AUTOMATED TRANSFER FUNCTION MEASUREMENTS

by

TIRUVOOR RADHAKRISHNA RAMESH

B.E., University of Mysore, India, 1985

A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Electrical and Computer Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1989

Approved by:

M S P Lucas

Major Professor

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to thank my major professor, Dr. Michael S.P. Lucas, for his encouragement and advice during the course of my research and education at Kansas State University.

I wish to express my appreciation to Dr. Andrzej Rys and Dr. Sallie McNulty for consenting to sit on my committee. Also, I would like to thank all the faculty members for their guidance and instruction during my studies at Kansas State University.

Finally, I would like to thank all my family members for their support and inspiration throughout my education.

AUTOMATED TRANSFER FUNCTION MEASUREMENTS

1. INTRODUCTION

Frequency response or transfer function measurements are important to component and circuit performance. The frequency response of a linear system is a frequency dependent relation, in both gain and phase difference, between the steady state sinusoidal input and output of a circuit or device.

Oscilloscopes can be used to make gain and phase-shift measurements because of their voltage and timing properties. With the oscilloscope, the user can observe the waveforms under test. The gain of the circuit is determined by visually measuring the peak amplitudes of the input and output waveforms and calculated using the relation

$$\text{Gain(dB)} = 20 \log_{10}(V_{out}/V_{in})$$

The phase-shift between the two equivalent frequencies is based on the difference between the zero crossings in a particular direction of the two signals and can be measured by the dual trace method or Lissajous pattern method. Measurements made on a conventional oscilloscope require repeated setup, user interpretation of the CRT screen, and calculations. Such measurements consume time and suffer from low repeatibility.

It is therefore necessary to automate these measurements for more consistency, accuracy and speed than manual methods. Several automated techniques are discussed in the literature. This thesis discusses some techniques for automating frequency response measurements using computer controlled lock-in systems. Also, a software simulation of the measurement procedures is described and compared with the results obtained earlier.

A brief introduction to lock-in systems is presented in section 2. Section 3 deals with the measurement procedures, practical implementation of the procedures on two test circuits and probable error sources in the measurements. Section 4 deals with the software techniques for frequency response measurements based on the input and output samples of a test circuit.

2. LOCK-IN SYSTEMS

Transfer function measurements may involve the measurement of signals obscured by high levels of noise and interference. In many experiments, the noise level due to thermal noise alone may be several orders greater than the signal of interest. The amplitude and phase variations introduced by the circuit or device have to be estimated under adverse conditions of signal-to-noise ratio.

The problem is essentially one of 'signal recovery'. This problem can be better understood by an examination of the signal and noise voltages that appear at the output of a typical experimental system. A frequency domain transformation of the output helps in identifying the signal of interest, noise and interference which are obscured in the time domain as shown in fig 1. In evaluating the effect of noise on a signal, the distribution of noise components with frequency is more important than the total noise power accompanying the signal.

A true signal recovery system must be capable of responding to the signal 'buried in noise'. Moreover, it must be able to determine the amplitude and phase variations over the desired frequency range. Fig 2 shows the characteristics of a signal recovery system to be used in transfer function measurements.

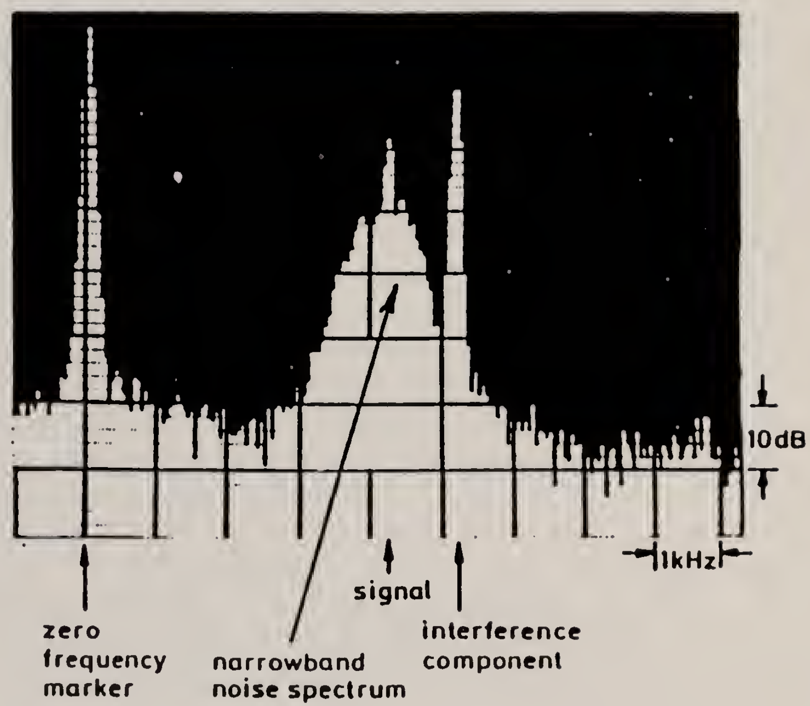

Fig 1. Frequency domain view of a signal recovery problem
(Meade M.L, Lock-in amplifiers: principles and applications, 1983, p12)

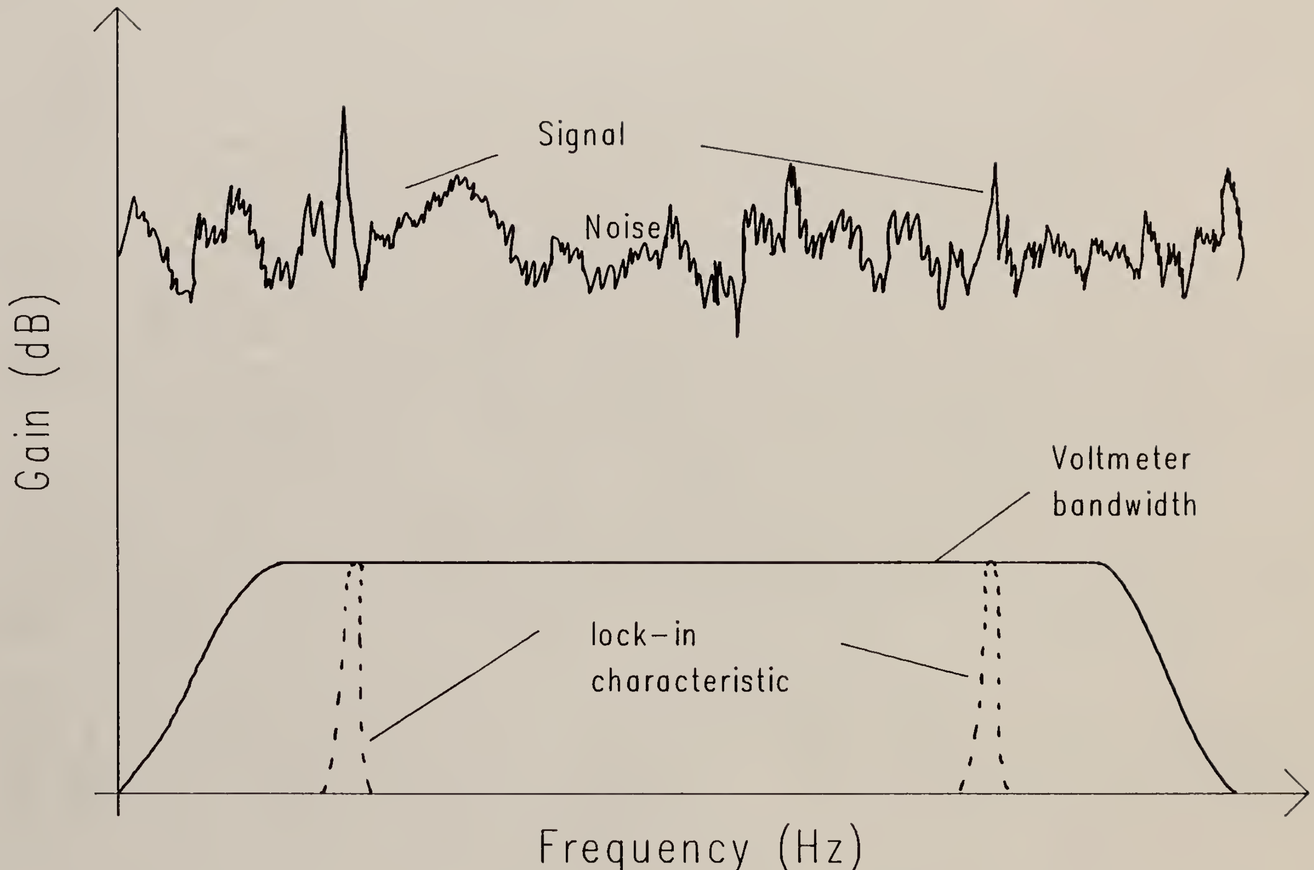

Fig 2. Signal recovery characteristic for transfer function measurements

Phase Sensitive Detectors (PSD) are commonly used in signal recovery applications. A phase sensitive detector provides an output proportional to the amplitude of the signal and the phase difference between the signal and the reference. In transfer function measurements, a PSD based system must lock on to the signal of interest over the desired frequency range. Systems operating on the PSD principle are termed lock-in systems.

2.1 Principles of operation:

The requirements of a basic lock-in system are shown in fig 3. Lock-in systems invariably use a reference signal derived from the excitation source. The correlation between the signal of interest and the reference signal is tested by multiplying together the two inputs to form the product

$$V_p(t) = r(t) \; (\; s(t) + n(t) \;)$$

where

r(t) --> reference signal

s(t) --> signal of interest

n(t) --> noise and interference

The higher products of multiplication are suppressed by a lowpass filter at the system output. The final output will be a constant voltage proportional to the signal

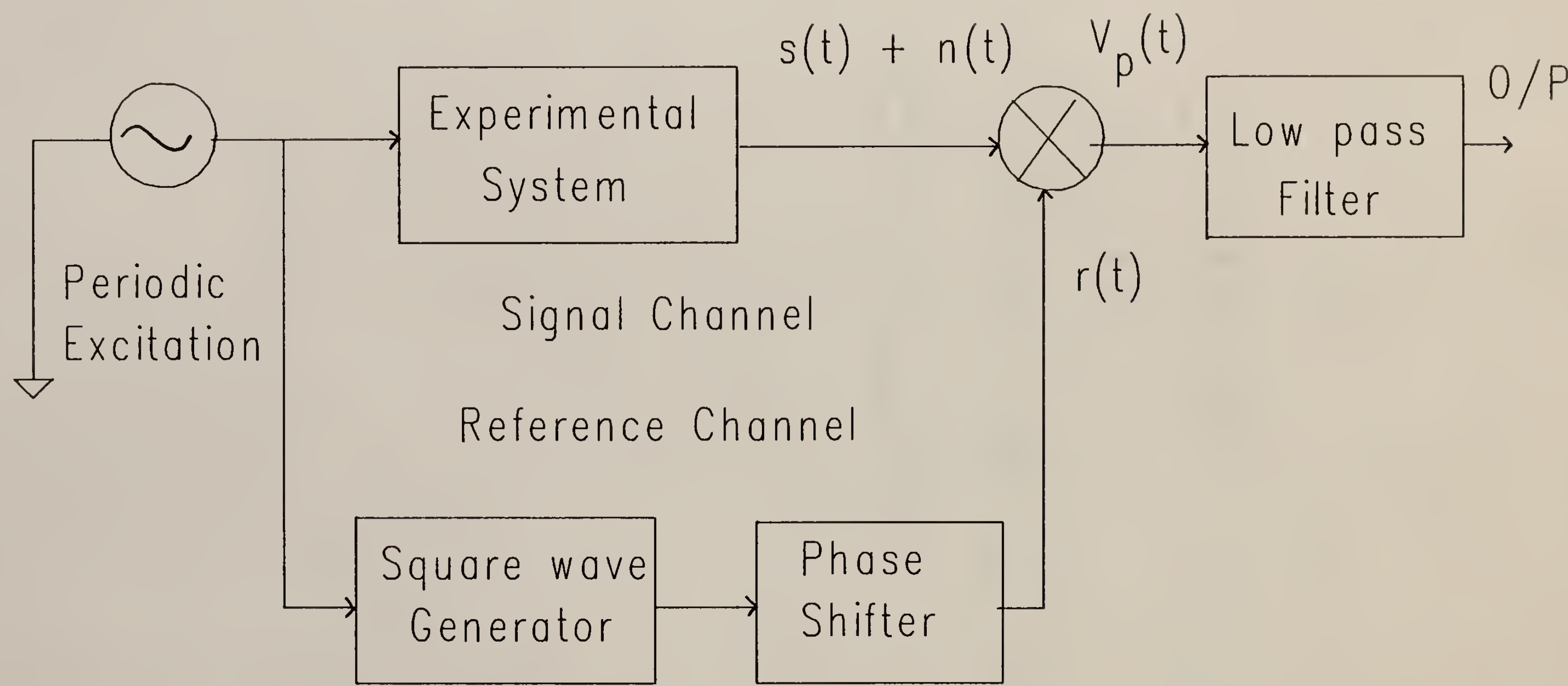

Fig 3. Block diagram of basic lock-in system

level when the signal of interest and the reference signal are closely correlated. The product of n(t) and r(t) averaged over a period of time will be zero.

2.2 Signal channel:

The signal channel mainly consists of the experimental system driven by a sinusoidal source. Commercial lock-in systems are usually provided with optional preamplifiers not merely to increase the gain but also to provide a noise match to the signal source. Signal conditioning filters are sometimes provided to increase system sensitivity. There is no need to clean the signal prior to detection in a synchronous system. The introduction of filters in the signal channel poses problems in phase measurements. The phase shifts introduced by these filters have to be compensated in the reference channel. Therefore, filtering has to be kept to a minimum.

2.3 Reference channel:

The main purpose of the reference channel is to provide a signal synchronous with the signal applied to the experimental system. The reference signal may be either a square wave or a sine wave determined by the design of the phase sensitive detector.

A phase shifter is provided in the reference channel to control the relative phase of the two inputs to the multiplier. The phase shifter forms a major component in transfer function measurements. It has to provide accurate phase shifts over the desired frequency range. The phase shifters available are of the voltage control type requiring the tuning of either a resistor or capacitor to obtain the proper phase shift. Several phase shifter circuits using transistors, operational amplifiers, Phase locked loops are available in the literature. In precision phase measurements, a calibration procedure is necessary to estimate the phase shift.

2.4 Phase sensitive detector:

The multiplier and the lowpass filter together constitute the phase sensitive detector. The phase sensitive detectors usually used are of the switching multiplier type as shown in fig 4 due to its exceptionally wide dynamic range and operational simplicity. The reference signal is a square wave which controls an electronic switch. The output of the switch is passed through a lowpass filter which provides a dc voltage proportional to the amplitude of the signal.

Fig 5 shows the waveforms at the output of the phase sensitive detector for different phase relations between

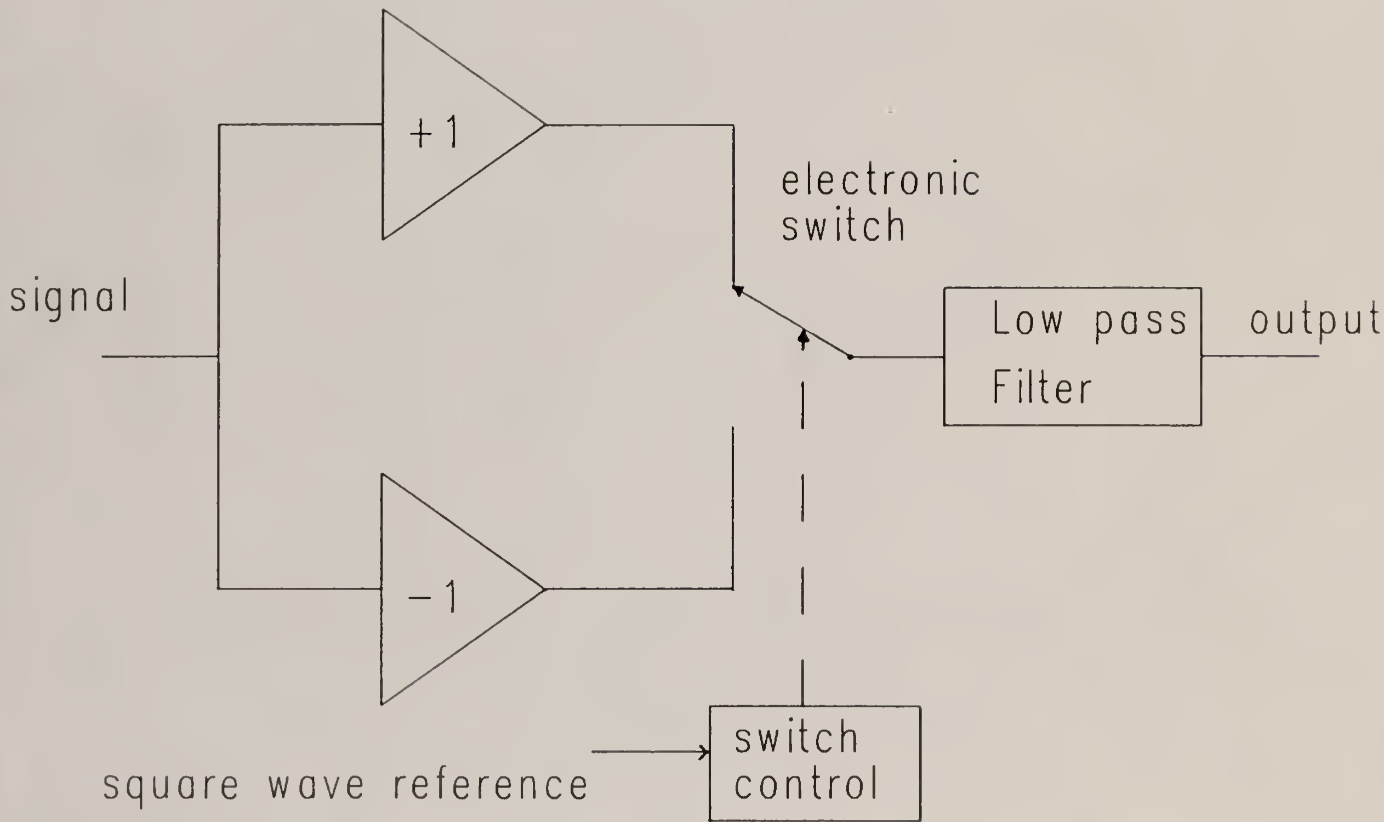

Fig 4. Block diagram of a phase sensitive detector
(Meade M.L, Lock–in amplifiers: principles and applications, 1983, p32)

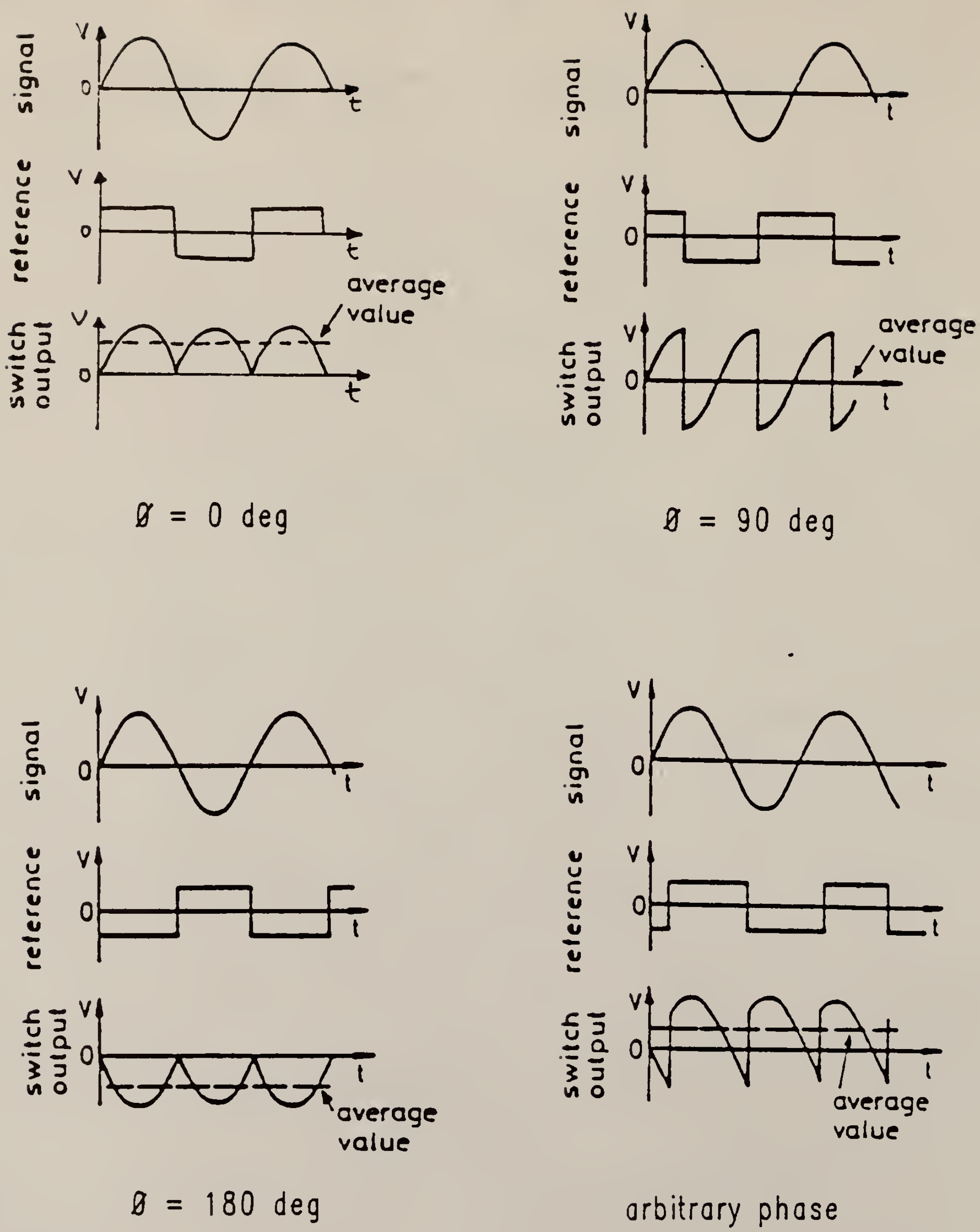

Fig 5. Waveforms in a phase sensitive detector for different phase relations

(Meade M.L, Lock-in amplifiers:principles and applications, 1983, p32-33) the inputs of the multiplier. The output depends not only on the amplitude of the signal but also on the phase difference between the signal and the reference. The response of the lock-in system is maximum when the inputs to the multiplier are precisely in phase. A mathematical analysis of the lock-in system for both square and sinewave references is given in Appendix A.

The output lowpass filter provides a major means of improving the signal to noise ratio in lock-in systems. The purpose is to filter the higher products of multiplication at the final output. The bandwidth of the filter is typically around 1 Hz. The settling time of the filter (inversely proportional to the bandwidth) is important otherwise the response will be sluggish. The lowpass filter may be a single section or double section RC filter with a roll off of 6 dB/Octave or 12 dB/Octave respectively beyond the cut off frequency.

2.5 A Practical lock-in system:

The Function Generator provides the sinusoidal input to the lock-in system. Most of the function generators available have the 'Sync output' capability. In addition to producing the desired signal, they also produce a square wave synchronous with the zero crossing of the signal. The levels of the square wave depend on the termination on the

'Sync output' line. The HP3325A is one such Function Generator and can be used to provide the input and reference signals for the lock-in system.

The switching multiplier was implemented using the CMOS Analog Switch DG305 (SPDT) as shown in fig 6. This implementation requires inverting and non-inverting unity gain amplifiers in the signal channel which causes errors in precision phase measurements. In addition, the switch does not work properly at high frequencies or small signal amplitude.

An alternative implementation is to use a precision IC Multiplier. The AD534 is a four quadrant analog multiplier without any external trimming for offset and gain accuracy. The generalized transfer function is given by

$$V_{out} = A \left[\frac{(X1 - X2)(Y1 - Y2)}{SF} - (Z1 - Z2) \right]$$

where

A = open loop gain of output amplifier, typically 70 dB at dc.

X,Y,Z = differential input voltages, high impedance

SF = Scale factor, pretrimmed to 10 V.

The operation of the AD534 as a multiplier is described by the equation

$$(X1 - X2)(Y1 - Y2) = 10\ V(Z1 - Z2)$$

The small signal bandwidth is typically around 1 MHz with offset voltage $\pm$ 5 mV. The lock-in system using an AD534 is shown in fig 7.

