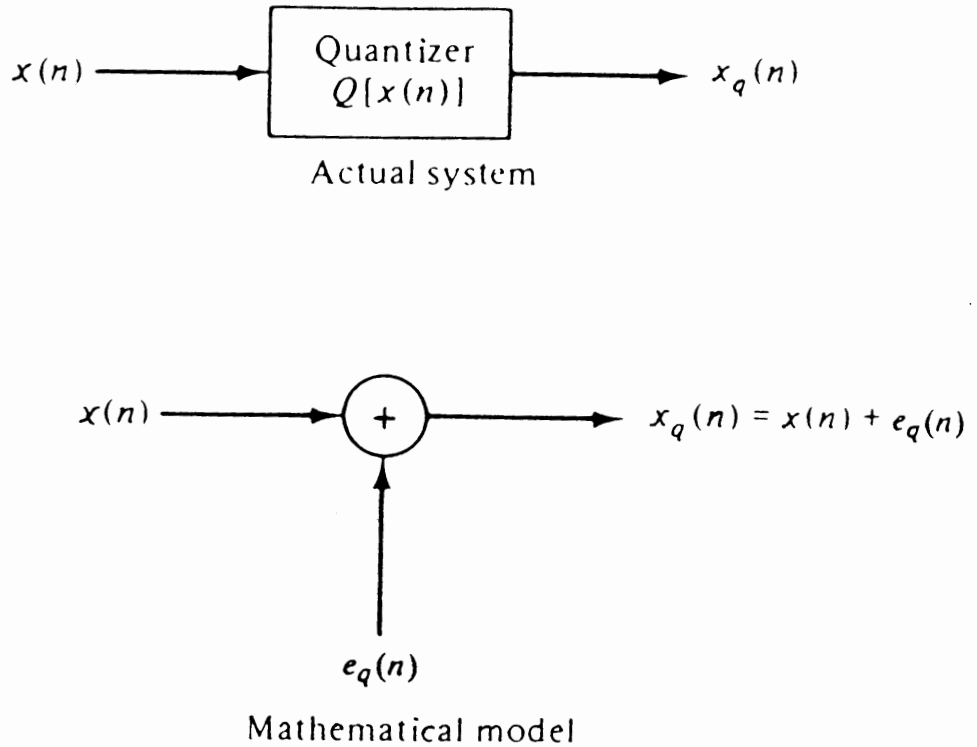


Figure 4. Modeling distorting effects of a practical ADC.

This model can be described mathematically by the following equation:

$$y(n) = x(n) + e(n) \quad (2.3)$$

The model reflects how each sample of the input sinewave is distorted by the analog-to-digital conversion process. The error signal ( $e$ ) can represent the effect of several practical transfer characteristics.

Since the purpose of this study is to determine an improved method for parameter estimation and not to examine details of ADC behavior, the principle types of distortion

introduced by ADCs will be represented by one of two broad categories -- linear or non-linear. "Linear" includes the effects of quantization error, offset error, and gain error (discussed earlier). "Nonlinear" represents both integral and differential nonlinearities, such as those which occur when transition levels are not uniform. For modeling purposes in this study, each of the two broad categories of distortion will be treated as a separate type of additive noise source.

After accounting for these errors introduced by the process of analog-to-digital conversion, the actual input data available to a sinewave parameter estimation algorithm generally resembles either:

$$y[n] = A \cdot \sin [2 \pi f n + \theta ] + D + e_{\text{lin}} [n] \quad (2.4)$$

when linear distortion ( $e_{\text{lin}}$ ) is present, or

$$y[n] = A \cdot \sin [2 \pi f n + \theta ] + D + a_h \cdot \sin ([ h 2 \pi f n ] + \theta_h [n]) \quad (2.5)$$

for nonlinear error sources,

where  $h$  = harmonic frequency (a multiple of the sinewave's fundamental frequency),  
 $a_h$  = amplitude of the harmonic and  $\theta_h$  = random phase angle of the harmonic.

After describing the estimation problem and clarifying the type of input data that is available to work with, the next step is to examine some previously-developed methods for extracting the sine parameters. Studying the existing methods can provide some insight to the most successful techniques and their expected level of performance. Because the sinewave has such a well-defined mathematical representation, traditional approaches to the parameter estimation problem have employed a curve-fitting technique.

## CHAPTER III

### TRADITIONAL APPROACHES TO THE PROBLEM -- ITERATIVE ALGORITHMS

Traditional attempts at sinewave parameter estimation usually resorted to some form of iterative algorithm. This technique was usually quite slow, due to its basic "trial-and-error" approach. In addition, the errors which occurred could not be expressed readily in closed form and therefore, were difficult to control.

### CURVE FITTING

A typical example of these earlier estimation algorithms is the so-called "curve-fit" method, described in [1]. This algorithm employs an iterative, "gradient search" technique to determine each of the sine parameters. The motivation behind this algorithm was finding a means to characterize the dynamic performance of an ADC. "The result of this [curve fit] test is a figure of merit called the number of effective bits for the ADC" [1].

The basic curve-fit process consists of the following steps. First, a record of data is selected for analysis (often collected from the output of an ADC). Then, a sinewave is generated in software that is a best fit to the data record. The process of generating this best-fit sinewave for comparison gives the method its "curve fit" title. The accuracy of this key step governs the ultimate accuracy of the parameters calculated.

The basis for computing the best-fit sinewave is the formula for actual RMS (root mean squared) error. This is simply the square root of the sum of the squared errors between the measured data points and the fitted sinewave.



The actual RMS error is calculated from:

$$E = \sum_{k=1}^m [x_k - A \cos(\omega t_k + P) - C]^2 \quad (3.1)$$

where  $E$  is the error,  $x_k$  and  $t_k$  are data points,  $m$  is the total number of data points in the record, and the fitted sinewave parameters are:

A - amplitude

$\omega$  - frequency

C - offset

P - phase

Although the curve-fit equations are derived with cosines, the process is valid for all sinusoids. Using this formula, the curve-fit algorithm attempts to find the best-fit sinewave by minimizing the error  $E$ .

First, the partial derivative of "E" is taken with respect to each of the four sinewave parameters. Then, all four of the derivatives are set equal to zero (which defines the point of minimum error). Performing some substitutions and rearranging terms eventually yields two nonlinear equations. These are solved simultaneously by iteratively adjusting the fit parameters (frequency, phase, gain, and offset), until a solution is achieved within the specified tolerance. Any difference between the data record and the best-fit sinewave is assumed to be error.

This approximation algorithm requires initial guesses for frequency and phase, which are close to the actual solution values, to ensure convergence to the best-fit sinewave. For the initial frequency estimate, the frequency of the generator output is typically used. The initial phase value is usually based on an examination of the data record by a software routine [1].

## ADVANTAGES

The curve-fit algorithm will detect error components in the ADC such as harmonic distortion, white noise, and aperture uncertainty [1]. Gain, offset, and phase errors do not affect the results. Harmonic distortion is usually a nonlinear function of amplitude and frequency. The amplitudes of harmonics or aliased harmonics present in the error residue can be extracted with this technique by fitting the error with best-fit sinewaves of the major harmonic frequencies.

## DISADVANTAGES

The greatest pitfall of the curve-fit test is the potential for selecting an initial input frequency that is a submultiple of the sample frequency. This could cause a false convergence. Other disadvantages of this method range from slow convergence to complete lack of convergence (when data is poor or computational resolution is inadequate). Also, the initial guesses for frequency and phase required by this algorithm severely limit its usefulness for sinewaves which are completely unknown.

## A RECENT SOLUTION

A more recent approach to the problem of sinewave parameter estimation attempts to overcome some of these difficulties with a completely different procedure [4]. By combining a nonlinear method to determine sinewave frequency and phase angle, with a windowing technique to determine amplitude and dc offset, the iterative process is replaced by "closed form" equations. Before proceeding with a further description of this non-iterative algorithm, the next chapter outlines the methodology that will be used throughout this study to compare the performance of estimation algorithms.

## CHAPTER IV

### A METHODOLOGY TO COMPARE THE ALGORITHMS

Before proceeding to investigate some of the more recently developed algorithms for sinewave parameter estimation, a means for comparing their performance must be established. The means adopted for this study consist of software simulations and empirical conclusions drawn from the results. This approach was found to be more suitable than complex theoretical predictions for the following reasons:

- 1) Although several of the existing algorithms have accompanying formulas for the expected error, these equations are usually derived assuming pure sinewave inputs (no distortion). For practical applications, where some distortion is inevitable, the theoretical equations no longer characterize actual performance.
- 2) With simulations, it is possible to model and test a broad range of input conditions. These results offer a direct performance comparison, without the additional process of deriving equations to match the responses.

A fair comparison requires a uniform test (or tests) which can be applied to each algorithm and a consistent set of assumptions governing the process. The following discussion presents each of the key assumptions made for the purpose of this study and the justification for each one.

#### THE TEST SINEWAVE

Since the goal of this study is improved sinewave parameter estimation, a pure sinusoid is chosen as the basic test signal. Unit amplitude, zero DC offset and zero phase angle were chosen for the analog sinewave parameters. These values were selected both

to simplify the analysis, and to remain consistent with assumptions established for Jenq's frequency and phase estimation technique [2]. The fourth and final parameter, frequency, is allowed to vary over a specified range of values. This range will be determined from constraints described in the next chapter.

Mathematically, the basic analog test signal is described by:

$$s(t) = A \cdot \sin(2\pi f t + \theta) + D \quad (4.1)$$

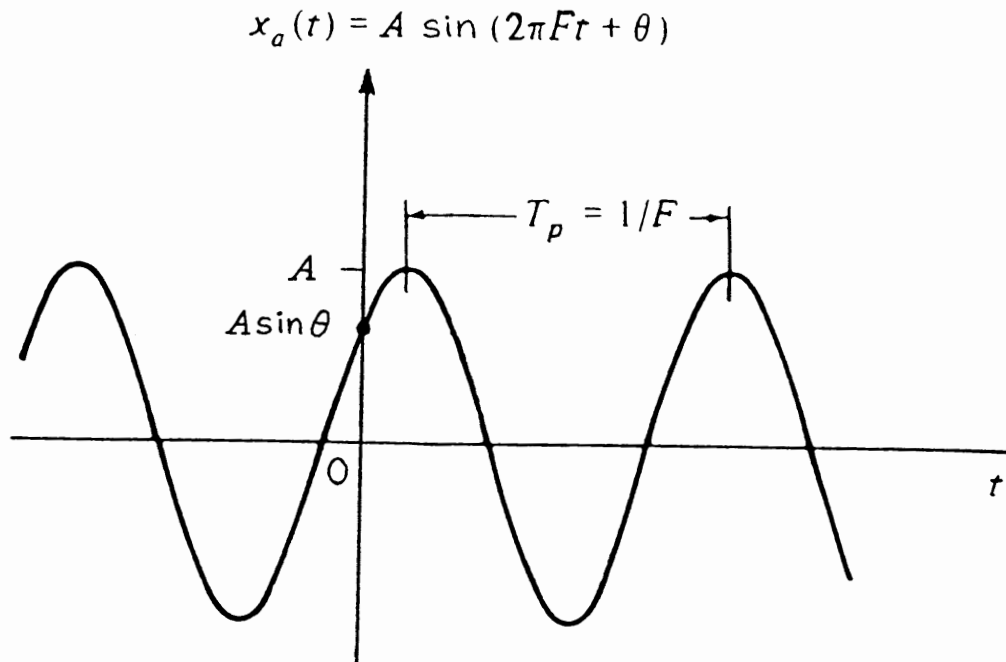
where  $A$  = amplitude

$D$  = dc offset

$f$  = frequency

$\theta$  = phase angle

Figure 5 depicts this basic test signal.



**Figure 5.** Analog sinewave.

## SAMPLING

As discussed in Chapter II, the typical input to parameter estimation algorithms is a record of digital values. Converting practical analog test signals into the necessary digital sequence involves sampling. To ensure a consistent performance comparison, the input data (and thus the sampling process) must be the same for each algorithm.

There are two bounds to establish for this process: the "length" (in sample points) of the data record, and the "rate" at which the data is recorded. In general, if it were possible to store an unlimited number of samples, the higher the sampling rate (more samples per cycle), the better the digital representation will be. However, practical records are limited to a finite length, so lower sampling rates (fewer samples per cycle) acquire data over more cycles of the waveform. The resulting record will be a better average of the entire signal and provide better parameter estimates. For this study, the length of the data record is fixed at 1024 points to balance the advantages of a large record for better signal representation, with a shorter length for faster data processing. Furthermore, since this value is an even power of two (specifically  $2^{10}$ ), it permits the use of the Fast Fourier Transform (FFT) -- an efficient algorithm for computing the discrete Fourier transform of the data.

The sampling rate chosen for the simulations is 1024 samples per second. Actually, the "exact numerical" value of this rate is not as critical as its value "relative" to the frequency of the analog waveform being sampled. For instance, a 1 Hz (Hertz) sinewave sampled at 4 Hz, or a 1 GHz sinewave sampled at 4 GHz, has the same relative effect (i.e. the same number of samples per cycle). The most important factor in choosing the sampling rate is satisfying the well-known Nyquist Sampling Theorem. Briefly, this theorem requires that an analog signal must be sampled at a rate of at least twice the highest frequency component of the input signal to allow reconstruction of the original signal from the discrete samples without aliasing. The judicious selection of the range of

frequencies for the sinusoidal test signal will ensure that the chosen sampling rate (1024) satisfies the Nyquist criteria.

## NOISE MODELS

The previous sections established assumptions for the analog sinewave and its digital processing. However, practical signals usually suffer some distortion from this process. For a realistic test of parameter estimation performance, this distortion must be incorporated into the model for input sinewave. Chapter II described a common model for the distortion (using an additive noise source) that will be adopted for this analysis (see Figure 4 and equation (2.3)). In order to apply this additive noise model, the error signal ( $e$ ) must be defined for each type of distortion represented. The principle types of distortion imposed on an input signal by a practical ADC can be divided roughly into two categories -- linear and nonlinear. The next step is to determine what type of signal best represents each general category.

### Linear Noise

Beginning with linear distortion (such as quantization noise), a common model for this type of error is uniformly distributed white noise [6, 7]. "White" implies a random mix of frequencies is present in the signal (this assures "unbiased" distortion for a wide range of test signal frequencies). Uniformly distributed means that the possible range of noise levels is uniform -- centered about the origin with values constrained between  $\pm 1/2$  of the mean amplitude chosen.

### Nonlinear Noise

The second distortion category, "nonlinear," is often modeled with harmonics of the input signal. For simplicity, the model in this study is based on the primary harmonics -- second and third. The basic harmonic parameters were selected as follows. "Zero" dc

offset is chosen to be compatible with the input sinewaves. The harmonic "frequencies" are relative to the fundamental frequency of the input test signal ( $2f_0$  and  $3f_0$ ). To avoid biased data, harmonics are generated with random phase angles. The final harmonic parameter, "amplitude," varies over a range of values. The following sections describe how this range was selected.

### Average Power

A realistic analysis of parameter estimation algorithms involves testing their performance with practical input signals. Practical signals may contain linear and nonlinear distortion. In this study, the effect of each type of distortion (on estimation accuracy) is considered separately. To ensure a consistent performance comparison, roughly equivalent levels of each type of distortion should be tested. The problem is determining what is *equivalent*. Because linear distortion is represented by a random signal (white noise), and nonlinear by deterministic signals (harmonics), equal amplitudes does not imply equal distortion. Instead, the average signal *power* provides a common reference for these incompatible types. In the statistical approach commonly adopted, average signal power refers to the mean-square power or variance.

The power of uniformly distributed white noise (representing linear distortion), is calculated from this random signal's Probability Density Function (PDF). Figure 6 shows the PDF for this signal.

By definition, the average power (or variance) of a uniform random variable equals the integral of its PDF. The total area under the PDF equals one (shown in Figure 6). For consistent derivations, let  $q_n = \Delta =$  the peak-to-peak noise amplitude, and  $e =$  error. Then,

$$P_e = \int_{-q_n/2}^{q_n/2} e^2 \cdot \frac{1}{q_n} de = \frac{1}{q_n} \left[ \frac{1}{3} e^3 \right]_{-q_n/2}^{q_n/2} = \frac{q_n^2}{12} \quad (4.2)$$

computes the power of the random noise signal.

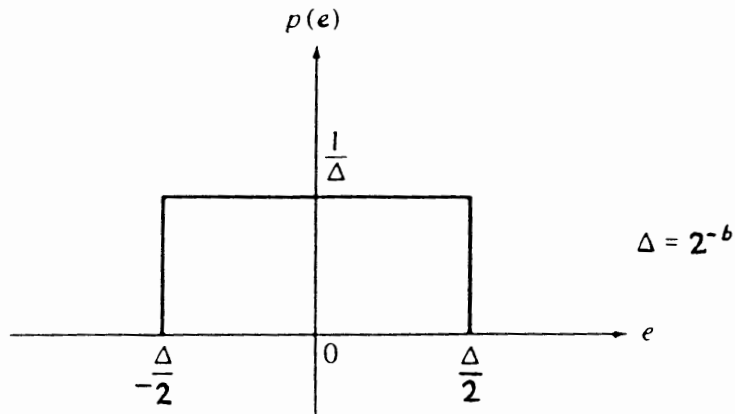


Figure 6. PDF for a uniform random variable.

The corresponding result for sinewaves can be derived as follows. Letting  $\tau$  = one sinewave period,  $A_s$  = sinewave amplitude and  $q_s$  = peak-to-peak amplitude, the sinewave power (variance) is given by:

$$P_s = \frac{1}{\tau} \int_0^{\tau} (A_s \sin \omega_0 t)^2 dt = \frac{A_s^2}{2} = \frac{\left(\frac{q_s}{2}\right)^2}{2} = \frac{q_s^2}{8} \quad (4.3)$$

Finally, the signal-to-noise ratio (SNR) for the test sinewave *signal* combined with uniformly distributed random *noise* is:

$$\text{SNR} = \frac{\text{sine power}}{\text{noise power}} = \frac{P_s}{P_e} = \frac{\frac{q_s^2}{8}}{\frac{q_n^2}{12}} = \frac{3}{2} \frac{q_s^2}{q_n^2} \quad (4.4)$$

Converting to decibels (dB):

$$\begin{aligned} \text{SNR}_{\text{dB}} &= 10 \log(\text{SNR}) \\ &= 10 \log\left(\frac{3}{2} \frac{q_s^2}{q_n^2}\right) \\ &= 10 \log\left(\frac{3}{2}\right) + 20 \log\left(\frac{q_s}{q_n}\right) \\ &\approx 1.760913 + 20 \log\left(\frac{q_s}{q_n}\right) \text{ dB} \end{aligned} \quad (4.5)$$



where the "q's" can represent either peak, or peak-to-peak amplitude; as long as the interpretation is consistent.

Next, an equivalent result will be derived for nonlinear distortion. Harmonics are deterministic signals with the same characteristics as the test sinewave. This simplifies the SNR calculation. From equation (4.3) the power of a sinusoid is given by:  $P_s = \frac{A_s^2}{2}$ , where  $A_s$  = sinewave amplitude. Since a harmonic is another sinewave, its power can also be represented by:  $P_h = \frac{A_h^2}{2}$ , where  $A_h$  = harmonic amplitude. So, the SNR for sinewave signals with nonlinear (harmonic) distortion is:

$$\text{SNR} = \frac{\text{sine power}}{\text{harmonic power}} = \frac{P_s}{P_h} = \frac{\frac{A_s^2}{2}}{\frac{A_h^2}{2}} = \frac{A_s^2}{A_h^2} \quad (4.6)$$

or expressed in dB:

$$\text{SNR}_{\text{dB}} = 10 \log(\text{SNR}) = 10 \log\left(\frac{A_s^2}{A_h^2}\right) = 20 \log\left(\frac{A_s}{A_h}\right) \text{ dB} \quad (4.7)$$

Using equations (4.5) and (4.7) for comparison, the next step is to select a reasonable range of linear distortion levels and compute equivalent harmonic levels to test.

### Noise Levels

The objective of the simulations is to represent practical conditions for testing the estimation algorithms. Therefore, levels of distortion in the input sinewave should be similar to levels that could occur in actual test signals. Since ADCs commonly introduce some distortion into practical signals, their effects will provide bounds for the simulated noise range.

In the worst case (poorest resolution), an ideal one-bit ADC would have a maximum quantization error equal to 1/2 of the input signal's full scale amplitude (or SNR = 2.0). At the opposite end of the scale, ADCs with greater resolution (more bits) would

generate much smaller quantization errors (producing higher SNR). From these general guidelines, the selected amplitude range for white noise (linear distortion) is: [0.05, 0.1, 0.2, 0.3, 0.4, 0.5]. Table I compares the amplitudes with their equivalent signal-to-noise ratios computed from (4.5).

**TABLE I**  
**COMPARING SIGNAL STRENGTHS: SINE AND WHITE NOISE**

SINE AMPLITUDE (Constant)		NOISE AMPLITUDE (Variable)		SIGNAL-TO- NOISE RATIO (SNR)		NOISE-TO- SIGNAL RATIO (NSR)	
PEAK (PK)	PEAK-TO-PEAK (P-P)	PEAK (PK)	PEAK-TO-PEAK (P-P)	AMPLITUDE	dB	AMPLITUDE	dB
1.0	2.0	0.05	0.1	20.00	27.78	0.05	-24.26
1.0	2.0	0.10	0.2	10.00	21.76	0.10	-18.24
1.0	2.0	0.20	0.4	5.00	15.74	0.20	-12.22
1.0	2.0	0.30	0.6	3.33	12.22	0.30	-8.70
1.0	2.0	0.40	0.8	2.50	9.72	0.40	-6.20
1.0	2.0	0.50	1.0	2.00	7.78	0.50	-4.26
1.0	2.0	0.75	1.5	1.33	4.26	0.75	-0.74
1.0	2.0	1.00	2.0	1.00	1.76	1.00	1.76

Corresponding levels of nonlinear distortion were calculated by equating signal-to-noise ratios (equations (4.5) and (4.7)) and determining the required harmonic amplitude. The resulting range of nonlinear distortion levels is shown in Table II.

**TABLE II**  
**COMPARING SIGNAL STRENGTHS: SINE AND HARMONICS**

SINE AMPLITUDE (Constant)		HARMONIC AMPLITUDE (Variable)		SIGNAL-TO- HARMONIC RATIO (SHR)		HARMONIC-TO- SIGNAL RATIO (HSR)	
PEAK (PK)	PEAK-TO-PEAK (P-P)	PEAK (PK)	PEAK-TO-PEAK (P-P)	AMPLITUDE	dB	AMPLITUDE	dB
1.0	2.0	0.05	0.1	20.00	26.02	0.05	-26.02
1.0	2.0	0.10	0.2	10.00	20.00	0.10	-20.00
1.0	2.0	0.20	0.4	5.00	13.98	0.20	-13.98
1.0	2.0	0.30	0.6	3.33	10.46	0.30	-10.46
1.0	2.0	0.40	0.8	2.50	7.96	0.40	-7.96
1.0	2.0	0.50	1.0	2.00	6.02	0.05	-6.02
1.0	2.0	0.75	1.5	1.33	2.50	0.75	-2.50
1.0	2.0	1.00	2.0	1.00	0.00	1.00	0.00

At this point, some comments are in order regarding the simulation data.

- 1) Throughout the remaining text (in tables, graphs and discussion) the following terms will be used interchangeably:
  - SNR (signal-to-noise ratio) and NSR (noise-to-signal ratio). These are merely inverse ratios and although graphs may plot estimation errors versus increasing noise levels (NSR), results are usually described in terms of SNR for familiarity.
  - SHR (signal-to-harmonic ratio) and HSR (harmonic-to-signal ratio). These terms are the equivalent of SNR and NSR (and may be used interchangeably), but refer specifically to nonlinear distortion. Results will be presented generically in terms of SNR, but tables and figures may use SHR or HSR to distinguish nonlinear distortion.
- 2) The analysis of parameter estimation algorithms is based on their performance for the levels of distortion described above. However, results of simulations with noise levels as high as 2.0 or 3.0 peak-to-peak may be listed in later tables. These additional simulations were only run to clarify performance trends. The levels are unrealistic for practical signals and will not be incorporated in the final evaluations.

## EXPERIMENTAL PROCEDURE

### Software

Simulations were performed using the MATLAB software package. MATLAB is a general purpose tool for mathematics computation. In addition, it features several built-in functions adapted specifically for circuit analysis and signal processing tasks. MATLAB also allows users to create their own customized code for special applications

not covered by the built-in functions. Custom code was written to implement each of the parameter estimation algorithms studied.

### Trials

The majority of simulations in this study represent "sensitivities" (or the effect of changing only *one* variable on the algorithm's performance). In these tests, as one of the input parameters is varied incrementally through a range of values, the results of the algorithm being tested, are recorded for each increment. Due to the random component of the noise distortion which is present in most practical signals, several simulations (or "trials") using the same range of input conditions, are required to obtain representative results. Taking the "average" of the outcomes helps to "cancel out" the effects of random noise. Determining an adequate number of trials to accomplish this, is influenced by two conflicting requirements. There should be enough trials to obtain a representative range of values, but not so many that computation time for the necessary iterations becomes prohibitive. With this tradeoff in mind, five trials per experiment was selected as a reasonable number.

### Error Analysis

Once the simulations are completed, the accuracy of parameter estimation algorithms is evaluated empirically by comparing how closely the estimated parameters approximate the actual analog values. Any differences represent errors. For performance comparisons, a consistent method must be determined for calculating this estimation "error."

There are several standard definitions to choose from, but a basic one, sufficient for this study, is *relative error*. Given an estimated quantity ( $e$ ) determined from simula-

tions, and an actual value ( $a$ ) which is assumed to be "known," the *relative error* of the estimated value compared to the actual value is:

$$E_{REL} = \frac{e - a}{a} \quad (4.8)$$

which can also be expressed as a percentage:

$$E_{\%REL} = E_{REL} \times 100 = \frac{e - a}{a} \times 100 \quad (4.9)$$

There are two exceptions to this chosen reference which arise in the course of this study. In some instances, the *actual* value ( $a$ ) is equal to "zero" (such as the test sinewave's *dc offset* or *phase angle*). For these special cases, applying the formula for relative error results in an illegal division by zero. Instead, the equation for *absolute error* ( $E_{ABS} = e - a$ ) is used for comparisons. The other special case occurs when the actual value ( $a$ ) is equal to "one" (such as the test sinewave's *amplitude*). Here, the relative error formula reduces to the equation for absolute error by default since the divisor is "one."

## SUMMARY

Now that the key assumptions and the simulation process have been defined, we have the means to gather data to compare the performance of various parameter estimation algorithms. This analysis begins with an algorithm recently developed by Jenq [4], which is a non-iterative technique for parameter estimation.

## CHAPTER V

### THE FOUR-PARAMETER ALGORITHM

A different approach to the sinewave parameter estimation problem, developed by Jenq [4], overcomes several of the disadvantages of traditional curve-fit methods. It successfully combines two separate techniques for partial parameter estimation into a complete four-parameter sinewave estimator. One component of this method is a high performance frequency and phase estimator based on a weighted least squares method [2]. However, the uniqueness of this method lies in the process of applying a window to the digitized data preparatory to estimating the amplitude and dc offset. The combination of this windowing technique with the results of the previous frequency and phase estimator produces a method capable of estimating all four parameters of an unknown sinusoid.

The following sections describe how this algorithm was derived. This description is presented in the following order. First, an intuitive explanation of the process and the logic behind it. Then, a more rigorous mathematical derivation is presented. This explanation is a summary; the full description is contained in [4].

#### ESTIMATING AMPLITUDE AND DC OFFSET

##### Intuitive

Since the four-parameter algorithm is, by definition, a method for estimating *all* of the parameters of an unknown sinewave, it is assumed that the only data available to work with is a record of samples obtained from the analog test signal. Given only this

input, the most straightforward parameter to compute is the sinusoid's "dc offset" (or average value). A rough estimate could be obtained by summing the *values* of all the data points in the record and dividing this sum by the total *number* of samples in the record. The four-parameter algorithm modifies this basic procedure by first windowing the data record for increased accuracy. The next section will detail this process.

The process for estimating a sinewave's "amplitude" follows similar reasoning. The first step is to subtract the previously-estimated dc offset from each windowed sample in the data record. The result is a new record with dc offset adjusted to zero. The square root of the sum of the squares of these values is computed to determine the data's average *magnitude* independent of the *sign* of the samples. This completes the first phase of the four-parameter estimation algorithm.

### Derivation

From an intuitive understanding of the process used by the four-parameter algorithm, a more rigorous derivation can be developed. The initial input to the algorithm is typically a digitized data record of the form:

$$S = s_k = s(k/f_s) = A \cdot \sin(k2\pi f/f_s + \theta) + D, \quad k = 0, 1, \dots, N-1 \quad (5.1)$$

This record consists of a sequence of data points obtained from sampling and quantizing the original test sinewave:

$$s(t) = A \cdot \sin(2\pi f t + \theta) + D \quad (5.2)$$

at the rate of  $f_s$  samples per second.

Next, a window sequence is defined by sampling the four-term Blackman-Harris window:

$$W = \left\{ w_k, \quad k = 0, 1, \dots, N-1 \right\},$$

where  $w_k$  is computed by the following equation:

$$w_k = 0.35875 - 0.48829 \cos(k2\pi/N) + 0.14128 \cos(k4\pi/N) - 0.01168 \cos(k6\pi/N), \quad k = 0, \dots, N - 1 \quad (5.3)$$

The windowing process is what distinguishes this algorithm from other four-parameter estimators. For a better understanding of this step, it would be instructive to pause temporarily from the derivation and explore the advantages of windowing.

If the input data record contained an infinite number of sinewave samples, it would be relatively straightforward theoretically to determine the amplitude and dc offset from the Fourier transform of the data. Figure 7 shows how the value of these parameters could be measured directly since all of the spectral energy is located at a single frequency (or bin).

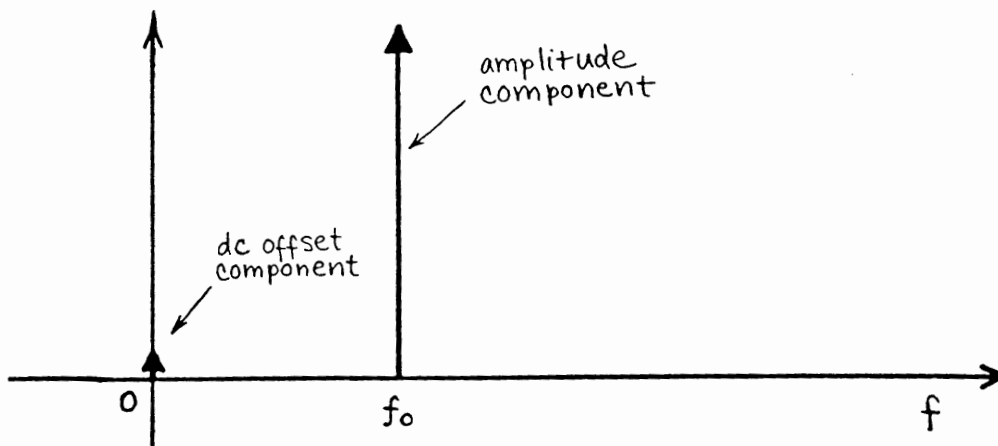


Figure 7. Frequency spectrum of a typical input sinewave.

However, the actual data available is contained in a finite record of fixed length. Applying the Fast Fourier Transformer (FFT) to this data will compute its frequency spectrum. However, there are some complexities involved in working with a limited

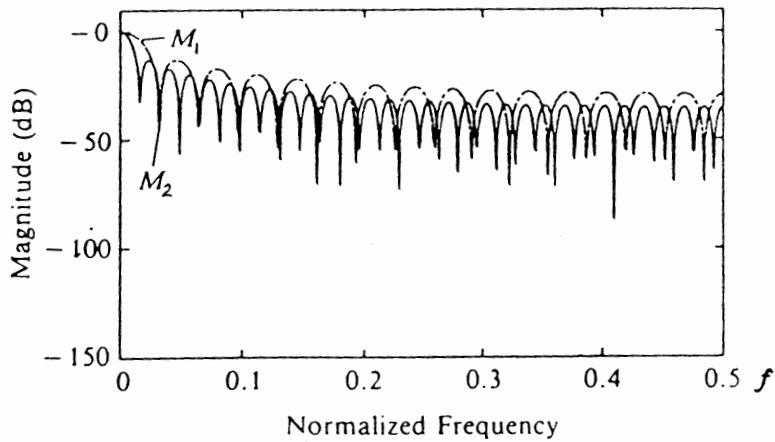


amount of data. In computing the FFT (or any discrete Fourier transform), the samples are assumed to be periodic, which produces a periodic frequency spectrum. The  $n$  samples input to the FFT can be thought of as an infinite sequence of samples to which a rectangular window has been applied -- resulting in a truncated data record. However, this windowing has some adverse effects on the frequency spectrum of the input data.