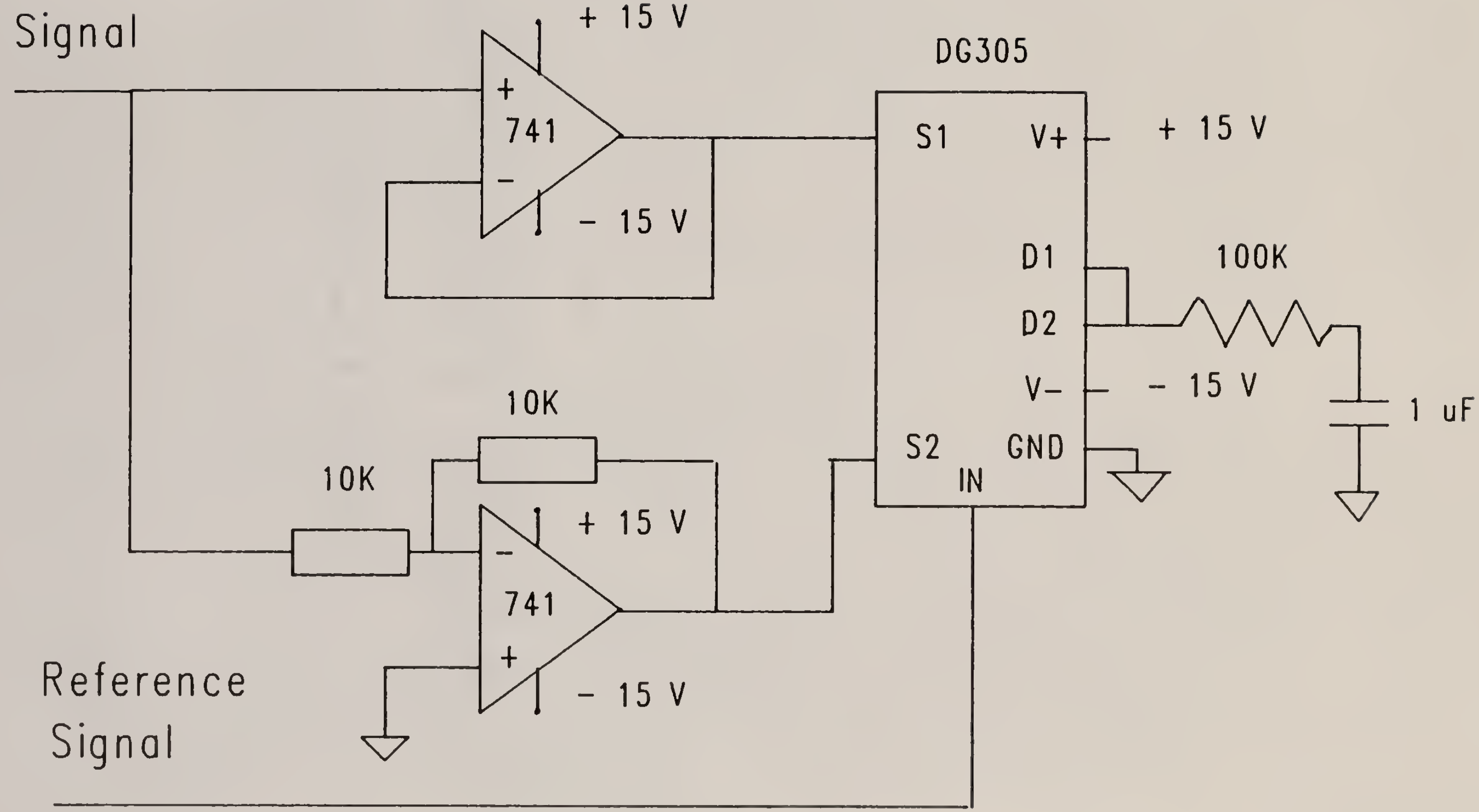

Fig 6. Phase sensitive detector using an analog switch

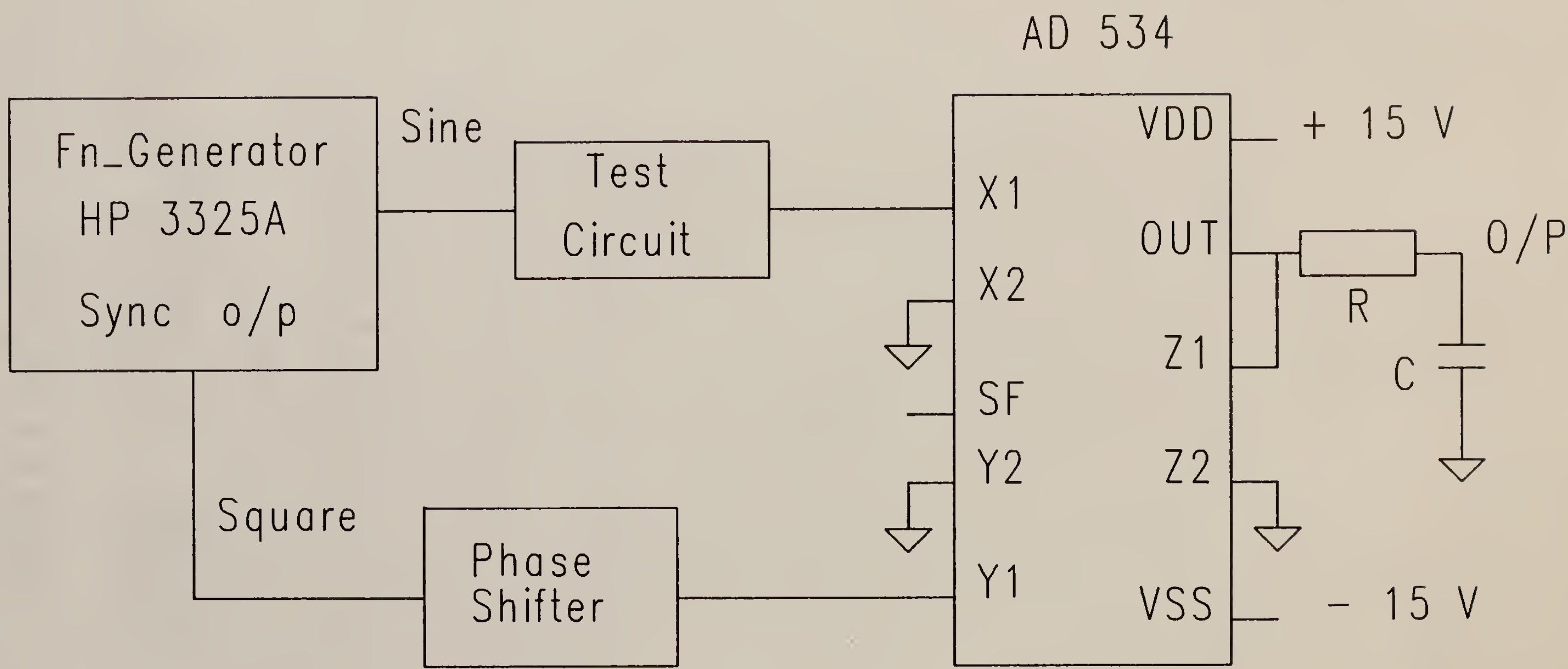

Fig 7. Block diagram of a practical lock-in system

3. GAIN PHASE MEASUREMENTS

Frequency response measurements refer to the performance evaluation of a test circuit in amplitude and phase when it is subjected to a time varying input voltage. The distortion in the input signal has to be minimized for accurate measurements.The analog input voltage usually provided in these measurements is a sinusoidal waveform. This is because of availability of sinusoidal waveform generators with low distortion and the sinewave allows the performance to be specified at discrete frequencies.

3.1 Measurement procedures:

Two measurement procedures will be discussed.

1. Simultaneous sampling method.
2. Using lock-in systems.

3.1.1 Simultaneous sampling method:

This method consists of simultaneously sampling the input and output signals of a test circuit for a particular frequency. For proper measurements, at least two cycles of the signals have to be sampled. The gain and phase-shift of the circuit for a particular frequency is obtained in software based on these samples. The maximum amplitudes for both the signals are determined and the ratio of the maximum output amplitude to the maximum input amplitude gives the gain of the circuit for that frequency. The phase-shift introduced is estimated using the difference in zero crossing of the two signals in a particular direction. A knowledge of the sampling frequency is necessary to obtain the magnitude of the phase-shift. The sign of the phase-shift depends on which one of these signals reaches the positive peak first. However, it becomes difficult to interpret the phase relationship if both the signals attain the peak in the same sampling interval. Also, the phase measurement is sensitive to any offset introduced by the circuit. The accuracy of gain and phase measurements using this method is entirely dependent on the sampling frequency. The higher the sampling frequency, more accurate are the measurements . A program to obtain simultaneous sampling of input and output signals using the Keithley 194A is provided in Appendix C.

3.1.2 Using lock-in systems:

The key to the measurement procedure using lock-in systems is to obtain the maximum average output signal. The following methods are possible.

3.1.2.1 Reference channel variations:

For any frequency setting, vary the phase of the reference channel from 0 to 360 degrees in steps depending on the accuracy of measurements required. Obtain the average output voltages for each phase setting. From these readings, the maximum average output voltage and the corresponding phase-shift can be obtained. If the phase-shift obtained is greater than 180 degrees, the output signal leads the input signal. If the phase-shift is less than or equal to 180 degrees, the output lags the input. The time taken by this method to perform a frequency response measurement for a reasonable phase accuracy of ± 1 degree is quite large. (at least an hour!). The offset introduced by the circuit affects the gain measurements but not the phase measurements.

3.1.2.2 Null shift procedure:

The phase sensitive detector response is proportional to $\cos\theta$, where θ represents the relative phase of the signal and reference at the phase sensitive detector input. (Appendix A). The variation at the output of the phase sensitive detector as the reference channel phase is varied from 0 to 360 degrees is shown in fig 8. The cosine nature of the output can be exploited to obtain the phase-shift introduced by the circuit. The procedure is outlined below.

- Start from an arbitrary initial phase condition (0 deg).
- The phase sensitive detector output is nulled by adjusting the reference phase. (quadrature phase). This establishes a quadrature condition at the phase sensitive detector.

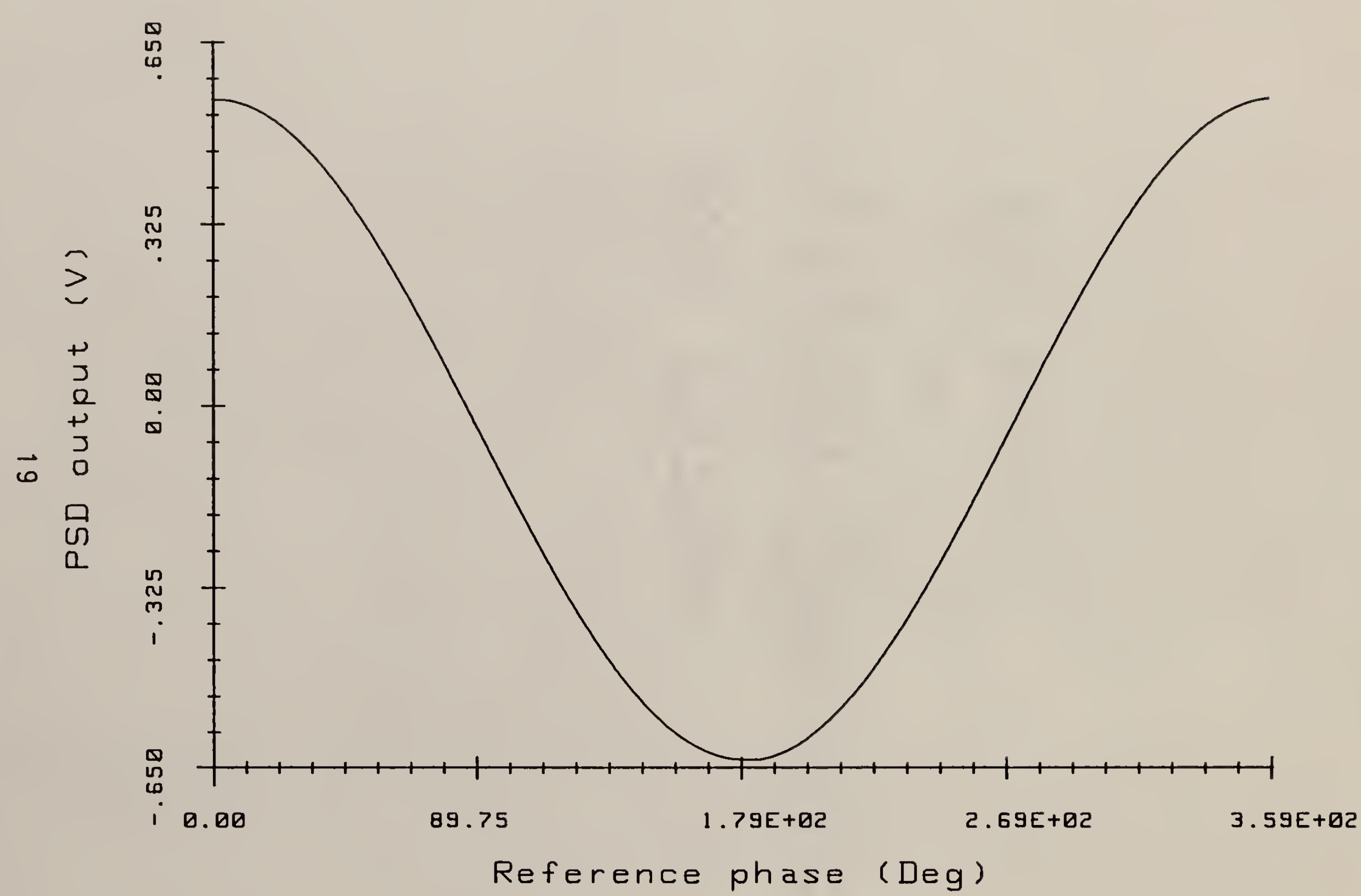

Fig 8. Phase sensitive detector output variation with reference channel phase

- The phase-shift introduced by the circuit is then established by shifting the quadrature phase by 90 degrees.

Frequency response measurements done using this method are susceptible to any offset introduced by the lock-in system.

3.2 Implementation:

3.2.1 Computerized Test system:

The basis of frequency response measurements using lock-in systems is to precisely adjust the phase shifter in the reference channel until the inputs to the phase sensitive detector are in phase. The phase shifter circuits are frequency dependent and physical tuning of circuit elements is undesirable in a computer controlled lock-in system. In order to automate these measurements, function generators with phase shifters and computer control capability are necessary.

The block diagram of a computerized test system is shown in fig 9. The Synthesizer and the Digital Voltmeter can be programmed by the controller HP9836 over the IEEE 488 bus. A Printer and Plotter are also connected to represent data and results.

The HP8904A Multifunction Synthesizer (Option 002) is a dual channel synthesizer with the channels programmed individually. The relevant specifications of the

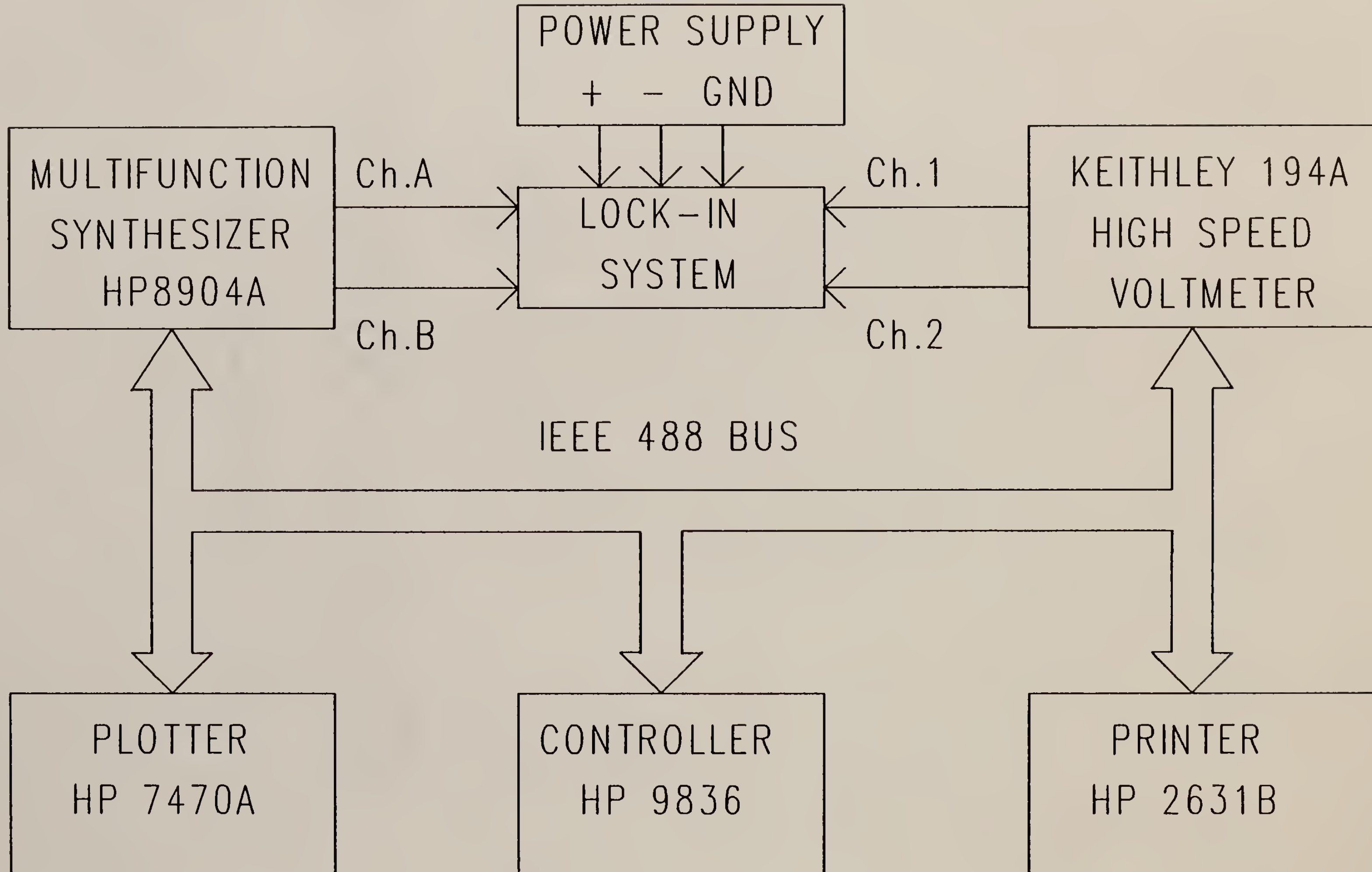

Fig 9. Block diagram of a computerized test system

synthesizer are provided in Appendix B. Channels A & B on the synthesizer are used as the signal and reference channels of the lock-in system respectively. The phase of the signals in both the channels can be programmed from 0 to 359.9 degrees in steps of 0.1 degree.

The Keithley 194A High Speed Voltmeter is a dual channel digital voltmeter. The channels can be programmed individually to obtain the samples of the signal and store them in a buffer. Several mathematical functions such as Average, TRMS, Peak etc., are available which are computed based on the samples obtained. The sampling rate and the number of samples can be programmed . More details regarding the architecture and dual channel operation of Keithley 194A are provided in Appendix C.

The frequency range desired is upto 100 KHz. The reference signal when using the IC Multiplier as the phase sensitive detector can be either a square or a sine wave. The upper limit on the frequency range is decided based on the reference signal chosen. This is due to the maximum frequency specifications for sine and square waves on the HP8904A synthesizer (Appendix B). Accordingly, the measurements are carried out from 10 Hz to 100 KHz for sinewave reference and from 10 Hz to 50 KHz for a square wave reference.

3.2.2 Modifications to the lock-in system:

The sensitivity of the system is diminished considerably due to the attenuation by 10 at the multiplier output. This is due to the scale factor, SF, being 10. The SF can be trimmed down to 3 V. From the manufacturer's specifications, the maximum input allowed is 1.25 SF. Decreasing the SF would not help a great deal in increasing the sensitivity. A much lower scaling voltage can be achieved without any reduction of input signal range using a feedback attenuator. The elimination of scale factor can be done in software by multiplying every reading by the scale factor. Hardware scale factor elimination improves the range in which the voltmeter has to operate. However, two problems are associated with this:

1. The bandwidth of the multiplier is reduced to about 80 KHz in the presence of the peaking capacitor. This does affect the performance when a sine wave reference is used.
2. The output offset voltage increases by a factor of 10.

The offset affects the determination of the quadrature phase especially when the signal strength and the offset are comparable. The error is around 10 to 20 degrees. The effect of offset on phase determination is shown in fig 10. The offset can be removed in software as well as hardware. The offset is subtracted from every reading in software. Both the techniques require the estimation of offset prior

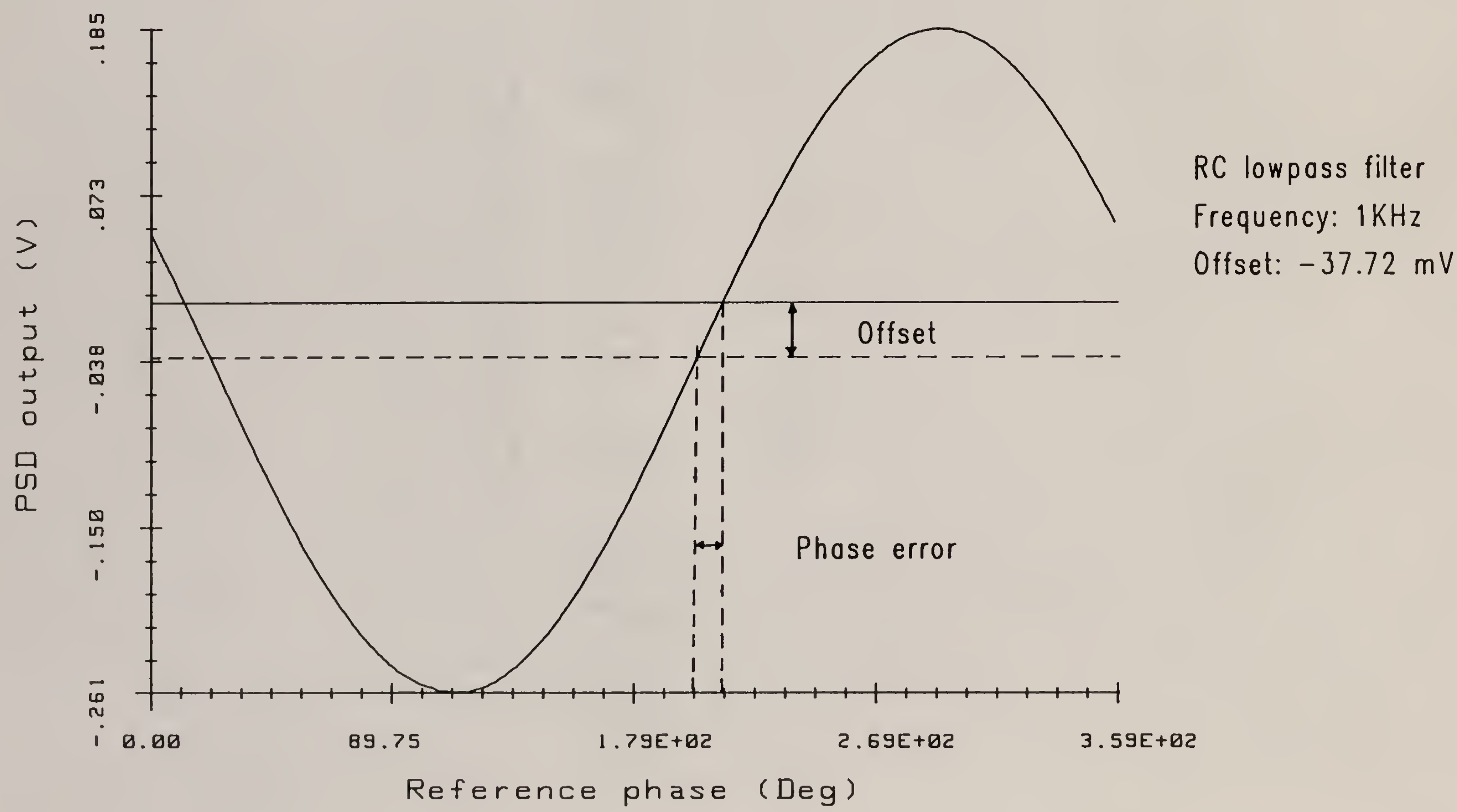

Fig 10. Effect of offset on phase determination

to the measurement process. Fig 11 shows the modified multiplier circuitry with unity SF and offset correction. In our implementation, the SF and offset are eliminated in hardware and software respectively.

3.2.3 Algorithm:

The null shift procedure outlined in section 3.1.2.2 with either a square or sine wave reference signal is used in this implementation.

An important criterion to be considered is to identify whether the output signal leads or lags the input signal. The relationship between the reference signal, input and output signals of the test circuit for the four general phase shifts is shown in fig 12. It is apparent that when the reference channel is shifted by more than 180 degrees, the output signal leads the input or else it lags.

The offset introduced by the lock-in system is removed in software by using the ZERO feature on the Keithley 194A Voltmeter.

The algorithm for frequency response measurements using the above test system is shown in the flowcharts given in figs 13 & 14. The program written in HP BASIC 3.0, using the above algorithm, is provided in Appendix D.

Autoset routine:

This routine incorporates both autophase selection and autosensitivity. The quadrature phase occurs at two phases

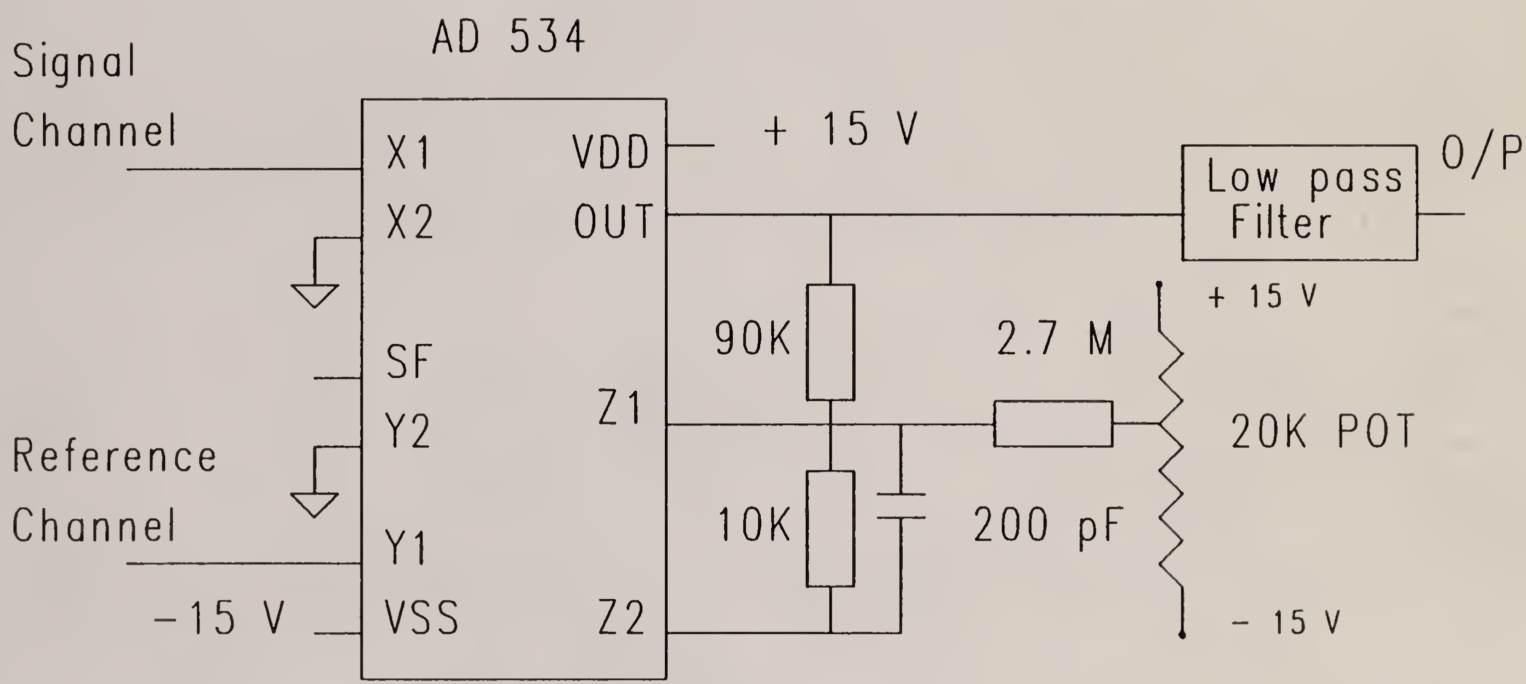

Fig 11. Modified multiplier circuitry with unity Scale factor and offset correction

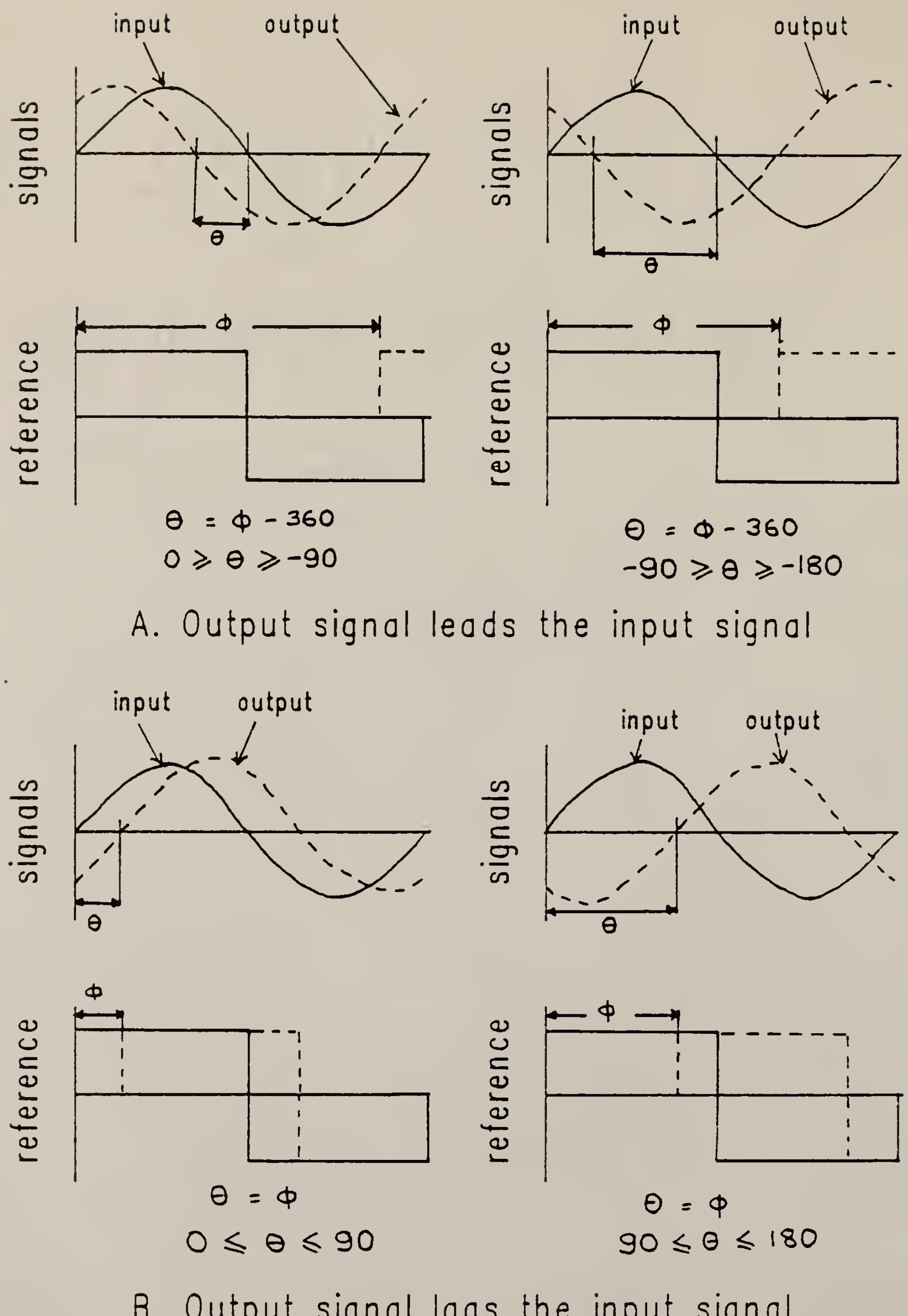

Fig 12. General phase relationships in a transfer function measurement

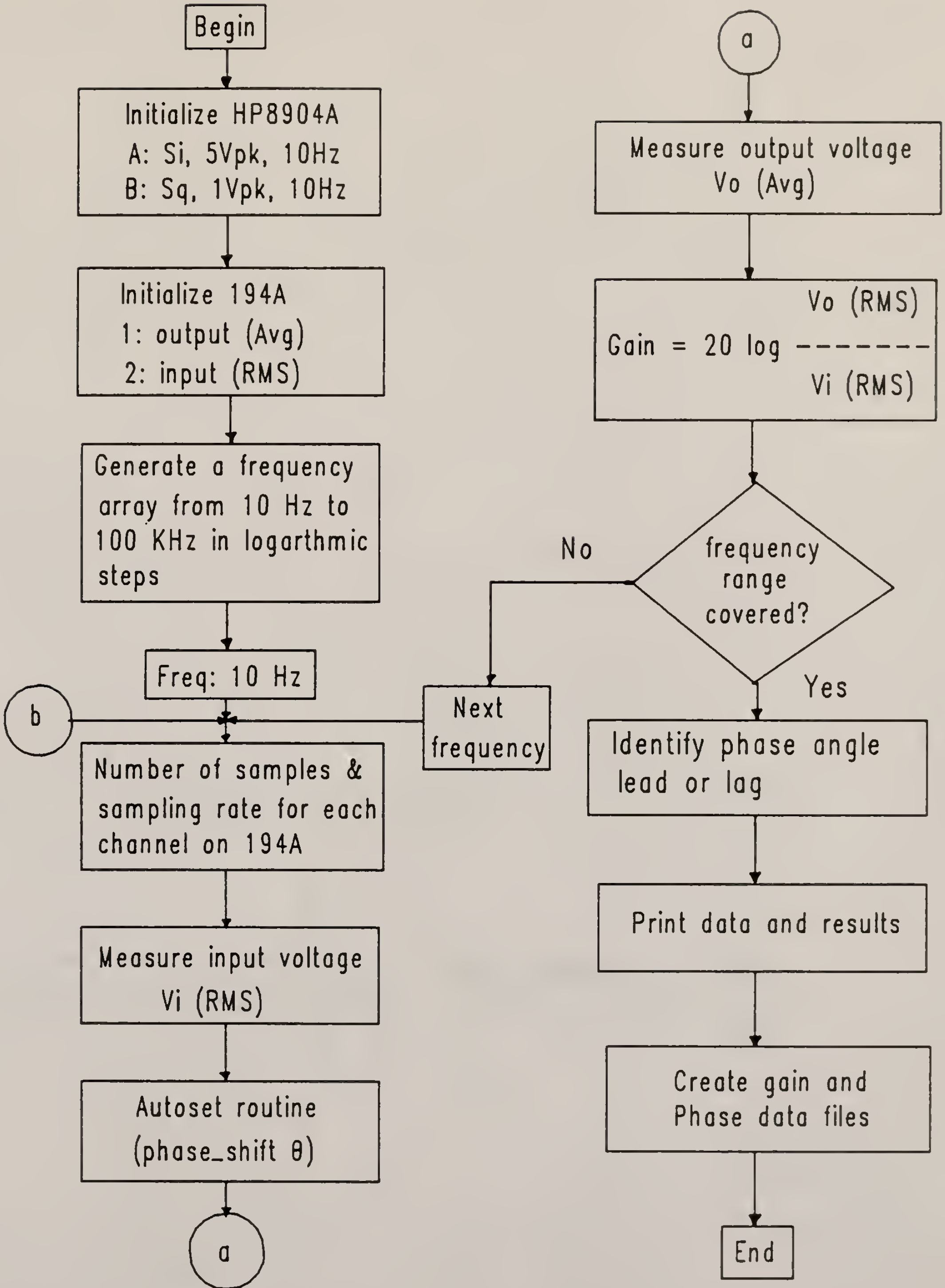

Fig 13. Main flowchart for transfer function measurements using lock-in systems

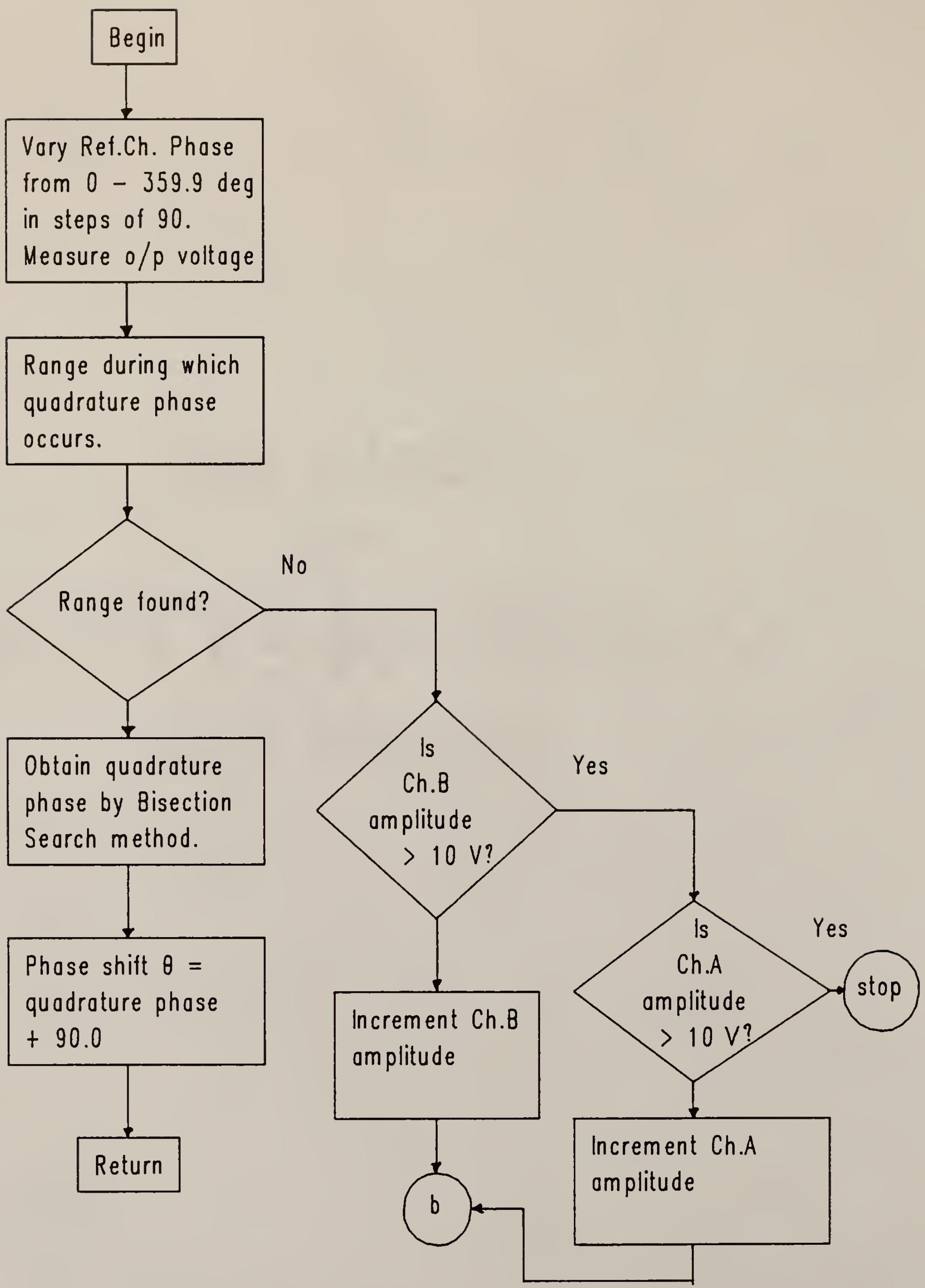

Fig 14. Autoset routine

180 degrees apart. The quadrature phase corresponding to the positive zero crossing of the output is determined using the 'Bisection Search Method'. In situations wherein the quadrature phase cannot be determined properly, the sensitivity of the system is increased by varying the amplitudes on both the signal and reference channels.

3.3 Frequency response of test circuits:

Gain and phase measurements using the procedure outlined in section 3.2.3 were carried out on the following test circuits.

1. RC lowpass filter
2. RC highpass filter

3.3.1 RC lowpass filter:

The transfer function of a RC lowpass filter is given by

$$\frac{V_o}{V_i} = \frac{1}{1 + jwRC}$$

With R = 22 K and C = 0.022 uF, the cut off frequency is 328.83 Hz. The amplitude and phase responses are determined using both sine wave and square wave references. These responses are compared with the ideal obtained from the transfer function. Figs 15 & 16 shows the amplitude and phase responses of a RC lowpass filter. The phase errors with sine wave and square wave references are illustrated in fig 17.

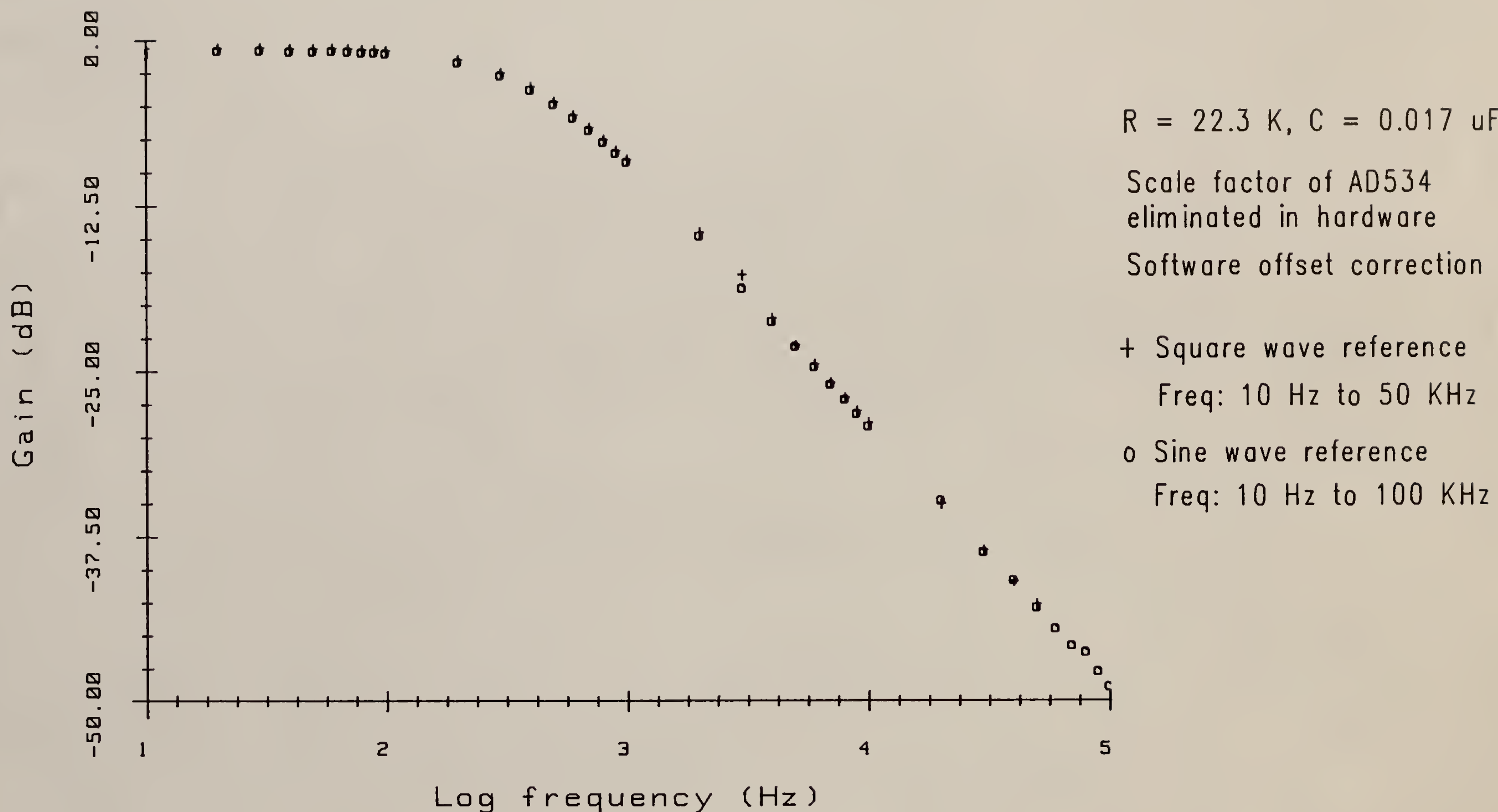

Fig 15. Amplitude response of RC lowpass filter

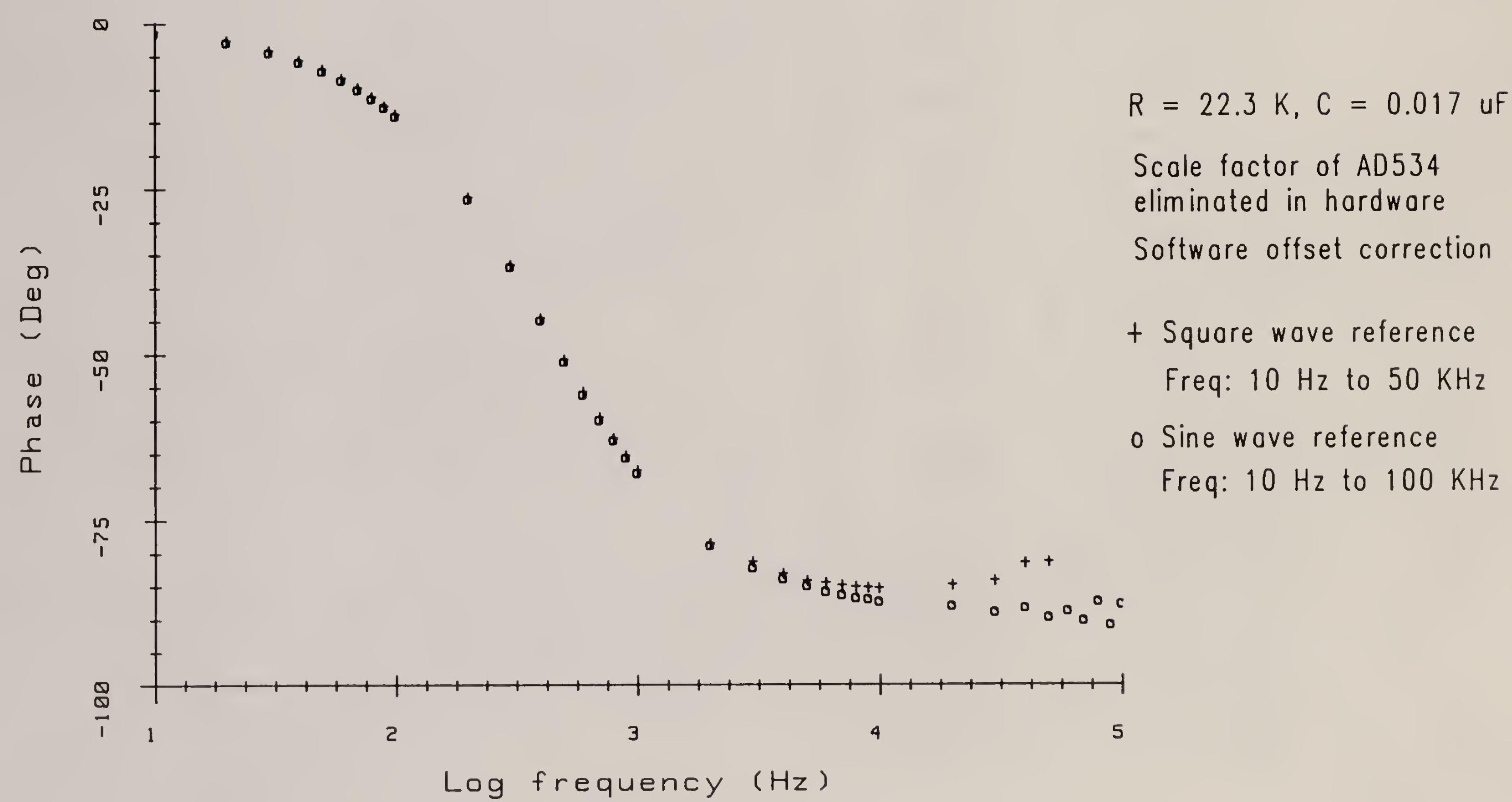

Fig 16. Phase response of RC lowpass filter

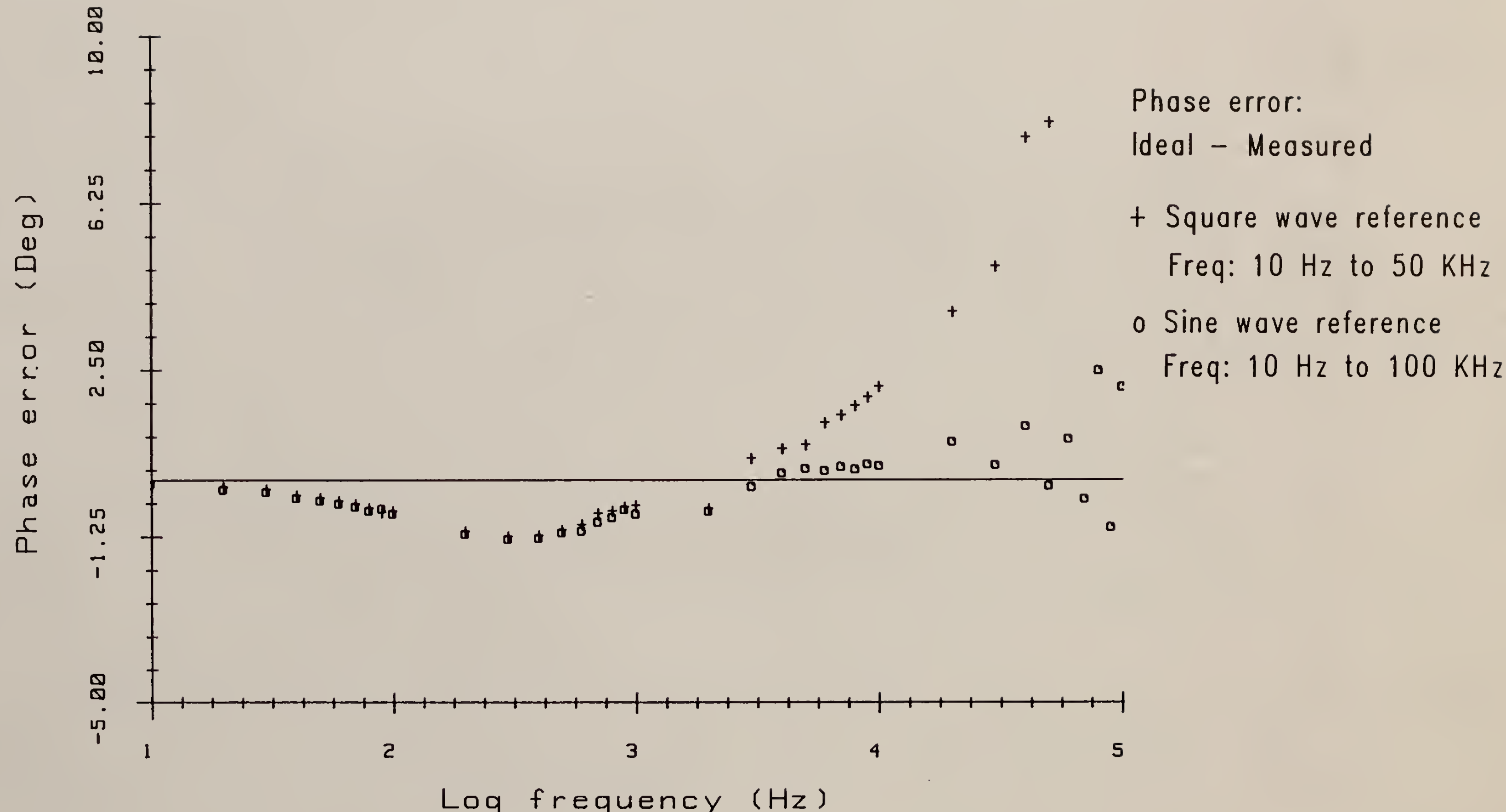

Fig 17. Error in phase measurements of RC lowpass filter

The phase error tends to increase in the high frequency range when the signal amplitude is very small. The error is less when a sine wave is used as the reference.

3.3.2 RC highpass filter:

The transfer function of a RC highpass filter is given by

$$\frac{V_o}{V_i} = \frac{jwRC}{1 + jwRC}$$

With R = 22 K and C = 0.022 uF, the cut off frequency is 328.83 Hz. The amplitude and phase responses of the highpass filter are shown in figs 18 & 19. The phase errors are shown in fig 20.

The phase error is again large in the high frequency range where the signal strength is large.