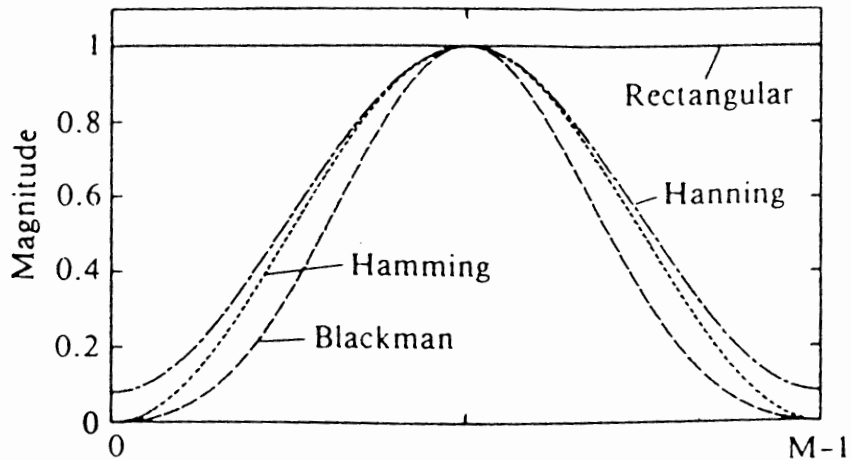
One of these effects is due to the shape of the window's main lobe. It is called "spectral smearing" because a spectrum that would normally be an impulse located at a specific frequency, will have its power spread or "smeared" across the surrounding bins after windowing. This results in a loss of resolution in the frequency spectrum. The extent of the smearing is proportional to the width of the window's main lobe. The other effect, called "spectral leakage," is due to the properties of the window's sidelobes. Spectral leakage occurs when frequency components of the input signal do not fall exactly on one of the bin frequencies of the FFT. When that happens, the spectral energy of those components will leak out to the surrounding bins, distorting their true magnitudes.

Clearly, windowing affects the input data, and working with a finite record of samples introduces rectangular window characteristics -- wide main lobe and large sidelobes (see Figure 8).

However, these inevitable negative effects can be offset somewhat by the proper choice of a window function to apply *before* estimating the sinewave's parameters. With this choice, spectral smearing can be confined to a narrow range of bins and spectral leakage can be limited to distant bins. There are many windows available to choose from, each with slightly different properties. Figure 9 shows the time domain and Figure 10, the frequency domain characteristics for a variety of common window functions.

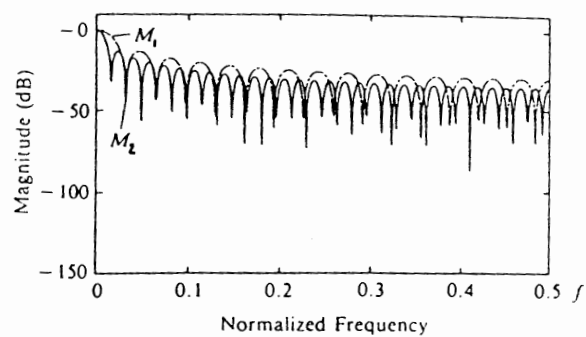


**Figure 8.** Frequency domain characteristics of the rectangular window.

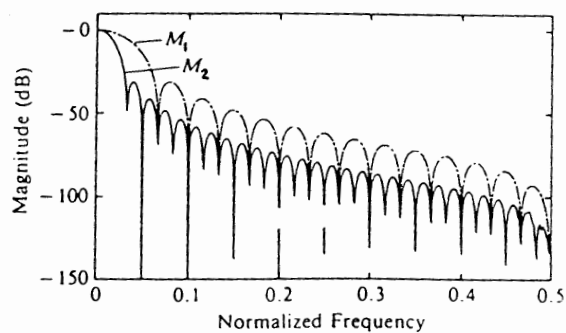


**Figure 9.** Time domain characteristics of several common windows.

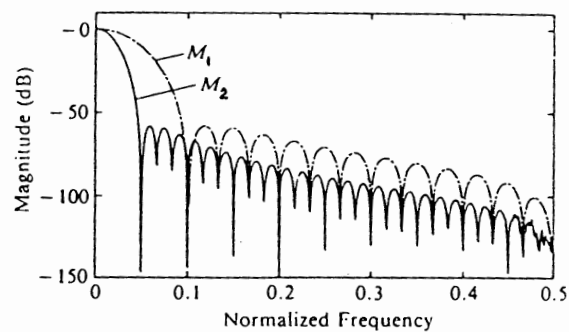
From the available functions, the Blackman-Harris window was chosen for a some of its unique properties. Figure 11 depicts its magnitude spectrum [4]. First, it is evident from the figure that the window has a clearly defined main lobe confined between the frequency bins 0 to 4. This limits spectral smearing of an impulse to a width of nine bins (the original bin with smearing of  $\pm 4$  bins around it). Secondly, this window is designed for very small sidelobes which do not exceed -92 dB after the fifth bin. This reduces spectral leakage by 92 dB within 5 bins from an impulse. So, the Blackman-



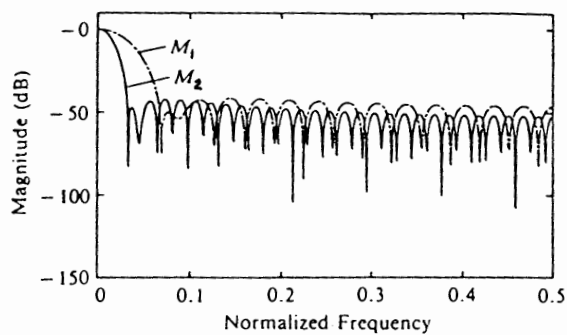
(a) Rectangular window



(b) Hanning window



(c) Blackman window



(d) Hamming window

**Figure 10.** Frequency domain characteristics of several common windows.

Harris window achieves a good compromise between the slight loss of resolution for frequency components occurring exactly at bins and excellent reduction of spectral leakage for all other components.

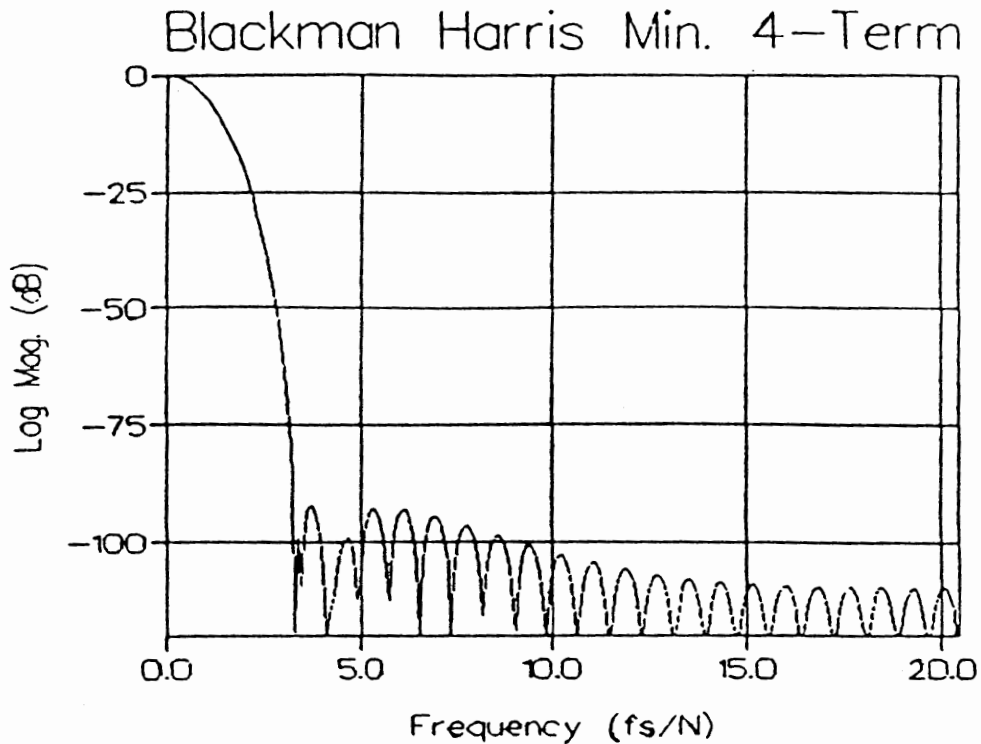


Figure 11. Magnitude spectrum of the Blackman-Harris window.

What this means in terms of improvement to the parameter estimates is illustrated in Figure 12. As this figure shows, when the condition that the data record (S) contains more than 4 cycles of the sinewave is met, the frequency component (impulse) representing the sinewave's *amplitude* falls outside the main lobe of the Blackman-Harris window. This reduces the spectral leakage from this component by 92 dB, insuring minimal distortion to the dc offset component after windowing. Consequently, the true value of the sinewave's dc offset will be determined more accurately. The amplitude estimate will be similarly improved.

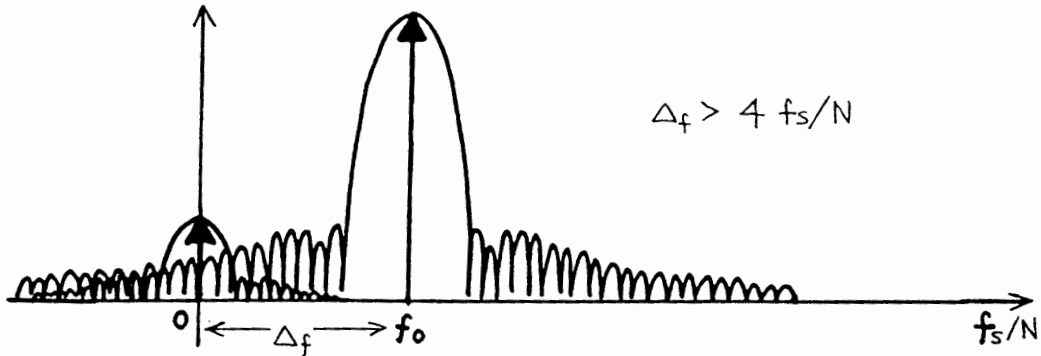


Figure 12. Windowing the sinewave spectrum.

Returning to the derivation, applying the Blackman-Harris window ( $w_k$ ) to the data record ( $s_k$ ) yields the dc offset estimator ( $\tilde{D}$ ):

$$\tilde{D} = \frac{\sum_{k=0}^{N-1} w_k s_k}{\sum_{k=0}^{N-1} w_k} \quad (5.4)$$

and the equation for the amplitude estimator ( $\tilde{A}$ ) follows:

$$\tilde{A} = \left[ 2 \frac{\sum_{k=0}^{N-1} w_k (s_k - \tilde{D})^2}{\sum_{k=0}^{N-1} w_k} \right]^{1/2} \quad (5.5)$$

### ESTIMATING FREQUENCY AND PHASE ANGLE

With estimates for amplitude and dc offset calculated, the remaining two sinewave parameters (frequency and phase) must be determined to complete the algorithm.

## Intuitive

A method known as "phase unwrapping" is used to extract the frequency and phase parameters from the sinewave data record. This process will be described in greater detail in the following section, but the result is a sequence of phase values computed from the sample points. These phase angles (in radians) can be plotted versus the sample points in an ever-increasing plot like Figure 13. This figure shows that the *frequency* of the sinusoid (or "rate of change" of the phase angle graph) is estimated by calculating the derivative (slope) of the linear plot. The point where this graph intercepts the vertical axis provides an estimate of the sinewave's *phase*.

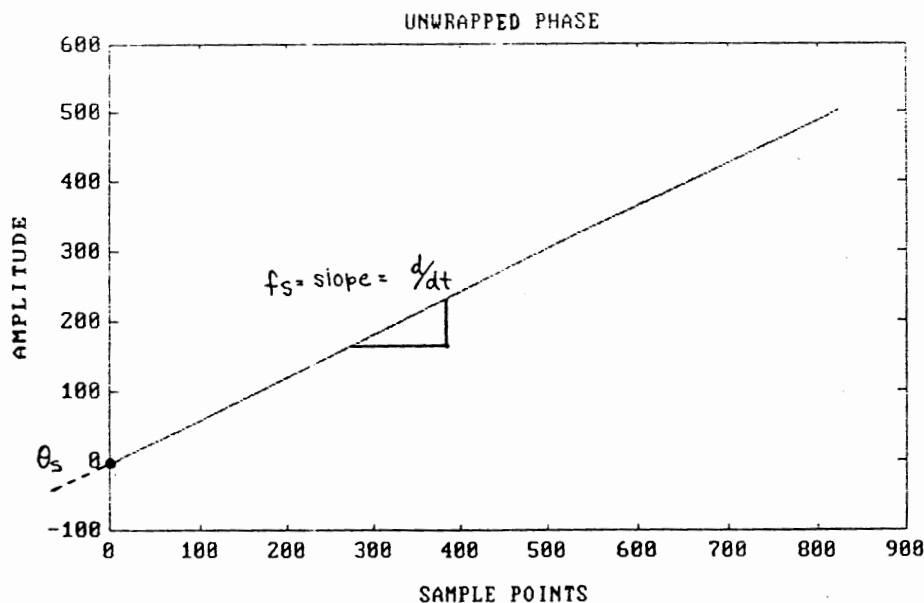


Figure 13. Estimating frequency and phase.

## Derivation

To express the "phase unwrapping" process mathematically, some additional sequences must be defined. Starting with the original data sequence ( $s_k$ ), subtracting the estimated value for dc offset and dividing by the estimated amplitude, yields a new

sequence ( $\hat{s}_k$ ) defined by:

$$\hat{s}_k = \frac{(s_k - \tilde{D})}{\tilde{A}}, \quad k = 0, 1, \dots, N - 1 \quad (5.6)$$

with dc offset adjusted to zero and amplitude normalized to unity.

The next step is to "unwrap" the phase of the sequence just created. The results of this process provide the data necessary for estimating the sinusoid's frequency and phase angle. In order to accomplish this critical step, it is first necessary to understand exactly what is meant by the term "unwrapping." The term is illustrative of the effect of computing the angular argument of each sinusoid sample, plotting these angles successively around the unit circle, and then "unwrapping" this wound string of values to form a linear graph. Figure 14 illustrates this concept.

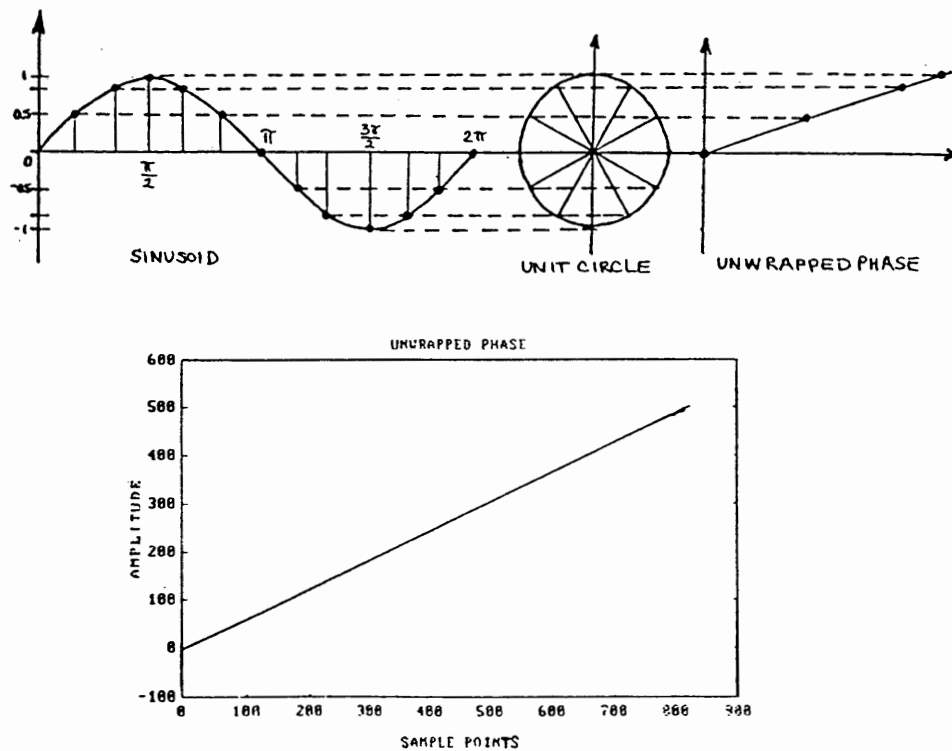


Figure 14. Phase Unwrapping.

Since the digitized sequence ( $\hat{s}_k$ ) represents a sinusoid, the value of each data point in the sequence is the *sine* of some angle ( $\theta$ ). To "unwrap" the phase angles from this sequence requires reversing the computation to determine the associated angle, or *arcsine*, of the data.

Because a sinewave  $s(t)$  is a periodic function, it is characterized by the following property:

$$s(t + \tau) = s(t)$$

where  $\tau$  is the fundamental period of the sinewave. This property describes how the sine function repeats the same sequence of values at intervals separated by one period ( $\tau$ ). Attempting to extract the phase angles from the data record by computing the arcsine of each sample directly, results in a periodic sequence of angles constrained between  $[-\frac{\pi}{2} \leq \sin^{-1}(\theta) \leq \frac{\pi}{2}]$ . Values within this range are known as the "principal" values of the arcsine. However, the four-parameter algorithm's formulas are based on the "absolute" value of arcsine (which ranges over the unit circle  $[0, 2\pi]$ ).

To determine the absolute arcsine of a sample (and not merely the principal value) requires two separate equations depending on which portion of the sinewave cycle the sample was taken from. As Figure 15 illustrates, each cycle of a sinewave can be roughly divided into two regions -- one of increasing (or positive) slope, and the other of decreasing (negative) slope.

Within the region of positive slope, the principal value of the arcsine equals the absolute value. However, within the region of negative slope, the principal value must be subtracted from  $\pi$  to obtain the correct angle. Table III shows the need for this correction.



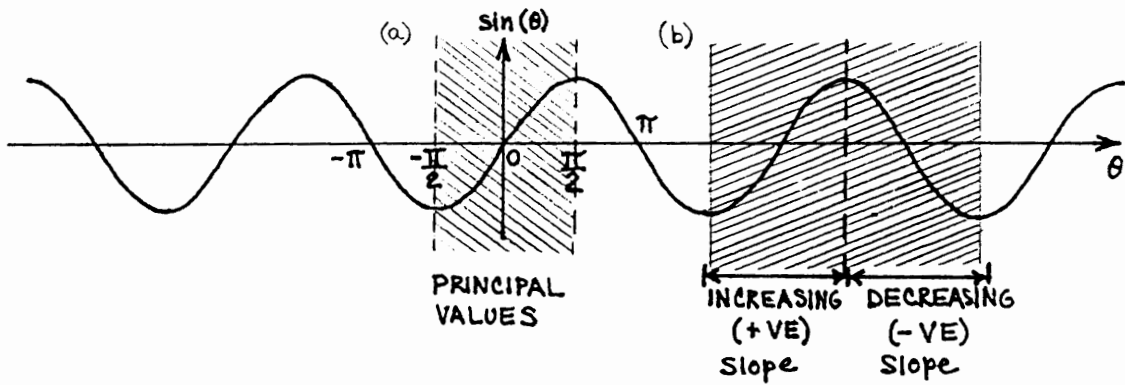


Figure 15. Regions of a sinewave cycle.

TABLE III  
PHASE UNWRAPPING

FUNCTION	FUNCTION VALUES ( $\theta$ in radians)		
	Positive Slope ( $\theta_a = \theta_p$ )	Negative Slope ( $\theta_a = \pi - \theta_p$ )	...
$y = \sin(\theta)$	$-1 \quad \frac{-\sqrt{2}}{2} \quad 0 \quad \frac{\sqrt{2}}{2} \quad 1$	$\frac{\sqrt{2}}{2} \quad 0 \quad \frac{-\sqrt{2}}{2} \quad -1$	...
$\theta_p = \sin^{-1}(y)$ <b>Principal Value</b> ( $-\frac{\pi}{2} \leq \theta_p \leq \frac{\pi}{2}$ )	$\frac{-\pi}{2} \quad \frac{-\pi}{4} \quad 0 \quad \frac{\pi}{4} \quad \frac{\pi}{2}$	$\frac{\pi}{4} \quad 0 \quad \frac{-\pi}{4} \quad \frac{-\pi}{2}$	...
$\theta_a = \sin^{-1}(y)$ <b>Absolute Value</b> ( $0 \leq \theta_a \leq 2\pi$ )	$\frac{-\pi}{2} \quad \frac{-\pi}{4} \quad 0 \quad \frac{\pi}{4} \quad \frac{\pi}{2}$	$\frac{3\pi}{4} \quad \pi \quad \frac{5\pi}{4} \quad \frac{3\pi}{2}$	...

In order to apply the appropriate arcsine formula, the next task is to detect which region of the sinewave, each sample was taken from. In the software simulations, this is accomplished with three pointers into the data record:

- $x[n - 1]$  - the previous sample
- $x[n]$  - the current sample
- $x[n + 1]$  - the next sample

where  $n$  is an index into the record. Advancing through the record, the relative positions of these three points (measured by the sine amplitude of the particular samples) identify the region of the sinewave and correspondingly, which arcsine formula to apply.

After the phase unwrapping process is complete, the resulting angles (in radians) are plotted; starting with the first value and adding successive values to the cumulative total in an ever-increasing function. Figure 16 shows the result.

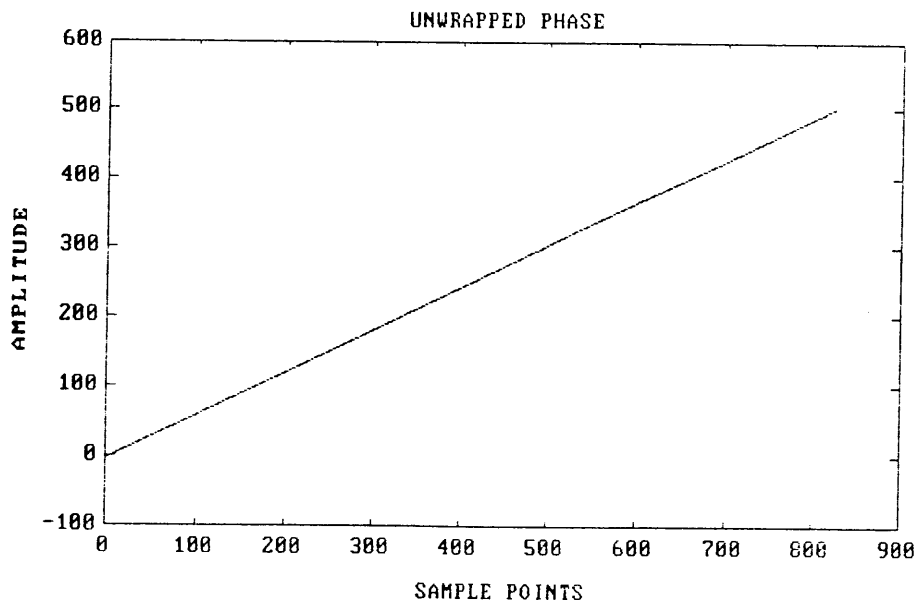


Figure 16. Plot of unwrapped phase angles.

This entire "unwrapping" process can be summed up mathematically by the conditional sequence:  $x_k = \sin^{-1}(\hat{s}_k)$ ,  $k = 0, 1, \dots, N - 1$ , subject to the monotonic condition.

Defining two further sequences:

$$u_k = 1 - (\hat{s}_k)^2, \quad k = 0, 1, \dots, N - 1$$

$$t_k = k/f_s, \quad k = 0, 1, \dots, N - 1$$

completes the terms necessary for the following summations.

$$\text{Sum } U = \sum u_k$$

$$\text{Sum } UT = \sum u_k t_k$$

$$\text{Sum } UTT = \sum u_k t_k t_k$$

$$\text{Sum } UX = \sum u_k x_k$$

$$\text{Sum } UTX = \sum u_k t_k x_k$$

$$\text{and } \Delta = \text{Sum } U \cdot \text{Sum } UTT - \text{Sum } UT \cdot \text{Sum } UT$$

Finally, the formulas for estimating frequency and phase are expressed by:

$$\tilde{f} = \frac{1}{2\pi} \left[ \frac{\text{Sum } U \cdot \text{Sum } UTX - \text{Sum } UT \cdot \text{Sum } UX}{\Delta} \right] \quad (5.7)$$

$$\tilde{\theta} = \left[ \frac{\text{Sum } UX \cdot \text{Sum } UTT - \text{Sum } UT \cdot \text{Sum } UTX}{\Delta} \right] \quad (5.8)$$

The estimation formulas derived in [4] for amplitude (5.5), dc offset (5.4), frequency (5.7), and phase angle (5.8) combine to form the complete four-parameter estimation algorithm.

## SIMULATIONS

Now that the basic procedure for the four-parameter algorithm has been described, this section is devoted to characterizing its performance through a series of software simulations. The results of these simulations will be used to identify potential areas for improving the algorithm. The following sensitivities:

- Performance with respect to increasing levels of "linear" distortion.
- Performance with respect to increasing levels of "nonlinear" distortion.

were investigated to characterize the algorithm's performance with input signals commonly encountered in practical applications.

In setting up the simulations, the general assumptions established in Chapter IV (Methodology) will be applied. However, that chapter deferred the selection of a specific range of frequencies for the sinusoidal test signal. Before proceeding further, this selection must be made.

Since the sampling rate is already established (1024 points per second), the *upper* bound for the frequency range is limited by the Nyquist criteria. To satisfy the criteria with the given sampling rate, the maximum sinusoid frequency must not exceed  $\frac{f_{\text{sample}}}{2}$  or 512 Hz (Hertz). To comply with this restriction, 200 Hz was selected as the sinewave's upper frequency bound. At the opposite end of the scale, the lower bound for the range of test frequencies is determined by the Blackman-Harris windowing function. It is a property of the Blackman-Harris window that if the data record ( $S$ ) contains more than four complete cycles of the input sinewave, dc offset estimates will have negligible error ( $\Delta D < |A| \times 10^{-4.6}$ ) [4]. Applying the same restriction will achieve the minimum error for the amplitude estimator (normalized error  $\Delta A/A < 10^{-2.3}$ ) [4]. To satisfy this

requirement, the lowest allowable frequency for the test sinewave is:

$$\frac{1024 \text{ samples}}{\text{second}} \times \frac{1 \text{ record}}{1024 \text{ samples}} \times \frac{4 \text{ cycles}_{\min.}}{\text{record}} = 4 \text{ cycles per second (Hertz).}$$

Choosing 5 Hz as the minimum frequency will remain within this lower limit imposed by windowing. After selecting these upper and lower limits, the range of simulation frequencies chosen for the test sinewave becomes [200, 100, 50, 25, 10 and 5] Hz.

## LINEAR SIMULATIONS

Initial simulations were made to test the algorithm's "linear" sensitivity -- the accuracy of parameter estimates when the input sinewave is distorted by varying levels of uniformly distributed white noise). This type of distortion represents the inevitable quantization errors. The following process is repeated for each sequence of simulations.

Given a user-defined time interval ( $t = [0, 1.024]$  seconds), MATLAB is used to generate a sinewave sequence with unit amplitude, zero dc offset, zero phase angle, and variable frequency selected from the established range: [200, 100, 50, 25, 10, 5]. Using the same time interval, MATLAB also generates a sequence of uniformly distributed random noise values with zero mean and a peak-to-peak amplitude from the range: [0.1, 0.2, 0.4, 0.6, 0.8, 1.0]. Combining these two sequences models the effect of processing a sinewave with a practical ADC. Figure 17 shows the stages in creating this model.

Next, this noisy sinewave is input to the four-parameter algorithm and the resulting estimates are recorded and averaged. Estimation errors are calculated from these averages. Then, a new (higher amplitude) random noise sequence is generated and the estimates repeated. This process continues until all combinations of sine frequencies and noise levels have been tested.

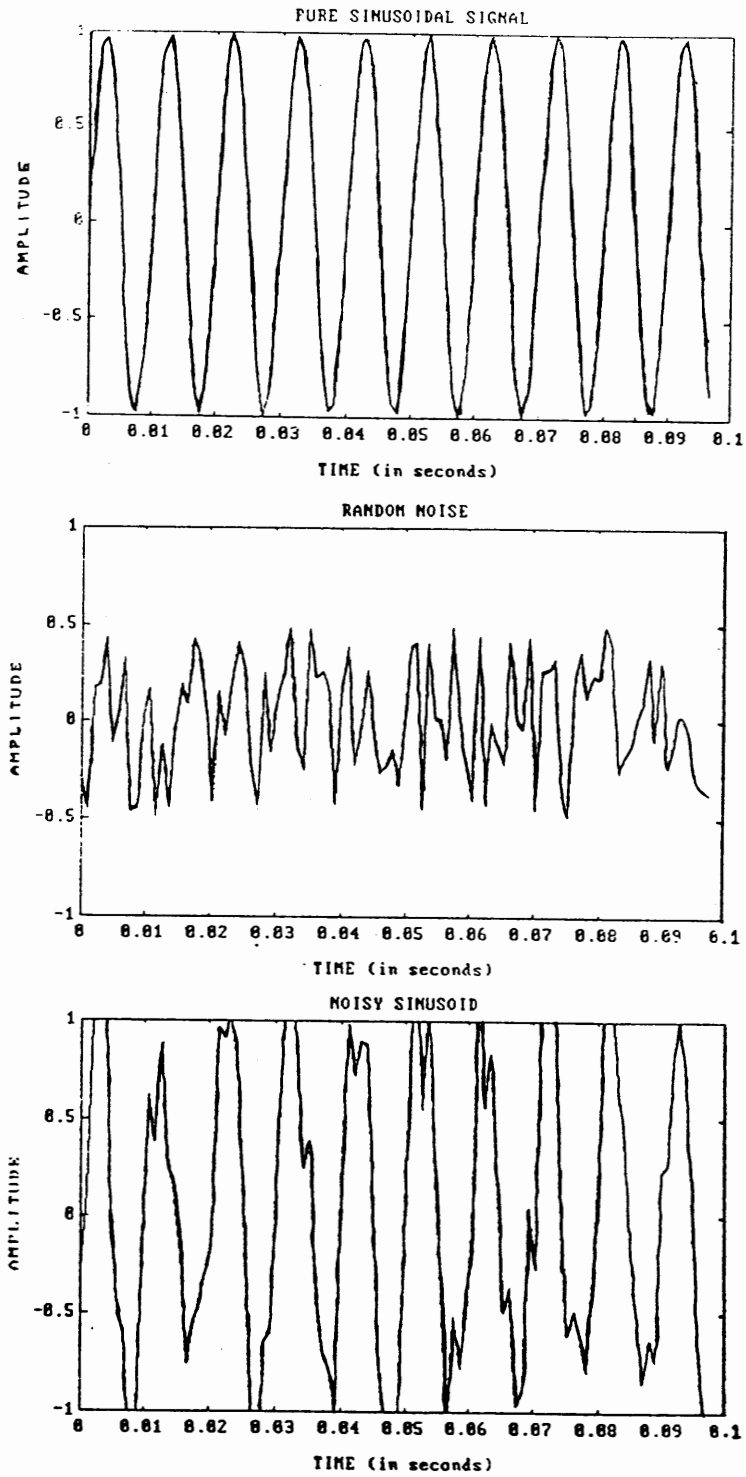


Figure 17. Modeling linear distortion in the test sinewave.

## LINEAR RESULTS

In this and all successive discussion of simulation results, distortion levels will be stated in terms of the standard reference, SNR (signal-to-noise ratio). Figures and graphs however, are plotted with respect to the inverse relation NSR (noise-to-signal ratio) to emphasize errors increasing with distortion.

Beginning with *amplitude* estimates, average errors using the four-parameter algorithm were on the order of  $10^{-2}$ . Table A.I shows the range of amplitude estimation errors for the various input combinations tested. As expected, estimation errors increase with higher noise levels (Figure A.1), but show no recognizable trend with respect to sampling rate.

*DC offset* estimates follow similar trends, but are more accurate than amplitude estimates overall. Average errors are on the order of  $10^{-3}$  for noise levels below 10 dB SNR. At higher levels of distortion, errors increase up to a factor of 10. Table A.II lists the exact values. Figure A.2 shows the relation between dc offset estimation errors and noise levels. There is no definite relation between these errors and sampling rates.