The amplitude response obtained using lock-in systems is a function of the phase response due to the nature of the measurement process. As evident from the plots, the gain errors are very small compared to the phase errors.

3.4 Probable error sources in phase measurements:

The principal sources of error affecting phase measurements are the following.

1. Signal and applied reference not strictly in phase.
2. Errors due to oscillator distortion.
3. Offset and drift in phase sensitive detector.
4. Component sensitivity.

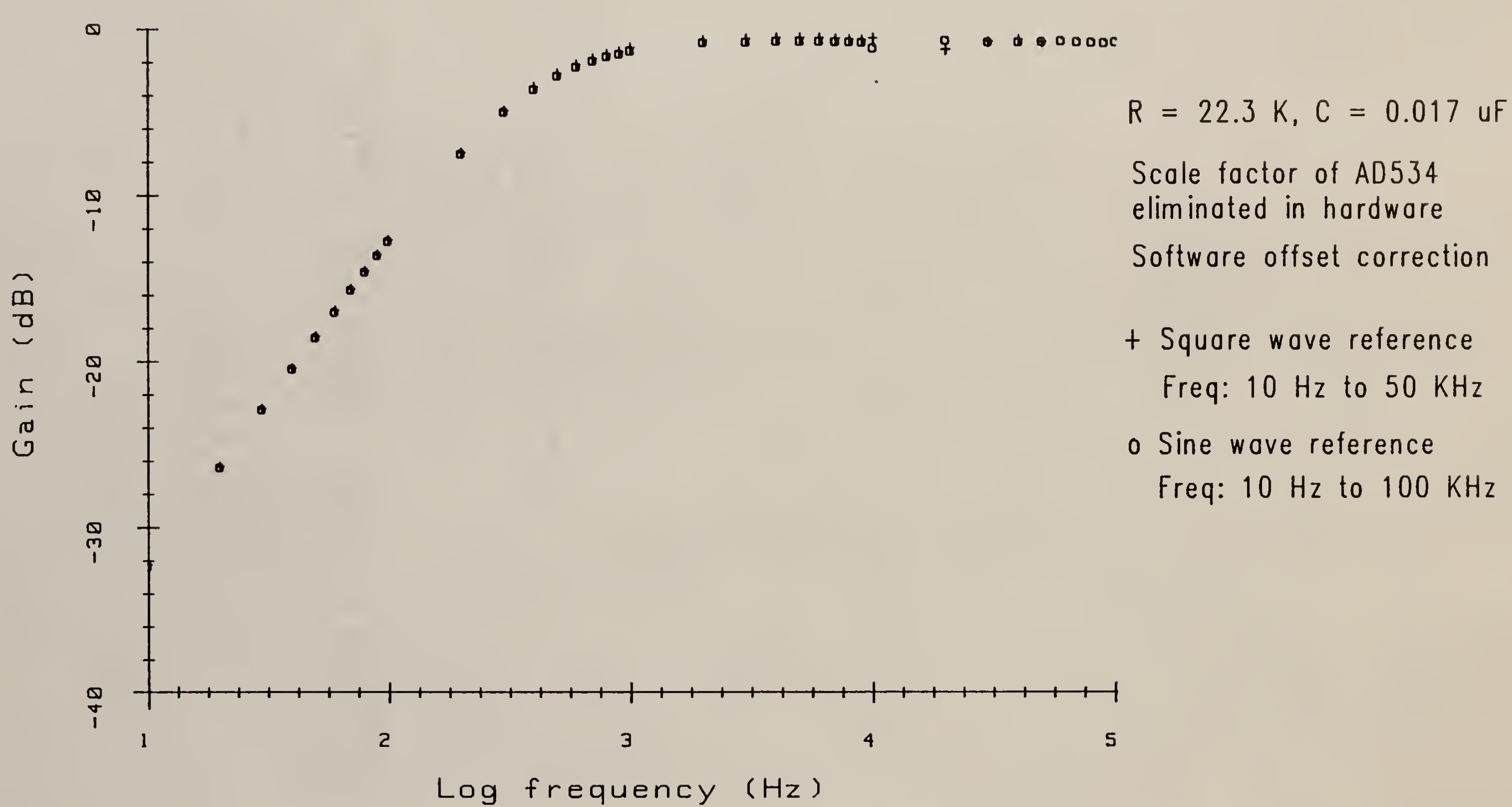

Fig 18. Amplitude response of RC highpass filter

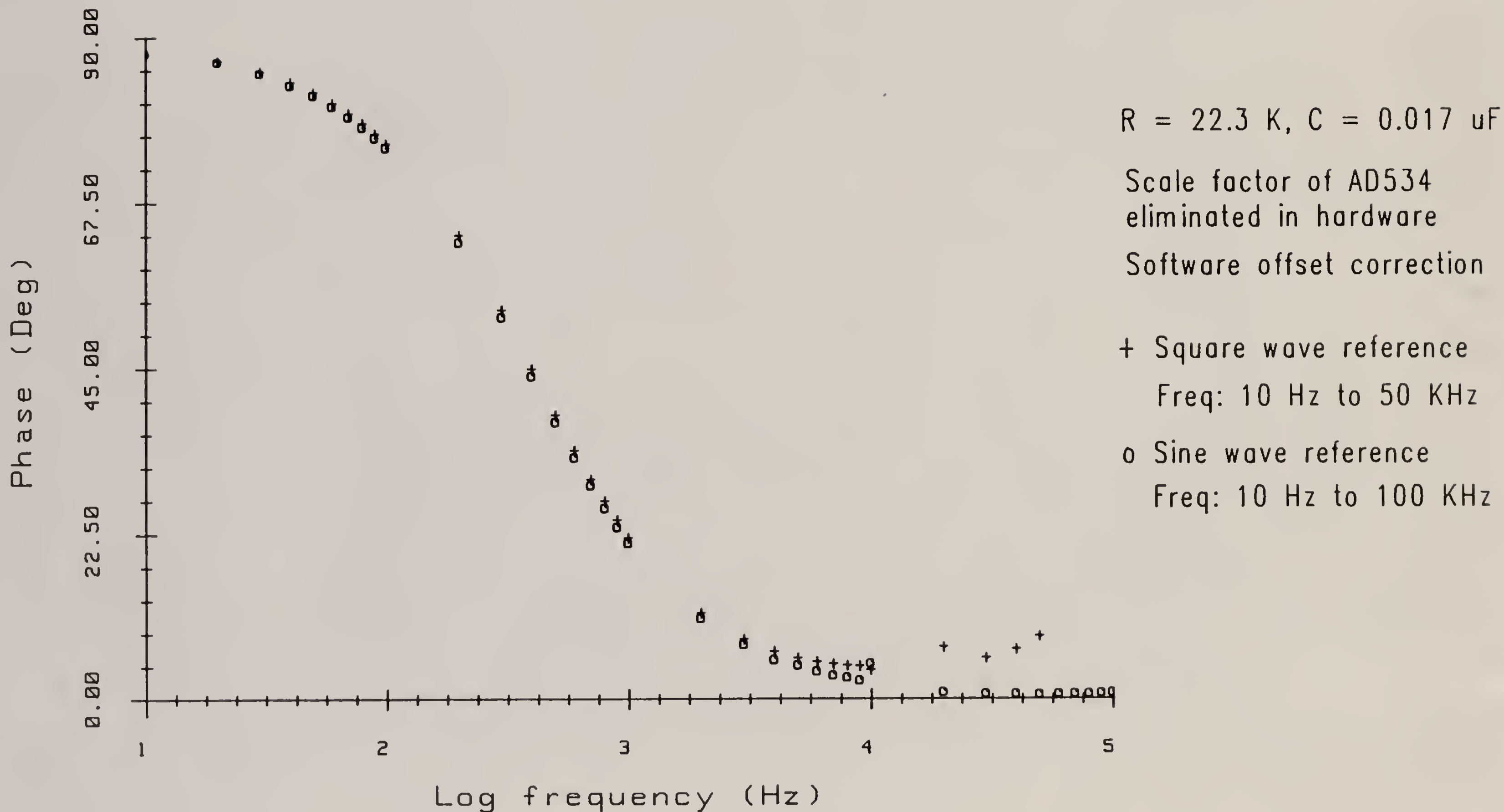

Fig 19. Phase response of RC highpass filter

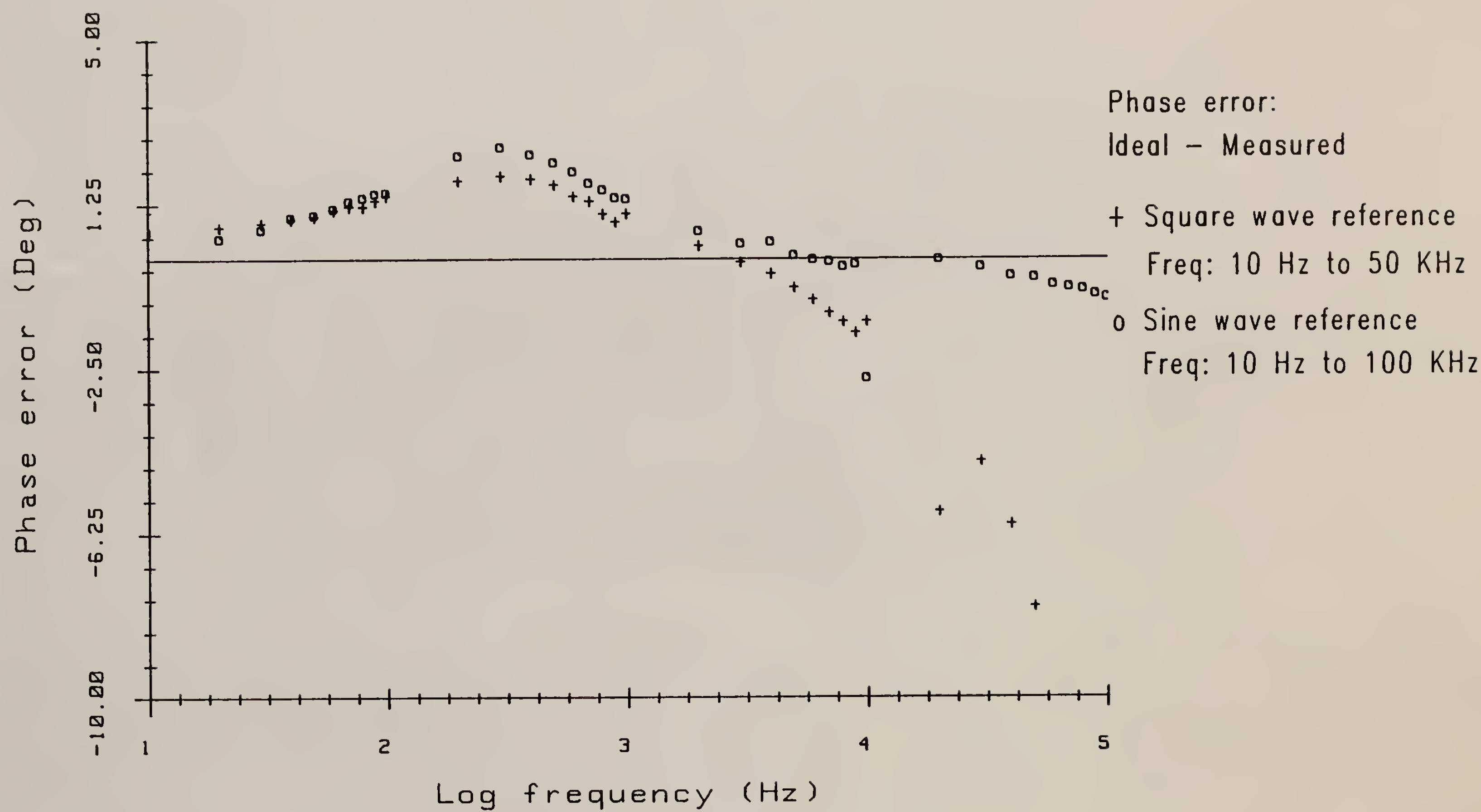

Fig 20. Error in phase measurements of RC highpass filter

3.4.1 In-phase signal and reference:

The applied signal and the derived reference have to be strictly in phase for accurate phase measurements. Trigger errors may occur when the reference is derived from the signal. In our case, the two channels are independent and according to the manufacturer's specifications, the phase-shift between the two channels is within $\pm$ 0.1 degree for sinewave signals. The phase-shifts between the two channels for different signals were tested using a dedicated Gain-Phase meter HP3575A. The results of the test are tabulated in Tables 1 & 2.

The phase-shifts for equal amplitude sinewaves was found to be well within the specifications. For unequal amplitudes, the phase-shift increases as the frequency increases. The phase-shifts for sine and square waves were found to be large at very low and very high frequencies. The maximum phase-shift was as high as 6 degrees.

The phase-shift between the signal and the reference channel on the synthesizer was added to the phase-shift determined earlier for the highpass and lowpass filters. The resultant errors in the phase responses for both the filters are shown in figs 21 & 22.

3.4.2 Errors due to oscillator distortion:

The response of the phase sensitive detector to a sinusoidal input is cosine in nature. If the input signal is not strictly sinusoidal, the output may not follow the

Table 1

Phase shift between channels on the HP8904A for sinusoidal signals

Frequency (Hz)	Phase shift (Deg) Ch A: 1 V_{pk} Ch B: 1 V_{pk}	Ch A: 5 V_{pk} Ch B: 1 V_{pk}
10	0.0	0.0
20	0.0	0.0
30	0.0	0.0
40	0.0	0.0
50	0.0	0.0
60	0.0	0.0
70	0.0	0.0
80	0.0	0.0
90	0.0	0.0
100	0.0	0.0
200	0.0	-0.1
300	0.0	-0.1
400	0.0	-0.1
500	0.0	-0.1
600	0.0	-0.1
700	0.0	-0.1
800	0.0	-0.1
900	0.0	-0.1
1000	0.0	-0.1
2000	0.0	-0.2
3000	0.0	-0.2
4000	0.0	-0.2
5000	0.0	-0.2
6000	0.0	-0.2
7000	0.0	-0.3
8000	0.0	-0.3
9000	0.0	-0.3
10000	0.0	-0.3
20000	0.0	-0.4
30000	0.0	-0.5
40000	0.0	-0.6
50000	0.1	-0.7
60000	0.1	-0.7
70000	0.1	-0.8
80000	0.1	-0.8
90000	0.1	-0.8
100000	0.2	-0.9

Note: Phase shifts are measured with respect to channel A

Table 2

Phase shift between channels on the HP8904A for sine and square waves

Frequency (Hz)	Phase shift (Degrees)			
	Ch A: Sine wave Ch B: Square wave		Ch A: Square wave Ch B: Sine wave	
	A: 1 V_{pk} B: 1 V_{pk}	A: 5 V_{pk} B: 1 V_{pk}	A: 1 V_{pk} B: 1 V_{pk}	A: 1 V_{pk} B: 5 V_{pk}
10	-0.8	-0.9	-0.9	-0.9
20	-0.4	-0.4	-0.4	-0.4
30	-0.2	-0.3	-0.3	-0.3
40	-0.1	-0.2	-0.1	-0.2
50	-0.1	-0.1	-0.1	-0.1
60	0.0	-0.1	-0.1	-0.1
70	0.0	-0.1	0.0	-0.1
80	0.0	-0.1	0.0	-0.1
90	0.0	0.0	0.0	0.0
100	0.0	0.0	0.0	0.0
200	0.1	0.0	0.1	0.0
300	0.1	0.0	0.1	0.0
400	0.1	0.0	0.1	0.0
500	0.1	0.0	0.1	0.0
600	0.1	0.0	0.1	0.0
700	0.1	0.0	0.1	0.0
800	0.1	0.0	0.1	0.0
900	0.1	0.0	0.1	0.0
1000	0.1	0.0	0.1	0.0
2000	0.1	-0.1	0.0	-0.1
3000	-0.1	-0.3	-0.1	-0.3
4000	-0.2	-0.5	-0.3	-0.5
5000	-0.4	-0.6	-0.4	-0.6
6000	-0.5	-0.8	-0.6	-0.8
7000	-0.7	-0.9	-0.7	-0.9
8000	-0.8	-1.1	-0.8	-1.2
9000	-0.9	-1.2	-1.0	-1.3
10000	-1.1	-1.4	-1.2	-1.5
20000	-2.5	-2.9	-2.6	-3.0
30000	-3.6	-4.0	-3.8	-4.3
40000	-4.5	-5.0	-4.9	-5.5
50000	-5.6	-6.0	-5.9	-6.6

Note: Phase shifts are measured with respect to sine wave

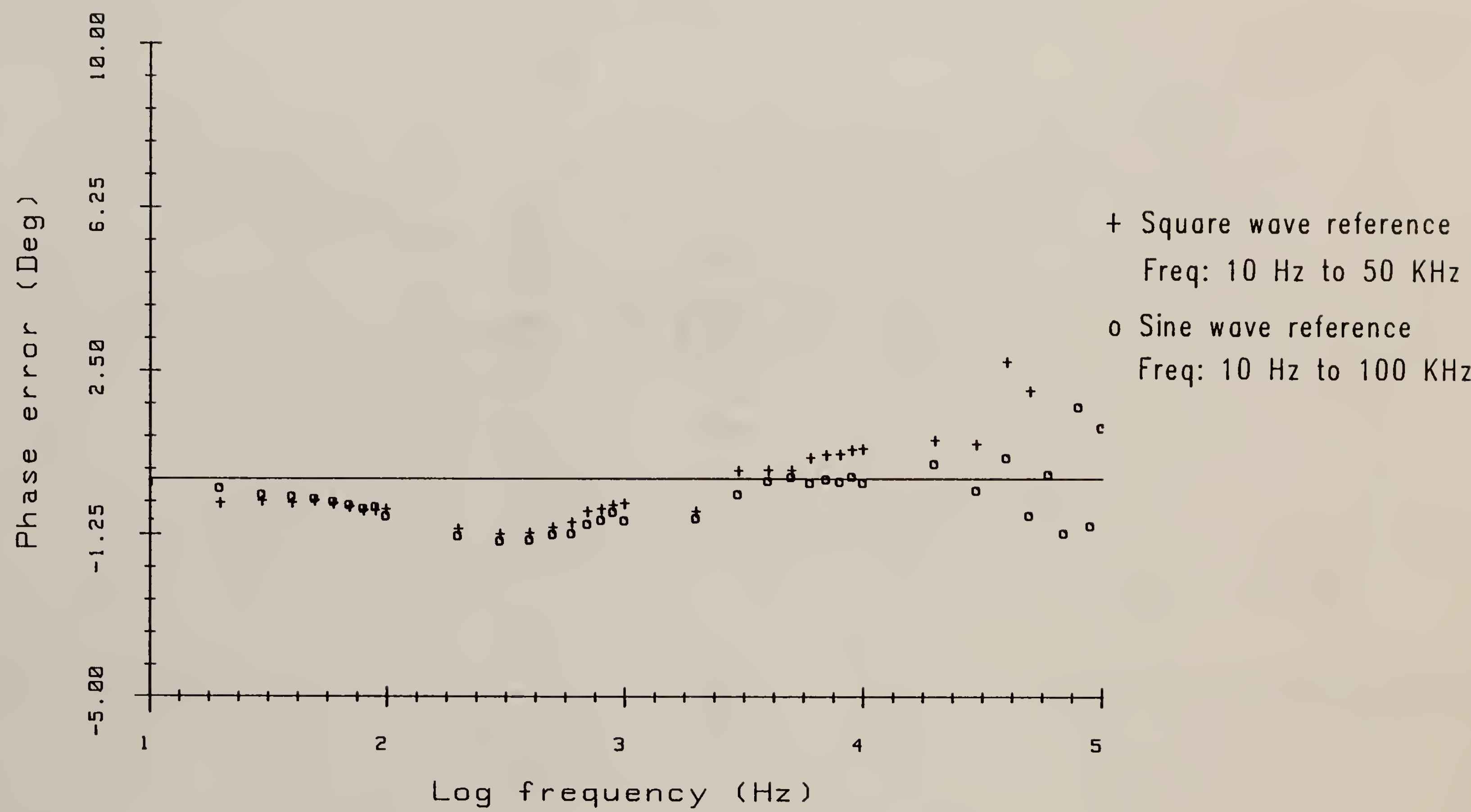

Fig 21. Phase error of RC lowpass filter with synthesizer phase correction

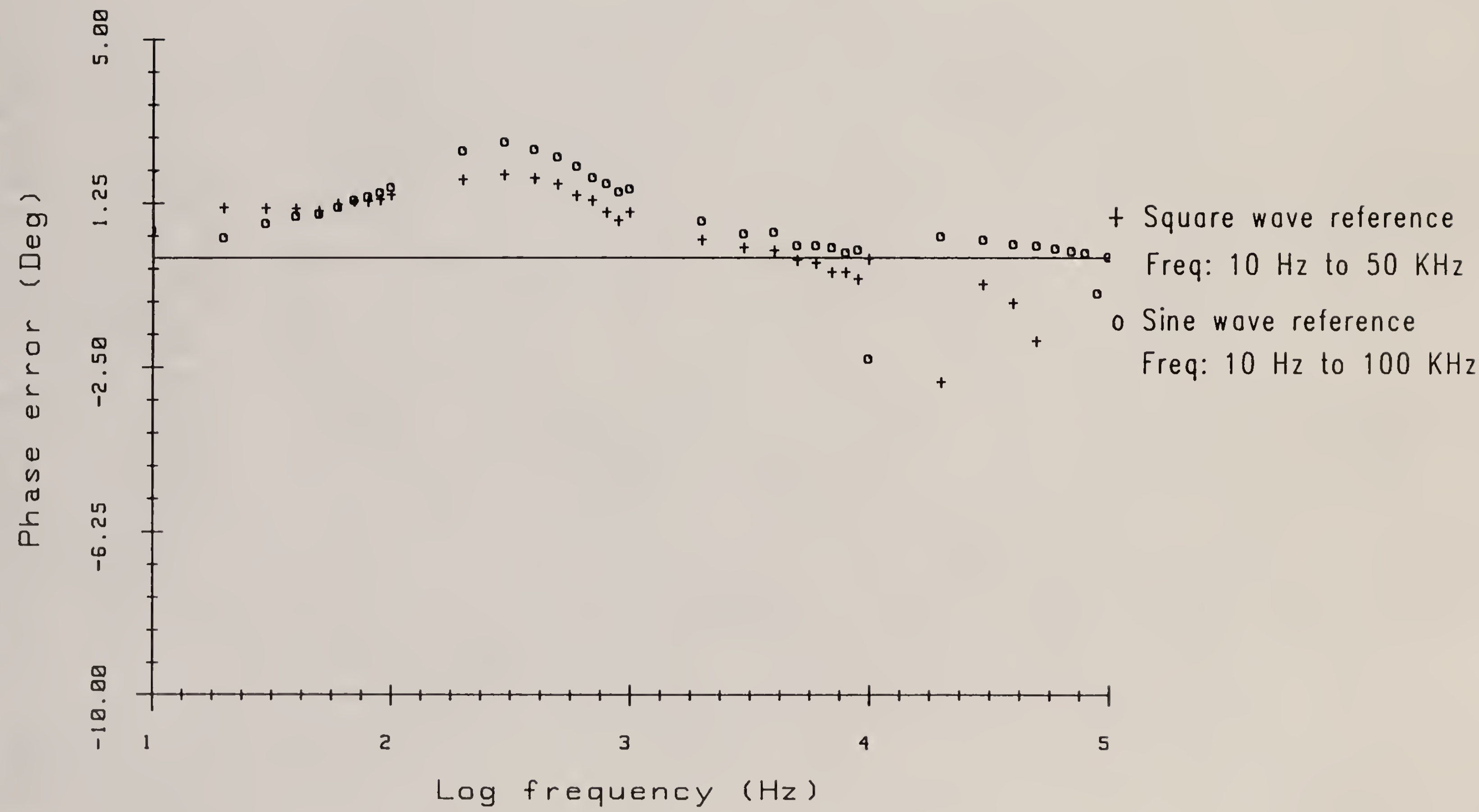

Fig 22. Phase error of RC highpass filter with synthesizer phase correction

Cosθ law. It is necessary for precision phase measurements, the distortion of the input signal be very small.

Oscillator distortion results in harmonics. The effect of harmonic components is to shift the zero crossings of the signal relative to its fundamental component. As a result, there exists an arbitrary phase relationship between the input and the reference signal causing errors in measurements.

3.4.3 Offset and drift in phase sensitive detector:

The offset voltage is a main source of error in phase measurements based on the zero crossing of the signals. The effect of offset was shown in section 3.2.

The output drift in the phase sensitive detector and phase drift in the reference channel can cause errors when the reference phase is being adjusted for a null output.

3.4.4 Component sensitivity:

The measurements carried out have to be compared if possible with the ideal results in order to determine the phase error. The ideal results are calculated based on the transfer function of the test circuit. The circuit components that are actually used have to be measured precisely in order to determine the errors. The resistor and capacitor for the highpass and lowpass filters were supposed to be 22 K and 0.022 uF. The measured values were 22.3K and 0.017 uF respectively. This corresponds to a 22%

decrease in the RC product. The phase errors for different percentage changes in RC product for both highpass and lowpass filters is shown in figs 23 & 24. It is apparent that the phase errors are dependent on frequency, circuit components and function of the circuit.

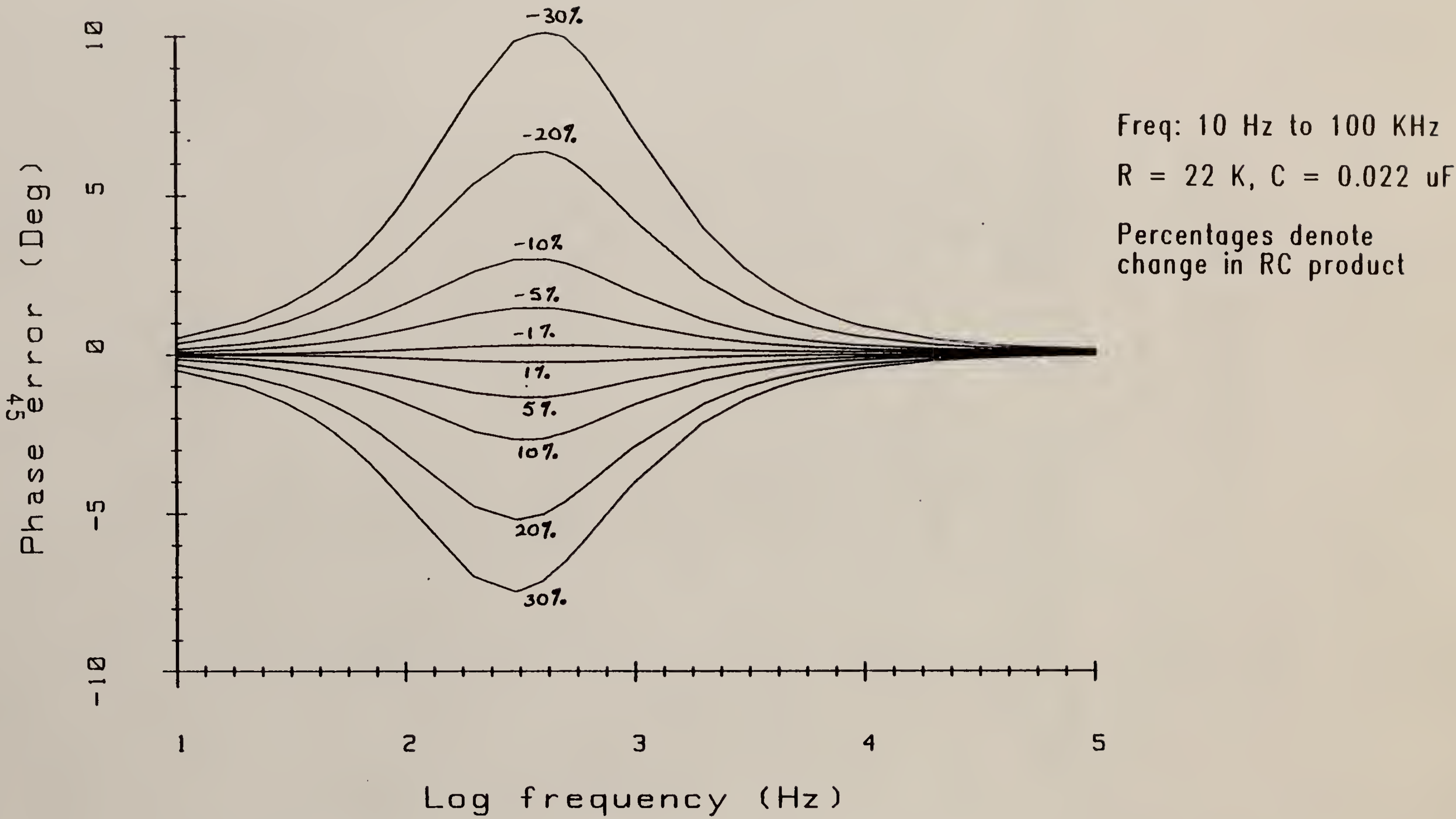

Fig 23. Phase error of RC lowpass filter as a function of frequency with component sensitivity as a parameter

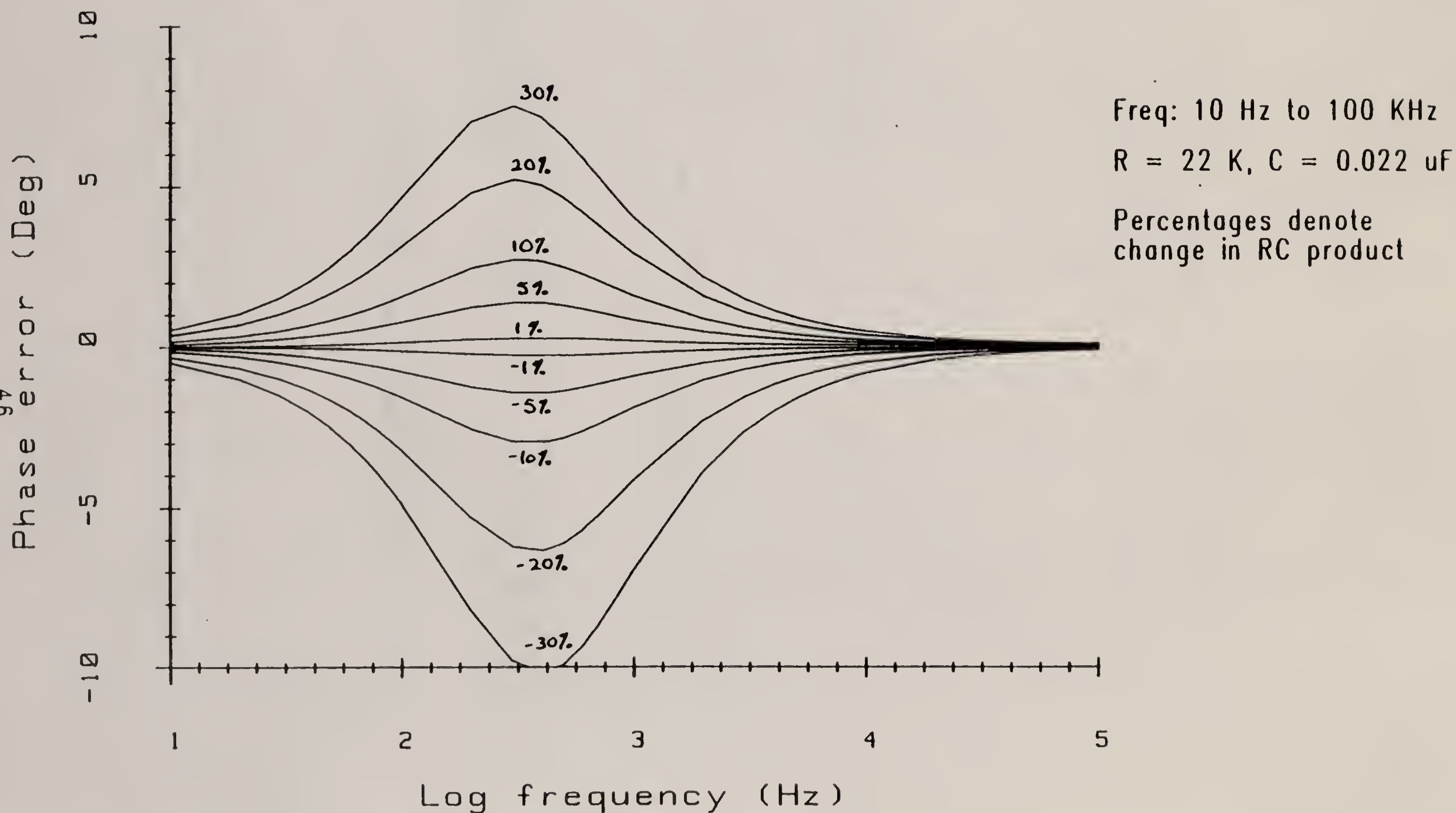

Fig 24. Phase error of RC highpass filter as a function of frequency with component sensitivity as a parameter

4. SOFTWARE SIMULATION OF GAIN PHASE MEASUREMENTS

The determination of amplitude and phase responses of linear systems is based on the simultaneous sampling of input and output signals of the system. Since the input and output signals are sinusoidal, sine wave curve fitting routines can be used to obtain the gain and phase-shift.

The sine wave curve fitting routines provide estimates of the amplitude, frequency, offset and phase to a digital data record. Several curve fitting routines, both closed form and iterative, are available in the literature. They include the closed form algorithm [2], the three-parameter (known frequency) least squares fit algorithm and the four-parameter least squares fit algorithm [3]. In our application, since the input frequency is known, the three parameter least squares fit algorithm would be ideal.

An alternative approach would be to test the correlation between the input and output signals of the system. The basis of transfer function measurements using lock-in systems was to test the correlation of the signals at the input of the multiplier.

A brief dicussion of the above algorithms is provided in the next section followed by the implementation of the algorithms on two test circuits.

4.1 Three parameter least squares fit algorithm:

The three parameter algorithm estimates the amplitude, offset and phase of the digital data record for known frequency. The algorithm provides a closed form solution and the accuracy of the estimations will be poor if the actual frequency of the signal sampled differs from the frequency used in the algorithm.

Algorithm:

A detailed description of the algorithm along with its derivation may be found in [3]. The input to the algorithm consists of a data record of N samples y_n taken at times t_n, where n = 0,1,...N-1.

The algorithm provides the solution in the form

$$y_n' = A \, Cos(wt_n) + B \, Sin(wt_n) + C$$

where

w --> known angular input frequency

t --> sample times

However, we are interested in obtaining a solution of the form

$$y_n' = R \, Sin(wt_n + 0) + C$$

The amplitude R and the phase 0 of the sinewave can be obtained by using the following relations

$$R = (A^2 + B^2)^{1/2}$$

$$0 = \tan^{-1}(A/B)$$

The offset present in the data record is estimated by C.

The amplitude and offset are estimated properly. The input signal phase range is from 0 to 360 degrees, while the estimated phase does not exceed $\pm$ 90 degrees. A suitable phase correction scheme has to be applied to the algorithm to obtain the proper phase shift. In applications such as transfer function measurements, where we need to estimate the phase-shift between two signals, it is necessary that the phase be defined from a common reference. It is convenient to determine the phase using the positive B axis as the reference.

The phase correction scheme depends on the signs of A and B. The actual phase 0 may be obtained as follows:

$\phi = \theta$ --> 1st quadrant

$\phi = \pi + \theta$ --> 2nd quadrant

$\phi = \pi + \theta$ --> 3rd quadrant

$\phi = 2\pi + \theta$ --> 4th quadrant

The above scheme is illustrated in fig 25.

The gain and phase measurements for a particular frequency using the three parameter least squares fit method is implemented as follows:

1. Inputs: input samples[y_{in}], output samples[y_{out}], number of samples[num], sample times[x_n] and frequency[freq].
2. A least squares fit of the input samples provides estimates of the input amplitude, offset and phase.

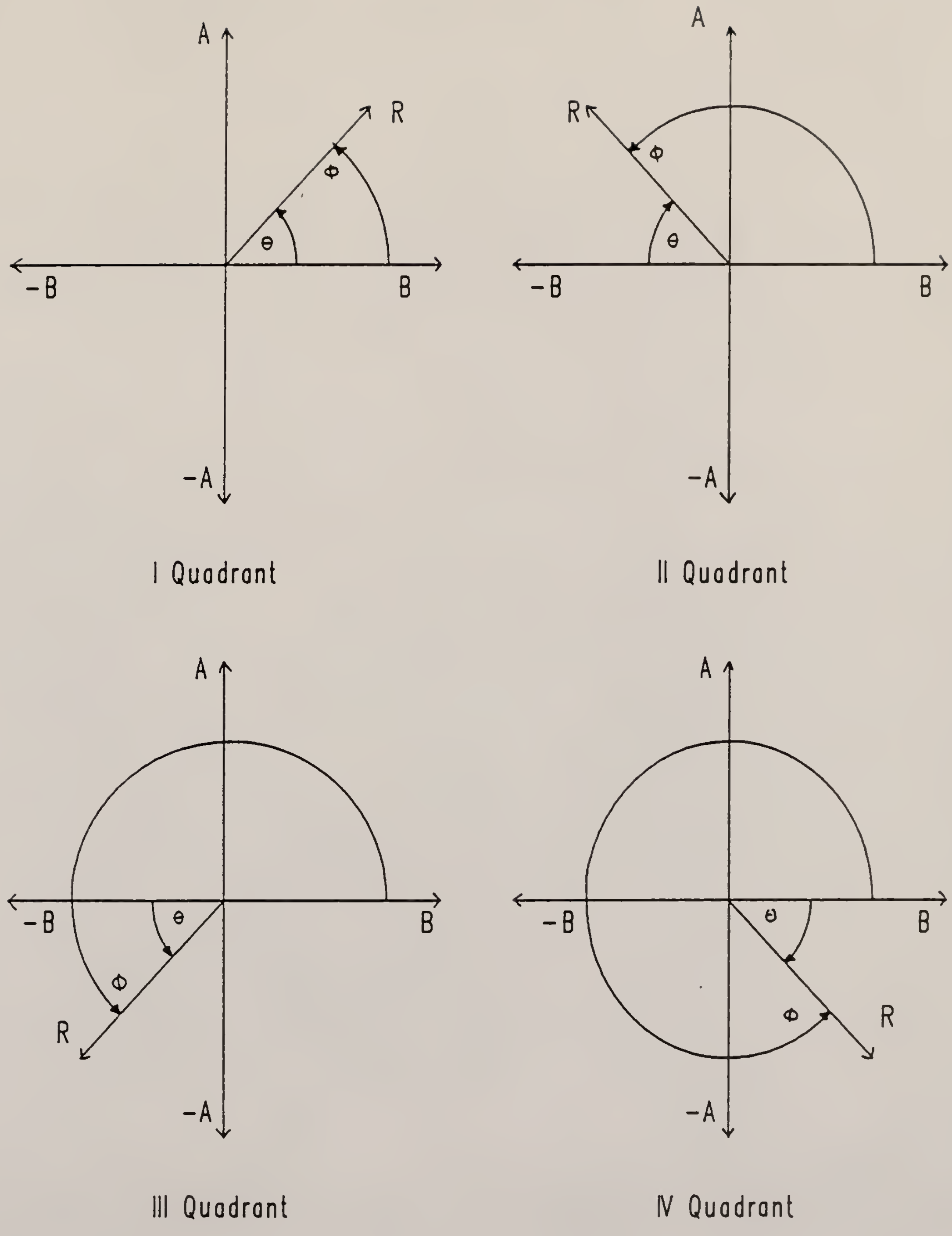

Fig 25. Phase correction scheme used in three parameter least squares fit algorithm

3. A least squares fit of the output samples provides estimates of the output amplitude, offset and phase.

4. $$\text{Gain(dB)} = 20 \log \frac{\text{output amplitude}}{\text{input amplitude}}$$

5. Phase-shift = output phase - input phase

4.2 Cross correlation using Fast Fourier Transforms:

Correlation tests the similarity of two data sets. The correlation between input and output samples is tested by comparing them both directly superposed. The correlation will be large at some value of time t if the output signal lags the input signal. Likewise, the correlation will be large for some negative value of t if the output leads the input.

The correlation between the input and output samples is computed as follows:

1. Obtain the Discrete Fourier Transform (DFT) of the two data sets.
2. Multiply one resulting transform by the complex conjugate of the other.
3. Obtain the Inverse Discrete Fourier Transform (IDFT) of the product.

The result (vector r_k) will formally be a complex vector of the sample length N. The correlation at zero lag is in r_o, the first component; the correlation at lag 1 is in r_1, the second component; the correlation at lag -1 is in r_{N-1}, the last component etc. The gain can be calculated by obtaining the maximum amplitude of input and output samples for a particular frequency and using the relation

$$\text{Gain(dB)} = 20 \log \frac{\text{max. output amplitude}}{\text{max. input amplitude}}$$

Discrete Fourier Transform:[4][5]

The DFT of a function h_k from a finite number of sampled points N is given by

$$H_n = \sum_{k=0}^{N-1} h_k \, e^{2\pi ikn/N} \qquad n = -N/2 \, .. \, N/2$$

The IDFT is given by

$$h_k = \frac{1}{N} \sum_{n=0}^{N-1} H_n \, e^{-2\pi ikn/N}$$

The only differences in the above two equations are

(i) Changing the sign in the exponential

(ii) Dividing the result by N

Therefore, a single routine for calculating DFT can also, with slight modifications, calculate the IDFT.

The computation of DFT using the above equation results in redundant calculations [4]. This redundancy can be overcome by computing the DFT using an algorithm called the Fast Fourier Transforms (FFT).

The computation of the DFT using the FFT is based on the Danielson Lanczos Lemma which states that a DFT of length N can be obtained from the two discrete fourier transforms, each of length N/2. One of the two is formed from the even numbered points of the original N, the other from the odd numbered points. The Danielson Lanczos Lemma can be used recursively until we have subdivided the data all the way down to transforms of length 1. The fourier transform of length one is just the identity operation that copies its one input number into its output slot. The output slot for a particular decomposition is the location obtained by bit reversal of the original sample location. The DFT can be computed quickly but the number of samples has to be an integer power of 2.

The above algorithms have been implemented in a 'C' VAX/VMS environment and the source code listings are provided in Appendix D.

4.3 Frequency response of test circuits:

The RC lowpass and highpass filters (R = 22.3K, C = 0.017uF) used in section 3.3 were used again as the test circuits. The input to the test circuits was a sinewave of amplitude 5 V_{pk}. The input and output samples over the frequency range 10 Hz to 100 KHz for both the circuits were obtained using the program provided in Appendix C. Since the number of samples for correlation has to be an integer power of 2, the samples were obtained accordingly. The maximum sampling frequency on the Keithley 194A placed restrictions on the number of samples per cycle. Table 3 gives the number of points sampled per cycle for different frequencies.

Table 3

Number of points sampled for different frequencies

Frequency (Hz)	Samples per cycle
10	512
.	.
.	.
1K	512
2K	256
3K	256
4K	128
5K	128
6K	128
7K	128
8K	64
9K	64
10K	64
20K	32
30K	32
40K	16
50K	16
60K	16
70K	8
80K	8
90K	8
100K	8

The phase accuracy when using the cross correlation method is dependent on the number of points sampled per cycle. Table 4 shows the phase resolution (in degrees) as a function of the number of points per cycle.

Table 4

Phase resolution as a function of points sampled per cycle in cross correlation method

Points per cycle	Phase resolution (degrees)
1024	± 0.3516
512	± 0.7031
256	± 1.4063
128	± 2.8125
64	± 5.625
32	± 11.25
16	± 22.5
8	± 45.0

The amplitude and phase responses of the RC lowpass filter obtained using both the software techniques implemented are shown in figs 26 & 27 respectively. The amplitude response of the filter using least squares fit method provides better results than the correlation method. This is because the least squares fit provides an amplitude estimate for the data record whereas the cross correlation method uses the maximum amplitude in the data record.

The phase error in the least squares method is generally less than the cross correlation method. The phase error in correlation increases when the sample size decreases (at high frequencies). However, the phase error at 10 KHz is extremely large (around 22 degrees). This error is of the same magnitude using both the methods implying sampling error on part of the Keithley 194A Voltmeter.

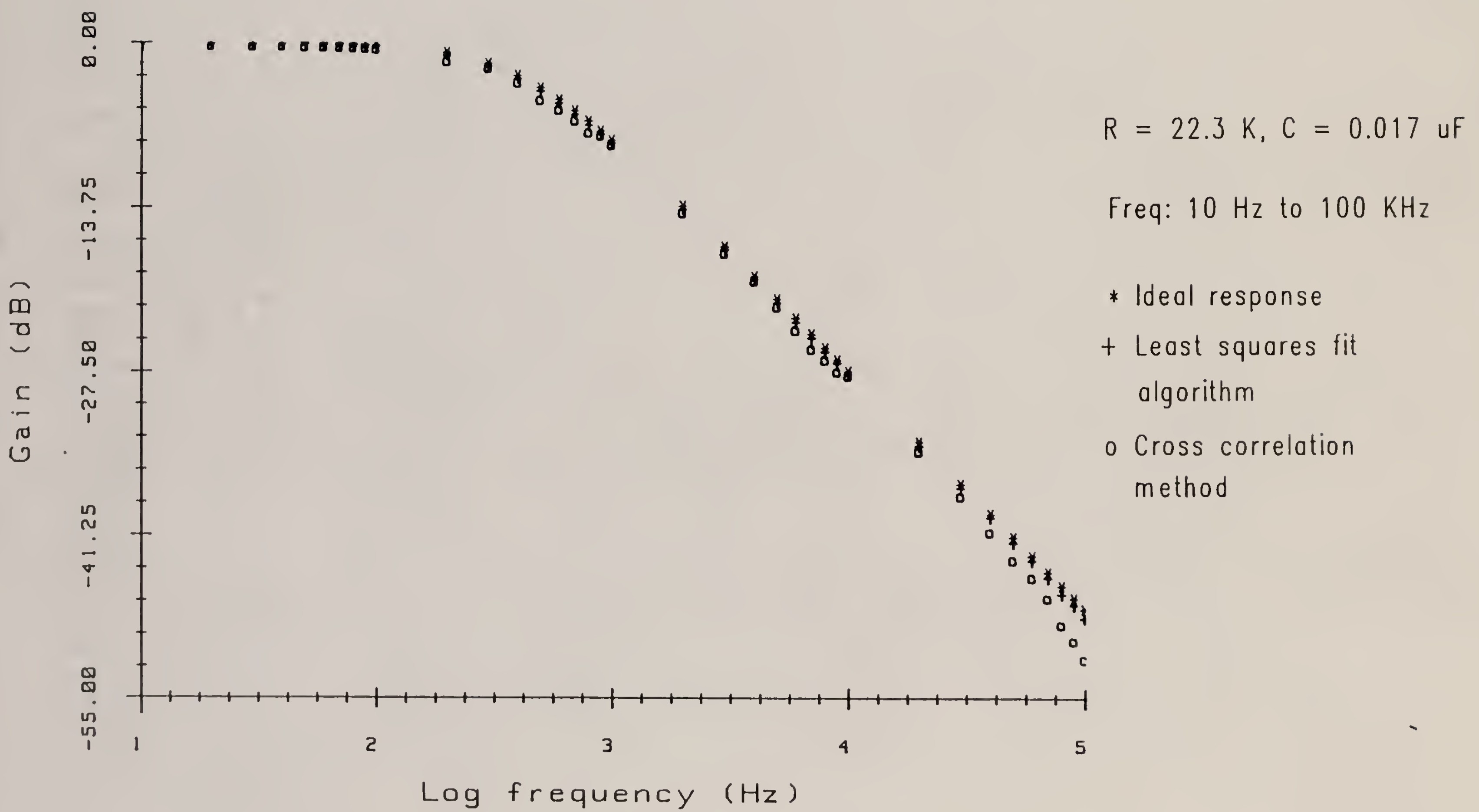

Fig 26. Amplitude response of RC lowpass filter using software methods

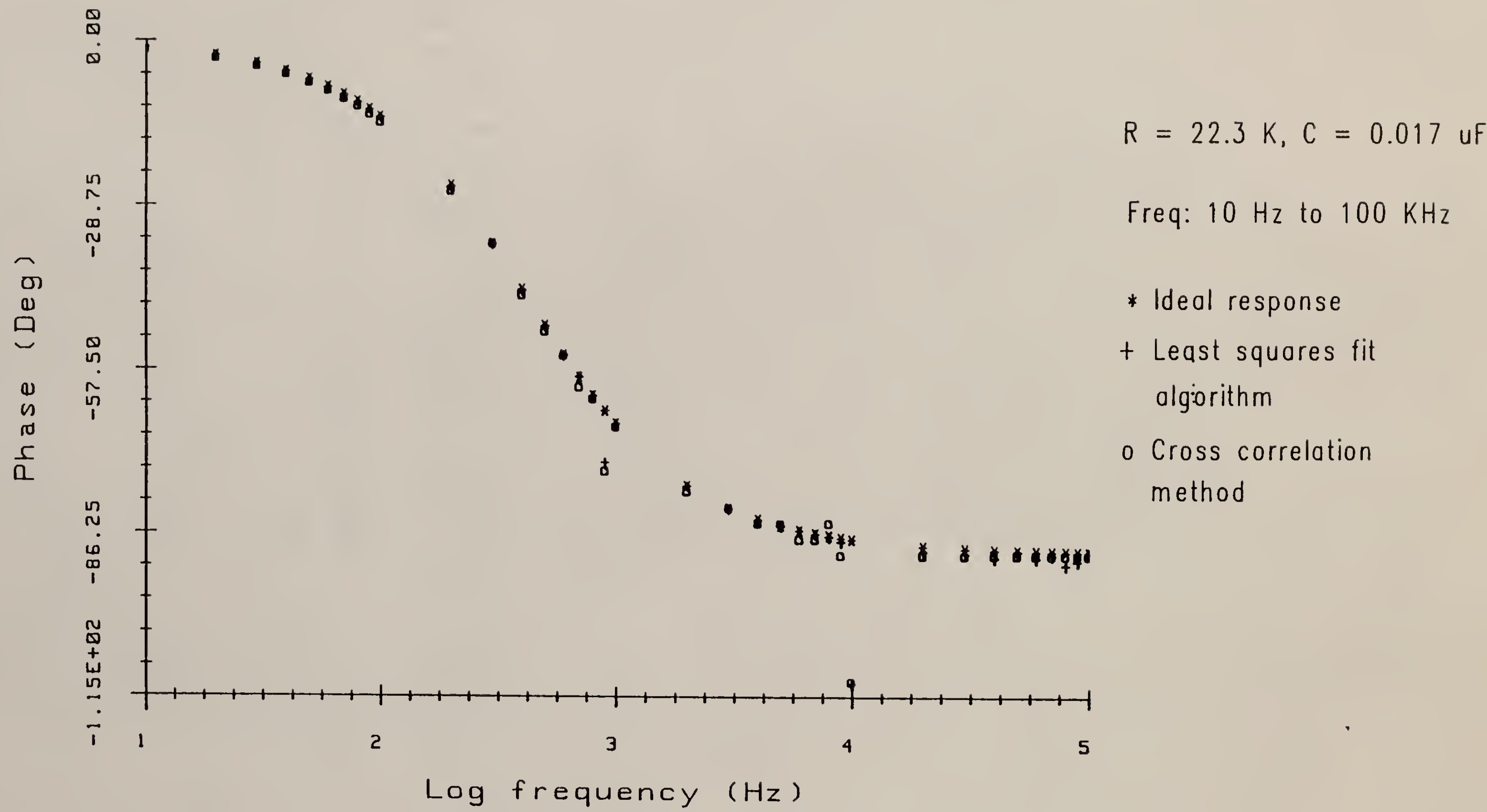

Fig 27. Phase response of RC lowpass filter using software methods

The amplitude and phase responses of the RC highpass filter are shown in figs 28 & 29 respectively. The observations are similar to that made for the lowpass filter. Extremely large phase errors are observed at 30 & 400 Hz in both the methods due to sampling error.