The accuracy of sinewave *frequency* estimates follows two distinct trends. Errors increase not only with the level of noise distortion in the test sinewave (Figure A.3), but also with the applied sampling rate (Figure A.4). The latter trend reveals the principle weakness of the four-parameter algorithm -- a sudden breakdown in frequency estimates. At low noise levels (below 12 dB SNR) and relatively coarse sampling rates (less than 40 samples per cycle), the error of frequency estimates remains at a respectable level of  $10^{-3}$  (or 0.1%). As noise levels rise (with fixed sampling rate), estimation errors rise by as much as two factors of 10. However, the degradation is even more dramatic for

increasing sampling rates (referred to as *finer* sampling). As sampling rates rise above 40 samples per cycle, errors rise sharply to values on the order of  $10^1$  or more. Table A.III lists the range of average errors measured.

Following the same trends, *phase* estimates decline in accuracy for both high noise levels and sampling rates. At low noise levels (below 12 dB SNR) and sampling rates below 40 samples per cycle, average errors are on the order of  $10^{-1}$ . Higher noise levels increase the error by approximately one factor of 10 (Figure A.5). However, for the combined increase of distortion and sampling rate, errors grow up to the order of  $10^2$  (Figure A.6). Table A.IV lists the average errors.

## NONLINEAR SIMULATIONS

These simulations test the four-parameter algorithm's performance when the input sinewave is corrupted by nonlinear distortion (modeled by harmonics). This scenario represents the effect of using a practical ADC, with differential or integral nonlinearities, to generate the test signal. The basic procedure used for these simulations is the same as described under the "Linear Simulations" section. The only difference is the additive noise model. To model nonlinear distortion, a MATLAB-generated harmonic sequence is added to the original sinewave sequence before applying the estimation algorithm. Figure 18 shows the stages to create this model. Both second and third harmonics of the original sine were tested. For consistency, all harmonics shared common parameters: zero dc offset, random phase and variable peak-to-peak amplitude.



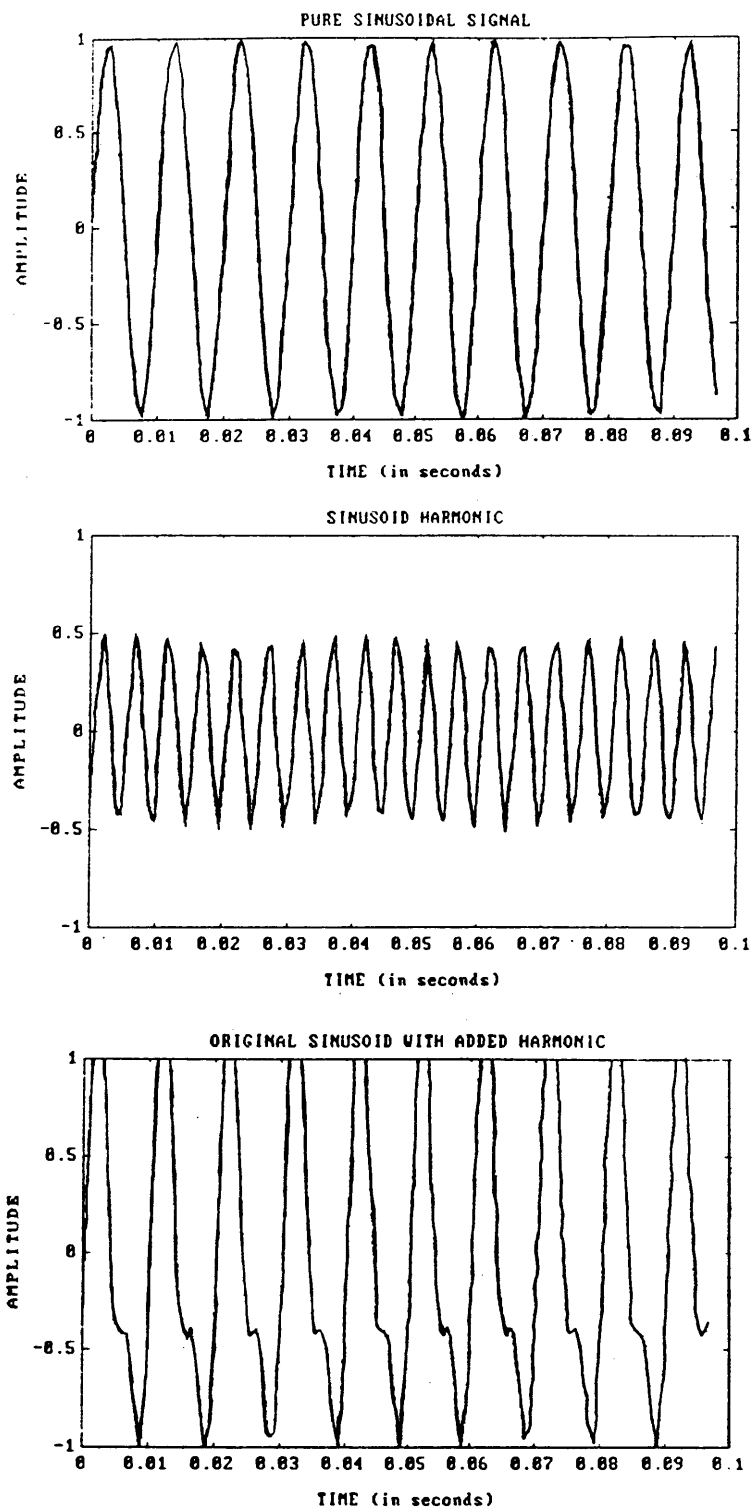


Figure 18. Modeling nonlinear (harmonic) distortion in the test sinewave.

## NONLINEAR RESULTS

Errors in estimating sinewave *amplitude* in the presence of nonlinear distortion, show one consistent trend. As Figure A.7 reveals, estimation errors increase solely with the level of nonlinear distortion present in the input signal regardless of the sinewave or harmonic frequency tested. The average error magnitude is on the order of  $10^{-2}$ . Table A.V lists all of the simulation averages.

Estimates of the sinewave's *dc offset* are extremely accurate (errors on the order of  $10^{-14}$ ) throughout the range of input signals tested. Table A.VI lists these error values. Figure A.8 reveals a general trend of slight error increases with rising noise levels. The erratic nature of the relation is due primarily to imprecision in measuring such small differences ( $10^{-14}$  or below) rather than inconsistency in computing the estimates.

*Frequency* estimates once again reveal the four-parameter algorithm's weakness. Table A.VII shows average estimation errors on the order of  $10^{-3}$  for harmonic levels below 10.5 dB SNR. As harmonic levels rise, average errors increase by a factor of 10. In Figure A.9, errors that jump rapidly to 100% near the end of the simulation range, do *not* represent a complete breakdown of the algorithm. Actually, that is the point where the magnitude of the distortion exceeds half the strength of the input test signal and thus becomes the dominant signal to the estimation algorithm. The large error occurs, because the algorithm is now estimating the frequency of the "harmonic" rather than the original sinewave. This magnitude of nonlinear distortion is unrealistic in practical situations, but simulation results were plotted to confirm dominant trends.

*Phase* estimates also deteriorate at high distortion levels. Errors on the order of  $10^{-1}$  (for distortion levels below 10.5 dB SNR), climb to the order of  $10^0$  as distortion rises. The effect is accelerated for higher frequency input sinewaves. Table A.VIII lists

average errors for phase estimates. Figures A.10 illustrates performance trends.

## CONCLUSIONS

The results of simulations reveal that the four-parameter algorithm provides quite accurate estimates under most conditions. Its major weakness is the rapid breakdown of the frequency estimator for certain input conditions. The critical conditions are:

- very high sampling rates relative to the sinewave frequency
- high levels of noise distortion present in the input sinewave

Problems resulting from the first condition can be avoided by careful selection of the sampling rate. To correct the second condition, one possibility is to eliminate some of the distortion present in the input before proceeding with the estimates. This idea will serve as the fundamental hypothesis for developing an improved estimation algorithm.

However, before investing a lot of effort to reduce distortion, it seems prudent to determine whether this plan will yield measurable improvement. Fortunately, another algorithm has been developed which can provide some idea. Chapter VI explores the performance of this "three-parameter algorithm."

## CHAPTER VI

### THE THREE-PARAMETER ALGORITHM

The Three-Parameter (or Known Frequency) Sine-Wave Curvefit Algorithm is a well-documented technique for estimating three sinewave parameters (amplitude, dc offset and phase) assuming the sinewave frequency is known and input to the algorithm [3]. This method's performance will indicate how much accuracy can be expected for the remaining estimates when the frequency is as accurate as possible (i.e. "known precisely"). This method has some important advantages. First, it yields a "closed form" solution for sinewave curve fitting in cases where the frequency of the recorded data is known. Second, unlike the four-parameter algorithm, it is a "linear" technique (i.e. estimates are computed entirely with linear functions such as "sums") which increases its accuracy. The performance of this algorithm will justify whether or not it is worthwhile to pursue improving the frequency estimate of the four-parameter method. In addition, the results of this algorithm will establish a benchmark against which the accuracy of the proposed (new) algorithm can be compared.

The following section presents a brief derivation of the three-parameter algorithm. A complete description is given in [3].

### THE PROCESS

The input to the three-parameter algorithm consists of a data record containing  $M$  samples of an input sinusoid measured at times  $t_n$ . The solution to the curve-fit problem

is assumed to have the form [3]:

$$y'_n = A \cos(\omega t_n) + B \sin(\omega t_n) + C \quad (6.1)$$

where  $\omega$  = known angular input frequency and  $t_n$  = sample times.

Although the method was derived with cosines, the algorithm is valid for all sinusoids. The variables A, B, and C are used to solve for the three unknown parameters of the sinusoid (amplitude, phase angle and dc offset). From the sampled data record and the assumed closed-form solution, the total residual error of the measured data relative to the best-fit sine is given by:

$$\varepsilon = \sum_{n=1}^M [y_n - x_n]^2 = \sum_{n=1}^M [y_n - A \cos(\omega t_n) - B \sin(\omega t_n) - C]^2 \quad (6.2)$$

To minimize this total error, the partial derivatives are taken with respect to each unknown variable (A, B, and C). Then, each of the partial derivatives is set equal to zero and the three equations are solved simultaneously for the variables of interest (A, B, and C). The results are given below:

$$A = \frac{A_N}{A_D} \quad (6.3)$$

where

$$A_N = \frac{\sum_{n=1}^M y_n \alpha_n - \bar{y} \sum_{n=1}^M \alpha_n}{\sum_{n=1}^M \alpha_n \beta_n - \bar{\beta} \sum_{n=1}^M \alpha_n} - \frac{\sum_{n=1}^M y_n \beta_n - \bar{y} \sum_{n=1}^M \beta_n}{\sum_{n=1}^M \beta_n^2 - \bar{\beta} \sum_{n=1}^M \beta_n}$$

$$A_D = \frac{\sum_{n=1}^M \alpha_n^2 - \bar{\alpha} \sum_{n=1}^M \alpha_n}{\sum_{n=1}^M \alpha_n \beta_n - \bar{\beta} \sum_{n=1}^M \alpha_n} - \frac{\sum_{n=1}^M \alpha_n \beta_n - \bar{\alpha} \sum_{n=1}^M \beta_n}{\sum_{n=1}^M \beta_n^2 - \bar{\beta} \sum_{n=1}^M \beta_n}$$

and

$$B = \frac{B_N}{B_D} \quad (6.4)$$

where

$$B_N = \frac{\sum_{n=1}^M y_n \alpha_n - \bar{y} \sum_{n=1}^M \alpha_n}{\sum_{n=1}^M \alpha_n^2 - \bar{\alpha} \sum_{n=1}^M \alpha_n} - \frac{\sum_{n=1}^M y_n \beta_n - \bar{y} \sum_{n=1}^M \beta_n}{\sum_{n=1}^M \alpha_n \beta_n - \bar{\alpha} \sum_{n=1}^M \beta_n}$$

$$B_D = \frac{\sum_{n=1}^M \alpha_n \beta_n - \bar{\beta} \sum_{n=1}^M \alpha_n}{\sum_{n=1}^M \alpha_n^2 - \bar{\alpha} \sum_{n=1}^M \alpha_n} - \frac{\sum_{n=1}^M \beta_n^2 - \bar{\beta} \sum_{n=1}^M \beta_n}{\sum_{n=1}^M \alpha_n \beta_n - \bar{\alpha} \sum_{n=1}^M \beta_n}$$

and

$$C = \bar{y} - A \bar{\alpha} - B \bar{\beta} \quad (6.5)$$

where the component terms are defined by:

$$\alpha_n = \cos(\omega t_n)$$

$$\beta_n = \sin(\omega t_n)$$

$$\bar{y} = \frac{1}{M} \sum_{n=1}^M y_n$$

$$\bar{\alpha} = \frac{1}{M} \sum_{n=1}^M \alpha_n$$

$$\bar{\beta} = \frac{1}{M} \sum_{n=1}^M \beta_n$$

Substituting the variables into the curve-fit equation assumed earlier, yields the solution in the form:

$$y'_n = A\cos(\omega t_n) + B\sin(\omega t_n) + C \quad (6.6)$$

Finally, the equations:

$$A_{\cos} = (A^2 + B^2)^{1/2}$$

$$\theta = \tan^{-1} \left( -\frac{B}{A} \right)$$

will convert the solution into the more familiar form of amplitude, phase, and dc offset:

$$y_n = A \cos (\omega t_n + \theta) + C \quad (6.7)$$

## ALGORITHM PERFORMANCE

### The Experiments

The key sensitivities used to analyze the three-parameter algorithm's performance are:

- The algorithm's accuracy for increasing levels of white noise distortion in the input sinewave.
- The algorithm's accuracy for incremental variations in the input frequency.

Results of the first sensitivity will reveal the accuracy of a "linear" estimation algorithm. If the results are favorable (i.e. low estimation errors), the results of the second sensitivity will determine how accurate the input frequency needs to be to utilize this algorithm. This algorithm was not tested for input sinewaves with nonlinear (or harmonic) distortion. The dominant effect of harmonic distortion is to corrupt the *frequency* of the original sinewave. Since, the original sinewave's frequency is an *input* to the algorithm, this test would be trivial.

## Sensitivity to Linear Distortion

The first sensitivity was tested by adding varying levels of white noise (to simulate quantization error from a practical ADC) to each sinusoidal test signal. To ensure a consistent comparison, the tests are identical to those described for the four-parameter algorithm (see Chapter V -- Linear Simulations).

## Linear Results

The three-parameter algorithm yields consistent estimates even with increasing levels of noise distortion in the input sinewave. Errors in *amplitude* estimates remained on the order of  $10^{-3}$  for SNR down to 8 dB. Table B.I shows the range of amplitude estimation errors for all noise levels and sampling rate combinations tested. Figure B.1 plots these errors with respect to increasing noise levels. Although there is no clear trend with respect to sampling rate, estimation errors increase gradually with increased sinewave distortion.

*DC offset* estimates with the three-parameter algorithm also show consistent accuracy. Once again, errors remain on the order of  $10^{-3}$  for most input conditions (see Table B.II). Furthermore, the maximum error for dc offset estimates is only half the error magnitude of amplitude estimates. The errors grow as distortion increases (Figure B.2), but vary independent of sampling rates.

There are no *frequency* estimates to evaluate because this parameter is input to the algorithm.

Finally, *phase* estimates follow trends similar to the other parameters -- errors increasing with noise distortion (Figure B.3), and relatively independent of sampling rates. Average errors remain on the order of  $10^{-3}$ . Table B.III lists all of the phase esti-



mation errors from simulations.

### Sensitivity to Input Frequency Variation

To test the algorithm's sensitivity with respect to *variations* in the input frequency, the initial input *value* is not critical. Only the relative *variations* about this value are important. However, to ensure a representative characterization, variations about two separate sinewave frequencies were tested -- one (50 Hz) from the lower end and one (200 Hz) from the upper end of the designated range of test frequencies. Various increments of these values were added or subtracted and the result used as the input frequency for the three-parameter algorithm. These increments ranged in factors of 10 from  $\pm 0.00001$  to  $\pm 0.1$  of the original frequency. Table IV shows how these increments translate into actual frequency inputs.

TABLE IV  
INCREMENTAL FREQUENCY VARIATIONS

Incremental Variation	Original Frequency	
	200 Hz	50 Hz
	Actual Input $f_1$	Actual Input $f_2$
+0.00001	200.002	50.0005
+0.0001	200.02	50.005
+0.001	200.2	50.05
+0.01	202.0	50.5
+0.1	220.0	55
-0.00001	199.998	49.9995
-0.0001	199.98	49.995
-0.001	199.8	49.95
-0.01	198	49.5
-0.1	180	45

Each frequency input was tested with two levels of linear distortion (8 dB and 2 dB SNR). This avoids biasing the results with one particular noise level.

### Results

Tables B.IV, B.V and B.VI list the results of sensitivities to input frequency for amplitude, dc offset and phase estimates respectively. In these tables, the first column shows the incremental frequency variation tested. The remaining columns are grouped into two sets -- the first shows results of variations about an original sine frequency of 50 Hz and the second, 200 Hz. Within each group, the first column lists the actual frequency input to the three-parameter algorithm. The second and third columns show the error of the particular parameter estimate assuming fixed SNR of 8 and 2 dB respectively.

The results of these sensitivities show some decrease in accuracy (or increase in estimation errors) for increasing variations about the input frequency. However, these error differences are generally quite small (less than 1%). This indicates that the three-parameter algorithm's performance is not extremely sensitive to small variations in the input frequency. What it means for this study is that the frequency estimate from the four-parameter algorithm needs improvement, but the three-parameter algorithm can tolerate some inaccuracy and still yield reliable estimates.

## CONCLUSIONS

The accuracy of parameter estimates using a "linear" estimation technique (such as the three-parameter algorithm) is more consistent than "nonlinear" methods manage to achieve. However, to take advantage of this linearity, the frequency of the test sinewave

must be input to the algorithm -- hence, it must be known in advance. For an unknown sinusoid, it is not always possible to determine its exact frequency. However, after investigating the three-parameter algorithm's sensitivity to variations in the input frequency, it appears that the input frequency can vary significantly from the signal's actual value without serious degradation of the other three estimates. Based on this analysis, it seems worthwhile to proceed with improving the four-parameter algorithm's frequency estimate. The improved estimate will become the required input to the three-parameter algorithm to take advantage of its consistent accuracy. The following chapter presents the development of an improved four-parameter estimation algorithm which incorporates linear estimation formulas.

## CHAPTER VII

### THE PROPOSED ALGORITHM

Chapter V revealed that the basic weakness inherent to the four-parameter algorithm is the breakdown of frequency estimates under low sampling rates and/or high noise levels. Chapter VI demonstrated that more consistent parameter estimates could be obtained with a linear technique such as the three-parameter algorithm. The disadvantage of this method is that it requires the frequency of the test sine wave as an input. In typical test situations with unknown sine waves as input, this is one of the parameters we are trying to find.

Previous simulations in this study have revealed basic characteristics of some existing techniques for parameter estimation. Armed with this information, one way to develop an improved algorithm is to build on their best features. It would be especially beneficial to utilize the linearity of the three-parameter algorithm by finding a way to provide the initial frequency input. Since the existing four-parameter algorithm can estimate the frequency of an unknown sinusoid, this estimate could serve as the required input. The proposed method adopts this approach. Therefore, the first step is to improve the four-parameter algorithm's frequency estimate.

### FILTER DESIGN AND APPLICATION

With the four-parameter method, the distorting effects of additive noise (found in practical test signals) degraded the accuracy of the frequency estimate. One way to

improve this estimate is to eliminate some of the distortion by filtering. Then, the four-parameter algorithm can be applied to the filtered signal to obtain a better frequency estimate.

The *type* of filter needed for this purpose can be determined by taking a look at the characteristics of the signal we want to preserve. Figure 19 shows the frequency spectrum of a sinusoid. Although only one is shown, the symmetrical spectrum consists of two impulses centered at the sinewave's fundamental frequency ( $f_0$ ); one on the positive and the other on the negative axis. Since negative frequencies have no physical meaning, it is common practice to plot only the positive half of the spectrum. This is the desired spectrum to preserve by filtering.

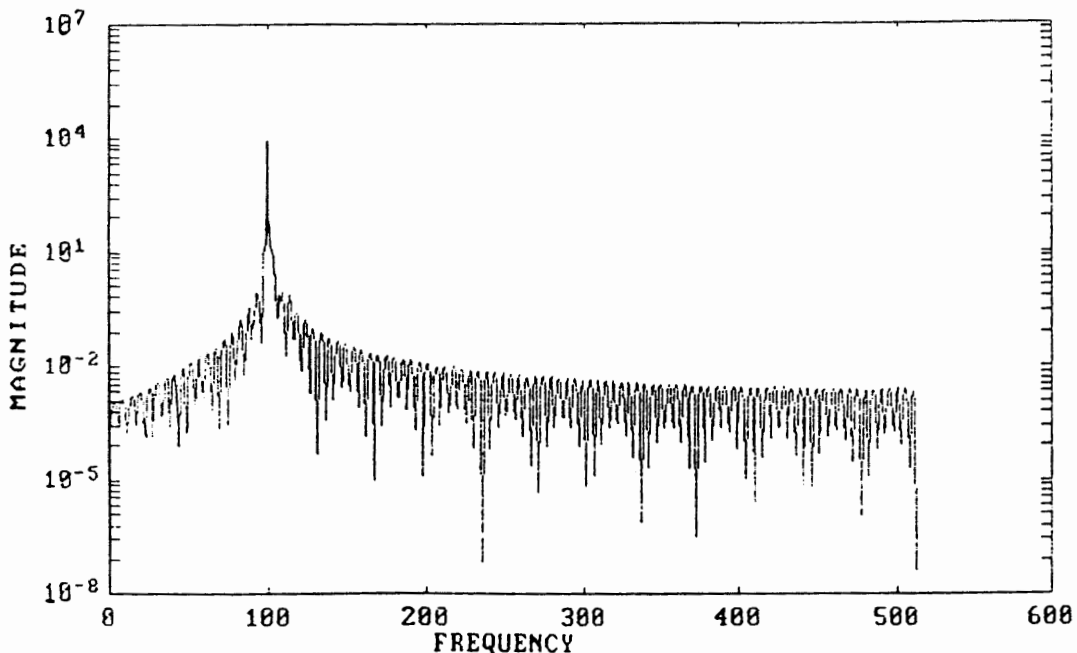
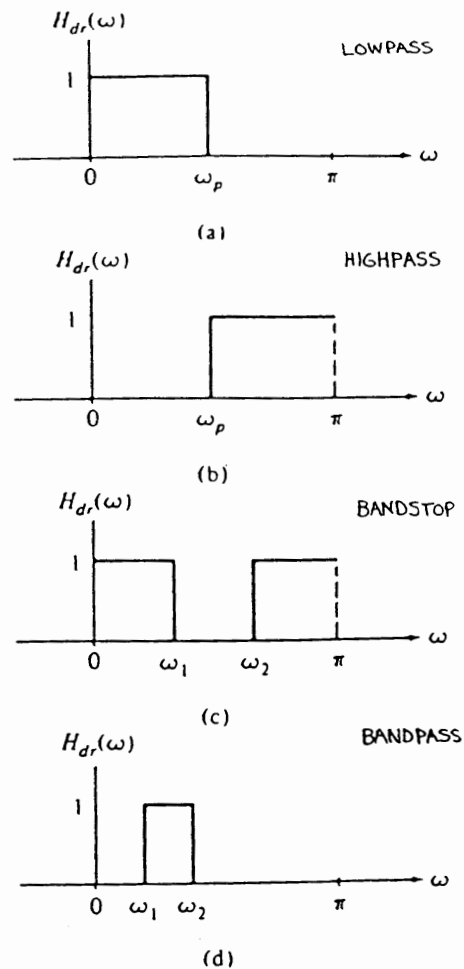


Figure 19. Frequency spectrum of the basic test sinewave.

Figure 20 shows the characteristics of some typical frequency selective filters. Based on the shape of the sinewave spectrum, a bandpass filter seems most appropriate.



**Figure 20.** Characteristics of some common frequency selective filters.

A bandpass filter is designed to pass all frequency components of the input signal within a certain range (the passband), and eliminate all frequency components outside of this range. This type of filter will preserve the narrow band of frequencies surrounding the sinewave impulses while eliminating any extraneous frequency components (due to noise) around them.

In order to design a bandpass filter, some basic parameters must be chosen. First, a value must be selected for the center of the passband, so the shape of the bandpass filter can be designed around this central point. Given only the data record for the original

noisy signal, there is a way to obtain a rough estimate of the sinewave's fundamental frequency. Applying the Fast Fourier Transform (FFT) to the noisy data, yields a spectrum similar to Figure 21.

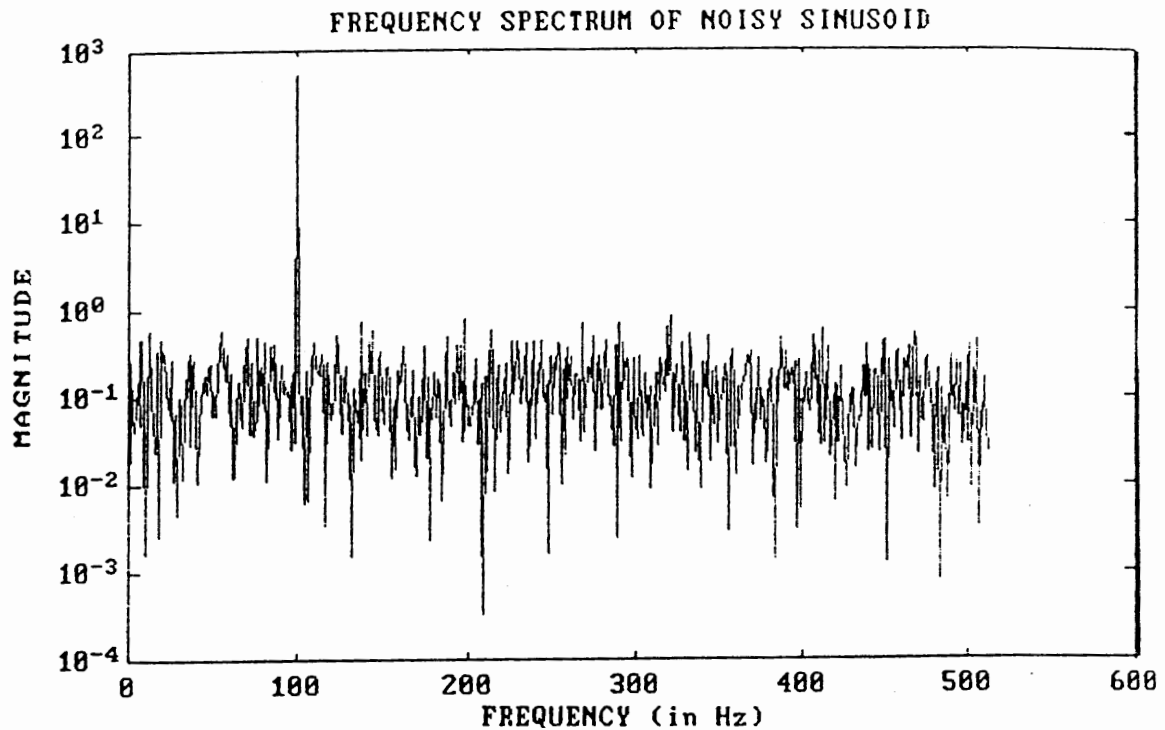


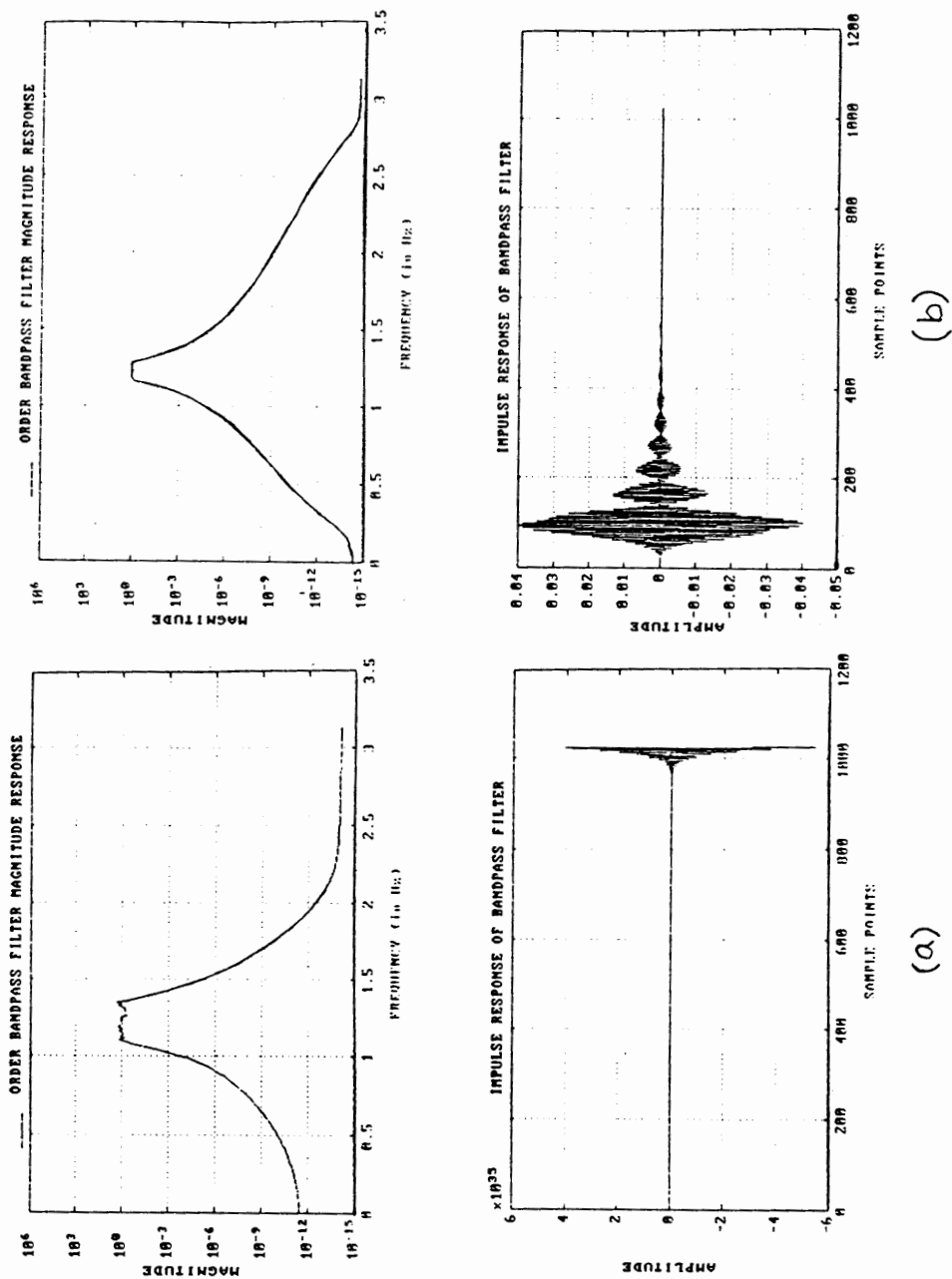
Figure 21. Frequency spectrum of a practical test sinewave (including distortion).

Although the spectrum has many components due to noise distortion, the impulse representing the sinewave is distinct. The location of this impulse provides a rough initial frequency estimate for bandpass filter design.

To keep design computations to a minimum, a basic Butterworth bandpass filter was chosen and simulated with MATLAB. The next step is to select values for the filter's order, bandwidth, and cutoff frequencies. In the following sections, the effect of each parameter will be considered individually.