The error in phase measurements for lowpass and highpass filters are shown in figs 30 & 31 respectively.

The sampling resolution of the Keithley 194A Voltmeter and the constraint on number of samples per cycle results in errors since a complete cycle of the signal is not sampled.

The correlation method was tested for the two circuits with simulated samples of 512 samples per cycle over the frequency range 1 KHz to 100 KHz. The error in phase measurements are tabulated in Table 5. As expected the errors are well within the maximum error (resolution/2).

The least squares fit method was tested for the two circuits with simulated samples of 8 samples per cycle over the frequency range 10 Hz to 100 KHz. The phase error for both the circuits were less than 0.1 degree.

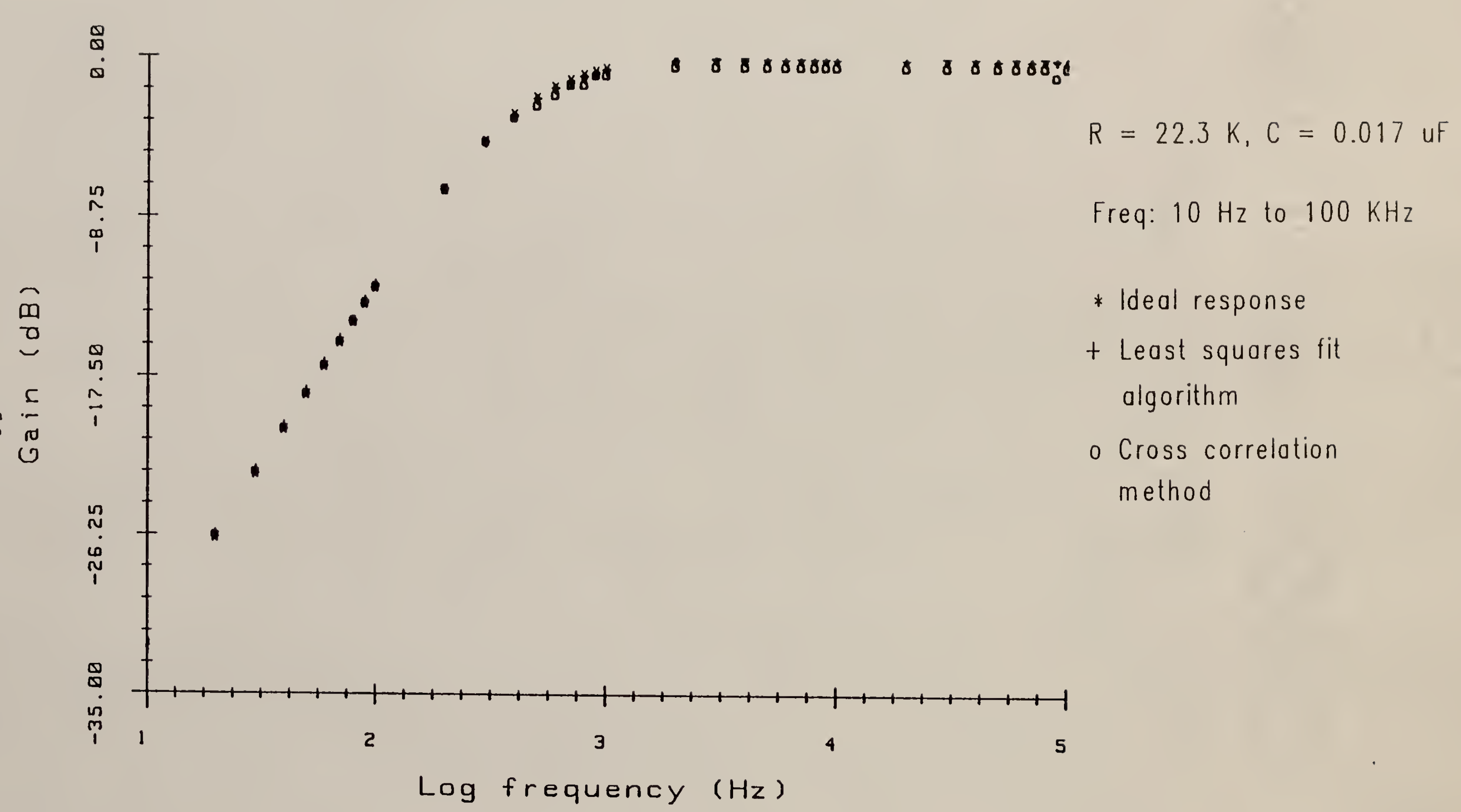

Fig 28. Amplitude response of RC highpass filter using software methods

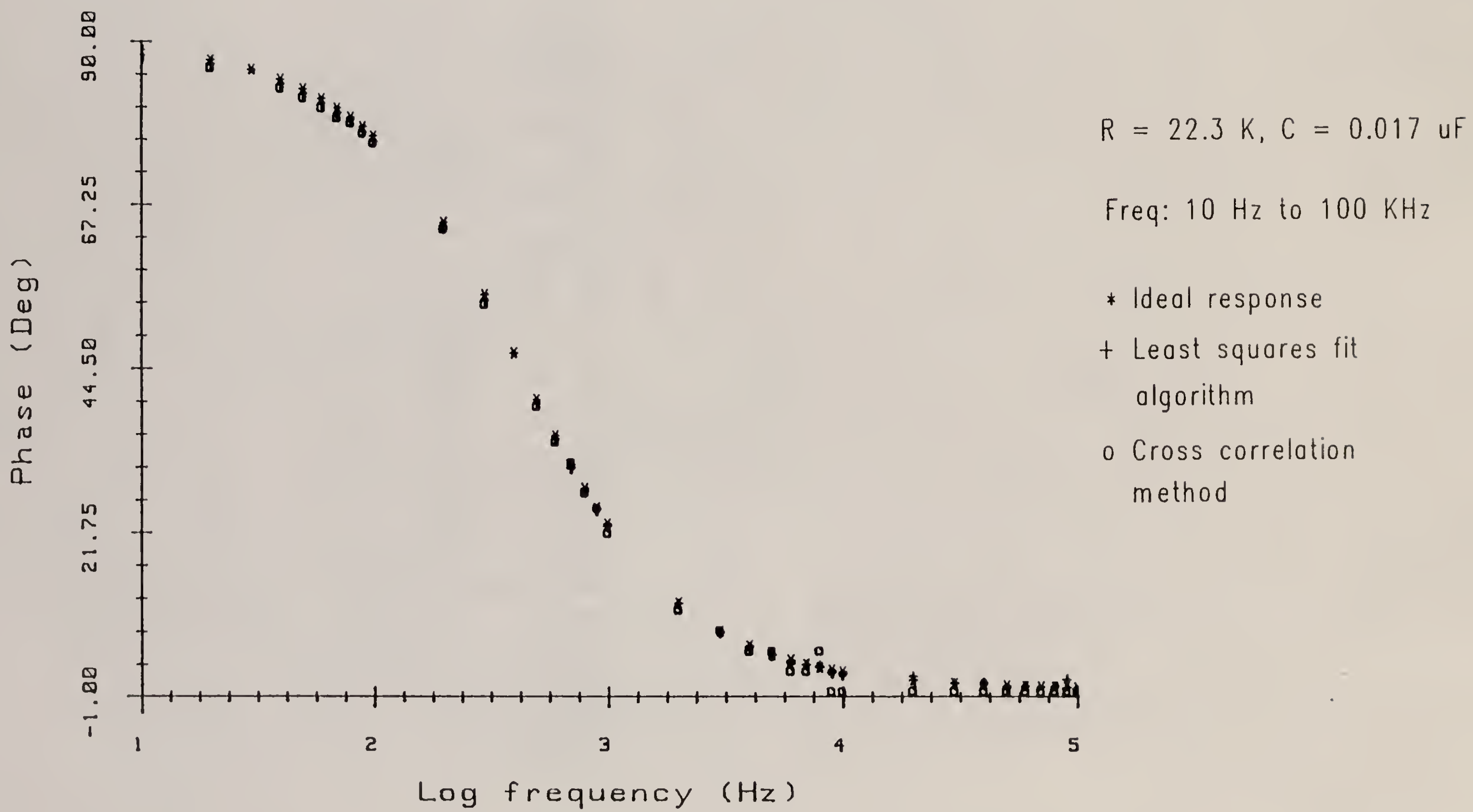

Fig 29. Phase response of RC highpass filter using software methods

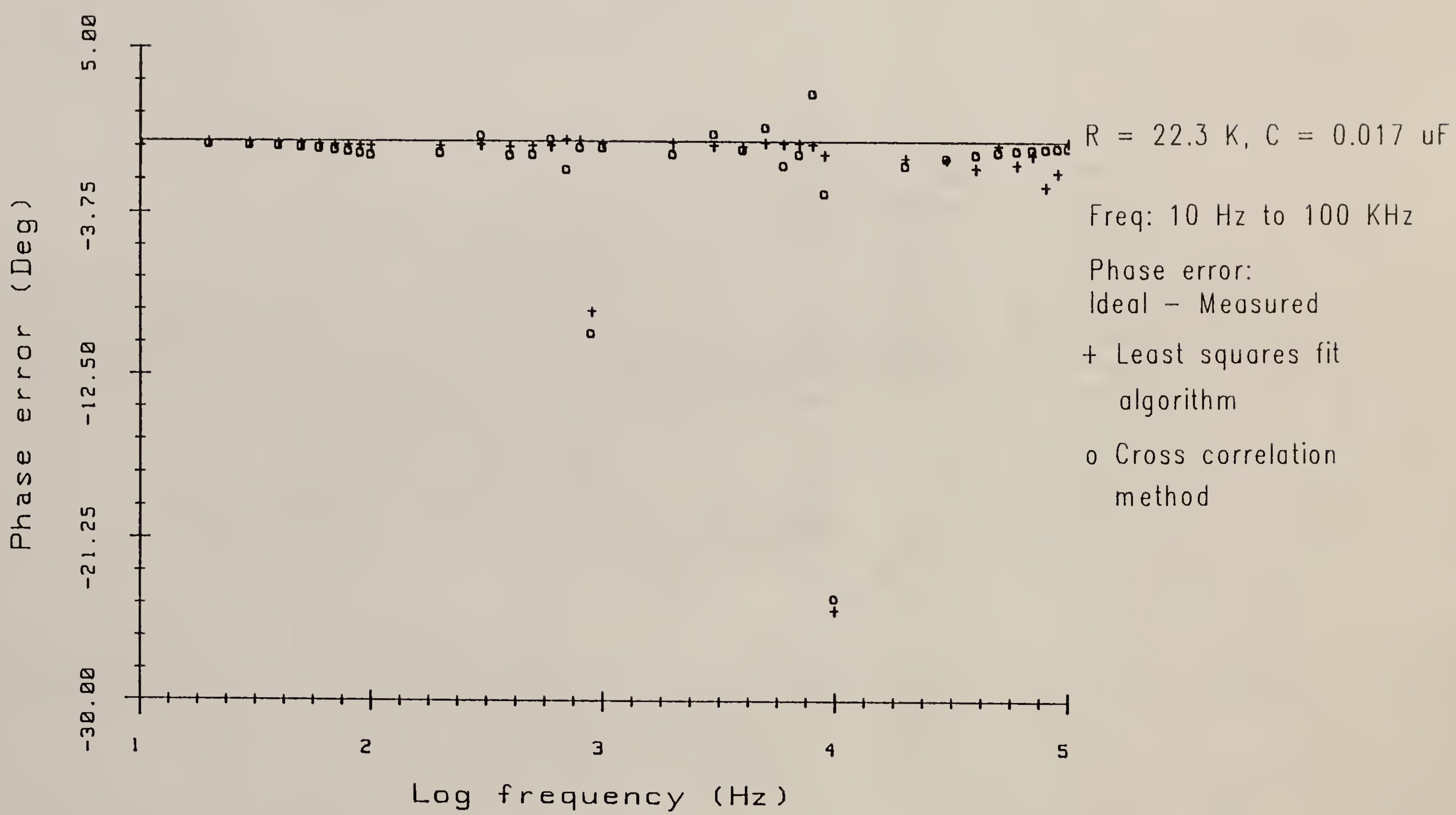

Fig 30. Error in phase measurements of RC lowpass filter using software methods

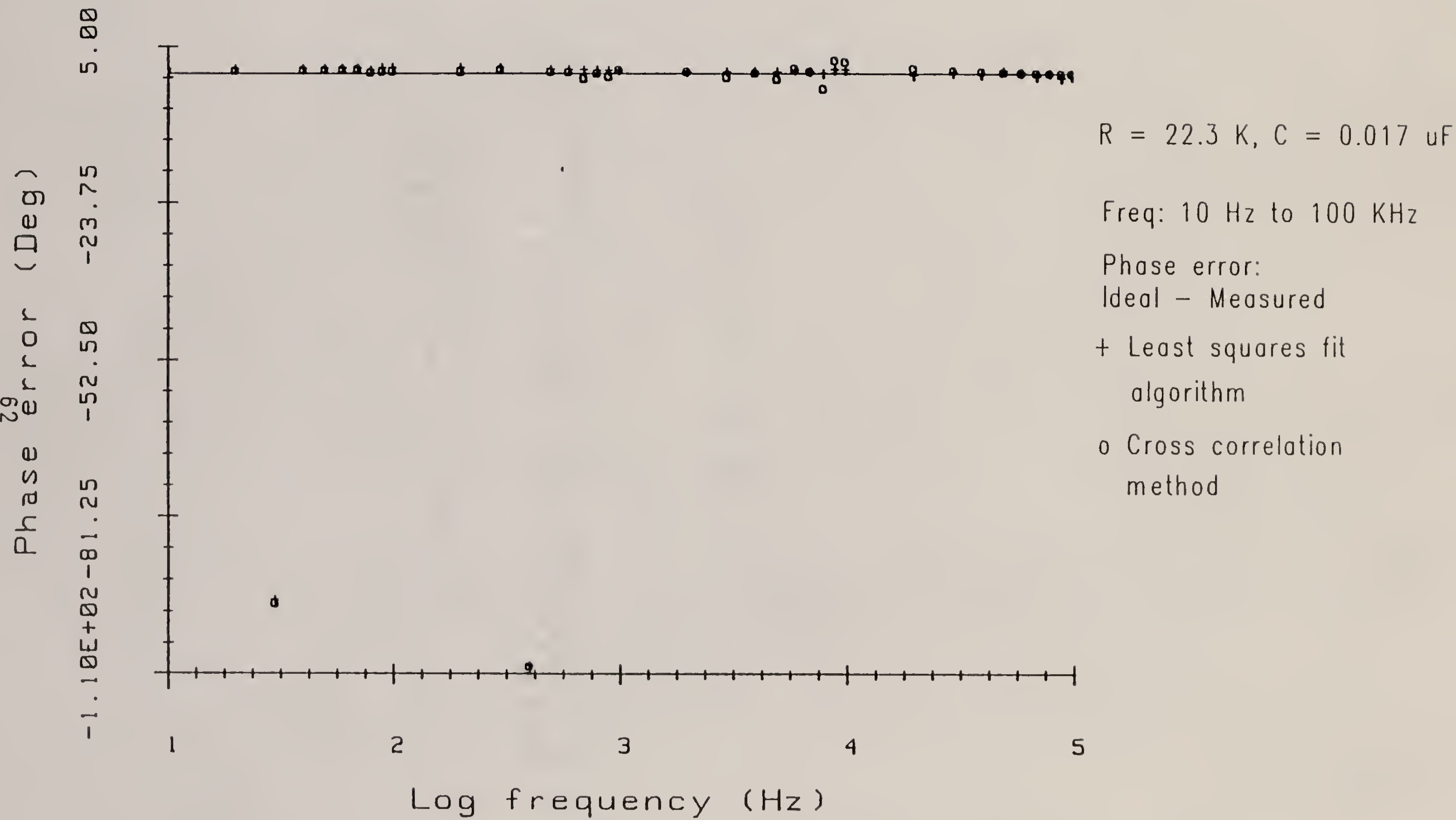

Fig 31. Error in phase measurements of RC highpass filter using software methods

Table 5

Phase error obtained for lowpass and highpass filters with simulated samples using cross correlation method

Frequency (KHz)	Phase error (degrees) RC lowpass filter	RC highpass filter
1	-0.27379	0.27379
2	0.098199	-0.098199
3	-0.231922	0.231922
4	0.336538	-0.336538
5	0.122314	-0.122314
6	0.216258	-0.216258
7	0.083437	-0.083437
8	-0.191506	0.191506
9	0.141759	-0.141759
10	-0.294623	0.294623
20	0.203722	-0.203722
30	-0.098626	0.098626
40	0.101795	-0.101795
50	0.222054	-0.222054
60	0.30223	-0.30223
70	-0.343626	0.343626
80	-0.300673	0.300673
90	-0.267266	0.267266
100	-0.24054	0.24054

Phase error = Ideal value - Measured value
Phase resolution for 512 samples per cycle is ±0.7031 deg.

5. CONCLUSIONS

The purpose of this work was to compare the performance of transfer function measurement schemes in hardware and software.

The hardware approach using computer controlled lock-in systems provides phase accuracy dependent on the resolution of the phase shifters available. Using the precision IC multiplier AD534 in the phase sensitive detector enables us to use either a square or sine wave reference. The response is generally better to a sine wave reference. It was found that the phase errors were primarily due to the phase shift between the signal and reference channels on the synthesizer. The method provides scope for varying signal and reference amplitudes, thereby increasing the sensitivity of the system. However, the user must be aware of the maximum output amplitude of the test circuit as this may exceed the range of input amplitudes on the multiplier. The amplitude response is a function of the phase response due to the nature of the measurement process.

The software techniques require simultaneous sampling of the input and output signals of the test circuit. The number of points sampled per cycle plays a critical role in phase estimation using the cross correlation method. There should be at least 180 samples per cycle for a phase accuracy of ± 1 degree. The cross correlation method using

Fast Fourier Transforms requires that the number of sample points be an integer power of 2 in computing the Discrete Fourier Transform. The computation of the DFT using straight forward method does not place any restriction on the sample size, but is slower. However, the phase resolution still depends on the number of points per cycle.

The least squares fit method is more robust and it provides phase accuracy to within ± 0.1 degrees with 8 samples per cycle. The only constraint on this method is that the frequency used in the algorithm be the same as that of the signal sampled.

The software techniques rely on the accurate simultaneous sampling of input and output signals. The sampling resolution of the Keithley 194A Voltmeter and the constraint on the number of points per cycle causes difficulties in sampling a complete cycle. An improvement to the sampling program would be to sample two cycles of the signal and then select samples of one cycle in software. Care should be taken to satisfy the constraint on number of points per cycle. Controlled triggering on the Keithley 194A may alleviate the problem.

REFERENCES

[1] Meade M.L, Lock-in amplifiers: principles and applications, (Published by: Peter Peregrinus Ltd., London UK, 1983)

[2] Y.C.Jenq and P.Crosby, "Sinewave parameter estimation algorithm with application to waveform digitizer effective bits measurement," IEEE Proceedings of the Instrumentation and Measurement Conference 1988, CH2569-2, pp 218-221, 1988

[3] Waveform measurements and Analysis Committee of the IEEE Instrumentation and Measurement Society, "Trial use standard for digitizing waveform recorders," Working Draft, P1057/D11, May 1988

[4] Samuel D. Stearns, Digital Signal Analysis, (Rochelle Park, NJ, Hayden Book Company, Inc., 1975)

[5] William H. Press, Brian P. Flannery, Saul A. Teukolsky and William T. Vetterling, Numerical Recipes in C: The Art of Scientific Computing, (New Rochelle, NY, Cambridge University Press, 1988)

[6] Siliconix Analog Switch & IC Product Data Book, Siliconix incorporated, January 1982, pp 3-79 - 3-80

[7] Data Acquisition Databook, Vol.1, Analog Devices, Inc., 1984, pp 6-27 - 6-35

[8] Keithley Model 194A High Speed Voltmeter Instruction Manual, Keithley Instruments, Inc., Cleveland, Ohio, 1985

[9] HP8904A Multifunction Synthesizer, Operation and Calibration Manual, Hewlett-Packard Company, Spokane, Washington, 1987

APPENDIX A

Mathematical Analysis of lock-in systems

The phase sensitive detector of the lock-in system can be modelled as an ideal multiplier followed by a lowpass filter. The excitation of the experiment is sinusoidal at frequency w_s and provides a reference which is to be used for detection of the signal at the output of the experimental system. This output signal is at the same frequency as the excitation, but suffers a phase-shift 0_s in the experiment. The reference is applied to the multiplier via a phase shifter. The mathematical relationship for sine and square wave references are derived below. The ideal multiplier model for a phase sensitive detector is shown in fig A1.

1. Square wave reference:

The signal and the reference voltages can be expressed as

$$s(t) = \sqrt{2}\, V_s \operatorname{Cos}(w_s t + \theta_s)$$

$$r(t) = \frac{4}{\pi} V_r \left[\operatorname{Cos}(w_r t + \theta_r) - \frac{1}{3} \operatorname{Cos} 3(w_r t + \theta_r) + \frac{1}{5} \operatorname{Cos} 5(w_r t + \theta_r) -+ ..\right]$$

where

V_s --> RMS value of the signal

V_r --> Peak value of the reference

The product $V_p(t)$ is given by

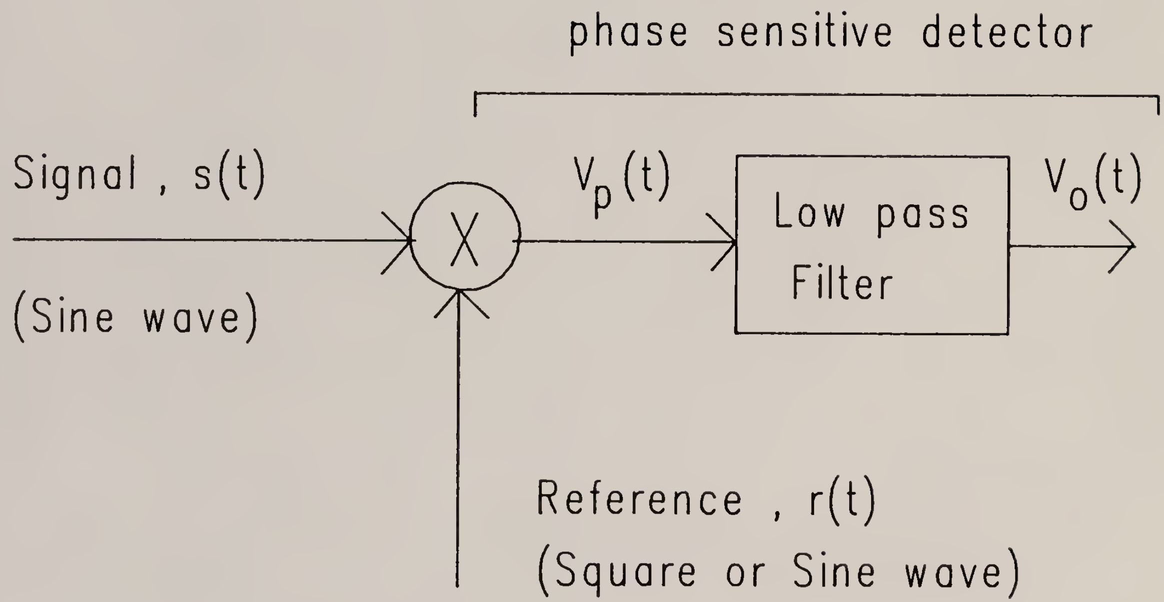

Fig A1. Ideal multiplier model for a phase sensitive detector

$$
\begin{aligned}
V_p(t) &= s(t) * r(t) \\
&= \frac{2\sqrt{2}}{\pi} V_s V_r \Big[\mathrm{Cos}(w_s t \pm w_r t + \theta_s \pm \theta_r) \\
&\qquad - \frac{1}{3} \mathrm{Cos}(w_s t \pm 3w_r t + \theta_s \pm 3\theta_r) \\
&\qquad + \frac{1}{5} \mathrm{Cos}(w_s t \pm 5w_r t + \theta_s \pm 5\theta_r) \\
&\qquad - + \, .. \Big]
\end{aligned}
$$

The signal and the reference are at the same frequency. The lowpass filter cuts off well below the reference frequency which eliminates the higher products of multiplication. The d.c component of the final output is given by

$$V_{avg} = \frac{2\sqrt{2}}{\pi} V_s V_r A_L(0) \mathrm{Cos}(\theta_s - \theta_r)$$

where $A_L(0)$ is the gain of the lowpass filter at dc. When the signal and the reference at the input of the multiplier are precisely in phase ($\theta_s = \theta_r$), the average output voltage is given by (assuming $A_L(0)$ and the peak reference voltage to be unity)

$$V_{avg} = \frac{2\sqrt{2}}{\pi} V_s$$

$$V_s = \frac{\pi}{2\sqrt{2}} V_{avg}$$

Therefore, the scale-factor to be used to obtain the RMS value of the output signal of the experimental system is $\frac{\pi}{2\sqrt{2}}$.

2. Sinewave reference:

The signal and the reference voltages can be expressed as

$$s(t) = \sqrt{2}\, V_s \, \mathrm{Cos}(w_s t + \theta_s)$$

$$r(t) = \sqrt{2}\, V_r \, \mathrm{Cos}(w_r t + \theta_r)$$

where V_s and V_r are the RMS values of the signals.

The product $V_p(t)$ is given by

$$V_p(t) = V_s\, V_r\, \mathrm{Cos}((w_s \pm w_r)t + \theta_s \pm \theta_r)$$

Since $w_s = w_r$ and neglecting higher products of multiplication, the final output is given by

$$V_{avg}(t) = V_s\, V_r\, A_L(0)\, \mathrm{Cos}(\theta_s - \theta_r)$$

when $\theta_s = \theta_r$,

$$V_{avg}(t) = V_s\, V_r \qquad \text{(assuming } A_L(0) \text{ to be unity)}$$

The signals on the HP8904A synthesizer are provided in peak or peak to peak values. Assuming $V_{r(pk)}$ to be unity,

$$V_s = \sqrt{2}\, V_{avg}$$

Therefore, the scale-factor to be used to obtain the RMS value of the output signal of the experimental system is $\sqrt{2}$.

APPENDIX B

HP8904A Multifunction Synthesizer Specifications [9]

Wave Form	Frequency Range	Frequency Resolution	Phase Range	Phase Resolution	Amplitude Range
	0 Hz to 600 kHz	0.1 Hz	359.9° or 6.282 rad	0.1° or 0.001 rad	0 to 10 V (p-p) (50Ω load)
	0 Hz to 50 kHz	0.1 Hz	359.9° or 6.282 rad	0.1° or [2] 0.001 rad	0 to 10 V (p-p) (50Ω load)
	0 Hz to 50 kHz	0.1 Hz	359.9° or 6.282 rad	0.1° or [2] 0.001 rad	0 to 10 V (p-p) (50Ω load)
	0 Hz to 50 kHz	0.1 Hz	359.9° or 6.282 rad	0.1° or [2] 0.001 rad	0 to 10 V (p-p) (50Ω load)
Gaussian Noise	(N/A)	(N/A)	(N/A)	(N/A)	0 to 10 V (p-p) (50Ω load)
	(N/A)	(N/A)	(N/A)	(N/A)	± 10 V (open circuit)

Table B1. HP8904A Multifunction Synthesizer Capabilities

Operating Considerations: Option 002

Destination Control

Option 002 does not allow you to change the Destination of the two channels. Channel A is always designated for Output 1; Channel B is always designated for Output 2.

Abbreviated Specifications: Option 002

Output 1 to Output 2 phase accuracy (sine waves of the same frequency):

+0.1 degree or 30 ns, 0.1 Hz to 100 KHz, whichever is greater.

APPENDIX C

Dual Channel operation of Keithley 194A [8]

The Keithley 194A is a high speed dual channel digital voltmeter. The channels can be programmed to obtain samples of the signal and store them in a buffer. Several mathematical functions such as Average, TRMS, Peak etc., are available. These functions are calculated based on the samples obtained. The sampling rate and the number of samples can be programmed.

The architecture of the Keithley 194A Dual Voltmeter is shown in fig C1. Each channel has its own A/D converter, which digitizes the analog input signal, and a 64K byte measurement buffer for storage of samples. The mathematical functions are calculated based on these samples and placed in a reading buffer. Each channel can process raw data independently. However, the two channels share the same microcomputer, reading buffer and the IEEE 488 bus. The voltmeter output always comes from the reading buffer whether or not that buffer has been enabled.

1. Reading buffer disabled: In this mode, the reading buffer has an effective size of one, with both channels sharing a same location. The location will be continously updated with data from either channel as the measurement from that channel is completed, and the reading is

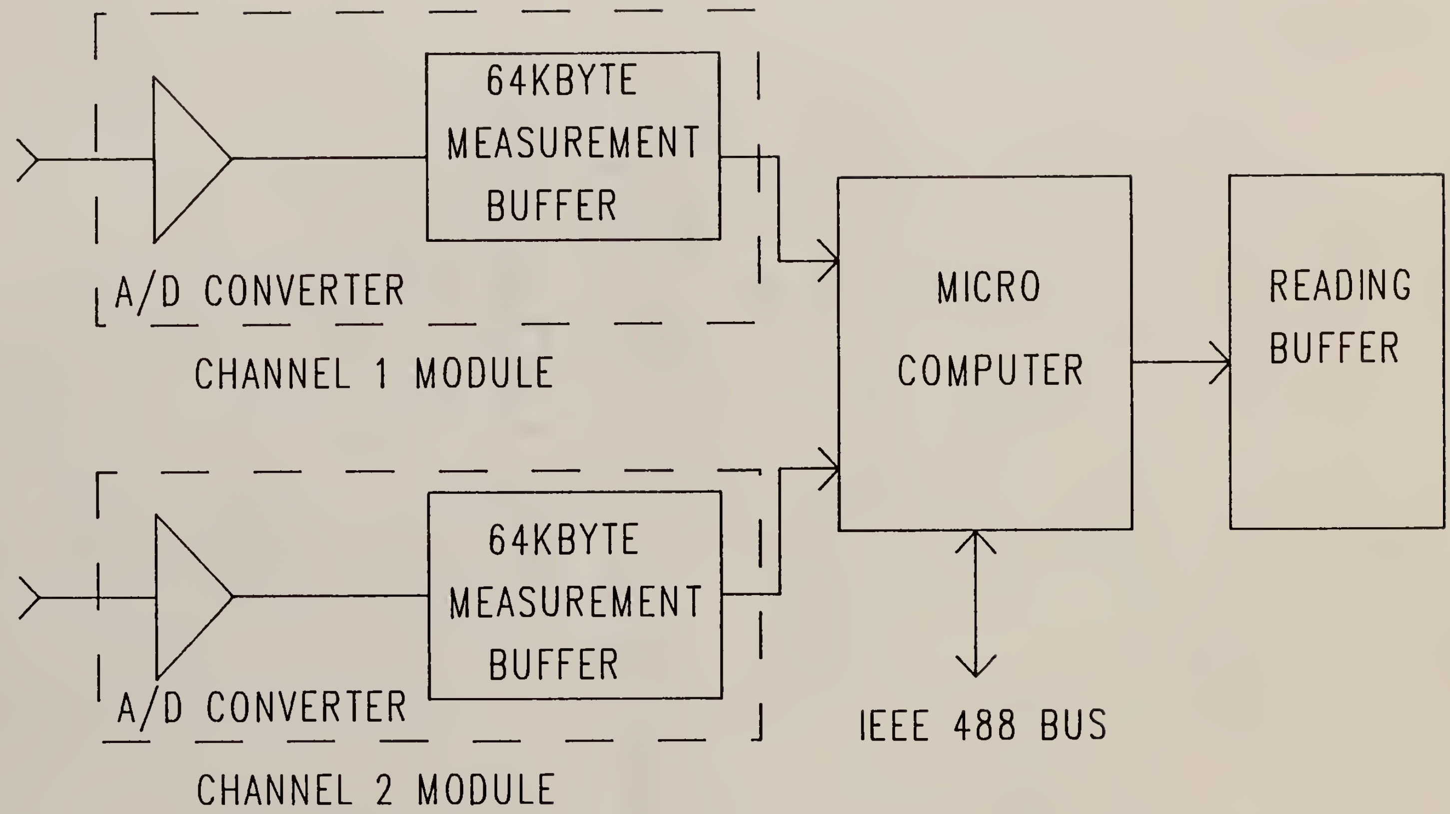

Fig C1. Keithley 194A dual channel architecture
(Model 194A High Speed Voltmeter Instruction Manual, p4–56)

processed. Any previous reading will be overwritten by the new one.

2. Reading buffer enabled: In this case, a maximum of 100 readings from both channels can be stored in the order the readings become available.

Therefore, care has to be taken to make certain that data sent over the bus comes from the desired channel. The following solutions are possible:

1. Turn off the unused channel.
2. Discretionary use of trigger modes for both channels.
3. Enable the reading buffer. Using suitable data formats, seperate the data using the channel number in the suffix in software.

Triggering:

There are two steps involved for initiating a measurement sequence. First, the A/D converter must be armed, so that it is running and processing data. Secondly, the unit must be triggered by the appropriate trigger stimulus (determined by the selected source) before it performs a measurement sequence.

Samples and Rate selection considerations:

The sampling rate has to be sufficiently high so as not to lose important information present in the original signal. The sampling frequency must be at least twice the highest frequency component in the measured signal. Otherwise, aliasing results. The sampling duration or the number of samples is based on how many cycles or how much of a single cycle we wish to sample. The maximum number of samples is 32,768. The overall resolution and accuracy of the measurement has to be considered when selecting the sampling rate. The A/D converter operates with 16 bit resolution for sampling rates less than or equal to 100 KHz. Above 100 KHz, it operates with 8 bit resolution.

The resolution of the programmed sampling interval is 0.1 usec. If the programmed sampling rate results in an interval below the resolution, the voltmeter automatically adjusts the frequency to the nearest whole interval. For example, a sampling rate of 35 KHz results in an interval of 28.571 usec. The voltmeter adjusts the interval to 28.5 usec giving a sampling rate of 35.0877193 KHz.

Simultaneous sampling of channels:

The architecture of the Keithley 194A poses problems for simultaneous sampling of channels. The channels are programmed in the waveform mode (F0X). The channels can be triggered simultaneously by using the following trigger commands.

Channel 1: T3X (Single, Group Execute Trigger)

Channel 2: T25X (Single, Other channel)

Channel 1 can also be programmed to trigger at a predetermined slope and trigger level.

The samples are stored in the respective measurement buffer in the locations pointed by the B1 and B2 pointers. Once the samples are transferred to the controller over the IEEE 488 bus, the channels have to be programmed to some other function (say F1X) in order to cancel the waveform output to enable further access to samples from location 0. The sampling rate for a particular frequency can be obtained from the voltmeter using the status command U6. The setup for simultaneous sampling is shown in fig C2. The program for simultaneous sampling is given in page C7.

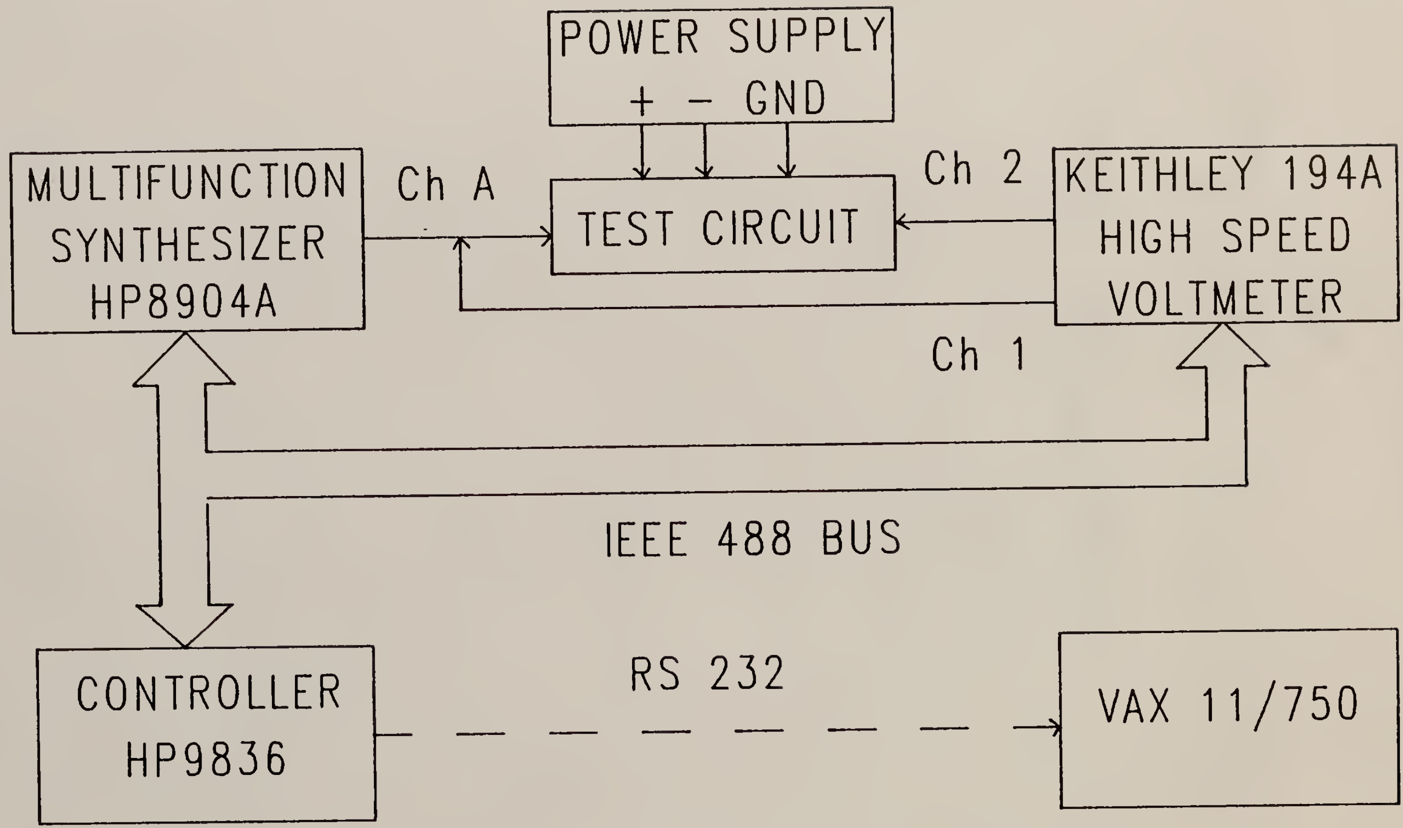

Fig C2. Set up for simultaneous sampling of input and output signals of a test circuit

```
10    ! PROGRAM TO OBTAIN SIMULTANEOUS SAMPLING OF CHANNELS
20    ! ON THE KEITHLEY 194A
30    !      T.R.RAMESH              FILE: TFM_DATA
40    !___________________________________________________
50    OPTION BASE 0
60    INTEGER I,J,Z,Length,Points
70    DIM A(2500),B(2500),Freq(100),Samples, D$[8], S$[8]
80    Fn_generator = 717        ! HP-IB address of HP8904A
90    Voltmeter = 726           ! HP-IB address of Keithley 194A
100   REMOTE Fn_generator
110   REMOTE Voltmeter
120   CLEAR Fn_generator        ! Reset HP8904A
130   CLEAR Voltmeter           ! Reset Keithley 194A
140   PRINTER IS CRT
150   !-------------------------------------------------!
160   ! Initialize HP8904A                              !
170   !-------------------------------------------------!
180   DISP "Initializing Fn_generator..."
190   OUTPUT Fn_generator;"PS"     ! Preset
200   OUTPUT Fn_generator;"FC1OF"  ! Float control Ch.A off
210   OUTPUT Fn_generator;"GM0"    ! Channel configuration
                                   ! mode
220   OUTPUT Fn_generator;"FRA10HZ APA5VL PHA0DG WFASI"
                   ! Ch.A: Frequency = 10 Hz, Phase = 0 Deg
                   ! Amplitude = 5 Vpk, Waveform = Sine
230   OUTPUT Fn_generator;"OO2OF >" ! Disable output B
300   !-------------------------------------------------!
310   ! Generate a frequency array from 10 Hz to 100 KHz in!
320   ! logarthmic steps                                !
330   !-------------------------------------------------!
340   DATA 10.0,20.0,30.0,40.0,50.0,60.0,70.0,80.0,90.0
350   J = 0
360   FOR Z = 0 TO 4
370      FOR I = 0 TO 8
380         READ Freq(I+J)
390         Freq(I+J) = Freq(I+J)*10^Z
400         IF Freq(I+J) >100000.0 THEN GOTO Start
410         Length = Length + 1
420      NEXT I
430      J = J+9
440      RESTORE
450   NEXT Z
500   !-------------------------------------------------!
510   ! Create a data file for storage of samples. The file!
520   ! will be stored on a disk in the right drive of the !
530   ! HP9836                                          !
540   !-------------------------------------------------!
550   CREATE ASCII "HPFILTER:INTERNAL,4",1000
560   ASSIGN @Ofile TO "HPFILTER:INTERNAL,4"
570   OUTPUT @Ofile;Length
```

```
600  !---------------------------------------------------------!
610  ! Obtain the input and output samples for all the       !
620  ! frequencies                                           !
630  !---------------------------------------------------------!
700  DISP "Initializing Voltmeter...."
710  FOR I = 0 TO Length-1
720     Points = 512                       ! Points per cycle
730     Samples = Points-1 * Freq(I)       ! Sampling frequency
740     IF Samples > 1.E+6 THEN
750        Points = Points/2
770        GOTO 730
780     END IF
790     REDIM A(Points-1),B(Points-1)
800     OUTPUT Fn_generator;"FRA"&VAL$(Freq(I))&"HZ"
810     OUTPUT Voltmeter;"C1G1M2R0X"   ! format,autorange
820     OUTPUT Voltmeter;"N0,"&VAL$(Points)&"S1,"&VAL$
              (Samples)&"HZX"
830     OUTPUT Voltmeter;"B1,0B2,"&VAL$(Points-1)&"X"
                       ! Start and end buffer pointers
840     OUTPUT Voltmeter;"C2G1M2R0X"   ! format,autorange
850     OUTPUT Voltmeter;"N0,"&VAL$(Points)&"S1,"&VAL$
              (Samples)&"HZX"
860     OUTPUT Voltmeter;"B1,0B2,"&VAL$(Points-1)&"X"
                       ! Start and end buffer pointers
870     PRINT "WAITING FOR BUFFER TO FILL "
880     OUTPUT Voltmeter;"C1T3X"  ! Trigger source
890     OUTPUT Voltmeter;"C2T25X" ! Trigger source
900     TRIGGER Voltmeter         ! Trigger both channels
910     WAIT Delay
920     OUTPUT Voltmeter;"C1F0X"
930     PRINT "CHANNEL 1 READINGS"
940     FOR J = 0 TO Points-1
950        ENTER Voltmeter;A(J)
960     NEXT J
970     OUTPUT Voltmeter;"C2F0X"
980     PRINT "CHANNEL 2 READINGS"
990     FOR J = 0 TO Points-1
1000       ENTER Voltmeter;B(J)
1010    NEXT J
1040   OUTPUT Voltmeter;"U6X"     ! Obtain sampling
1050   ENTER Voltmeter;D$         ! frequency
1070   Samples = VAL(D$[3,11])
1080   OUTPUT @Ofile;Points,Freq(I),Samples,A(*),B(*)
1090   OUTPUT Voltmeter;"C1F1XC2F1X" ! Cancel waveform
                                     ! output
1100  NEXT I
1110  ASSIGN @Ofile TO *      ! Close data file
1120  DISP "DATA COLLECTION COMPLETED"
1130  END
```