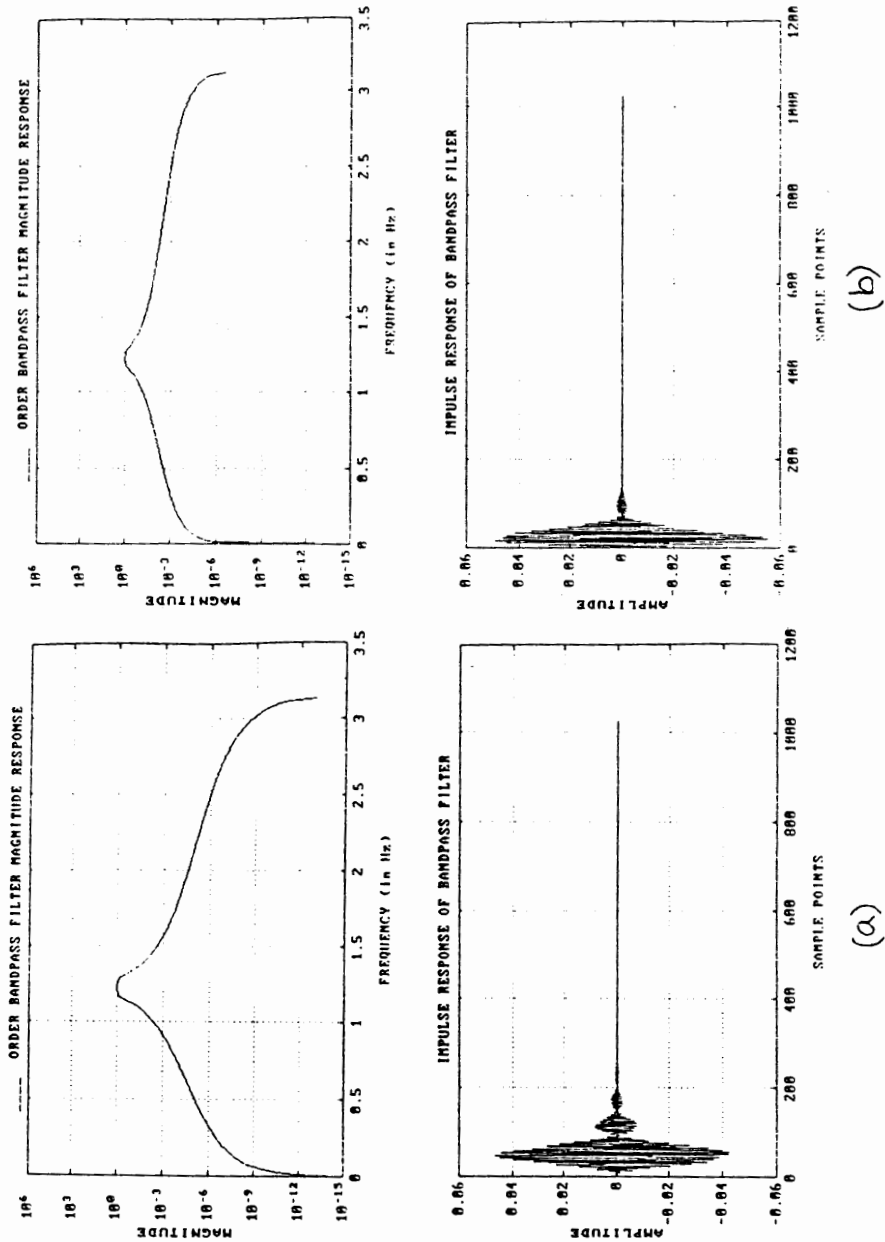
The filter's *order* basically determines its shape in the frequency domain (the sharp-

ness of its cutoff). Figures 22 and 23 show Butterworth bandpass filters of different orders which were generated with MATLAB. These figures illustrate how lower-order filters lose the desirable sharp cutoff.



**Figure 22.** Magnitude (upper) and impulse (lower) responses for (a) 16th order and (b) 8th order Butterworth bandpass filters.





**Figure 23.** Magnitude (upper) and impulse (lower) responses for (a) 4th order and (b) 2nd order Butterworth bandpass filters.

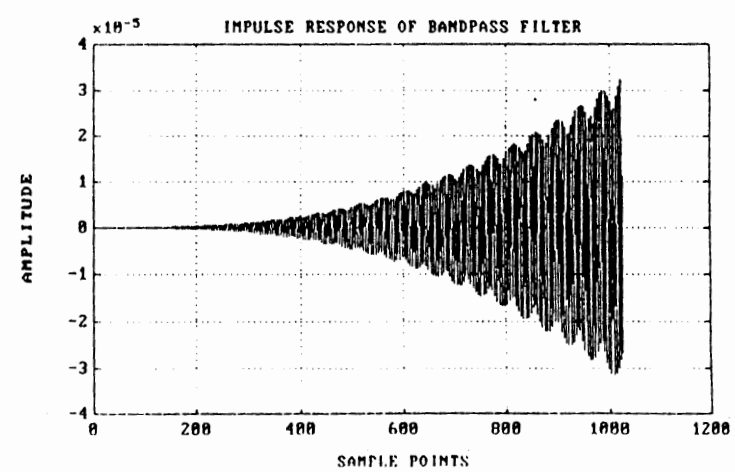
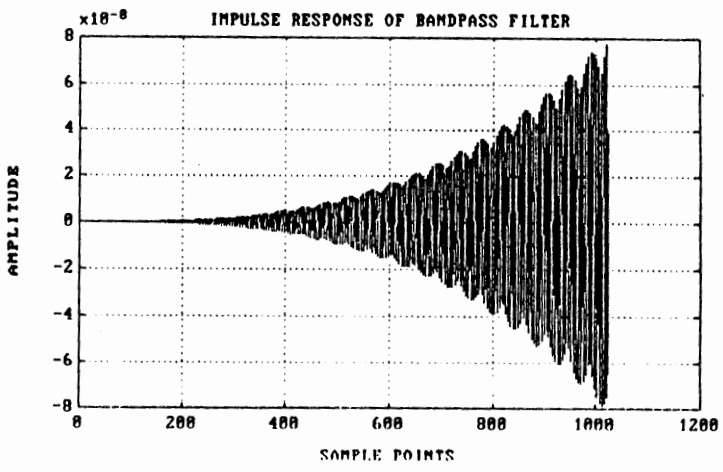
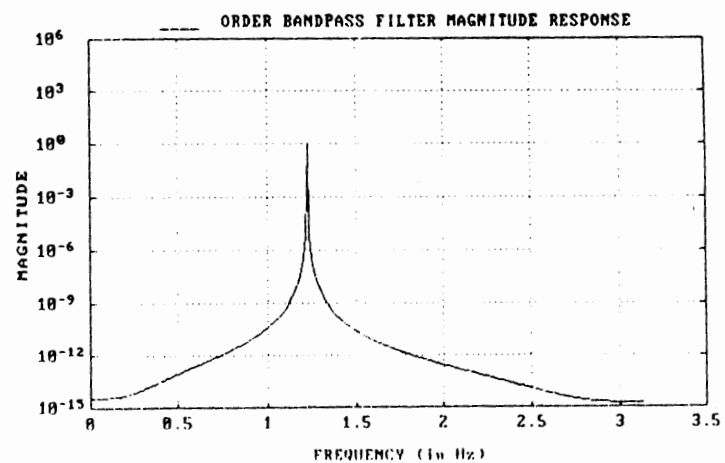
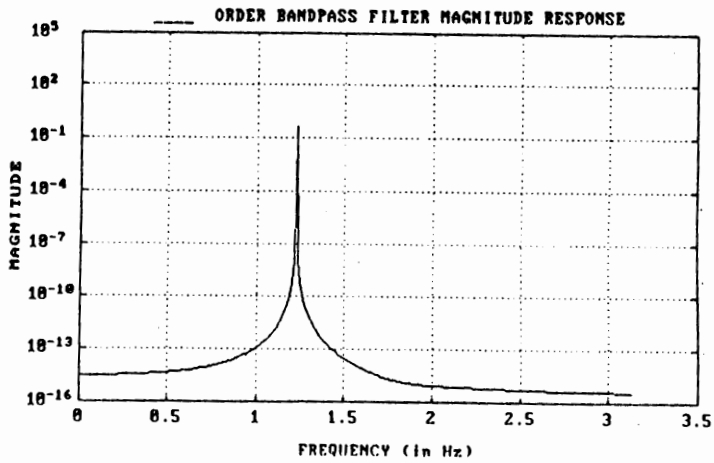
However, using MATLAB required a compromise in the ideal filter shape. With this software, the time required to generate a higher order filter of the desired length, (number of points), soon became prohibitively long for the iterative simulations required. Since the main goal is to improve the four-algorithm's frequency estimate and not

optimal filter design, a fourth order filter was chosen as a reasonable compromise. This value is high enough to ensure an adequate cutoff, yet low enough to generate rapidly with MATLAB.

Next, the *bandwidth* of the filter must be chosen. There are two main factors to consider in this choice. Narrow bandwidths, have the advantage that more of the distorting frequency components are outside of the passband, and therefore eliminated. However, since frequency and time are inverse quantities, a narrow frequency domain response has the disadvantage of causing a longer transient response in the time domain. The filtering process imposes these transients onto the original sinewave. Eliminating this interference requires truncating the transient samples from the data. Since a filter is applied to reduce distortion, the process itself shouldn't introduce unwanted components. The best compromise is to design the bandwidth as narrow as possible, without producing excessive time domain transients. In MATLAB, all filter design functions operate with "normalized frequencies" so that the system sampling rate is not required as an additional input argument. With this convention, frequencies in Hertz are normalized to half the sampling frequency (divide by  $\frac{f_{\text{sample}}}{2}$ ). For consistency throughout the remaining discussion, all filter parameters are assumed to be normalized (unit-less) values.

After experimenting with several bandwidth/truncation combinations, a normalized bandwidth of 20 was selected ( $\pm 10$  about the sinewave's fundamental frequency). Figures 24, 25 and 26 compare the frequency and time domain responses for a fourth order bandpass filter with a variety of bandwidths (from  $\pm 0.001$  to  $\pm 100$ ). The selected width ensures that the impulses in the sinewave spectrum will fall within the filter's passband, even if the central frequency estimate (from the FFT) is slightly off. Figure 27 shows the magnitude and impulse response of the final filter design.

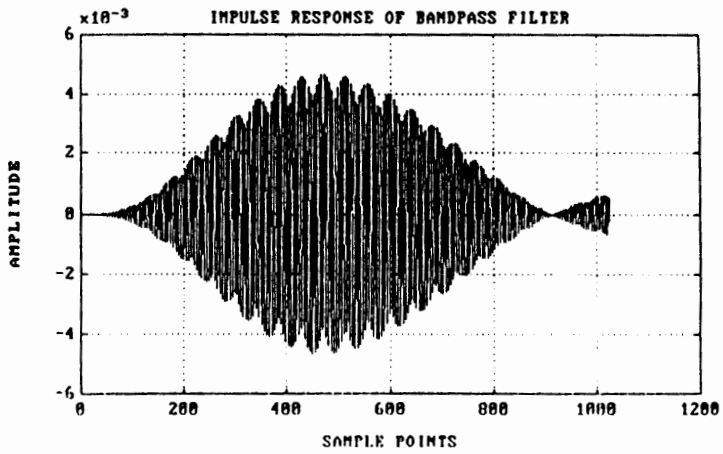
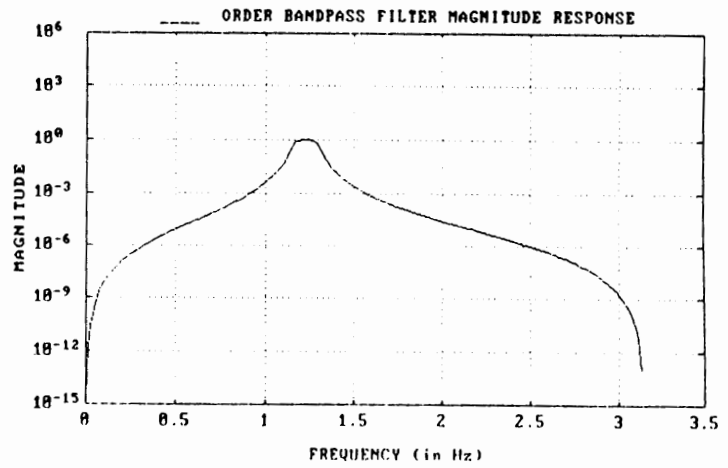
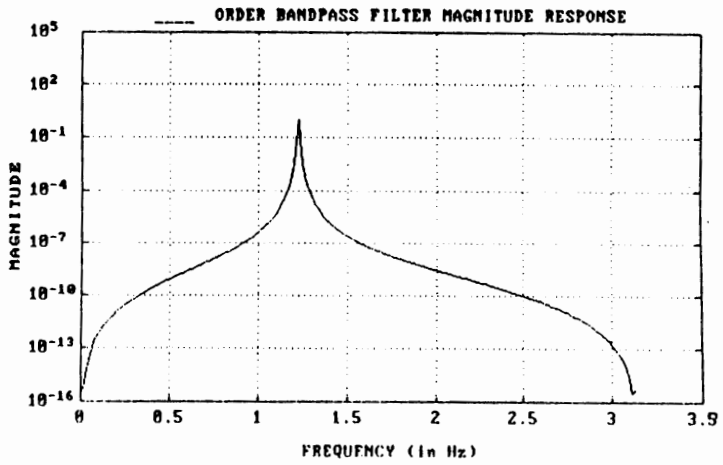
Figure 24. Magnitude (upper) and impulse (lower) responses for a bandpass filter with a bandwidth of (a)  $\pm 0.01$  and (b)  $\pm 0.1$ .



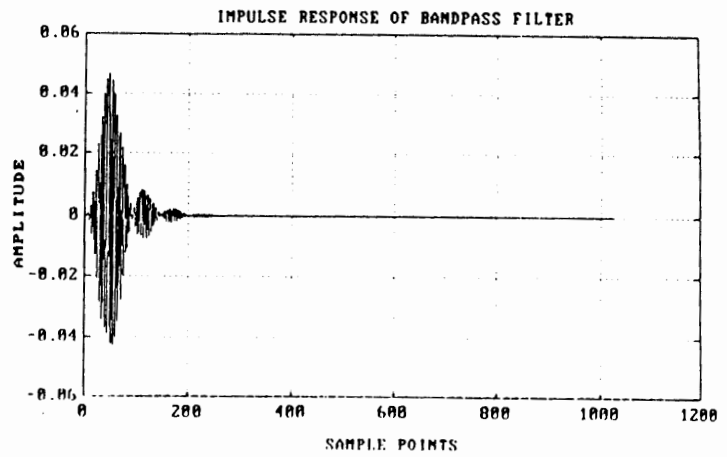
(a)

(b)

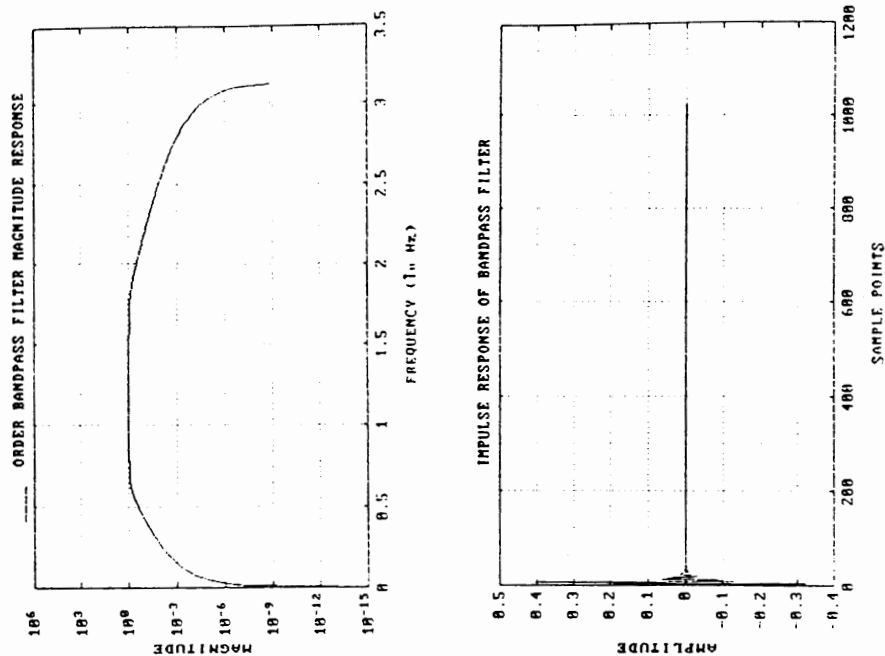
Figure 25. Magnitude (upper) and impulse (lower) responses for a bandpass filter with a bandwidth of (a)  $\pm 1.0$  and (b)  $\pm 10$ .



(a)



(b)



**Figure 26.** Magnitude (upper) and impulse (lower) responses for a bandpass filter with a bandwidth of  $\pm 100$ .

Eliminating filtering transients generated with the chosen bandwidth requires truncating roughly 100 sample points from each end of the filtered waveform to avoid corrupting the data. However, over 80% of the original data is preserved, with much less distortion present. Figure 28 compares the frequency spectrums of: the original (noisy) test sinewave, the signal after filtering, and after truncating filter transients. Filtering eliminates unwanted frequency components due to distortion and truncation eliminates any remaining sidelobes to produce the final smooth spectrum.

## THE PROCESS

After designing the bandpass filter, the next step is to apply it to eliminate some distortion from the original test signal. Using the filtered signal as input to the four-parameter estimation algorithm, yields an improved estimate of the original sinewave's

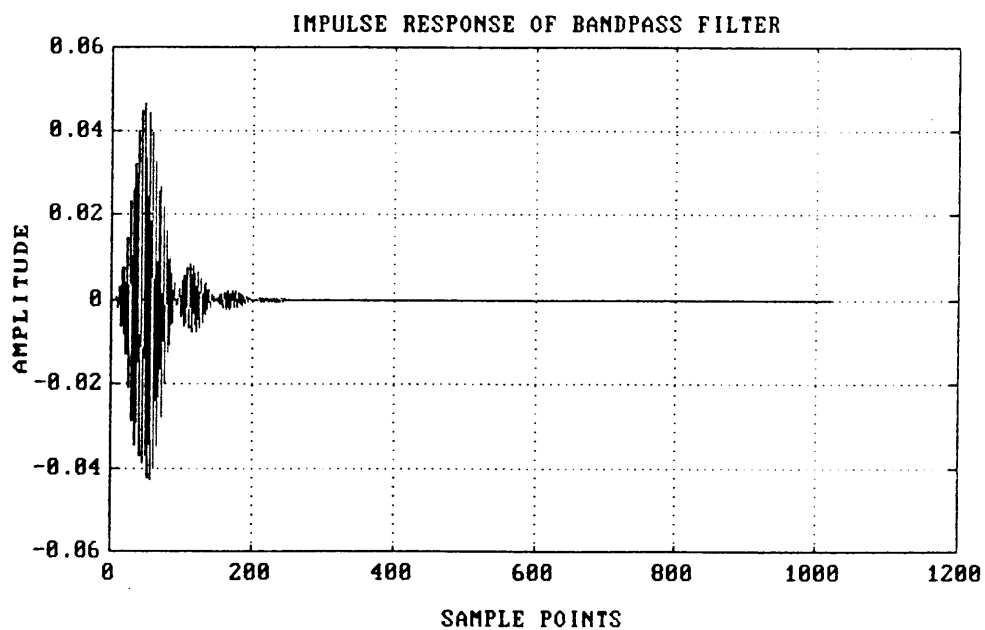
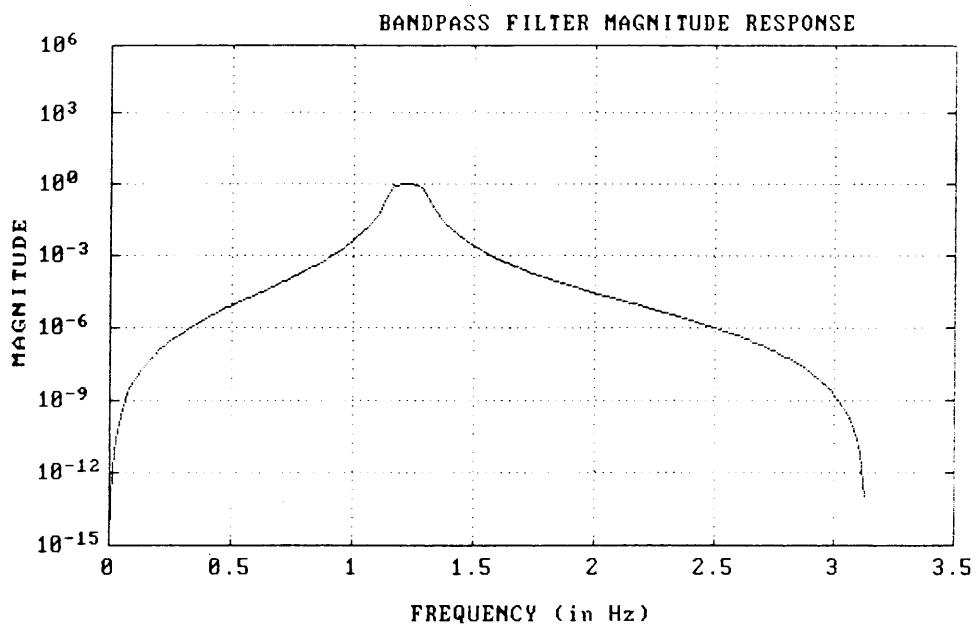
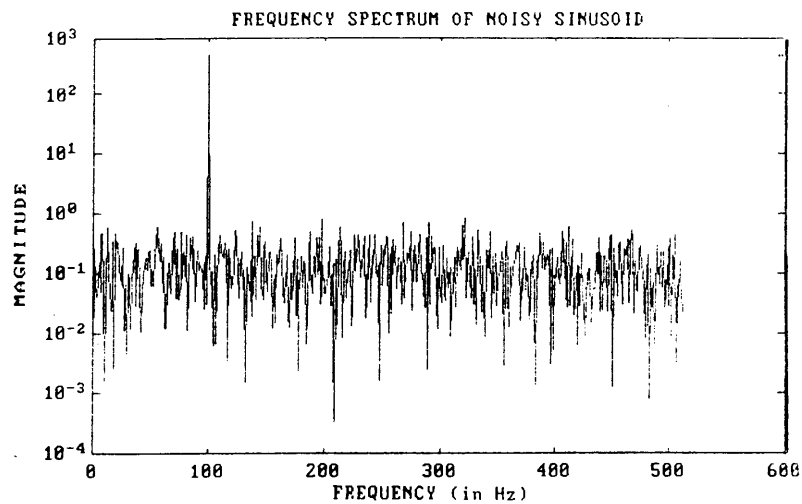
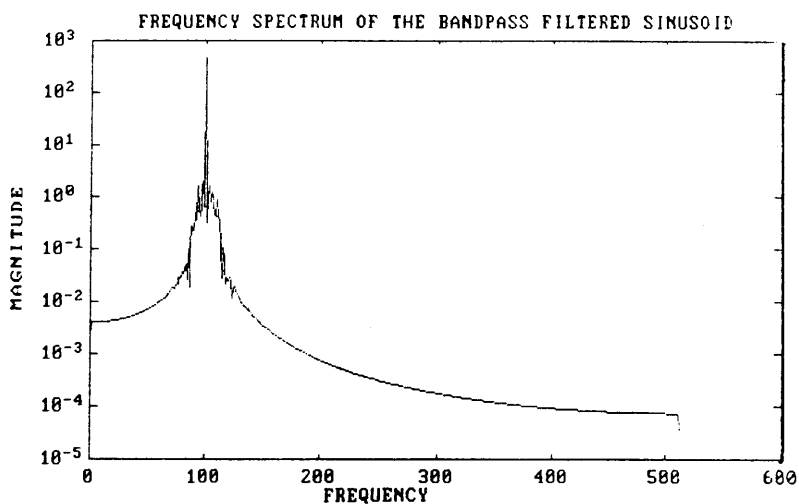


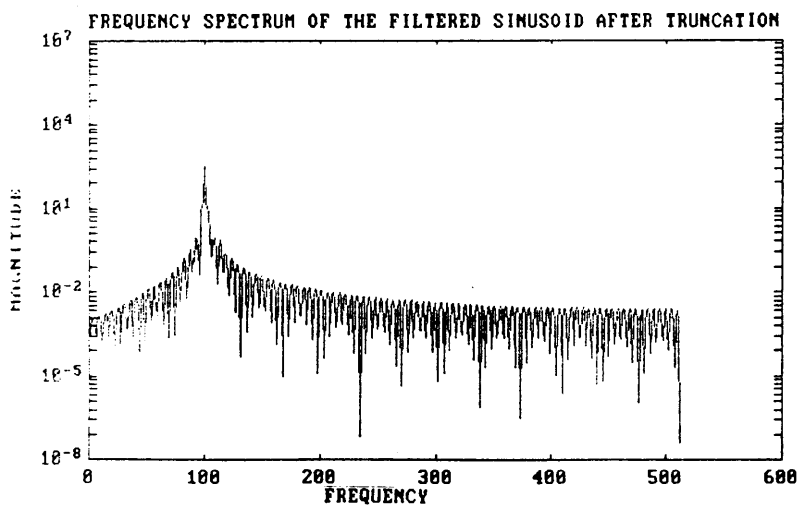
Figure 27. Magnitude (a) and impulse (b) responses of a fourth order Butterworth bandpass filter with bandwidth = 20.



(a)



(b)



(c)

**Figure 28.** Frequency spectrum of (a) the original noisy sinewave, (b) the filtered sinewave and (c) the result after truncating transients.

frequency. This value, together with the original (noisy) data record, are input to the three-parameter algorithm and values for the three remaining sinewave parameters (amplitude, dc offset, and phase), are computed.

This completes the description of the proposed algorithm. Figure 29 illustrates the process with a flowchart. The following section documents the results of simulations with this method.

## SIMULATIONS

To compare the performance of the proposed algorithm with Jenq's existing four-parameter method, the previous sensitivities were repeated with the new method.

- Performance with respect to increasing levels of "linear" distortion
- Performance with respect to increasing levels of "nonlinear" distortion

Chapter V discussed how these simulations were performed and presented the results for the four-parameter algorithm. The following sections present corresponding results for the proposed method. Chapter VIII will compare their performance.

### Linear Simulations

The first sensitivity was tested by applying the proposed algorithm to sinewaves with varying levels of uniformly distributed random noise added (to represent linear distortion). There is one difference between these simulations and those testing the four-parameter algorithm. Due to the filtering step in the proposed algorithm, input sinewaves with frequencies of either 10 or 5 Hz were not tested. The chosen bandpass filter design (bandwidth = 20) is too wide to filter sinewaves with these fundamental frequencies. Adequate data is provided by the other simulations to characterize the proposed algorithm's performance without testing sinewaves of these particular frequencies.



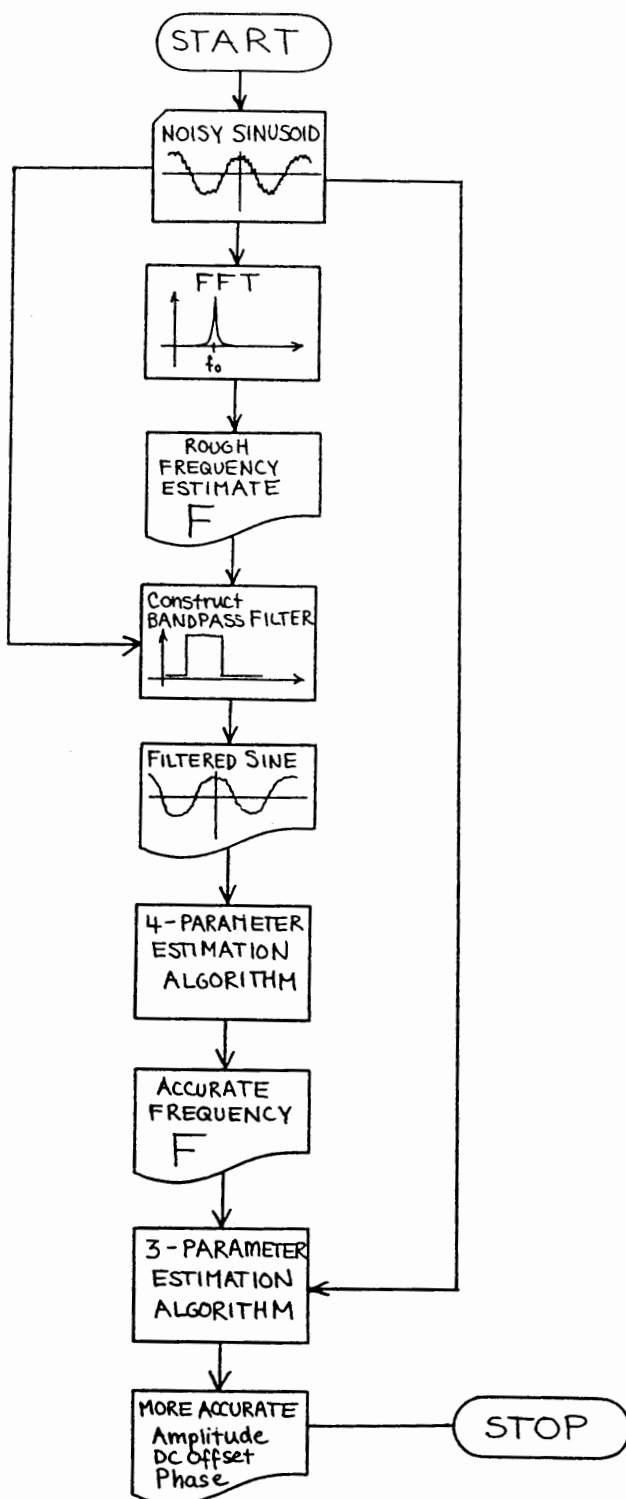


Figure 29. Process flowchart for the proposed algorithm.

Furthermore, altering the filter design for these two cases would result in larger time-domain transients and inconsistent comparisons with results of previously filtered waveforms.

### Linear Results

*Amplitude* estimates with the proposed algorithm remain fairly consistent over the range of noise level/sampling rate combinations tested. Average errors were on the order of  $10^{-3}$ . Table C.I lists the resulting errors from each simulation. Estimation errors increase gradually with rising noise levels (Figure C.1), but show no distinct trend with respect to varying sampling rates (Figure C.2).

*DC offset* estimates show even greater precision. Errors remain on the order of  $10^{-3}$  or lower throughout the simulations (Table C.II). Estimation errors vary independently of sampling rates (Figure C.4) and increase even more gradually than those of amplitude estimates, as noise levels rise (Figure C.3).

*Frequency* estimates obtained with the proposed algorithm show significant improvement from using a linear technique. Average estimation errors remained on the order of  $10^{-3}$ , increasing only gradually for either increased sampling rates (Figures C.6) or noise levels (Figure C.5). Table C.III lists average error values.

*Phase* estimates still show some inconsistency in performance. Average errors (Table C.IV) ranged from magnitudes of  $10^{-1}$  to  $10^0$  and showed rather erratic trends with respect to increasing sampling rates (Figure C.8) or noise levels (Figure C.7). This behavior for the proposed algorithm could be expected since trends are similar to those observed for the three-parameter method.

## Nonlinear Simulations

Next, the proposed algorithm is applied to sinewaves with varying levels of second or third harmonics added. The added harmonics represent nonlinear distortion (or nonlinear error) often present in practical test signals generated by an ADC. This will test the proposed method's sensitivity to nonlinear distortion present in the input test signal.

## Nonlinear Results

*Amplitude* estimates for sinewaves with nonlinear distortion were very accurate, with average errors between  $10^{-7}$  and  $10^{-5}$ . Table C.V shows the complete listing of simulation results. Errors remained fairly consistent until harmonic levels exceeded about 5 dB SNR, then gradually increased. Amplitude estimates decreased in accuracy with lower frequency input sinewaves. Reducing the frequency by one quarter of the previous value increased average errors by a factor of 10. Figure C.9 plots the results of each simulation.

*DC offset* estimates show similar accuracy. Average errors gradually rise from  $10^{-5}$  (for harmonic levels of 20 dB SNR or less) to  $10^{-4}$  (above 20 dB). Table C.VI lists these values. Errors show a definite increase with harmonic levels, but no distinct trends with respect to input sinewave frequencies (Figure C.10).

*Frequency* estimates are affected by both harmonic levels and test signal frequencies. Average errors are on the order of  $10^{-2}$  for an input sine frequency of 25 Hz,  $10^{-4}$  for 100 Hz frequency, and  $10^{-5}$  for 400 Hz. Table C.VII shows these values. Frequency estimations errors increase gradually with higher levels of nonlinear distortion in the input sinewave. Figure C.11 graphs the simulation results.

*Phase* estimates remain fairly consistent, with errors on the order of  $10^{-3}$  (Table C.VIII). Errors increase slightly with higher harmonic levels, but have no apparent relation to the frequency of the input sine. Figure C.12 plots these results.

In most cases, distortion from the second harmonic causes slightly larger estimation errors in cases where: the second harmonic amplitude is large, or the frequency of the input sinewave is high.

## CONCLUSIONS

The proposed algorithm attempts to combine the best features of the existing three and four-parameter methods with an additional filtering step to develop an improved technique for estimating all four parameters of an unknown digitized sinewave. The result is an algorithm which produces consistent estimates even for sinewaves corrupted by linear or nonlinear types of distortion. With the proposed method, estimation errors are quite low (suitable for effective bits measurement) and do not exhibit the extreme sensitivity to system sampling rate that the existing four-parameter method does.

## CHAPTER VIII

### COMPARISON OF ALGORITHMS

Each of the three preceding chapters focused on the performance of a specific algorithm for sinewave parameter estimation. After gathering data from many simulations and averaging the results, it is possible to compare the performance of the previous and proposed four-parameter algorithms empirically. This chapter presents a quantitative comparison with particular emphasis on the following issues:

- (1) The improvement in performance of the proposed algorithm over the previous four-parameter method.
- (2) How closely the accuracy of the proposed algorithm approaches the reference performance of the three-parameter algorithm.

This comparison is organized as follows. First, the performance of the existing and proposed four-parameter algorithms will be compared for input sinewaves corrupted by *linear* distortion (to evaluate issues (1) and (2)). Then, the same analysis will be repeated with *nonlinear* distortion present in the input sinewaves (to further evaluate issue (1)).

#### LINEAR DISTORTION

As described in Chapter IV (Methodology), each algorithm was simulated using an input sinewave with varying levels of uniformly distributed white noise added. The added distortion represents quantization errors which are inevitably introduced into the original sinusoid during the process of generating a digital data record.

## COMPARISONS

The proposed algorithm performed consistently better than the existing four-parameter method for sinewave *amplitude* estimates. Figure 30 shows that errors are lower with the proposed method at all noise levels. On average, errors are reduced by a full factor of 10 from the previous method. This accuracy approaches the three-parameter algorithm's performance (which represents the maximum expected from improved frequency estimates).

For estimates of *dc offset*, the proposed algorithm again displays lower errors overall. Figure 31 illustrates this comparison. In this case, the improvement is on the order of 0.01 and average errors are nearly at the level achieved by the three-parameter method. In fact, for noise levels below 12 dB SNR, estimates with the proposed method are slightly more accurate (differences on the order of  $10^{-3}$ ).

*Frequency* estimates exhibit mixed results. As Figure 32 shows, the existing four-parameter algorithm shows greater accuracy (lower errors) for distortion levels below about 6 dB SNR. However, there is a crossover around that point, and as distortion levels increase, the existing algorithm begins to break down while the proposed method's errors remain consistently lower (differences up to  $10^4$ ).

Figure 33 compares *phase* estimation errors for all of the algorithms. Again the results are mixed. At low distortion levels, the existing four-parameter method exhibits lower errors. However, this algorithm eventually breaks down as noise levels exceed 4 dB, while the proposed method maintains its initial accuracy (an improvement up to two factors of 10 over the existing method).

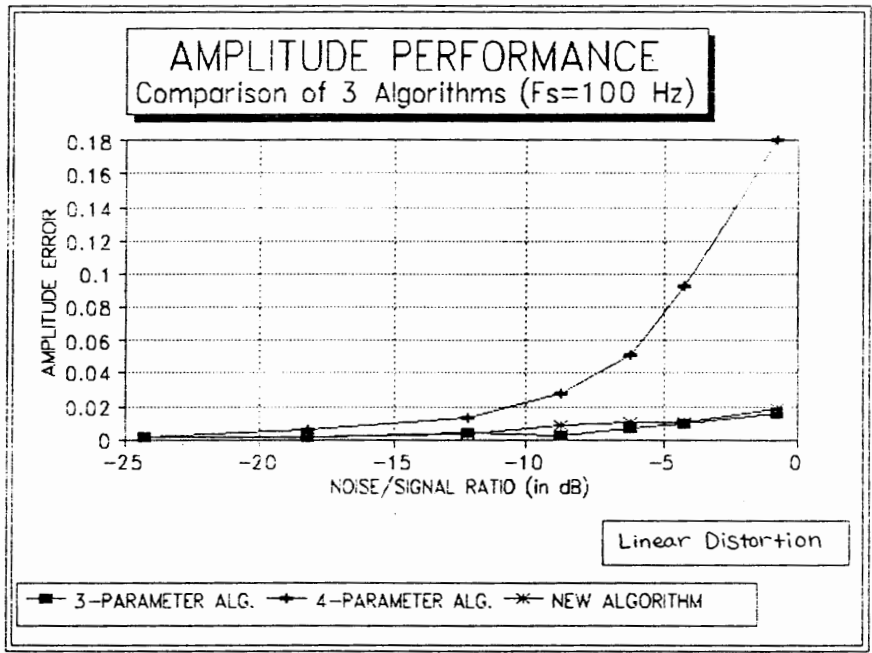


Figure 30. Comparison of amplitude estimation errors -- linear distortion.

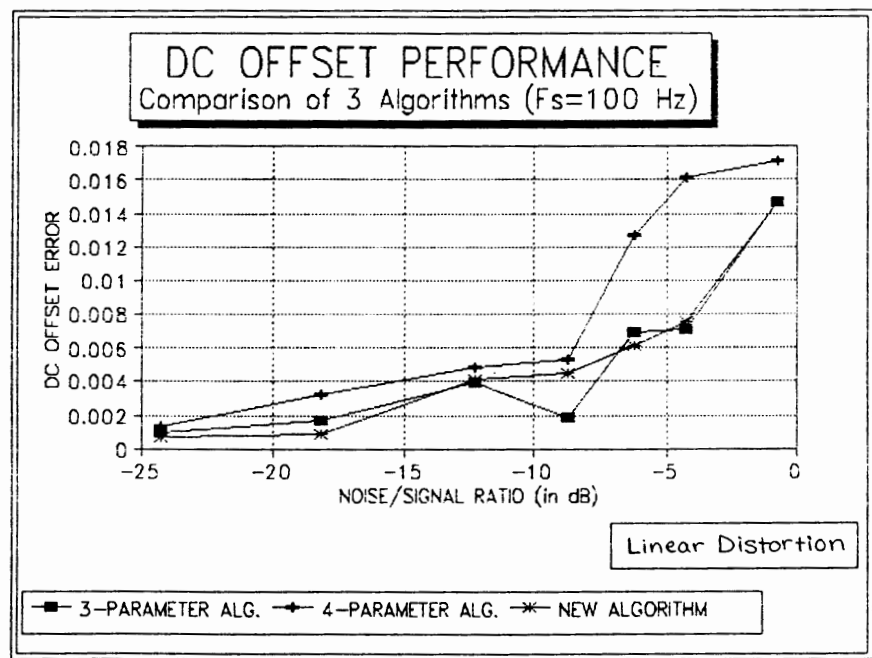
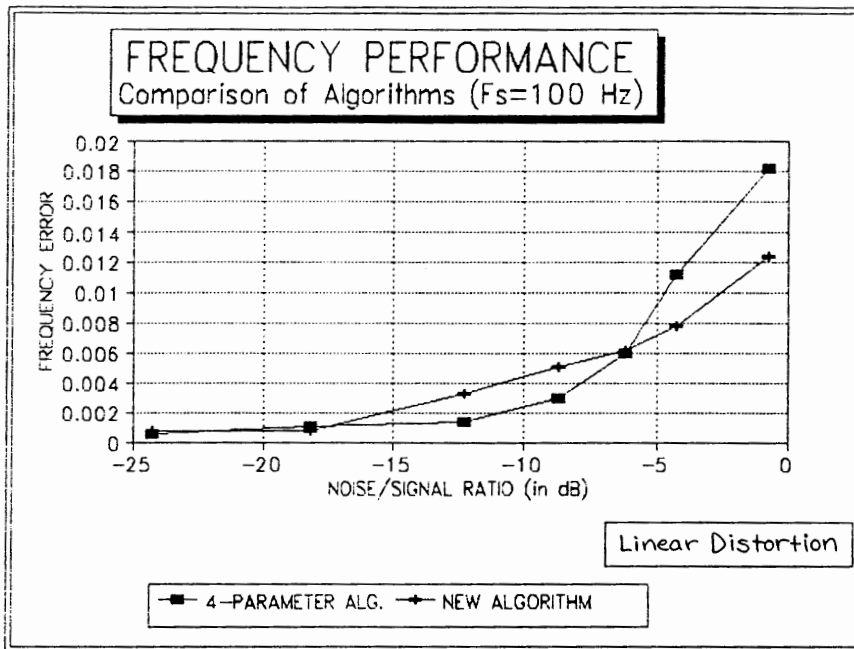
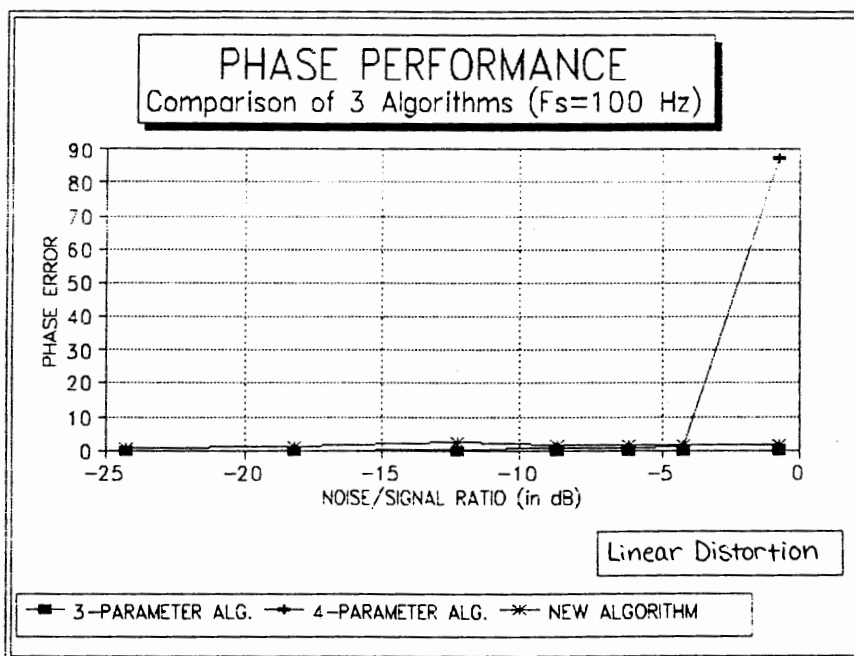


Figure 31. Comparison of dc offset estimation errors -- linear distortion.



**Figure 32.** Comparison of frequency estimation errors -- linear distortion.



**Figure 33.** Comparison of phase estimation errors -- linear distortion.



## NONLINEAR DISTORTION

In addition to analyzing each algorithm's performance in the presence of linear distortion, similar tests were also run assuming the test signals were corrupted by various levels of *nonlinear* distortion (represented by *harmonics* of the input sinewave). These simulations model the second major type of distortion that may occur to signals processed by practical ADCs. Because the principal distorting effect of harmonics comes from their frequency component, this type of distortion is not a meaningful test to characterize the performance of the three-parameter algorithm -- since the signal's true frequency is an input to the algorithm. Therefore, in the following discussion, only results from the four-parameter and proposed algorithms will be compared.

## COMPARISONS

Beginning with *amplitude* estimates, the proposed algorithm exhibits lower errors at every noise level tested. Figure 34 illustrates this comparison. Average errors with the proposed method are on the order of  $10^{-7}$  to  $10^{-5}$ ; an improvement up to four factors of 10 over the existing four-parameter algorithm.

For *dc offset* estimates, the existing four-parameter algorithm maintains an edge over the proposed method. As Figure 35 reveals, the existing algorithm produces lower errors at all noise levels tested, but both algorithms exhibit relatively low errors overall (magnitudes of  $10^{-4}$  for the proposed method and  $10^{-14}$  for the existing). The difference is caused by the filtering stage in the proposed method. Since the existing four-parameter algorithm does not impose filter transfer functions on the input signal, and harmonics with zero dc offset offer no corruption, extremely accurate dc offset estimates are achieved.

*Frequency* estimates follow a pattern similar to results from linear sensitivities. For nonlinear distortion below about 6 dB SNR, Figure 36 reveals that the existing and proposed four-parameter algorithms have similar accuracy (the maximum error difference being 0.0004 or less). At higher distortion levels (above 6 dB SNR), estimates with the existing four-parameter method deteriorate rapidly. The proposed algorithm does not break down at these levels and errors are on the order of  $10^{-4}$  to  $10^{-6}$  lower.

The proposed method's *phase* performance shows improvement over the existing four-parameter algorithm at all distortion levels (see Figure 37). Phase estimation errors from the proposed algorithm are consistently lower (on average by  $10^{-3}$ ) than those from the existing method.

## WRAPPING UP

This chapter was devoted to a relatively detailed "quantitative" comparison of the various algorithms' accuracy. The following chapter "qualitatively" summarizes this study's basic findings and suggests some topics for future research in this area.

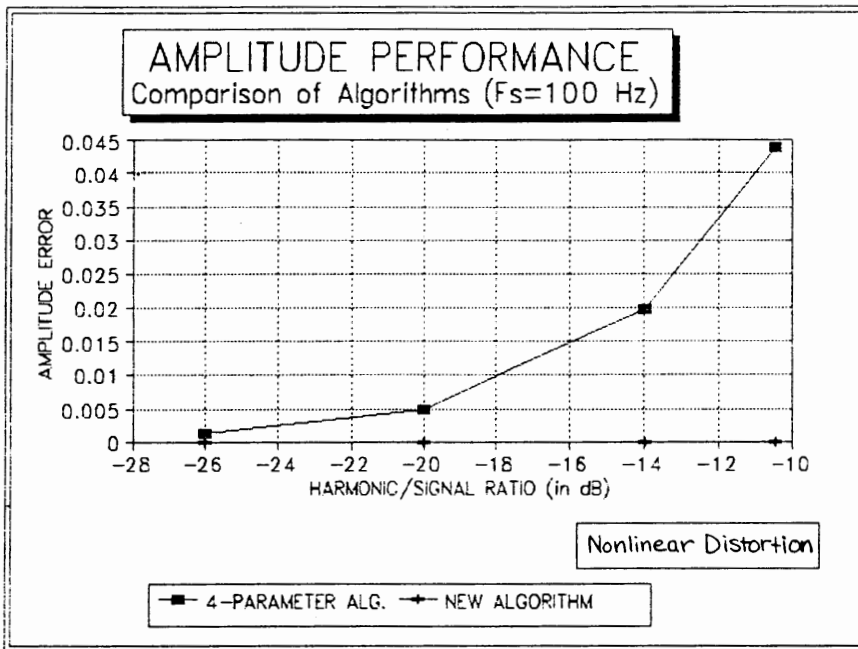


Figure 34. Comparison of amplitude estimation errors -- nonlinear distortion.

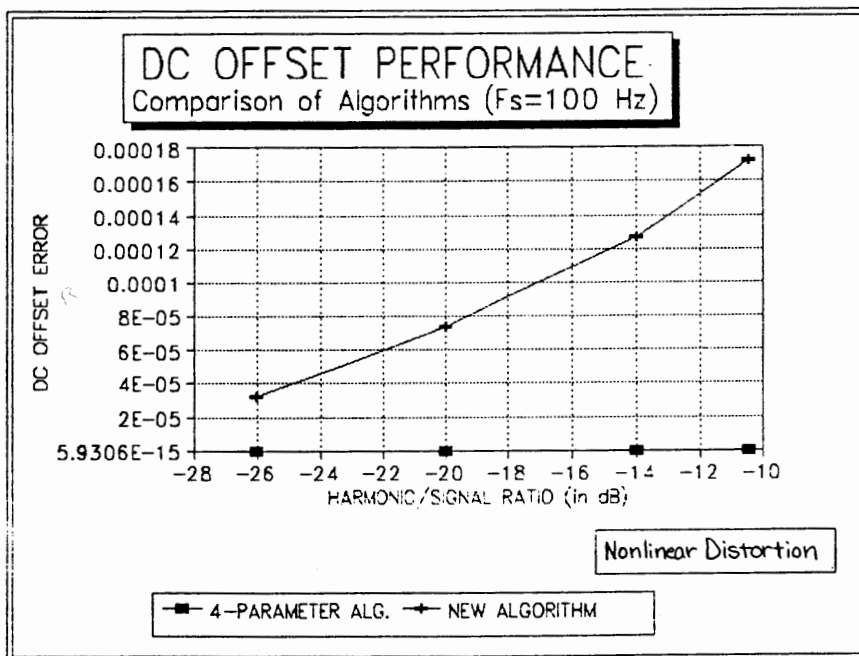
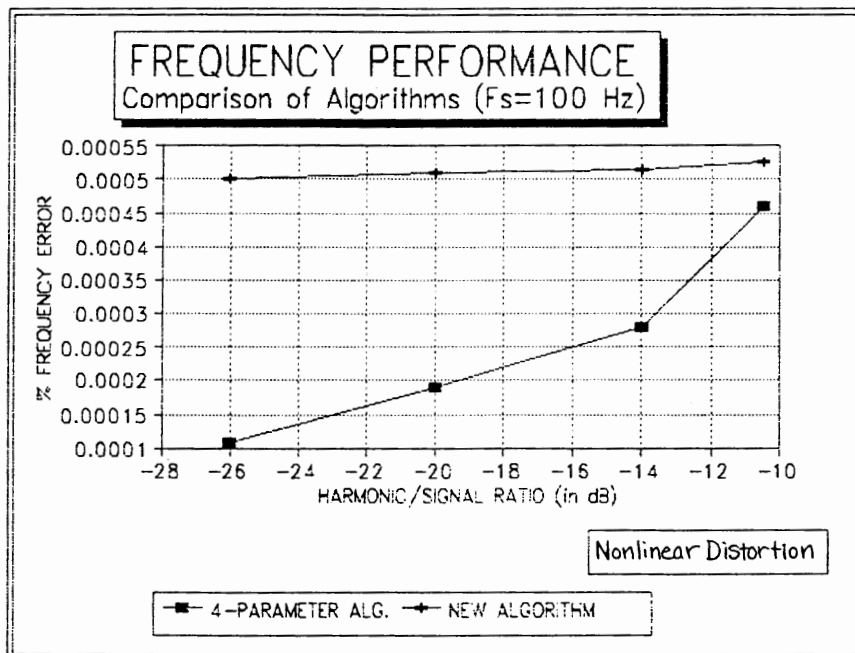
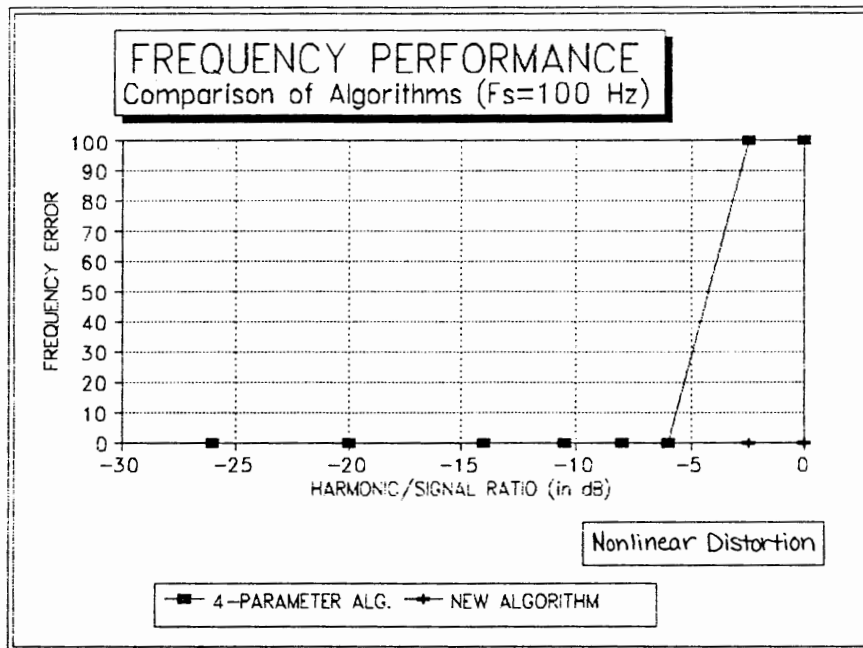
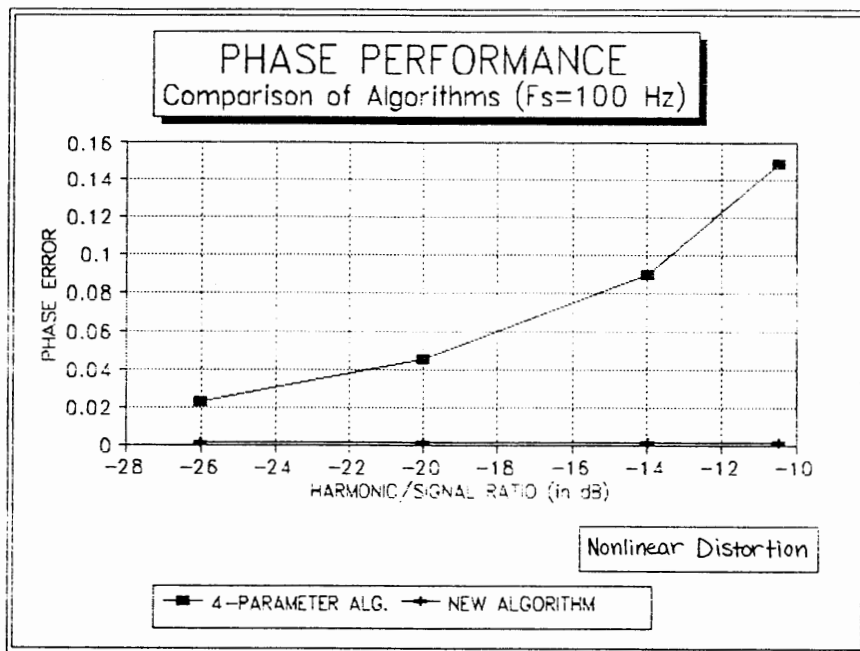


Figure 35. Comparison of dc offset estimation errors -- nonlinear distortion.



**Figure 36.** Comparison of frequency estimation errors -- nonlinear distortion.



**Figure 37.** Comparison of phase estimation errors -- nonlinear distortion.

## CHAPTER IX

### CONCLUSIONS

This study's main objective was to develop an improved algorithm for sinewave parameter estimation. The general process starts with a finite record of digitized data (obtained by sampling an unknown sinewave) and determines from it, the four parameters of the original analog signal -- amplitude, dc offset, frequency and phase angle. To accomplish this for the ideal case (where the input signal is a pure sinewave) is not easy, but the problem becomes even more complicated for practical applications in a noisy environment.

The sampling process (used to acquire the data record), is usually performed by an ADC. This device can introduce several types of distortion (errors) into the input signal depending on the way its transfer function deviates from the ideal 45 degree linear characteristic. The various types of distortion which may be present in the output of the ADC may be roughly divided into two categories (linear and nonlinear), and represented by an additive noise source. A good parameter estimation algorithm should produce accurate results even when the sinewave being analyzed suffers from some distortion.

Several approaches to the parameter estimation problem have been developed. Traditional methods used an iterative sinewave curve-fit process. These algorithms required accurate initial guesses to ensure convergence. A more recent approach applies a non-iterative, nonlinear technique which is quite accurate for low noise levels and relatively coarse (low) sampling rates. However, this method's estimates of sinewave fre-

quency (and associated phase angle) deteriorate rapidly in the presence of high distortion levels and/or sampling rates. An existing linear algorithm for parameter estimation avoids this problem, but requires advance knowledge of the sinewave's frequency for one of its inputs. The proposed algorithm attempts to combine the best features of the latter two techniques into a full four-parameter estimator. The new method is non-iterative, which eliminates difficulties in convergence and maintains a more consistent level of accuracy for increasing distortion in the input sinewave.

The proposed algorithm's process can be summarized as follows. The initial input is a finite record of digitized data obtained by sampling a practical sinewave (typically generated by an ADC). After this processing, the original data may be corrupted by various levels of linear or nonlinear distortion. Next, the discrete Fourier transform of the data is computed (via the FFT), which yields a rough estimate of the original sinewave's fundamental frequency. This value is used as the center of the passband for designing a Butterworth bandpass filter. Then, the noisy data is bandpass filtered to eliminate some of the distortion. The four-parameter algorithm is applied to the filtered data and yields an improved estimate of the sinewave's frequency. Finally, this value, together with the original data record, are input to the three-parameter algorithm which computes the remaining sinewave parameters.

## SUMMARY OF RESULTS

For sinewaves distorted with linear noise, the proposed algorithm's performance shows consistent improvement over the existing four-parameter method. Amplitude and dc offset estimates from the proposed algorithm exhibit lower errors at all noise levels. Frequency and phase estimates with the proposed algorithm are as accurate as the exist-

ing four-parameter method at low levels of distortion, and show significant improvement at higher levels.

When the sinewave distortion is nonlinear, the proposed algorithm continues to yield improved results. Amplitude and phase estimates are more accurate than the previous method at all distortion levels. While both algorithms provide adequate dc offset estimates, the existing four-parameter algorithm is more accurate. This is not surprising, since the filtering process involved in the proposed algorithm introduces distorting components into the sinewave which are not imposed by the nonlinear method. Frequency estimates with the proposed algorithm show distinct improvement over the existing method for increasing levels of distortion.

These results verify the improvement achieved by the proposed method over the existing four-parameter estimator. For several parameters, the estimation errors are lower with the proposed method at every noise level tested. For others, the improvement becomes evident at higher levels of distortion where the proposed method displays more consistent estimation accuracy. For input combinations where the existing estimation algorithm displayed a sudden breakdown in accuracy, the proposed algorithm not only produced reliable estimates over a broader range of test inputs, but also achieved a more gradual decline in accuracy at the limits of its useful range.

From these simulation results, it is possible to determine whether the proposed method's performance is adequate for real test situations. For practical applications, such as testing the effective bits of an ADC, the principal source of noise distortion comes from the digitizer. As a general rule, the output quality of an ADC (expressed as a signal-to-noise ratio) increases by approximately 6 dB for each additional bit of resolution. According to this rough guide, the resolution expected from an ADC can vary from



a minimum of 6 dB (for a 1-bit ADC), to 72 dB or better (for ADCs with 12 or more bits). Typical estimation errors from the proposed algorithm remain on the order of  $10^{-3}$  (or 60 dB SNR) throughout the range of distortion levels tested (30 to 0 dB SNR). This degree of accuracy (similar to a 10-bit ADC), suggests that the new technique can be applied to test a range of practical ADCs by ensuring that estimation errors will not mask the digitizer's inherent performance characteristics.

### TOPICS FOR FURTHER STUDY

The proposed algorithm attempts to improve parameter estimates by utilizing linear computations (avoiding the inaccuracies of nonlinear methods). In order to do this, bandpass filtering is used to improve the frequency estimate of the four-parameter algorithm to provide an input to the linear three-parameter method. Although this approach proved quite successful, there are some additional possibilities for improvement which were not investigated in this study.

- Enhancing the phase unwrapping process to use more pointers into the sampled data (possibly 5 instead of 3). This would help identify more precisely which portion of the sinewave cycle the data point is from, to ensure the correct arcsine formula is applied for unwrapping. Using more points to specify location would also decrease the chance of errors from trapping at a local extreme point (introduced by the noise). This could reduce estimation errors resulting from high sampling rates.
- Experimenting with different filter types to find the optimal combination of filter type, order, and bandwidth for the most effective elimination of distortion.
- Completely characterizing the method proposed in this study (i.e. more trials to test each sensitivity) and possibly improving its performance when high levels of

nonlinear distortion (harmonics) are present in the input test signal.

- Determining an effective method to handle statistical "outliers," or data points which fall outside the normal range of input values for the estimation algorithm. In practical situations, noise sources are not as well-controlled as assumed for the simulations. Individual samples with an unusually high level of noise could severely distort the estimates unless handled appropriately by the algorithm.
- Deriving closed form statistical equations to predict the estimation errors (for each parameter) from the proposed algorithm.

Each of these items could furnish topics for further study in the area of sinewave parameter estimation.

## REFERENCES

- [1] Bruce E. Peetz, Arthur S. Muto, and Martin J. Neil. Measuring waveform recorder performance. Hewlett-Packard Journal, 21-29. November 1982.
- [2] Yih-Chyun Jenq. High-precision sinusoidal frequency estimator based on weighted least square method. IEEE Transactions on Instrumentation and Measurement, IM-36(1):124-127, March 1987.
- [3] IEEE Standard 1057, Appendix A: Derivation of the three-parameter (known frequency) sine-wave curvefit algorithm, IEEE Trial-Use Standard for Digitizing Waveform Recorders, July 1989.
- [4] Yih-Chyun Jenq and Philip B. Crosby. Sinewave parameter estimation algorithm with application to waveform digitizer effective bits measurement. IEEE Transactions on Instrumentation and Measurement, 37(4):529-532, December 1988.
- [5] Yih-Chyun Jenq. Asynchronous dynamic test of A/D converters. Tektronix Handshake Journal, Summer. 1988.
- [6] Proakis G. John and Manolakis G. Dimitris. Digital Signal Processing: Principles, Algorithms, and Applications, 2nd Ed. Macmillian Publishing Co. New York 1992.
- [7] Oppenheim V. Alan and Schafer W. Ronald. Discrete-Time Signal Processing. Prentice-Hall, Inc. Englewood Cliffs, New Jersey 1989.
- [8] Papoulis Athanasios. Probability, Random Variables, and Stochastic Processes, 2nd Ed. McGraw-Hill Publishing Co. New York 1984.

## APPENDIX A

### FOUR-PARAMETER ALGORITHM PERFORMANCE

TABLE A.I

## AMPLITUDE ESTIMATION ERRORS : LINEAR DISTORTION

NOISE Pk-Pk	NOISE NSR-dB	SAMP/CYC						
		5.12 200	10.24 100	20.48 50	40.96 25	102.4 10	204.8 5	
0.1	-24.26	0.001675	0.001865	0.002276	0.001370	0.001989	0.002425	
0.2	-18.24	0.005023	0.006278	0.005783	0.004763	0.002927	0.003661	
0.4	-12.22	0.010033	0.013174	0.013436	0.012705	0.010963	0.015357	
0.6	-8.70	0.037664	0.028283	0.036646	0.018956	0.037585	0.025260	
0.8	-6.20	0.058057	0.050906	0.061417	0.060568	0.046813	0.051409	
1.0	-4.26	0.089493	0.092439	0.079409	0.076829	0.065565	0.098652	
1.5	-0.74	0.168713	0.179512	0.172894	0.165056	0.197578	0.168204	
2.0	1.76	0.284031	0.290341	0.258931	0.282813	0.295897	0.287568	

TABLE A.II

## DC OFFSET ESTIMATION ERRORS : LINEAR DISTORTION

NOISE Pk-Pk	NOISE NSR-dB	SAMP/CYC						
		5.12 200	10.24 100	20.48 50	40.96 25	102.4 10	204.8 5	
0.1	-24.26	1.22E-03	1.35E-03	1.83E-03	8.49E-04	1.28E-03	8.06E-04	
0.2	-18.24	0.003187	0.003179	0.002879	0.002439	0.002043	0.001601	
0.4	-12.22	0.005789	0.004916	0.003156	0.006214	0.003925	0.006368	
0.6	-8.70	0.005981	0.005245	0.004155	0.009061	0.009999	0.008031	
0.8	-6.20	0.006965	0.01274	0.006797	0.009458	0.011007	0.009374	
1.0	-4.26	0.008883	0.016121	0.009514	0.009720	0.016690	0.014385	
1.5	-0.74	0.013971	0.017133	0.017224	0.018069	0.020827	0.023528	
2.0	1.76	0.017529	0.017484	0.026926	0.026846	0.03139	0.028275	

TABLE A.III

## FREQUENCY ESTIMATION ERRORS : LINEAR DISTORTION

NOISE Pk-Pk	NOISE NSR-dB	SAMP/CYC					
		5.12 200	10.24 100	20.48 50	40.96 25	102.4 10	204.8 5
0.1	-24.26	0.000258	0.000577	0.001280	0.002100	0.010192	0.341787
0.2	-18.24	0.000250	0.001078	0.004248	0.011432	0.966693	19.28833
0.4	-12.22	0.001222	0.001360	0.004716	0.022632	25.26522	122.3833
0.6	-8.70	0.001390	0.002965	0.012556	2.851880	104.0710	211.8960
0.8	-6.20	0.002675	0.006046	0.094688	13.24307	121.0387	312.2141
1.0	-4.26	0.005027	0.011213	2.233324	25.38568	180.4460	440.2042
1.5	-0.74	0.005938	0.018252	10.75156	77.64162	337.1073	785.7622
2.0	1.76	1.69782	7.71202	38.63786	160.6356	433.1292	1048.981

TABLE A.IV

## PHASE ESTIMATION ERRORS : LINEAR DISTORTION

NOISE Pk-Pk	NOISE NSR-dB	SAMP/CYC					
		5.12 200	10.24 100	20.48 50	40.96 25	102.4 10	204.8 5
0.1	-24.26	0.098373	0.124345	0.161906	0.121944	0.234452	2.352660
0.2	-18.24	0.172661	0.187510	0.399560	0.108320	1.332236	149.5141
0.4	-12.22	0.228545	0.465847	0.466393	1.638737	6.417197	384.2696
0.6	-8.70	0.263395	0.544429	0.488016	37.35872	129.4184	453.2766
0.8	-6.20	1.289828	0.813523	1.100384	128.6215	255.5338	508.363
1.0	-4.26	2.225773	1.418073	109.1053	361.0451	308.6359	592.9817
1.5	-0.74	2.938592	87.6211	271.2254	393.291	393.0224	593.9628
2.0	1.76	172.9296	272.9335	402.7372	518.6275	525.5458	621.6342

TABLE A.V

## AMPLITUDE ESTIMATION ERRORS : NONLINEAR DISTORTION

HARMONIC Pk-Pk	HARMONIC HSR-dB	FREQUENCY					
		FS = 25 fh = 50	FS = 25 fh = 75	FS = 100 fh = 200	FS = 100 fh = 300	FS = 400 fh = 800	FS = 400 fh = 1200
0.0	-26.02	0.133582	0.042602	0.022642	0.031555	0.009270	0.117547
0.2	-20.00	0.246859	0.187069	0.045213	0.091039	0.032766	0.336577
0.4	-13.98	0.428400	0.375476	0.089846	0.216425	171.2468	3.402295
0.6	-10.46	0.559966	0.854182	0.148506	0.257115	642.074	4.338813
0.8	-7.96	0.637311	1.271314	0.212988	0.380102	208.3974	5.030159
1.0	-6.02	0.756585	1.421948	0.286147	0.424261	520.9202	5.631276
1.5	-2.50	1.197534	1.821767	0.940583	0.626762	1245.452	6.205044
2.0	0.00	1.670169	2.077714	0.933747	0.756606	411.6644	5.794618

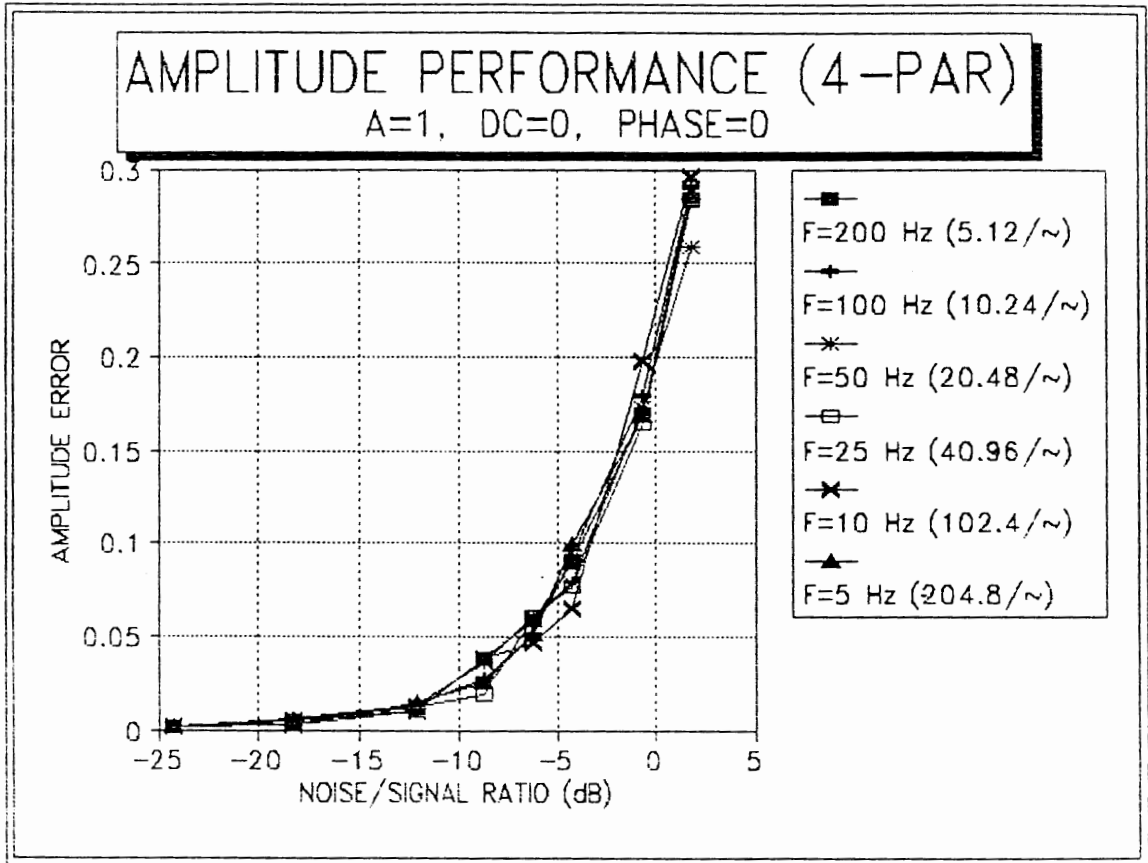
TABLE A.VI

## DC OFFSET ESTIMATION ERRORS : NONLINEAR DISTORTION

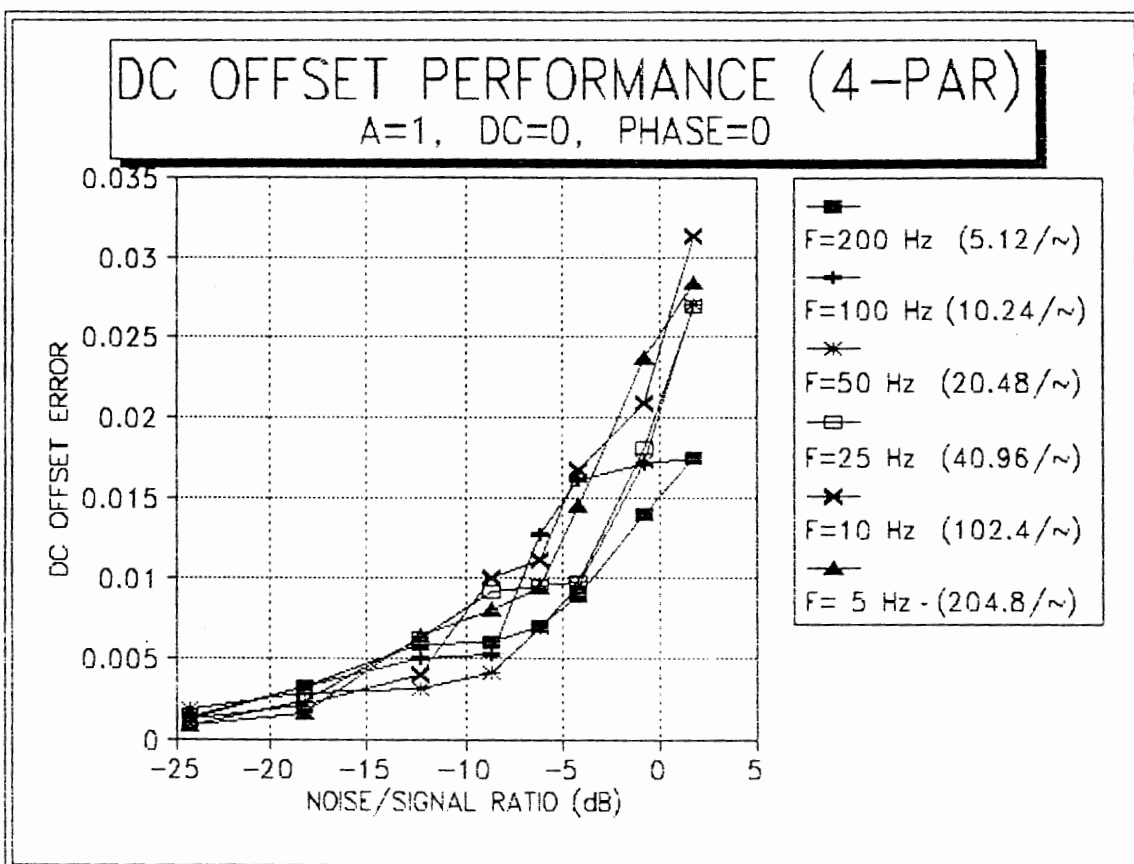
HARMONIC Pk-Pk	HARMONIC HSR-dB	FREQUENCY					
		FS = 25 fh = 50	FS = 25 fh = 75	FS = 100 fh = 200	FS = 100 fh = 300	FS = 400 fh = 800	FS = 400 fh = 1200
0.1	-26.02	0.002933	0.000880	0.000110	0.000050	0.000000	0.000000
0.2	-20.00	0.005440	0.002680	0.000190	0.000170	0.000025	0.000000
0.4	-13.98	0.009440	0.008400	0.000280	0.000320	5.957925	24.00470
0.6	-10.46	0.012320	0.018400	0.000460	0.000520	15.26103	55.99994
0.8	-7.96	0.014040	0.027480	0.000710	0.000680	29.91667	56.00029
1.0	-6.02	0.016720	0.032040	0.001090	0.000900	36.56061	56.00010
1.5	-2.50	100.1110	0.041040	99.9992	0.001700	47.89495	56.00010
2.0	0.00	99.9982	0.046760	99.9987	0.002100	46.1467	56.00008



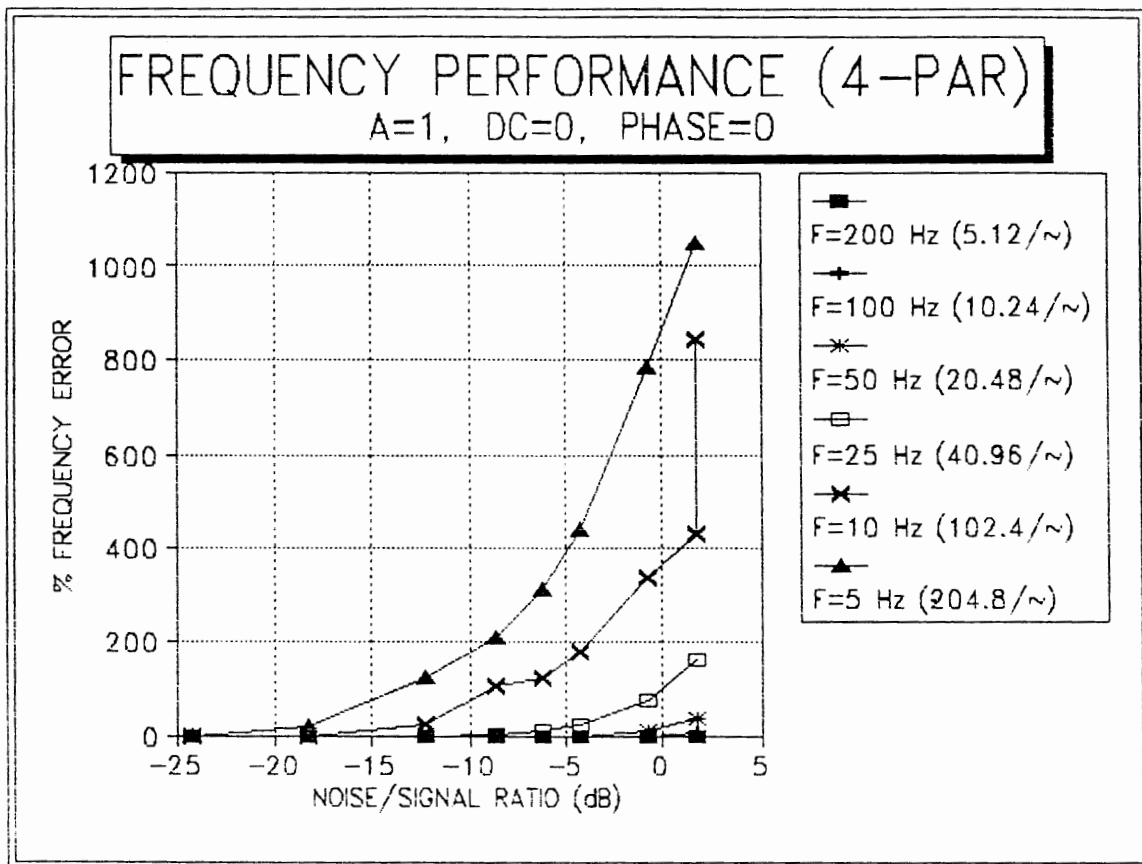




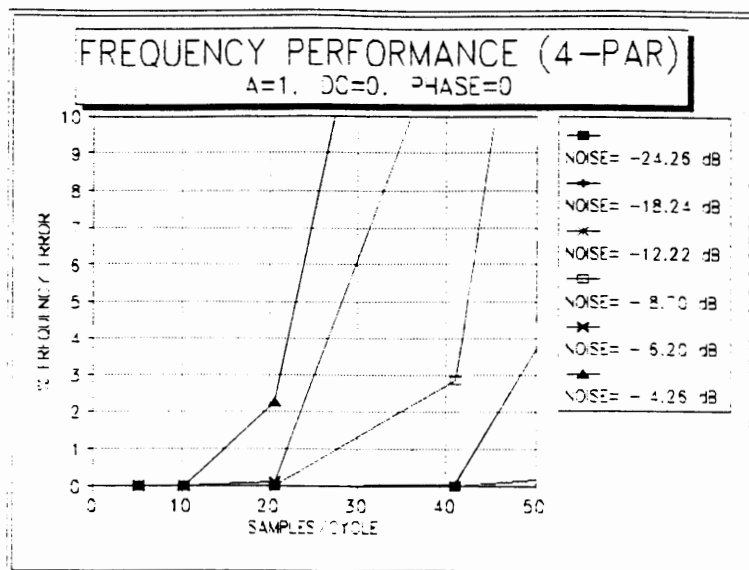
**Figure A.1.** Plot of *amplitude* estimation errors for increasing linear distortion.



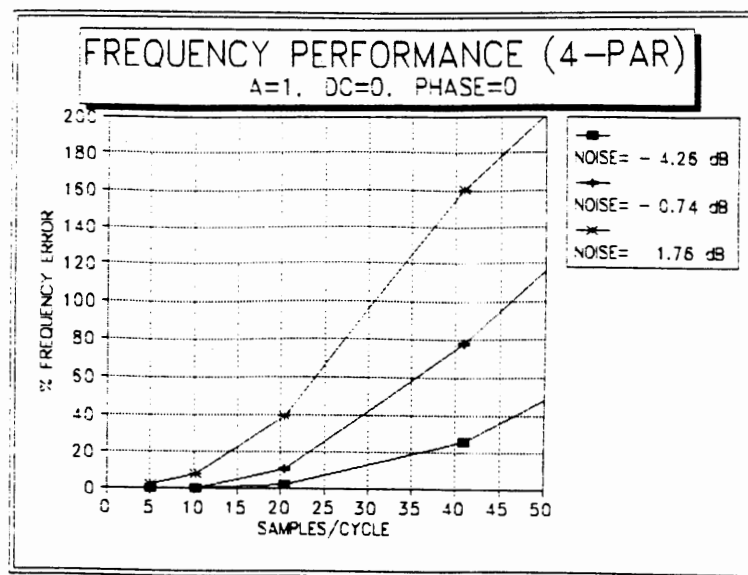
**Figure A.2.** Plot of *dc offset* estimation errors for increasing linear distortion.



**Figure A.3.** Plot of *frequency estimation errors* for increasing linear distortion.

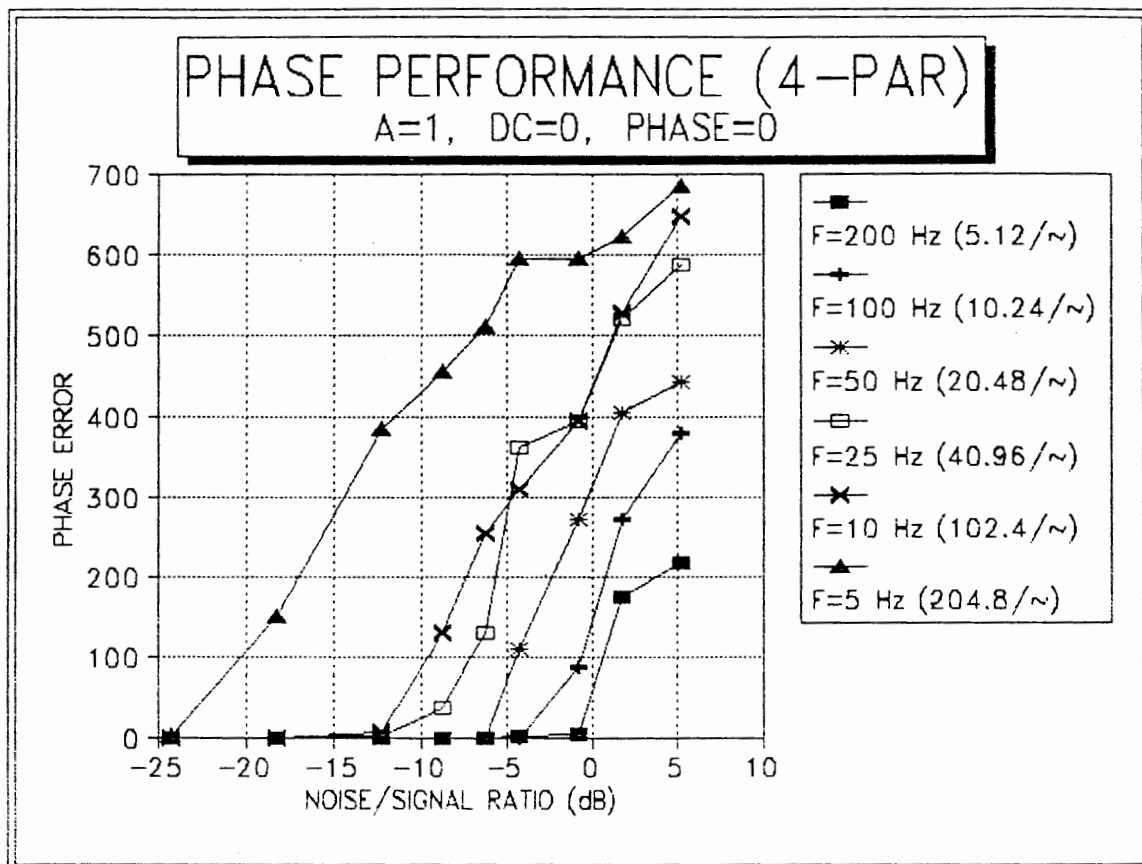


(a)



(b)

**Figure A.4.** Plot of *frequency estimation errors* for increasing sampling rate: (a) low and (b) higher noise levels.



**Figure A.5.** Plot of *phase* estimation errors for increasing linear distortion.

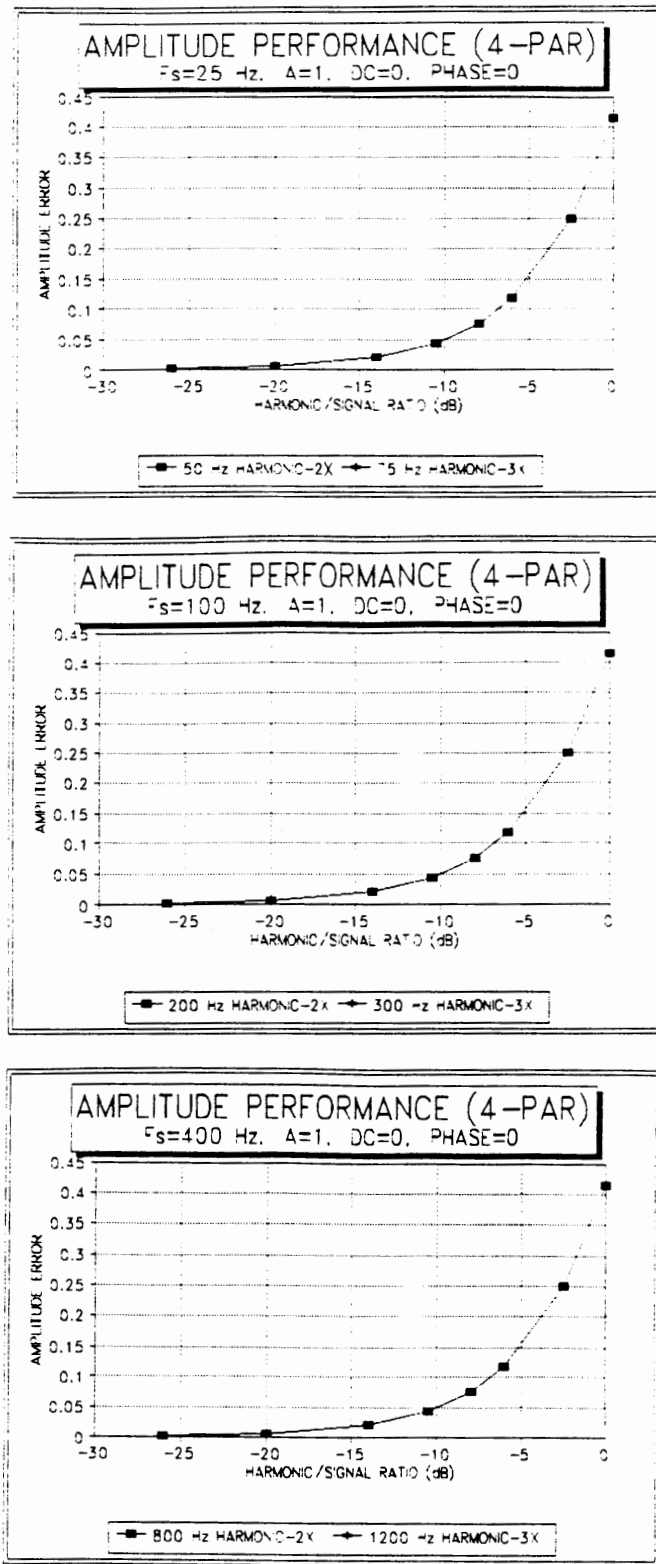


Figure A.6. Plot of *amplitude* estimation errors for increasing nonlinear distortion.

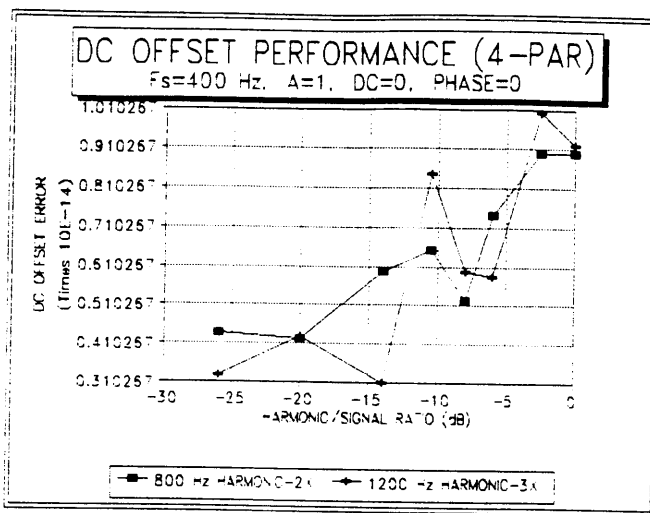
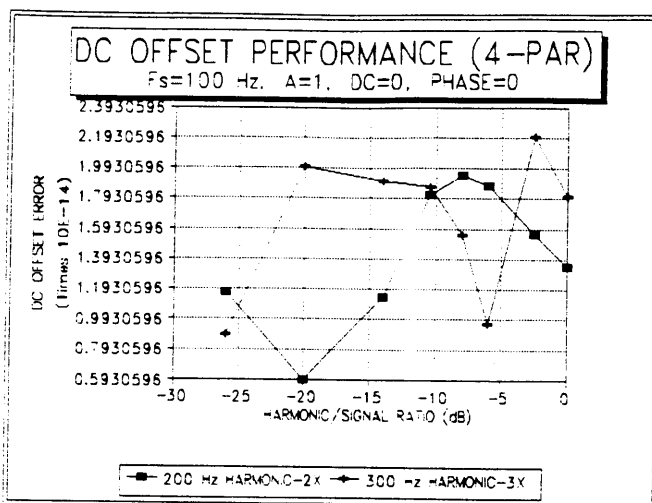
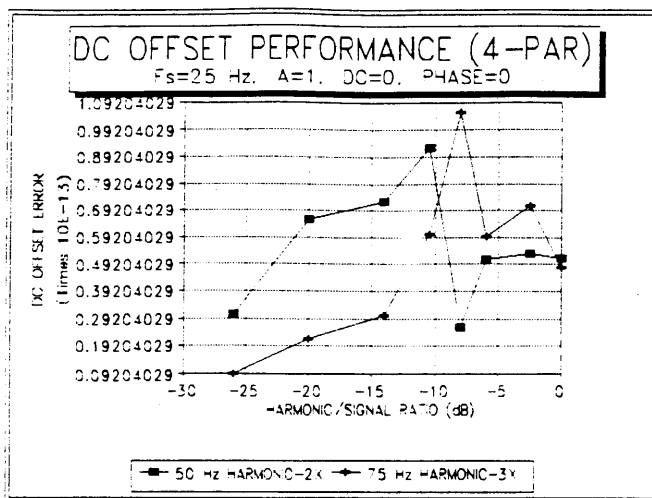
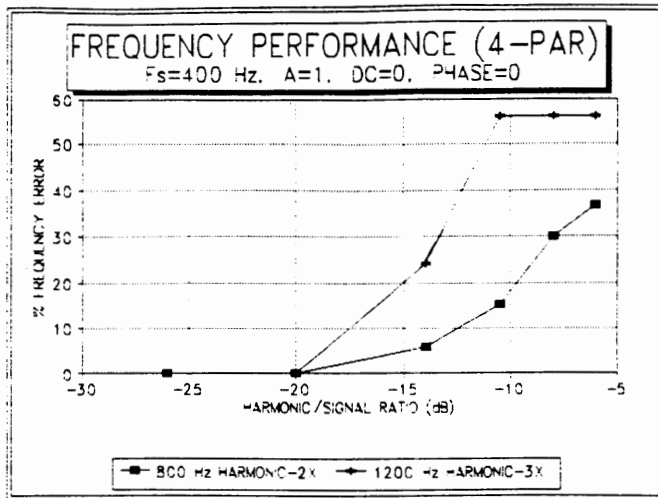
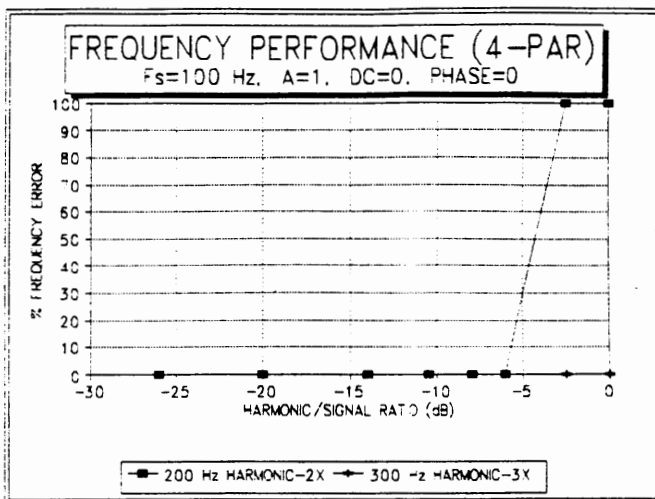
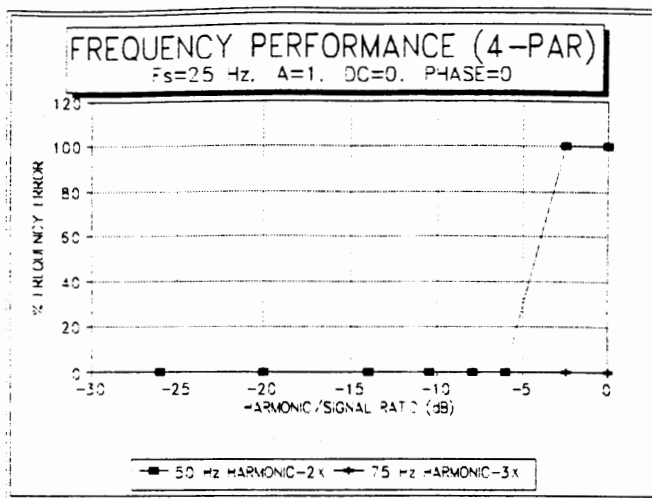


Figure A.7. Plot of *dc offset* estimation errors for increasing nonlinear distortion.



**Figure A.8.** Plot of frequency estimation errors for increasing nonlinear distortion.



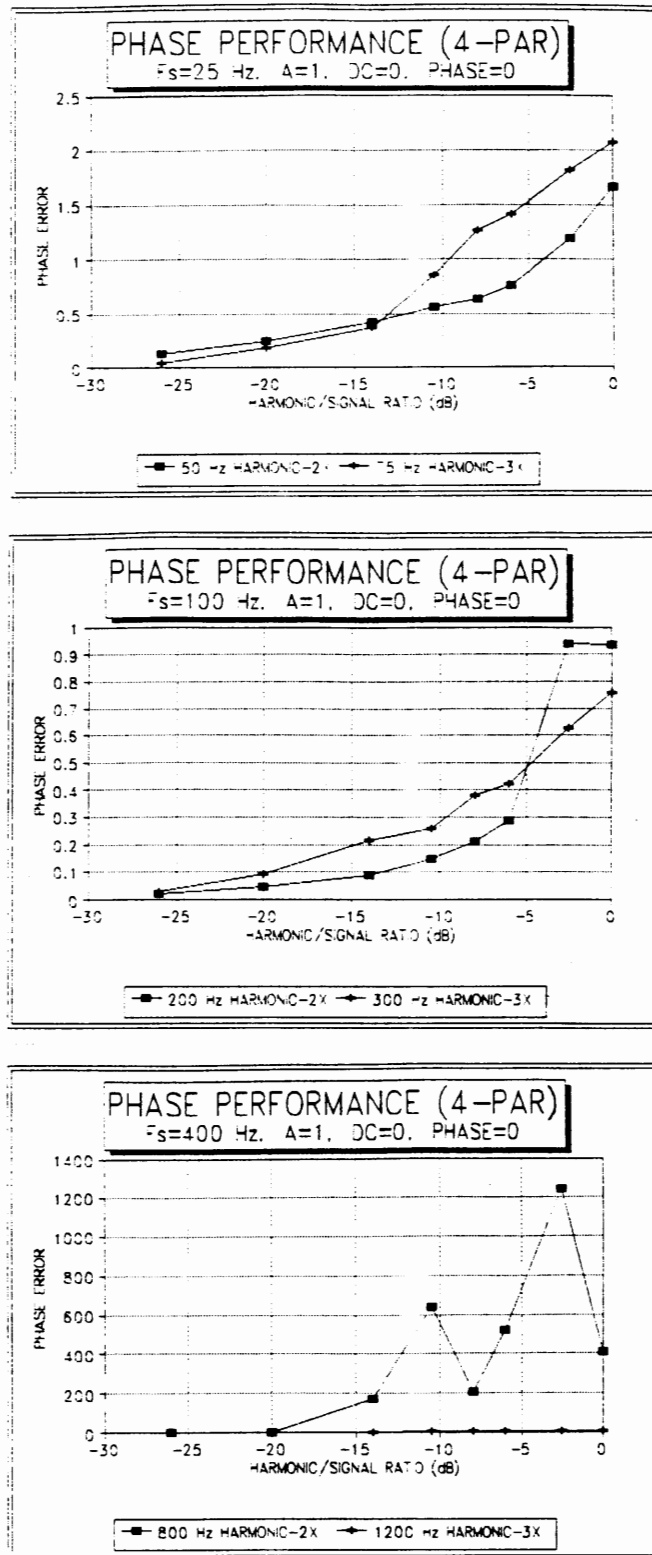


Figure A.9. Plot of phase estimation errors for increasing nonlinear distortion.

## APPENDIX B

### THREE-PARAMETER ALGORITHM PERFORMANCE

TABLE B.I

## AMPLITUDE ESTIMATION ERRORS : LINEAR DISTORTION

NOISE Pk-Pk	NOISE NSR-dB	SAMP/CYC						
		5.12 200	10.24 100	20.48 50	40.96 25	102.4 10	204.8 5	
0.1	-24.26	0.001142	0.00125	0.00091	0.001143	0.001263	0.001282	
0.2	-18.24	0.001226	0.002566	0.001704	0.002459	0.001321	0.001837	
0.4	-12.22	0.004170	0.004986	0.004298	0.002824	0.005478	0.005220	
0.6	-8.70	0.002800	0.006129	0.005458	0.008945	0.006482	0.004628	
0.8	-6.20	0.007318	0.006019	0.006941	0.007941	0.004782	0.009915	
1.0	-4.26	0.010135	0.012248	0.008476	0.013556	0.007015	0.011524	
1.5	-0.74	0.016574	0.018500	0.013630	0.017019	0.019140	0.024558	
2.0	1.76	0.01839	0.023004	0.024223	0.034375	0.009376	0.018412	

TABLE B.II

## DC OFFSET ESTIMATION ERRORS : LINEAR DISTORTION

NOISE Pk-Pk	NOISE NSR-dB	SAMP/CYC						
		5.12 200	10.24 100	20.48 50	40.96 25	102.4 10	204.8 5	
0.1	-24.26	0.000762	0.000979	0.000304	0.000868	0.000713	0.000587	
0.2	-18.24	0.001834	0.001740	0.001524	0.002458	0.001427	0.000641	
0.4	-12.22	0.002608	0.003917	0.003922	0.001692	0.002780	0.001687	
0.6	-8.70	0.004251	0.001825	0.001822	0.006545	0.003028	0.003928	
0.8	-6.20	0.006796	0.006944	0.006940	0.003983	0.003123	0.003858	
1.0	-4.26	0.008003	0.007141	0.007137	0.008967	0.004860	0.002880	
1.5	-0.74	0.009915	0.014688	0.004556	0.013023	0.010702	0.008801	
2.0	1.76	0.011424	0.011734	0.011730	0.016620	0.015514	0.014564	

TABLE B.III

## PHASE ESTIMATION ERRORS : LINEAR DISTORTION

NOISE Pk-Pk	NOISE NSR-dB	SAMP/CYC						
		5.12 200	10.24 100	20.48 50	40.96 25	102.4 10	204.8 5	
0.1	-24.26	0.000674	0.001044	0.000705	0.000846	0.001262	0.000440	
0.2	-18.24	0.002863	0.001517	0.001411	0.001531	0.002758	0.003268	
0.4	-12.22	0.004707	0.006095	0.003323	0.003328	0.005743	0.003833	
0.6	-8.70	0.004201	0.007330	0.003054	0.003324	0.008409	0.006336	
0.8	-6.20	0.006497	0.008974	0.006193	0.012014	0.014509	0.006844	
1.0	-4.26	0.006321	0.012888	0.010747	0.003959	0.005857	0.012781	
1.5	-0.74	0.010191	0.018694	0.007605	0.012991	0.014669	0.013888	
2.0	1.76	0.020383	0.023245	0.013648	0.024115	0.010699	0.035851	

TABLE B.IV

## SENSITIVITY OF AMPLITUDE ESTIMATES TO INPUT FREQUENCY VARIATION

FREQUENCY CHANGE	ACTUAL	FS = 200 N=0.5	FS = 200 p-pN=1.0	ACTUAL	FS = 50 N=0.5	FS = 50 p-pN=1.0
+VE						
1E-05	200.0020	0.003717	0.010476	50.0005	0.005375	0.0121
0.0001	200.0200	0.004263	0.005725	50.0050	0.004582	0.010811
0.001	200.2000	0.003510	0.015648	50.0500	0.004425	0.009560
0.01	202.0000	0.005497	0.012885	50.5000	0.003560	0.007631
0.1	220.0000	0.003412	0.009606	55.0000	0.005865	0.010692
-VE						
-1E-05	199.9980	0.002898	0.013646	49.9995	0.004249	0.015492
-0.0001	199.9800	0.002661	0.014457	49.9950	0.007703	0.012068
-0.001	199.8000	0.007456	0.010627	49.9500	0.002415	0.013829
-0.01	198.0000	0.006573	0.008792	49.5000	0.003503	0.009033
-0.1	180.0000	0.005109	0.010429	45.0000	0.006288	0.012016

TABLE B.V

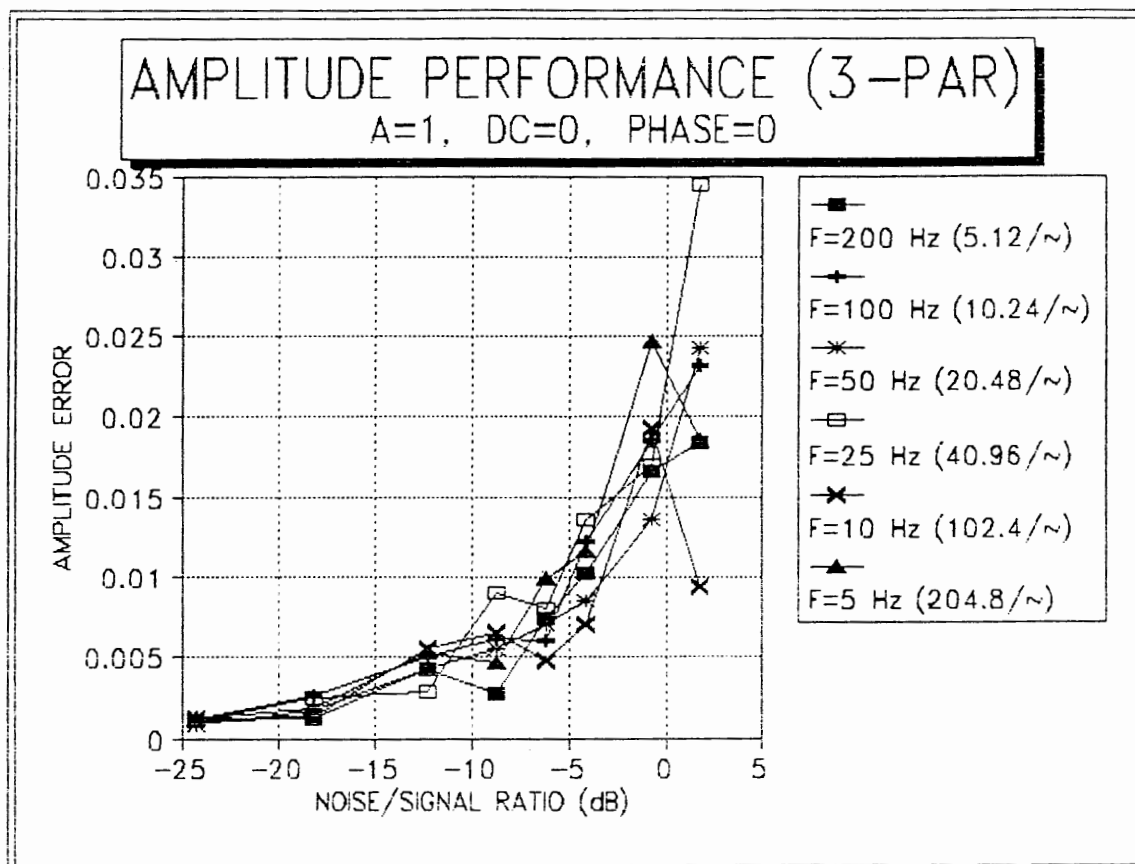
## SENSITIVITY OF DC OFFSET ESTIMATES TO INPUT FREQUENCY VARIATION

FREQUENCY CHANGE	ACTUAL	FS = 200 N=0.5	FS = 200 p-pN=1.0	ACTUAL	FS = 50 N=0.5	FS = 50 p-pN=1.0
+VE						
1E-05	200.0020	0.003894	0.005327	50.0005	0.004903	0.005865
0.0001	200.0200	0.003732	0.008971	50.0050	0.001519	0.009805
0.001	200.2000	0.003857	0.00497	50.0500	0.004335	0.003036
0.01	202.0000	0.004901	0.010907	50.5000	0.003577	0.008656
0.1	220.0000	0.001521	0.004226	55.0000	0.002933	0.007137
-VE						
-1E-05	199.9980	0.004338	0.006953	49.9995	0.003569	0.012298
-0.0001	199.9800	0.003568	0.005051	49.9950	0.006149	0.005865
-0.001	199.8000	0.002932	0.009804	49.9500	0.005454	0.012296
-0.01	198.0000	0.006151	0.003033	49.5000	0.002504	0.004276
-0.1	180.0000	0.002117	0.008693	45.0000	0.004482	0.0109

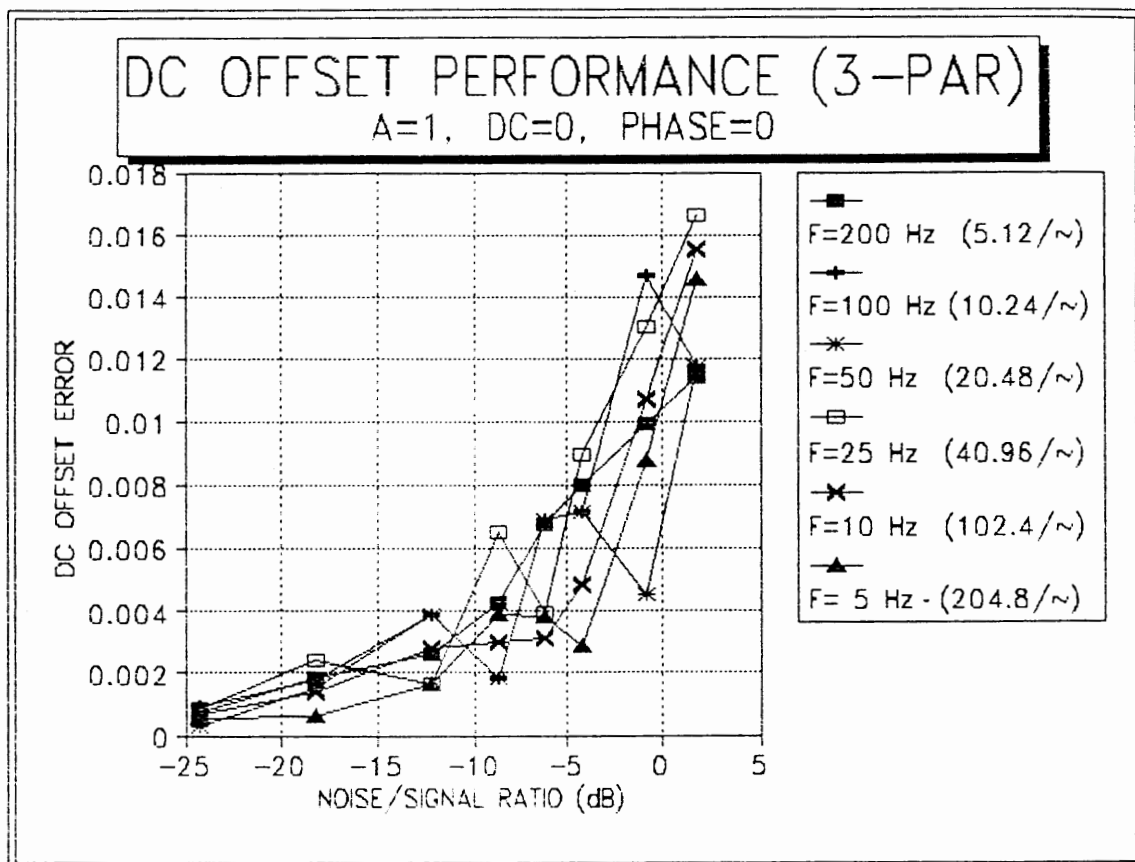
TABLE B.VI

## SENSITIVITY OF PHASE ESTIMATES TO INPUT FREQUENCY VARIATIONS

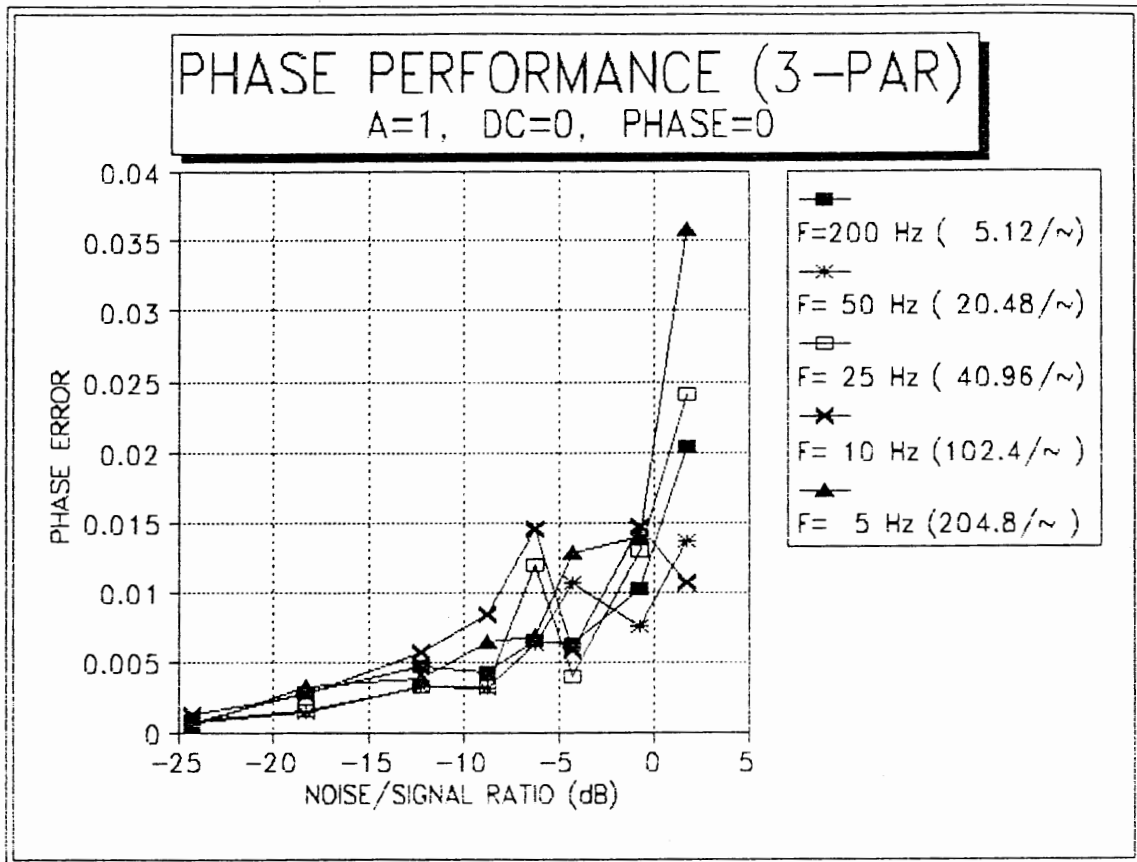
FREQUENCY CHANGE	ACTUAL	FS = 200		ACTUAL	FS = 50	
		N=0.5	p-pN=1.0		N=0.5	p-pN=1.0
+VE						
1E-05	200.0020	0.004612	0.009990	50.0005	0.004152	0.006814
0.0001	200.0200	0.003241	0.004542	50.0050	0.002454	0.008244
0.001	200.2000	0.007998	0.005752	50.0500	0.003517	0.004254
0.01	202.0000	0.003257	0.010413	50.5000	0.004679	0.011582
0.1	220.0000	0.003553	0.009045	55.0000	0.004636	0.010470
-VE						
-1E-05	199.9980	0.003587	0.015329	49.9995	0.005353	0.00899
-0.0001	199.9800	0.001294	0.013773	49.9950	0.004574	0.006908
-0.001	199.8000	0.004889	0.008571	49.9500	0.007488	0.01098
-0.01	198.0000	0.006447	0.013553	49.5000	0.006763	0.009938
-0.1	180.0000	0.00596	0.012921	45.0000	0.002531	0.010462



**Figure B.1.** Plot of *amplitude* estimation errors for increasing linear distortion.



**Figure B.2.** Plot of *dc offset* estimation errors for increasing linear distortion.



**Figure B.3.** Plot of *phase* estimation errors for increasing linear distortion.



## APPENDIX C

### PROPOSED (NEW) ALGORITHM PERFORMANCE

TABLE C.I

## AMPLITUDE ESTIMATION ERRORS : LINEAR DISTORTION

NOISE Pk-Pk	NOISE NSR-dB	SAMP/CYC			
		5.12 200	10.24 100	20.48 50	40.96 25
0.1	-24.26	0.001552	0.001093	0.001075	0.001048
0.2	-18.24	0.002635	0.002074	0.001730	0.002678
0.4	-12.22	0.004023	0.004046	0.003401	0.003175
0.6	-8.70	0.005934	0.008769	0.005379	0.006856
0.8	-6.20	0.006867	0.010431	0.007202	0.010212
1.0	-4.26	0.009401	0.010972	0.009902	0.010488
1.5	-0.74	0.014476	0.018461	0.012389	0.012115
2.0	1.76	0.030058	0.020366	0.023755	0.022867

TABLE C.II

## DC OFFSET ESTIMATION ERRORS : LINEAR DISTORTION

NOISE Pk-Pk	NOISE NSR-dB	SAMP/CYC			
		5.12 200	10.24 100	20.48 50	40.96 25
0.1	-24.26	3.05E-04	7.62E-04	9.81E-04	3.03E-04
0.2	-18.24	0.001671	0.000862	0.001735	0.001793
0.4	-12.22	0.004271	0.004126	0.002855	0.002132
0.6	-8.70	0.002203	0.004525	0.003519	0.004277
0.8	-6.20	0.005258	0.006095	0.004230	0.005052
1.0	-4.26	0.007616	0.007618	0.009839	0.005562
1.5	-0.74	0.008195	0.014694	0.009908	0.005842
2.0	1.76	0.024258	0.006081	0.01636	0.009722

TABLE C.III

## FREQUENCY ESTIMATION ERRORS : LINEAR DISTORTION

NOISE Pk-Pk	NOISE NSR-dB	SAMP/CYC			
		5.12 200	10.24 100	20.48 50	40.96 25
0.1	-24.26	0.003757	0.000805	0.002385	0.009562
0.2	-18.24	0.003830	0.000814	0.002894	0.012167
0.4	-12.22	0.003837	0.003284	0.004192	0.017837
0.6	-8.70	0.003954	0.005084	0.006948	0.028422
0.8	-6.20	0.004093	0.006204	0.008356	0.028675
1.0	-4.26	0.005859	0.007748	0.010718	0.030702
1.5	-0.74	0.005923	0.012398	0.026887	0.067389
2.0	1.76	0.008074	0.020667	0.037305	0.074778

TABLE C.IV

## PHASE ESTIMATION ERRORS : LINEAR DISTORTION

NOISE Pk-Pk	NOISE NSR-dB	SAMP/CYC			
		5.12 200	10.24 100	20.48 50	40.96 25
0.1	-24.26	3.117054	0.629871	0.631905	0.008227
0.2	-18.24	3.117375	1.257482	1.885958	0.636940
0.4	-12.22	3.115572	2.511933	2.513090	1.885963
0.6	-8.70	3.114840	1.884759	0.639282	1.259761
0.8	-6.20	3.109703	1.880916	1.269318	0.651838
1.0	-4.26	2.498787	1.879471	1.884199	1.878634
1.5	-0.74	2.483510	1.843857	1.861251	0.675416
2.0	1.76	3.083767	1.895717	1.272621	1.277250

TABLE C.V

## AMPLITUDE ESTIMATION ERRORS : NONLINEAR DISTORTION

HARMONIC Pk-Pk	HARMONIC HSR-dB	FREQUENCY					
		FS = 25 fh = 50	FS = 25 fh = 75	FS = 100 fh = 200	FS = 100 fh = 300	FS = 400 fh = 800	FS = 400 fh = 1200
0.1	-26.02	3.58E-05	3.73E-05	1.60E-06	1.68E-06	9.85E-07	9.12E-07
0.2	-20.00	3.31E-05	3.57E-05	1.47E-06	1.62E-06	1.03E-06	8.99E-07
0.4	-13.98	2.55E-05	3.20E-05	1.15E-06	1.46E-06	1.15E-06	8.63E-07
0.6	-10.46	1.97E-05	2.65E-05	7.35E-07	1.18E-06	1.23E-06	7.46E-07
0.8	-7.96	2.33E-05	2.40E-05	8.51E-07	1.25E-06	1.40E-06	8.94E-07
1.0	-6.02	2.09E-05	1.68E-05	1.26E-06	1.40E-06	1.54E-06	8.54E-07
1.5	-2.50	4.48E-05	1.81E-05	1.62E-06	7.69E-07	1.68E-06	7.03E-07
2.0	0.00	7.70E-05	3.34E-05	4.20E-06	2.64E-05	1.97E-06	7.24E-07

TABLE C.VI

## DC OFFSET ESTIMATION ERRORS : NONLINEAR DISTORTION

HARMONIC Pk-Pk	HARMONIC HSR-dB	FREQUENCY					
		FS = 25 fh = 50	FS = 25 fh = 75	FS = 100 fh = 200	FS = 100 fh = 300	FS = 400 fh = 800	FS = 400 fh = 1200
0.1	-26.02	3.71E-05	3.71E-05	3.18E-05	3.18E-05	2.86E-05	2.86E-05
0.2	-20.00	6.35E-05	7.41E-05	7.41E-05	6.87E-05	6.35E-05	6.35E-05
0.4	-13.98	1.48E-04	1.27E-04	1.27E-04	1.23E-04	1.15E-04	1.15E-04
0.6	-10.46	1.90E-04	1.72E-04	1.72E-04	1.53E-04	2.22E-04	1.72E-04
0.8	-7.96	2.29E-04	2.54E-04	2.30E-04	1.99E-04	2.54E-04	2.54E-04
1.0	-6.02	2.87E-04	2.86E-04	2.30E-04	3.51E-04	2.86E-04	2.86E-04
1.5	-2.50	4.76E-04	4.30E-04	5.90E-04	3.01E-04	4.31E-04	3.99E-04
2.0	0.00	5.73E-04	4.59E-04	5.94E-04	5.55E-04	5.51E-04	4.31E-04

TABLE C.VII

## FREQUENCY ESTIMATION ERRORS : NONLINEAR DISTORTION

HARMONIC Pk-Pk	HARMONIC HSR-dB	FREQUENCY					
		FS = 25 fh = 50	FS = 25 fh = 75	FS = 100 fh = 200	FS = 100 fh = 300	FS = 400 fh = 800	FS = 400 fh = 1200
0.1	-26.02	1.00E-02	1.00E-02	5.01E-04	4.97E-04	7.13E-05	7.29E-05
0.2	-20.00	1.04E-02	1.02E-02	5.09E-04	5.02E-04	7.08E-05	7.37E-05
0.4	-13.98	1.05E-02	1.04E-02	5.15E-04	5.07E-04	6.99E-05	7.63E-05
0.6	-10.46	1.13E-02	1.05E-02	5.26E-04	5.08E-04	6.62E-05	1.72E-04
0.8	-7.96	1.13E-02	1.06E-02	5.83E-04	5.13E-04	6.81E-05	7.96E-05
1.0	-6.02	1.20E-02	1.08E-02	5.87E-04	5.37E-04	6.71E-05	8.32E-05
1.5	-2.50	1.31E-02	1.13E-02	5.97E-04	5.60E-04	4.55E-05	7.25E-05
2.0	0.00	1.45E-02	1.18E-02	6.82E-04	39.9988	8.8000	22.4000

TABLE C.VIII

## PHASE ESTIMATION ERRORS : NONLINEAR DISTORTION

HARMONIC Pk-Pk	HARMONIC HSR-dB	FREQUENCY					
		FS = 25 fh = 50	FS = 25 fh = 75	FS = 100 fh = 200	FS = 100 fh = 300	FS = 400 fh = 800	FS = 400 fh = 1200
0.1	-26.02	0.007884	0.007860	0.001600	0.001589	0.000862	0.000882
0.2	-20.00	0.008263	0.007969	0.001588	0.001457	0.000834	0.000871
0.4	-13.98	0.008112	0.008290	0.001727	0.001513	0.000743	0.000824
0.6	-10.46	0.009035	0.008466	0.001648	0.001632	0.000866	0.001190
0.8	-7.96	0.009675	0.008543	0.001452	0.001745	0.000635	0.000780
1.0	-6.02	0.008825	0.008828	0.001858	0.001895	0.000506	0.000708
1.5	-2.50	0.010692	0.008177	0.001672	0.001742	0.001282	0.001622
2.0	0.00	0.012049	0.009291	0.001676	0.012800	0.273937	0.462516

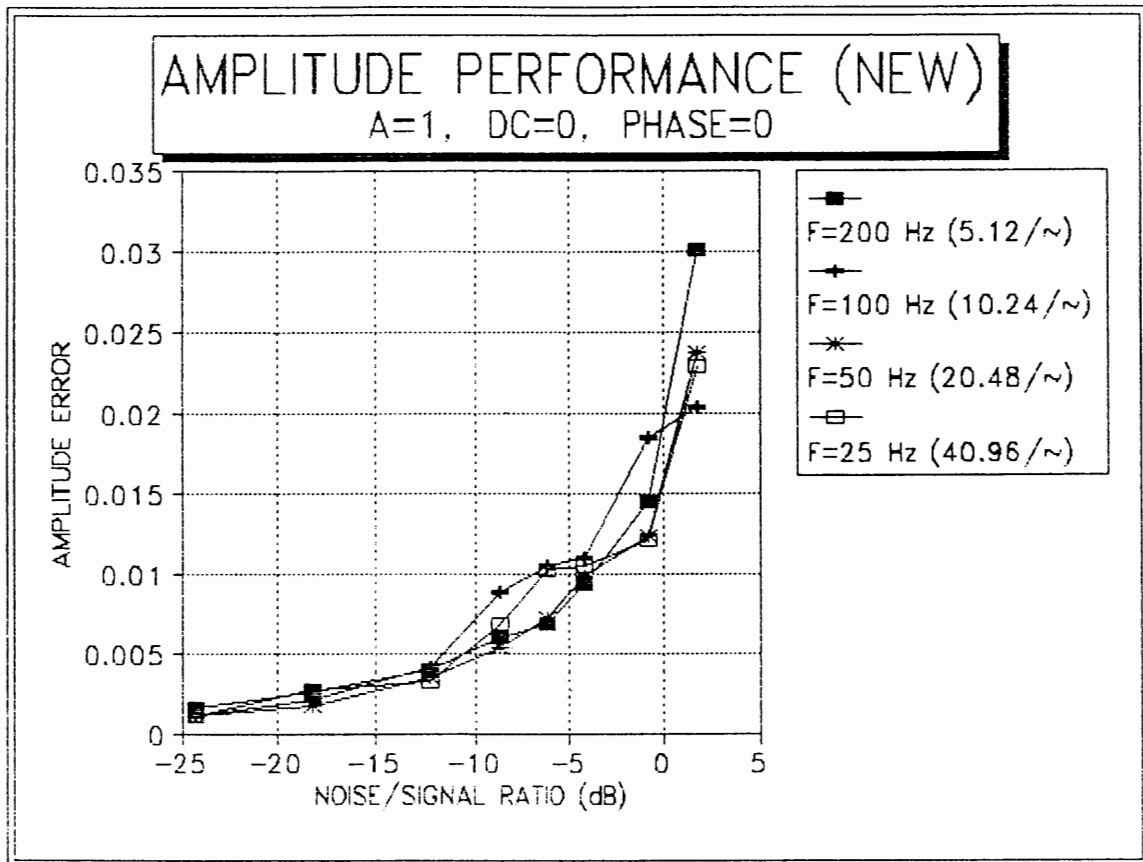
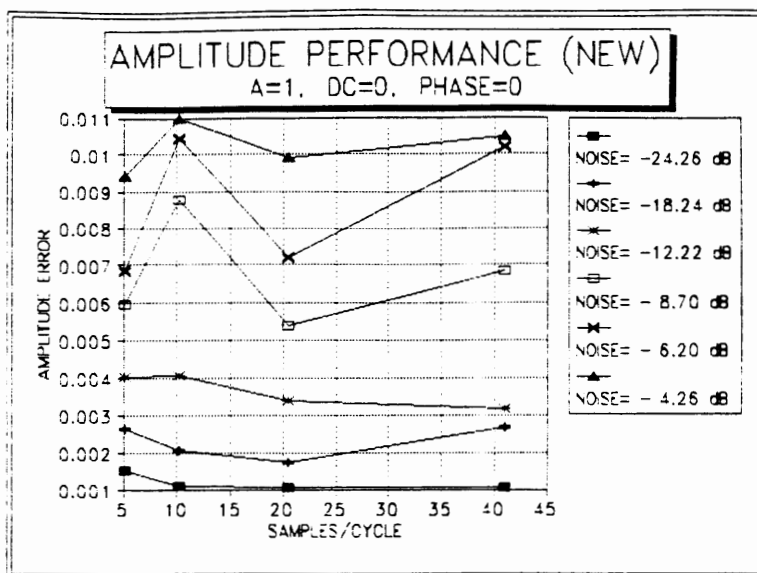
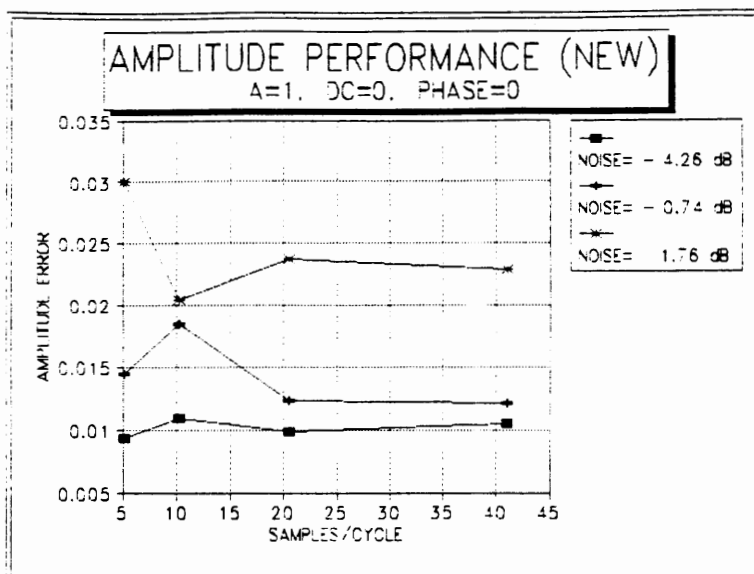


Figure C.1. Plot of *amplitude* estimation errors for increasing linear distortion.

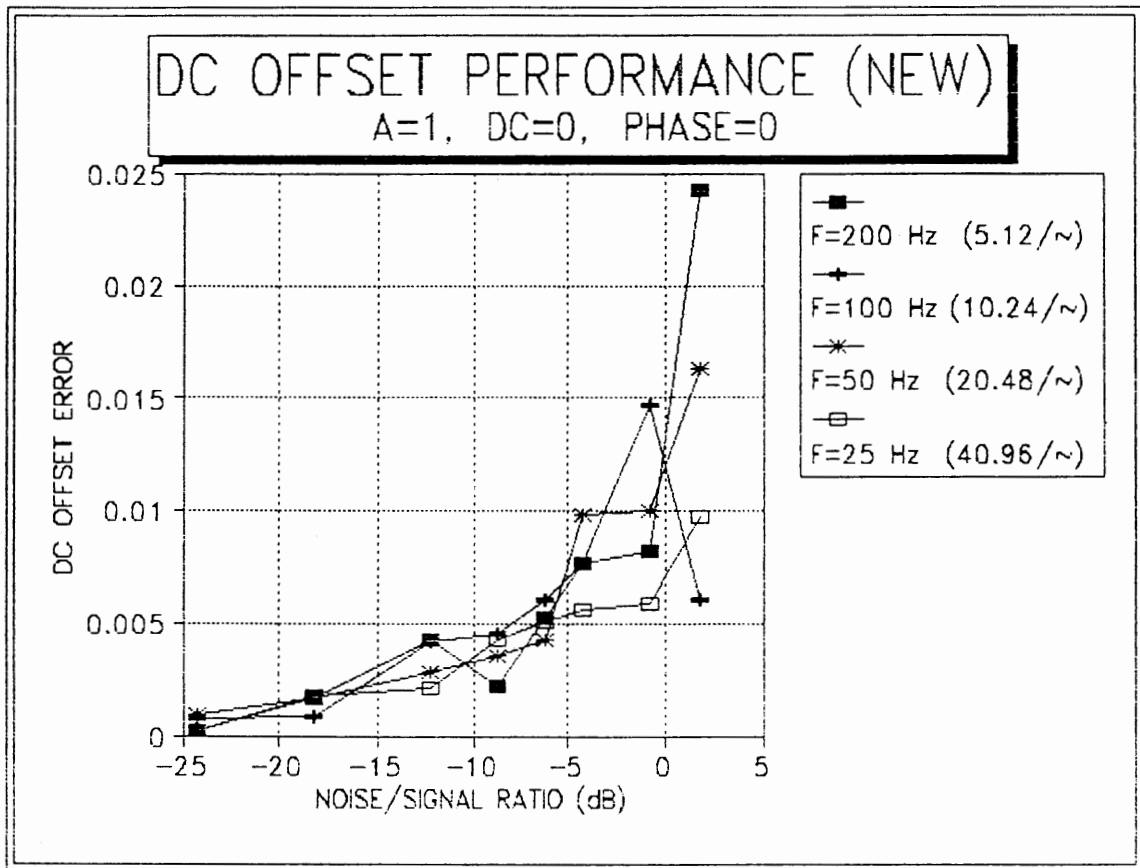


(a)



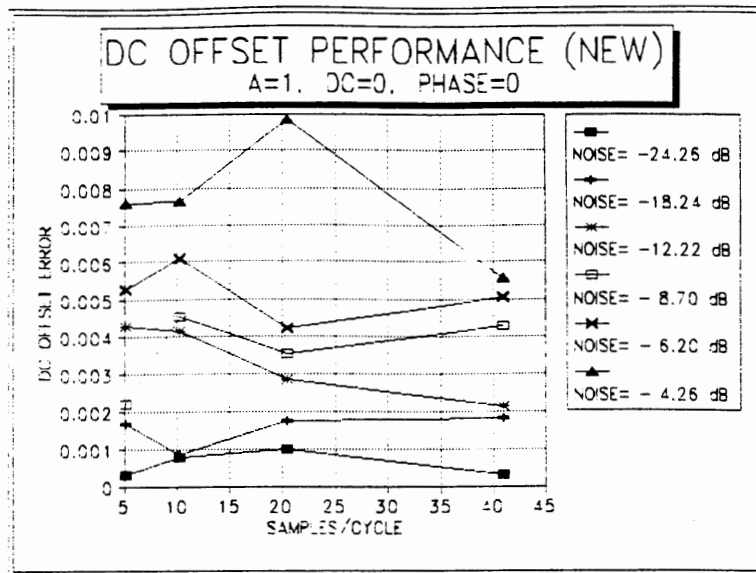
(b)

**Figure C.2.** Plot of *amplitude* estimation errors for increasing sampling rate: (a) low and (b) higher noise levels.

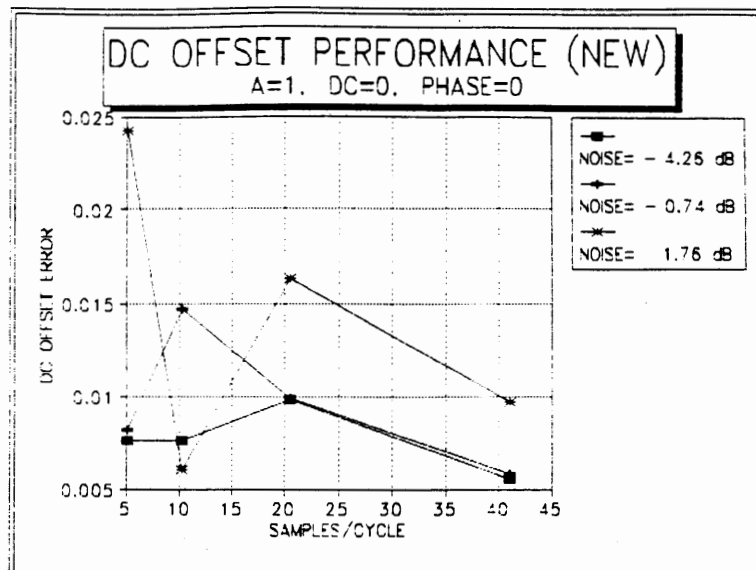


**Figure C.3.** Plot of *dc offset* estimation errors for increasing linear distortion.



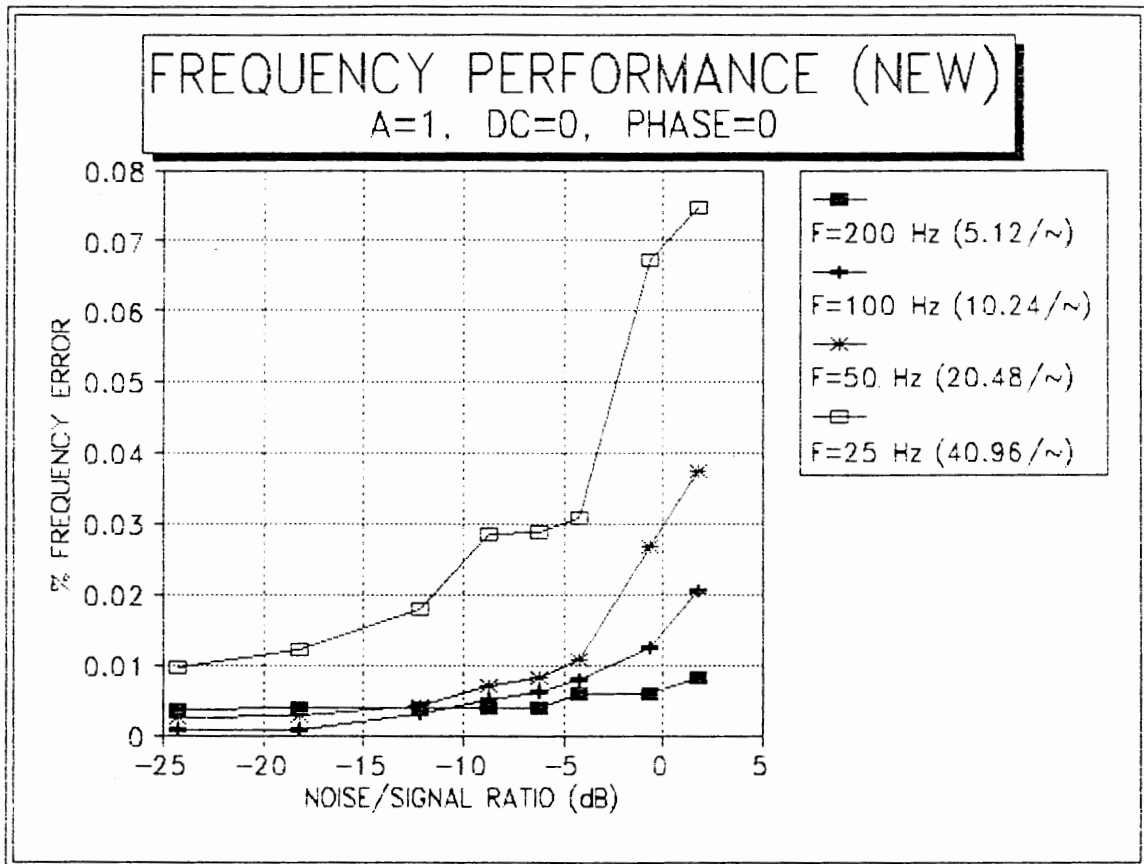


(a)

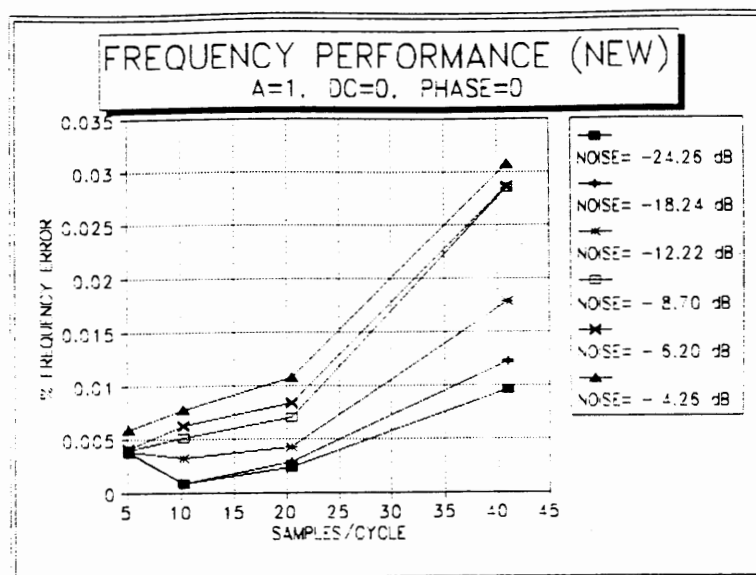


(b)

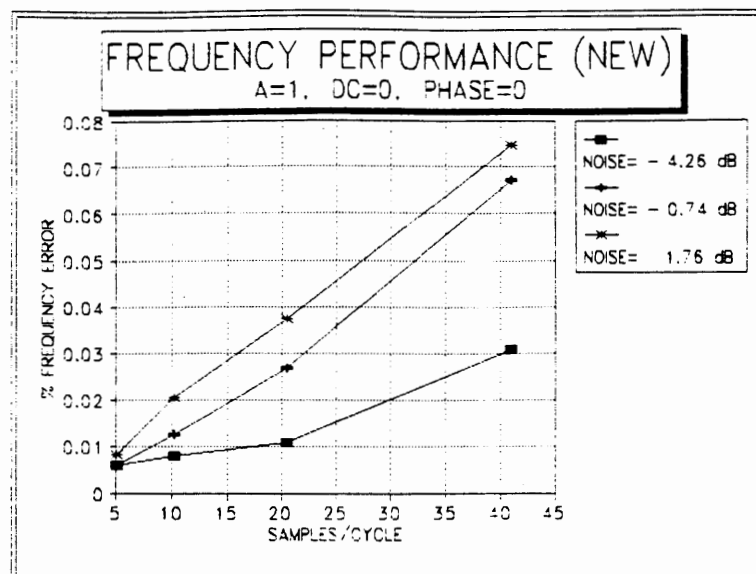
**Figure C.4.** Plot of *dc* offset estimation errors for increasing sampling rate: (a) low and (b) higher noise levels.



**Figure C.5.** Plot of *frequency estimation errors* for increasing linear distortion.

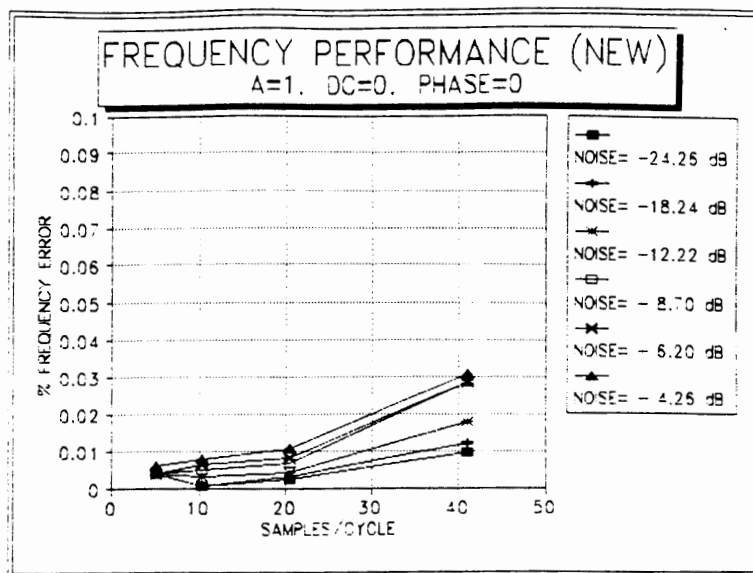


(a)

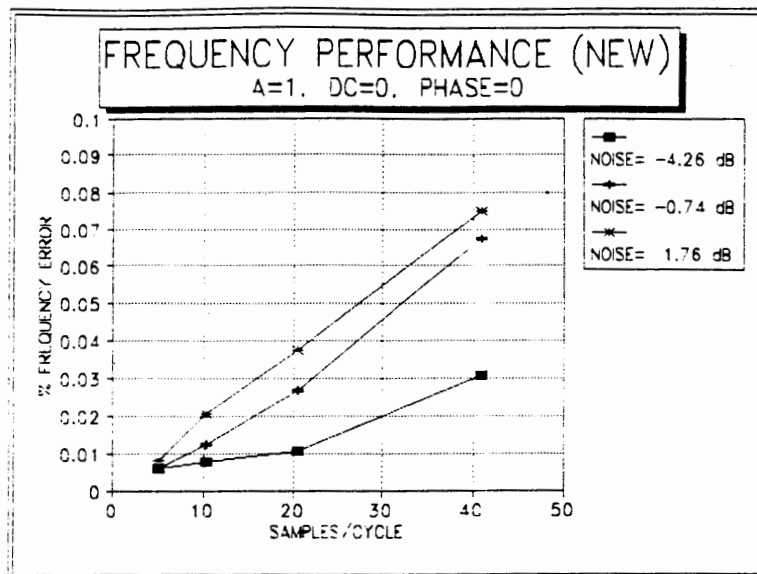


(b)

**Figure C.6.** Plot of *frequency* estimation errors for increasing sampling rate: (a) low and (b) higher noise levels.



(a)



(b)

**Figure C.6.** Plot of *frequency* estimation errors for increasing sampling rate: (a) low and (b) higher noise levels.

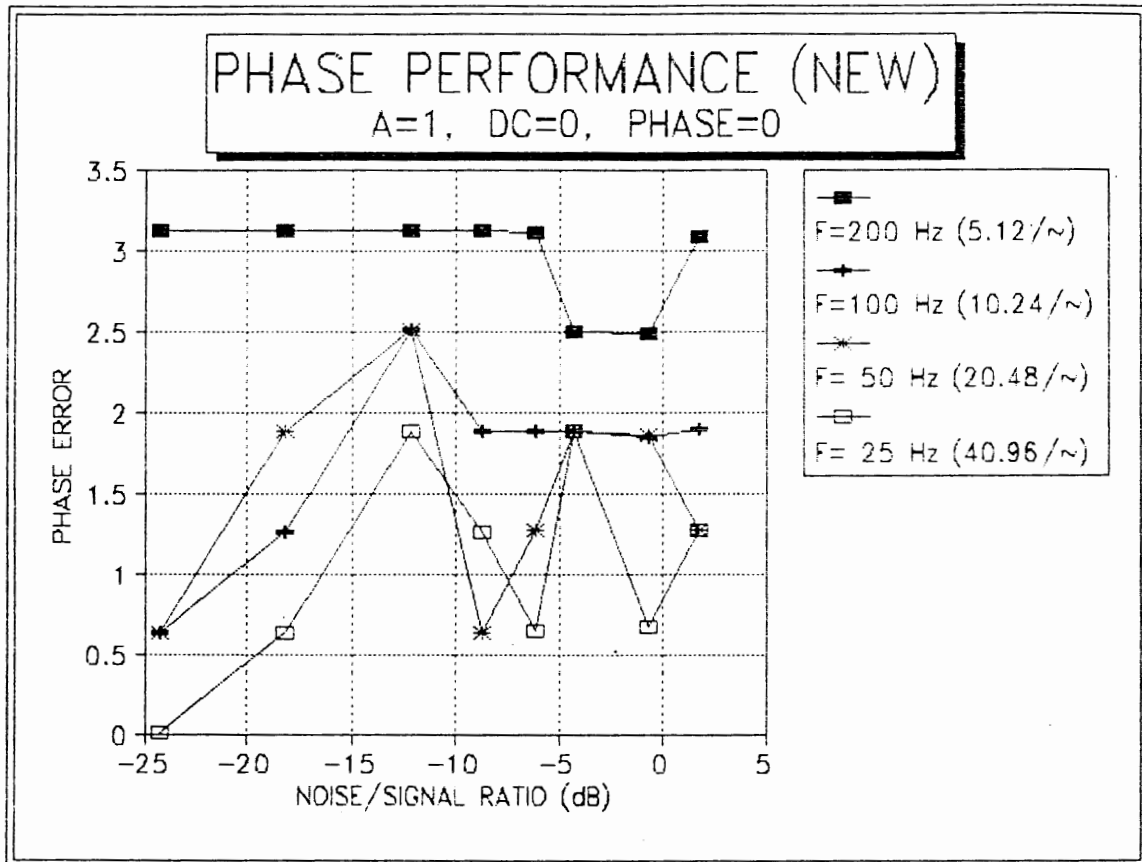
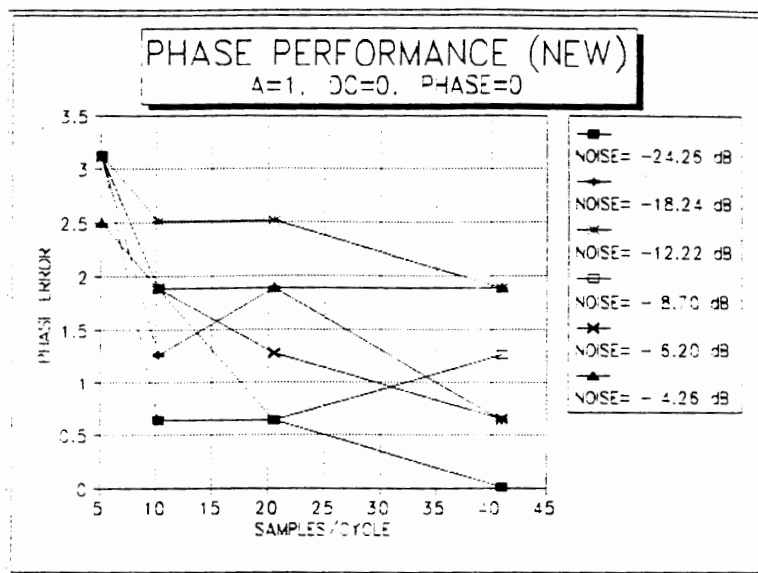
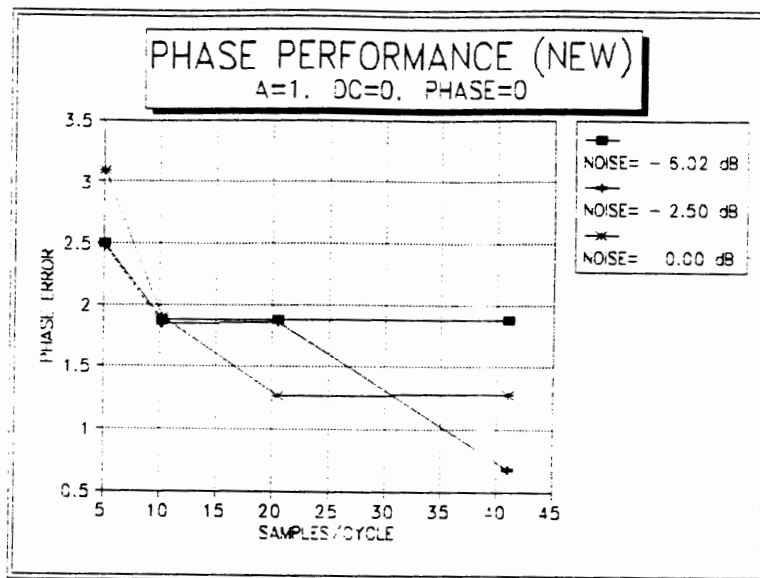


Figure C.7. Plot of *phase estimation errors* for increasing linear distortion.



(a)



(b)

**Figure C.8.** Plot of *phase* estimation errors for increasing sampling rate: (a) low and (b) higher noise levels.

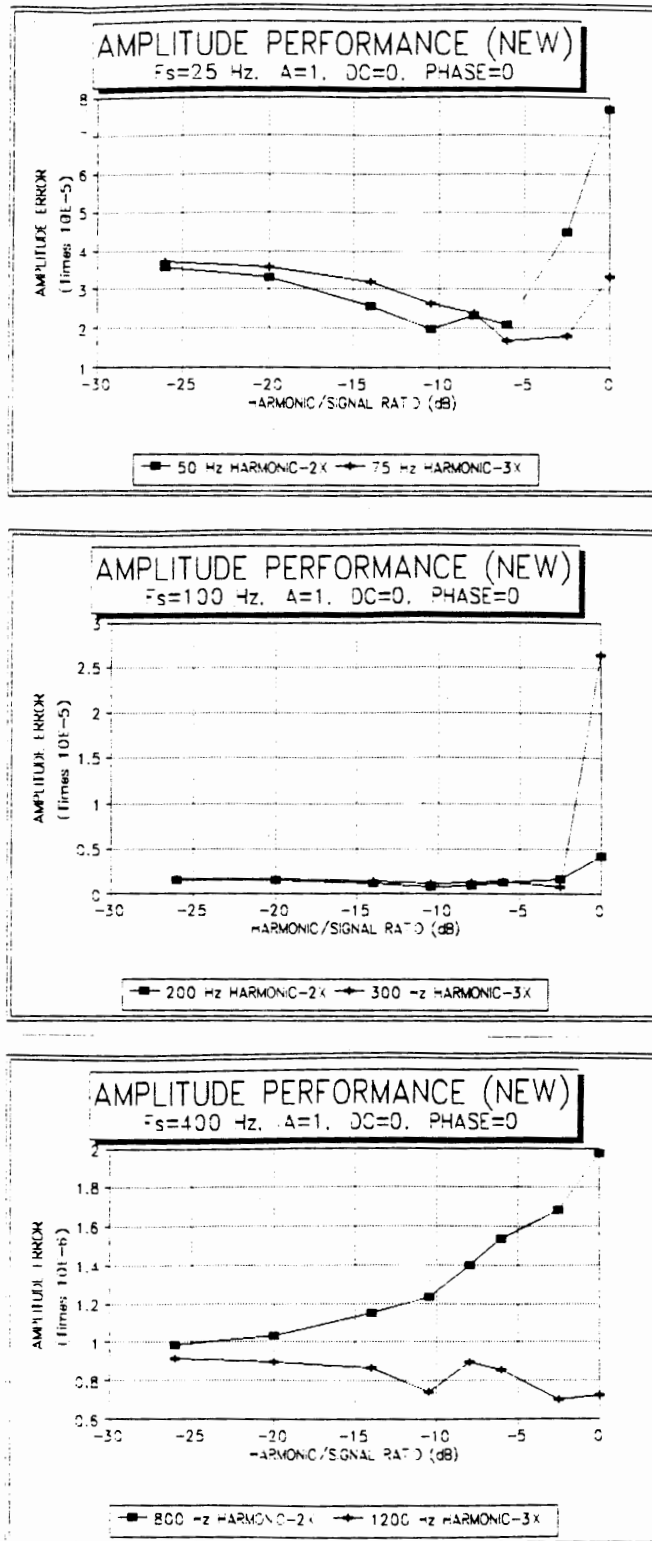


Figure C.9. Plot of *amplitude* estimation errors for increasing nonlinear distortion.

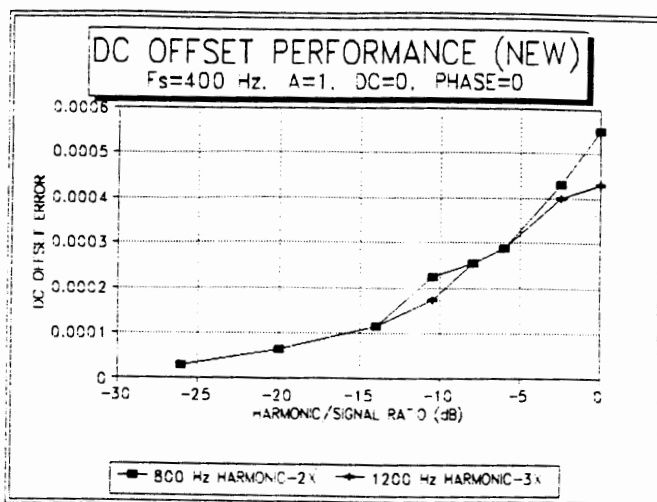
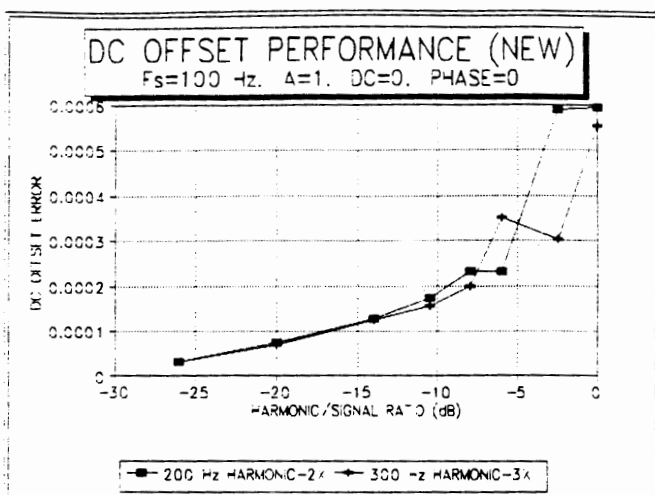
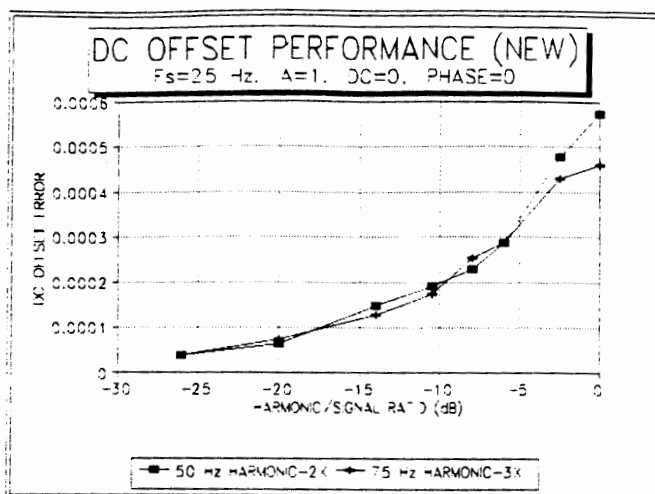


Figure C.10. Plot of *dc offset* estimation errors for increasing nonlinear distortion.



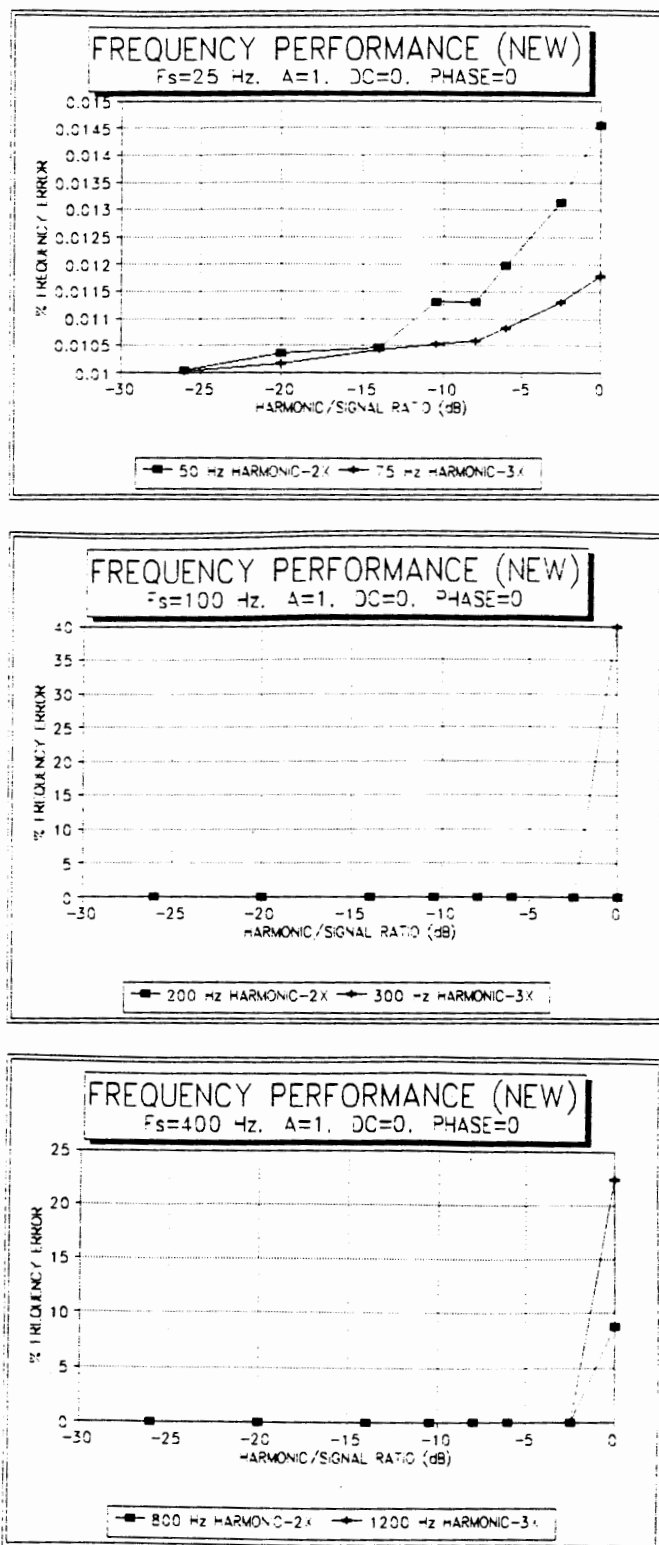


Figure C.11. Plot of frequency estimation errors for increasing nonlinear distortion.

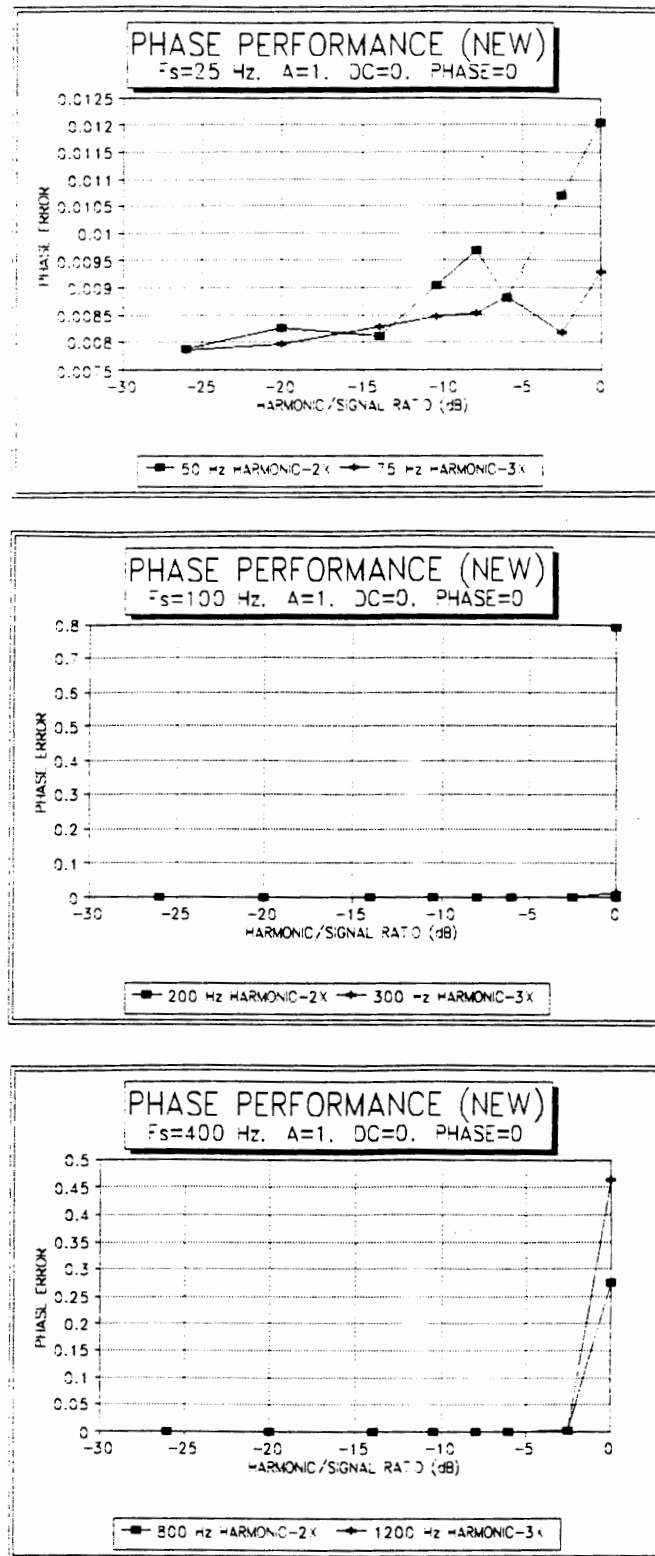


Figure C.12. Plot of *phase* estimation errors for increasing nonlinear distortion.