APPENDIX D

Source code listings

```
10    !*****************************************************
20    !    AUTOMATED TRANSFER FUNCTION MEASUREMENTS         *
30    !  This program determines the frequency response of *
40    ! circuits and devices using computer controlled      *
50    ! lock-in systems. The HP8904A Synthesizer provides   *
60    ! the input and reference signals of the lock-in      *
70    ! system. The Keithley 194A measures the input and    *
80    ! output voltages of the system. The program provides*
90    ! both autosensitivity and autophase selection.       *
100   !                                                     *
110   ! The following hardware connections are necessary:  *
120   !    HP8904A  Ch A:    Input signal                   *
130   !    HP8904A  Ch B:    Reference signal               *
140   !                                                     *
150   !    Keithley 194A Ch 1:   Output voltage             *
160   !    Keithley 194A Ch 2:   Input voltage              *
170   !                                                     *
180   !    6236B Power supply:   +15 V & -15 V              *
190   !                                                     *
200   ! Work station:  HP9836                               *
210   ! Language: HPBASIC 3.0                               *
220   !                                                     *
230   ! Programmer: T.R.Ramesh          03/27/89            *
240   !                                                     *
250   ! File name: ATFM                                     *
260   !*****************************************************
290   !
300   ! PROGRAM DECLARATIONS
310   !
320   OPTION BASE 0
330   INTEGER Voltmeter, Fn_generator, Printer
335   INTEGER Ref, Second, I,J,Z
340   DIM V_in(100), V_out(100), Db_gain(100), Freq(100)
345   DIM Phase(100),A(10),B(10),Offset(100)
350   DIM Mag$[8],Phase$[8],Name$[20],Title$[65],Date$[15]
370   Voltmeter = 708              ! HP-IB Address for 194A
380   Fn_generator = 717           ! HP-IB Address for HP8904A
390   Printer = 703                ! HP-IB Address for HP2631B
400   Second = 1                   ! Delay
410   Five_sec = 5                 ! Delay
420   !---------------------------------------------------!
430   ! Set up a frequency array from 10 Hz to 100 KHz in  !
440   ! logarthmic steps                                   !
450   !---------------------------------------------------!
460   GOSUB Initialize             ! Initialize 194A & HP8904A
470   DATA 10,20,30,40,50,60,70,80,90
480   J=0
490   FOR Z=0 TO 4
500      FOR I=0 TO 8
510         READ Freq(I+J)
520         Freq(I+J)=Freq(I+J)*10^Z
```

```
530         IF Freq(I+J) > Value THEN GOTO Again
540         Points=Points+1
550      NEXT I
560      J=J+9
570      RESTORE
580   NEXT Z
590   !--------------------------------------------------------!
600   ! Main data acquisition loop                             !
610   !--------------------------------------------------------!
620   Again: OUTPUT Voltmeter;"C2X"    ! Point to input of
630                                    ! lock-in system
640   ON ERROR GOTO Errors
650   FOR I = 0 TO Points-1
660      DISP "Data acquisition is occuring at this time"
670      DISP " ("&VAL$(Freq(I))&" Hz)"
680      BEEP
690      OUTPUT Fn_generator;"FRA"&VAL$(Freq(I))&"HZ"
700      OUTPUT Fn_generator;"FRB"&VAL$(Freq(I))&"HZ"
710      OUTPUT Fn_generator;"APA"&Ampa$&"VL APB"&Ampb$&"VL"
720   !
730   ! Program Sampling rate and the number of samples for
740   ! measurement
750      IF Freq(I)<100 THEN Samples = 1.E+3
760      IF Freq(I)>=100 AND Freq(I)<1E3 THEN Samples=1.E+4
770      IF Freq(I)>=1E3 AND Freq(I)<1E4 THEN Samples=1.E+5
780      IF Freq(I)>=1E4 THEN Samples = 5.E+5
790      Num = 100*Samples/Freq(I)
800  Here: OUTPUT Fn_generator;"PHB0DG"  ! Reset reference
805                                       ! phase
810      OUTPUT Voltmeter;"N0,"&VAL$(Num)&"XS1,"&VAL$
                 (Samples)&"X"
820      OUTPUT Voltmeter;"C2F2T3X"    ! RMS, GET Trigger
830      TRIGGER Voltmeter             ! Trigger
840      WAIT Second
850      ENTER Voltmeter;V_in(I)       ! Input reading (RMS)
860      OUTPUT Voltmeter;"C1Z0X"      ! Zero off
870      OUTPUT Voltmeter;"C1XN0,"&VAL$(Num)&"XS1,"&VAL$(
                 2*Samples)&"X"
880      OUTPUT Fn_generator;"APA0VL APB0VL"
890      OUTPUT Voltmeter;"C1T3X"
900      TRIGGER Voltmeter
910      WAIT Second
920      ENTER Voltmeter;Offset(I)     ! Obtain offset
930      OUTPUT Voltmeter;"C1Z4X"      ! Enable zero
940      GOSUB Autoset                 ! Determine phase
                                       shift
950      OUTPUT Fn_generator;"PHB"&VAL$(Middle(I))&"DG"
                               ! Shift reference phase by
                               ! the phase_shift
980      OUTPUT Voltmeter;"C1T3X"
990      TRIGGER Voltmeter
```

```
1000      WAIT Second
1010      ENTER Voltmeter;V_out(I)   ! Output reading (Avg)
1020      V_out(I) = V_out(I)/(VAL(Ampb$))
1030      V_out(I) = V_out(I)*Scale ! Convert Avg to RMS
1040      Db_gain(I) = 20*(LGT(((V_out(I)))/(V_in(I))))
                                     ! Compute gain
1050      OUTPUT Voltmeter;"C2X"     ! Point to input
1060    NEXT I
1090    !----------------------------------------------------!
1100    ! Identify whether the phase angle is lead or lag    !
1110    !----------------------------------------------------!
1120    REDIM Phase(Points-1),Offset(Points-1)
1130    Min = MIN(Phase(*))
1140    Max = MAX(Phase(*))
1150    Angle$="LAG"
1170    Off = 0.0
1180    FOR I=0 TO Points-1
1190       Off = Off+Offset(I)
1200    NEXT I
1210    Off = Off/Points
1220    IF Min > 181.0 THEN
1230       FOR I =0 TO Points-1
1240          Phase(I)=Phase(I)-360.0
1250          Angle$="LEAD"
1260       NEXT I
1270    END IF
1280    !----------------------------------------------------!
1290    ! Obtain parameters from the user                    !
1300    !----------------------------------------------------!
1310    DISP "Create Magnitude and Phase data files";
1320    INPUT Ans$
1330    Ans$=UPC$(Ans$)
1340    IF Ans$="Y" THEN
1350       DISP "Magnitude data file name";
1360       INPUT Mag$
1370       DISP "Phase data file name";
1380       INPUT Phase$
1390    END IF
1400    DISP "Enter your name";
1410    INPUT Name$
1420    DISP "Enter the title of the experiment";
1430    INPUT Title$
1440    DISP "Enter today's date";
1450    INPUT Date$
1460    !----------------------------------------------------!
1470    ! Create Magnitude and Phase data files              !
1480    !----------------------------------------------------!
1490    REDIM Freq(Points-1),Db_gain(Points-1),
              Phase(Points-1)
1510    IF Ans$="Y" THEN
```

```
1520     Create_file(Points,Freq(*),Db_gain(*),Mag$)
1530     Create_file(Points,Freq(*),Phase(*),Phase$)
1540  END IF
1550  !----------------------------------------------------------!
1560  ! Print data and results                                    !
1570  !----------------------------------------------------------!
1580  DISP "Print data and results";
1590  INPUT Print$
1600  Print$=UPC$(Print$)
1610  IF Print$="Y" THEN
1620     GOSUB Print_header
1630     FOR I = 0 TO Points-1
1640        PRINT USING "6D.D,5X,3D.4D,8X,3D.4D,5X,3D.4D,
            4X,3D.4D";Freq(I),V_in(I),V_out(I),Db_gain(I),
            Phase(I)
1650    NEXT I
1660  END IF
1670  GOTO Fin
1680  !----------------------------------------------------------!
1690  ! Initialize Fn_generator and Voltmeter                     !
1700  !----------------------------------------------------------!
1710 Initialize: !
1720  DISP "Initializing Voltmeter..."
1730  REMOTE Voltmeter         ! Place 194A in remote
1740  CLEAR Voltmeter          ! Reset 194A
1750  WAIT Second
1760  OUTPUT Voltmeter;"C1T7XC2T7X" ! Turn Ch 1 & Ch 2 off
1770  OUTPUT Voltmeter;"G1R0X" ! ASCII format,autorange
1780  OUTPUT Voltmeter;"Q0X" ! Disable reading buffer
1790  DISP "Initializing Fn_generator..."
1800  REMOTE Fn_generator    ! Place HP8904A in remote
1810  CLEAR Fn_generator     ! Reset HP8904A
1820  DISP "Signal amplitude (Peak) [ Output of test
          circuit should not exceed 11 V (Peak)]";
1825  INPUT Ampa$
1830  DISP "Reference Signal (1-SI, (Default) SQ)";
1840  INPUT Ref
1850  SELECT Ref
1860    CASE 1                 ! Sinusoidal reference
1870       Sig$="SI"
1880       Value = 100000.0 ! Maximum frequency
1890       Scale = SQR(2)    ! Scale factor
1900    CASE ELSE
1910       Sig$="SQ"           ! Square wave reference
1920       Value = 50000.0  ! Maximum frequency
1930       Scale = PI * 0.5/SQR(2) ! Scale factor
1940  END SELECT
1960  Ampb$="1"               ! Reference amplitude (Peak)
1970  OUTPUT Fn_generator;"PS GM0 FC1OF FC2OF"
                           ! Preset, Channel configuration
                           ! mode, Float controls off
```

```
2000  OUTPUT Fn_generator;"FRA10HZ APA"&Ampa$&"VL PHA0DG
           WFASI"             ! Ch A: Freq = 10 Hz, Amp = 5 V
                              ! Phase = 0 Deg, Wave = Sine
2040  OUTPUT Fn_generator;"FRB10HZ APB"&Ampb$&"VL PHB0DG
          WFBSQ"              ! Ch B: Freq = 10 Hz, Amp = 1 V
                              ! Phase = 0 Deg, Wave = Square
2080  OUTPUT Fn_generator;"> >" ! Simulate NEXT Key
2090  RETURN
2100  !
2110  !
2120  !-----------------------------------------------------!
2130  ! Autosensitivity and autophase selection             !
2140  !-----------------------------------------------------!
2150 Autoset: !
2160  !
2170  OUTPUT Fn_generator;"APA"&Ampa$&"VL APB"&Ampb$&"VL"
2180  OUTPUT Fn_generator;"PHBIS90DG"
2190  FOR J = 0 TO 4       ! Output voltage in steps of 90
2200                       ! degrees ref. channel phase
2210     OUTPUT Voltmeter;"C1T3X"
2220     TRIGGER Voltmeter
2230     WAIT Second
2240     ENTER Voltmeter;B(J)
2250     B(J) = B(J)/(VAL(Ampb$))
2260     IF J = 4 THEN GOTO Range
2270     IF J = 3 THEN
2280        OUTPUT Fn_generator;"PHB359.9DG"
2290     ELSE
2300        OUTPUT Fn_generator;"PHBUP"
2310     END IF
2320  NEXT J
2330  !
2340 Range: ! Identify the range during which quadrature
              phase occurs
2350  !
2360  Equal = 0
2370  FOR J = 0 TO 3
2380     IF SGN(B(J))=SGN(B(J+1)) THEN
2390        Equal = Equal + 1
2400     ELSE
2410        IF SGN(B(J))<0.0 THEN
2420          First = J*90.0
2430          GOTO Find
2440        END IF
2450     END IF
2460  NEXT J
2470  IF Equal = 4 THEN GOTO Senst
2480 Find: !
2490  Last=First+90.0
2500  IF Last=360.0 THEN Last=359.9
2510  !
```

```
2520  ! Find quadrature phase using Bisection Search method
2530  !
2540  OUTPUT Fn_generator;"PHB"&VAL$(First)&"DG"
2550  OUTPUT Voltmeter;"C1T3X"
2560  TRIGGER Voltmeter
2570  WAIT Second
2580  ENTER Voltmeter;A(0)
2590  A(0)=A(0)/VAL(Ampb$))
2600  OUTPUT Fn_generator;"PHB"&VAL$(Last)&"DG"
2610  OUTPUT Voltmeter;"C1T3X"
2620  TRIGGER Voltmeter
2630  WAIT Second
2640  ENTER Voltmeter;A(2)
2650  A(2)=A(2)/(VAL(Ampb$))
2660  IF (A(0)*A(2))>0.0) THEN GOTO Here
2670  IF (A(0)<=0.0) THEN
2680     Rtbis=First
2690     Dx=Last-First
2700  ELSE
2710     Rtbis=Last
2720     Dx=First-Last
2730  END IF
2740  FOR J=1 TO 40
2750     Dx=Dx*0.5
2760     Xmid=Rtbis+Dx
2770     OUTPUT Fn_generator;"PHB"&VAL$(Xmid)&"DG"
2780     OUTPUT Voltmeter;"C1T3X"
2790     TRIGGER Voltmeter
2800     WAIT Second
2810     ENTER Voltmeter;A(1)
2820     A(1)=A(1)/(VAL(Ampb$))
2830     IF (A(1)<=0.0) THEN Rtbis=Xmid
2840     IF (ABS(Dx)<0.5) THEN GOTO Found
2850  NEXT J
2860 Found: ! Obtain the phase_shift
2870  Middle(I) = Rtbis+90.0
2890  IF Middle(I)>359.9 THEN Middle(I)=Middle(I)-360.0
2900  GOTO A1
2910 Senst: ! Sensitivity variations
2930  IF VAL(Ampb$)<10 THEN
2940     Ampb$ = VAL$(VAL(Ampb$)+1)
2950  ELSE
2960     IF VAL(Ampa$)<10 THEN
2970        Ampa$=VAL$(VAL(Ampa$)+1)
2980     ELSE
2990        DISP "Unable to detect phase !!!"
3000        STOP
3010     END IF
3020  END IF
3030  GOTO Here
```

```
3040 A1: RETURN
3050  !
3060 Errors: !
3070  IF ERRN=28 THEN
3080     DISP "NEGATIVE VALUE FOUND !!!"
3090     OUTPUT Fn_generator;"PHB0DG"
3100     GOTO Here
3110  ELSE
3120     DISP "Error" ERRN "has occured";
3130     STOP
3140  END IF
3150  GOTO Fin
3160 Print_header: !
3170  !
3180  DISP "Select a printing device (1-CRT,2-PRINTER)";
3190  INPUT Device
3200  IF Device=2 THEN
3210     PRINTER IS Printer
3220  ELSE
3230     PRINTER IS CRT
3240  END IF
3250  PRINT "---------------------------------------------"
3260  PRINT Title$
3270  PRINT
3280  PRINT Name$
3290  PRINT "----------------------------------------------"
3300  PRINT "Frequency range: 10 Hz TO 100 KHz  Data points:
          ";Points
3310  PRINT "Output phase angle: ",Angle$
3320  IF Ans$="Y" THEN
3330     PRINT "Magnitude data file: ",Mag$,"  Offset:",Off
3340     PRINT "Phase data file    : ",Phase$
3350  END IF
3360  PRINT "----------------------------------------------"
3370  PRINT
3380  PRINT "Frequency  Input voltage  Output voltage
            Db_gain Phase "
3390  PRINT "  (Hz)         (RMS)           (RMS)
             (dB)   (Deg)"
3400  PRINT
3410  RETURN
3420  !
3430 Fin: PRINTER IS CRT
3440  END
3450  !
3460  SUB Create_file(Num,X(*),Y(*),Plot_title$)
3470   ! Num:  Number of data points
3480   ! X(*): X axis data
3490   ! Y(*): y axis data
3500   ! Plot_title$: Data file name
```

```
3510  OPTION BASE 0
3520  INTEGER I,J
3530  Records=INT((Num+1)/16)+1
3540  CREATE BDAT Plot_title$,Records
3550  ASSIGN @File TO Plot_title$
3560  OUTPUT @File;Num;X(*),Y(*)
3570  ASSIGN @File TO *
3580  SUBEND
```

Table D1

Sample output listing of transfer function measurement using lock-in systems

---

FREQUENCY RESPONSE OF A RC LOWPASS FILTER
T.R.RAMESH 25 March 1989

Frequency range: 10 Hz to 100 KHz Data points: 37
Output phase angle: LEAD Offset: -0.0372 V
Magnitude data file: LPMAG
Phase data file : LPPHA

Frequency (Hz)	Input voltage (RMS)	Output voltage (RMS)	Db_gain (dB)	Phase (Deg)
10.0	3.5320	3.2691	-.6718	-1.4063
20.0	3.5320	3.2617	-.6916	-2.9004
30.0	3.5320	3.2689	-.6724	-4.3006
40.0	3.5320	3.2601	-.6957	-5.8008
50.0	3.5320	3.2495	-.7242	-7.2070
60.0	3.5310	3.2370	-.7551	-8.6133
70.0	3.5310	3.2226	-.7939	-10.0195
80.0	3.5310	3.2065	-.8374	-11.4258
90.0	3.5310	3.1887	-.8857	-12.8320
100.0	3.5310	3.1719	-.9315	-14.1064
200.0	3.5300	2.9147	-1.6635	-26.6309
300.0	3.5290	2.6024	-2.6457	-36.8262
400.0	3.5280	2.2982	-3.7228	-44.8682
500.0	3.5270	2.0296	-4.7999	-51.1084
600.0	3.5270	1.8021	-5.8326	-56.0303
700.0	3.5250	1.6119	-6.7964	-59.8096
800.0	3.5250	1.4529	-7.6983	-63.0176
900.0	3.5250	1.3192	-8.5371	-65.6104
1000.0	3.5250	1.2171	-9.2367	-67.8076
2000.0	3.5240	.6482	-14.7063	-78.7939
3000.0	3.5250	.4487	-17.9049	-81.5625
4000.0	3.5250	.3073	-21.1915	-83.3203
5000.0	3.5260	.2434	-23.2202	-84.4189
6000.0	3.5260	.2059	-24.6705	-84.7266
7000.0	3.5280	.1759	-26.0475	-85.1221
8000.0	3.5290	.1542	-27.1937	-85.3418
9000.0	3.5300	.1382	-28.1429	-85.4736
10000.0	3.5440	.1247	-29.0735	-85.5176
20000.0	3.5490	.0618	-35.1825	-85.0342
30000.0	3.5330	.0411	-38.6831	-84.4189
40000.0	3.5200	.0312	-41.0340	-81.7383
50000.0	3.5060	.0256	-42.7370	-81.5186

```
/*****************************************************************************
 *
 *  SOURCE FILE:      xfer_fn.c
 *
 *
 *  FUNCTION:         main()
 *
 *
 *
 *  DESCRIPTION:      Obtains the frequency response of a linear
 *                    system by the following methods.
 *                      1. Three parameter least squared fit
 *                         algorithm
 *                      2. Cross-correlation method
 *
 *
 *  DOCUMENTATION
 *  FILES:            atfm.d
 *
 *
 *  ARGUMENTS:        None
 *
 *
 *  RETURN:           None
 *
 *
 *  FUNCTIONS
 *  CALLED:           corr(),fft(),lsf()
 *
 *
 *  AUTHOR:           T.R.Ramesh
 *
 *
 *  DATE CREATED:     26 March 1989               Version 1.00
 *
 *
*****************************************************************************/

#include 'atfm.h'                     /* Contains error-codes and      */
                                      /* definitions used by xfer_fn   */
                                      /* routines                      */

main()
{
int         i,c,j,k,                  /* General purpose indices       */
            length,                   /* Number of frequencies         */
            type,                     /* Selection variable            */
            freq,                     /* Index for frequency loop      */
            points;                   /* Points for each frequency     */

double      gain[MAX],                /* Amplitude response array      */
            phase_shift[MAX],         /* Phase response array          */
```

```
                resolution[MAX],      /* Resolution in degrees          */
                frequencies[MAX],     /* Frequency array                */
                yin[MAX], yout[MAX], /* Input and output samples       */
                samp_freq,            /* Sampling frequency             */
                x[MAX],               /* Time array                     */
                step,                 /* Step length                    */
                in_amp, out_amp,      /* Amplitude estimates            */
                in_off, out_off,      /* Offset estimates               */
                in_phase, out_phase, /* Phase estimates                */
                max,omax,imax;        /* Dummy variables                */

char            data_file[20],        /* Input data file                */
                mag_file[20],         /* Magnitude data file            */
                phase_file[20];       /* Phase data file                */

COMPLEX         in[MAX],              /* Input signal array             */
                out[MAX],             /* Output signal array            */
                result[MAX];          /* Correlation result vector      */

FILE            *input,               /* Data file pointers             */
                *mag,
                *pha;

extern corr(),fft(),lsf();

/*-----------------------------------------------------------------*/
/*  Obtain the input and output samples of the transfer function*/
/*  measurement                                                    */
/*-----------------------------------------------------------------*/

puts("\n AUTOMATED TRANSFER FUNCTION MEASUREMENTS");
puts("\n\n Refer to documentation file for data file format");

printf("\n Select one of the following methods:");
printf("\n  1. least squared method");
printf("\n  2. Cross-correlation");
printf("\n\n your choice?");
scanf("%d", &type);

printf("\n Data file name?");
scanf ("%s", &data_file);

input = fopen(data_file,"r");

fscanf(input,"%d", &length);
length=1;
for (freq=0; freq < length; freq++)
    {
    fscanf(input,"%d",&points);
```

```
   fscanf(input,'%lf %lf', &frequencies[freq], &samp_freq);
   for (i = 0; i < points; i++)
      fscanf(input,'%lf', &yin[i]);
   for (i = 0; i < points; i++)
      fscanf(input,'%lf', &yout[i]);
   step = 1.0/samp_freq;
/*------------------------------------------------------------------------*/
/*  Obtain amplitude and phase responses for each frequency              */
/*------------------------------------------------------------------------*/
   switch (type)
      {
      case LSF:                    /* Least squared method              */

         for (i = 0; i < points; i++)
            x[i] = i * step;
         if (sine_fit(points, x, yin, &in_amp, frequencies[freq],
            &in_off, &in_phase) != NORMAL)
            {
            printf('\n Error in sine fit to input data');
            exit(0);
            }
         if (sine_fit(points, x, yout, &out_amp, frequencies[freq]
            &out_off, &out_phase) != NORMAL)
            {
            printf('\n Error in sine fit to output data');
            exit(0);
            }
         gain[freq] = 20 * log10(out_amp/in_amp);

         if ((out_phase - in_phase) > 180.0)
            in_phase += 360.0;
         else
            {
            if ((in_phase - out_phase) > 180.0)
               out_phase += 360.0;
            }
         phase_shift[freq] = out_phase - in_phase;
         break;

      case CORR:                      /* Cross correlation method    */
         points =3;
         resolution[freq] = 360.0/(double)points;

         for (i = 0; i < points; i++)
            {
            in[i] = cmplx(yin[i],ZERO);
            out[i] = cmplx(yout[i],ZERO);
            }
```

```
        if (corr(in,out, points, result) != NORMAL)
           {
           printf("\n Error in correlation routine");
           exit(0);
           }
        max = 0.0;
        for (i = 0; i < points; i++)
           if (result[i].re > max) max = result[i].re;

        i = 0;
        while (i < points)
              {
              if (max == result[i].re) break;
              i += 1;
              }

        phase_shift[freq] = (double)i * resolution[freq];

        if (i >= points/2)
          phase_shift[freq] = phase_shift[freq] - 360.0;

        phase_shift[freq] *= -1.0;

        imax = 0.0;
        omax = 0.0;

        for (i = 0; i < points; i++)
           {
           if (yin[i] > imax) imax = yin[i];
           if (yout[i] > omax) omax = yout[i];
           }

        gain[freq] = 20.0 * log10(omax/imax);
        break;

  default:
        exit(0);
 }
}

/*-----------------------------------------------------------------*/
/* Create gain and phase data files                                */
/*-----------------------------------------------------------------*/

printf("\n Create data files? (Enter 1)");
scanf("%d",&c);
if (c == 1)
   {
   printf("\n Gain data file?");
   scanf("%s", &mag_file);
```

```
    printf("\n Phase data file?");
    scanf("%s", &phase_file);
    mag = fopen(mag_file,"w");
    pha = fopen(phase_file,"w");
    fprintf(mag,"%d",length);
    fprintf(pha,"%d",length);

    for (i = 0; i < length; i++)
        {
        fprintf(mag,"\n%f %f ",frequencies[i], gain[i]);
        fprintf(pha,"\n%f %f",frequencies[i],phase_shift[i]);
        }
    }
}
```

```
/*****************************************************************
 *
 *  SOURCE FILE:       corr.c
 *
 *
 *  FUNCTION:          corr(v_in, v_out, n, result)
 *
 *
 *  DESCRIPTION:       Determines the correlation of two complex
 *                     arrays
 *
 *
 *  DOCUMENTATION
 *  FILES:             None
 *
 *
 *  ARGUMENTS:
 *
 *     v_in            (input) COMPLEX
 *                     Array containing the input signal
 *
 *
 *     v_out           (input) COMPLEX
 *                     Array containing the output signal
 *
 *
 *     n               (input) int
 *                     Size of the array. (must be a power of 2)
 *
 *
 *     result          (output) COMPLEX
 *                     Array containing the correlation of the
 *                     two input arrays
 *
 *
 *  RETURN:             int
 *                      NORMAL      : Normal return
 *                      ERR_INDFT   : Error in input voltage DFT
 *                      ERR_OUTDFT  : Error in output voltage DFT
 *                      ERR_PRIDFT  : Error in product IDFT
 *
 *
 *  FUNCTIONS
 *  CALLED:            sin(),cos()
 *                     cadd(), csub(), cmult().
 *
 *
 *  AUTHOR:            T.R.Ramesh
 *
 *
 *  DATE CREATED:      23 Mar 1989              Version 1.00
 *
```

```
 *  REVISIONS:       None
 *
***************************************************************************/

#include 'atfm.h'                       /* Header file for xfer_fn        */
                                        /* routines                       */

int corr(v_in, v_out, n, result)

int          n;

COMPLEX      v_in[],
             v_out[],
             result[];

{
int          i,j;                      /* General purpose indices       */

extern  int fft();

/*-------------------------------------------------------------------*/
/*  Compute the DFT of both the input vectors                        */
/*-------------------------------------------------------------------*/

if (fft(v_in, n, DFT) != NORMAL)
   return (ERR_INDFT);

if (fft(v_out, n, DFT) != NORMAL)
   return (ERR_OUTDFT);
/*------------------------------------------------------------------*/
/*  Obtain the product of one vector with the complex               */
/*  conjugate of the other                                          */
/*------------------------------------------------------------------*/

for (i = 0; i < n; i++)
   {
   v_in[i] = cconj(v_in[i]);
   result[i] = cmult(v_in[i],v_out[i]);
   }
/*------------------------------------------------------------------*/
/*  Obtain the IDFT of the resultant vector                         */
/*------------------------------------------------------------------*/

if (fft(result, n, IDFT) != NORMAL)
   return (ERR_PRIDFT);
/*------------------------------------------------------------------*/
/*  Normal termination                                              */
/*------------------------------------------------------------------*/

return (NORMAL);
}
```

```
/**********************************************************************
 *
 *  SOURCE FILE:      dft.c
 *
 *
 *  FUNCTION:         fft(data, size, sign)
 *
 *
 *  DESCRIPTION:      Determines the Discrete Fourier Transform
 *                    (DFT) or Inverse Discrete Fourier Transform
 *                    (IDFT) of a complex array.
 *
 *
 *  DOCUMENTATION
 *  FILES:            None
 *
 *
 *  ARGUMENTS:
 *
 *     data           (input/output) COMPLEX
 *                    Input data array. data contains DFT or IDFT
 *                    upon return.
 *
 *
 *     size           (input) int
 *                    Size of the array. (must be a power of 2)
 *
 *
 *     sign           (input) int
 *                    DFT  :  Discrete Fourier Transform.
 *                    IDFT :  Inverse Discrete Fourier Transform.
 *
 *
 *  RETURN:            int
 *                     NORMAL  : Normal return
 *                     ERR_FFT : Number of data points is not a
 *                               power of 2.
 *
 *
 *  FUNCTIONS
 *  CALLED:           sin(),cos()
 *                    cadd(), csub(), cmult().
 *
 *
 *  AUTHOR:           T.R.Ramesh
 *
 *
 *  DATE CREATED:     20 Mar 1989               Version 1.00
 *
 *
 *  REVISIONS:        None
```

```
 *
*************************************************************************/


#include 'atfm.h'                  /* Header file for xfer_fn      */
                                   /* routines                     */


int fft(data, size, sign)

int       size,
          sign;

COMPLEX   data[];

{
int       i,j,k,l,                   /* General purpose indices     */
          mmax,                      /* Array size for recursive    */
                                     /* transform computations      */
          step;                      /* Step size to access groups */
                                     /* of arrays                   */

COMPLEX   w,                         /* A complex number            */
          dummy, temp;               /* Dummy variables             */

/*------------------------------------------------------------------*/
/*  Check if the size of the data array is a power of 2            */
/*------------------------------------------------------------------*/
    if ((size %  2) != NORMAL)
        return(ERR_FFT);
    k = size;

    while (k >1)
       {
       if ((k % 2) != NORMAL)
          return (ERR_FFT);
       k = k/2;
       }
/*------------------------------------------------------------------*/
/*  Perform bit reversal of the input data stream                  */
/*------------------------------------------------------------------*/
j = 0;
for (i = 0; i < size; i++)
   {
   if (i < j)
      {                               /* Swap the two complex        */
      dummy.re = data[i].re;          /* numbers                     */
      dummy.im = data[i].im;
      data[i].re = data[j].re;
      data[i].im = data[j].im;
```

```
      data[j].re = dummy.re;
      data[j].im = dummy.im;
      }

   k = size/2;
   while ( k >= 1 && k <= j)
      {
      j -= k;
      k = k/2;
      }
   j += k;
   }

/*-------------------------------------------------------------------*/
/* Perform FFT using the Danielson-Lanczos algorithm                  */
/* (successive doubling method)                                       */
/*-------------------------------------------------------------------*/
   mmax = 1;
   while (size >= (2 * mmax))
      {
      step = 2 * mmax;
      dummy = cmplx(ONE,ZERO);
      w = cmplx(cos(  PI/((double)mmax )), sin( PI/
             ((double)(sign * mmax ))));
      for (j = 0; j < mmax; j ++)
         {
         for (l = j; l < size; l += step)
            {
            temp = cmult(data[l+mmax], dummy);
            data[l+mmax] = csub(data[l], temp);
            data[l] = cadd(data[l], temp);
            }
         dummy = cmult(w, dummy);
         }
      mmax = step;
      }

/*-------------------------------------------------------------------*/
/*  Divide the DFT by the size of the array to obtain IDFT            */
/*-------------------------------------------------------------------*/
    if (sign == IDFT)
       {
       for(i = 0; i < size; i++)
          data[i] = cdiv(data[i], cmplx((double)size, ZERO));
       }

/*-------------------------------------------------------------------*/
/*  Normal termination                                                */
/*-------------------------------------------------------------------*/
    return(NORMAL);
}
```

```
/*********************************************************************
 *
 *  SOURCE FILE:       lsf.c
 *
 *
 *  FUNCTION:          sine_fit(np, x_dat, y_dat, amplitude, freq,
 *                          dc, theta)
 *
 *
 *  DESCRIPTION:       Provides the amplitude, phase and offset
 *                     estimates of a sampled sinewave using the
 *                     three parameter (known frequency) least
 *                     squared method.
 *
 *
 *  DOCUMENTATION
 *  FILES:             None
 *
 *
 *  ARGUMENTS:
 *
 *     np              (input)  int
 *                     Number of data points.
 *
 *
 *     x_dat           (input)  double *
 *                     Abscissa values of the sampled signal.
 *
 *
 *     y_dat           (input)  double *
 *                     Ordinate values of the sampled signal.
 *
 *
 *     amplitude       (output)  double *
 *                     Amplitude estimate of the sampled signal.
 *
 *
 *     freq            (input)   double
 *                     Frequency (known) of the sampled signal.
 *
 *
 *     dc              (output)  double *
 *                     dc offset estimate of the sampled signal.
 *
 *
 *     theta           (output)  double *
 *                     Phase angle estimate of the sampled signal.
 *
 *
 *  FUNCTIONS
 *  CALLED:            cos(), sin()
 *
 *
```

```
 *  AUTHOR:          T.R.Ramesh
 *
 *
 *  DATE CREATED:    21 Feb 1989                 Version 1.00
 *
 *
******************************************************************/

#include "atfm.h"


int sine_fit(np, x_dat, y_dat, amplitude, freq, dc, theta)

int       np;                           /* Number of points sampled */

double    x_dat[],                      /* Abscissa values          */
          y_dat[],                      /* Ordinate values          */
         *amplitude,                    /* Amplitude estimate       */
          freq,                         /* Frequency                */
         *dc,                           /* Offset estimate          */
         *theta;                        /* Phase angle estimate     */

{

int      i,j,k;                         /* General purpose indicies */

                                        /* Dummy variables          */
double   max, min, amp_est, off_est, sign, angle, phi, w,
         sum_yn, sum_an, sum_bn, sum_abn, sum_aan, sum_bbn,
         sum_yan, sum_ybn, sum_yyn, alpha, beta, ybar, alpha_bar,
         beta_bar, a_n, a_d, b_n, b_d, a, b, c;

/*---------------------------------------------------------------*/
/*  Find the initial amplitude, offset and phase estimates      */
/*---------------------------------------------------------------*/

    min = 100.0;                        /* Determine maximum and    */
    max = 0.0;                          /* minimum values in the    */
                                        /* sampled signal array     */
    for (i = 0; i < np; i++)
       {
       if (y_dat[i] > max) max = y_dat[i];
       if (y_dat[i] < min) min = y_dat[i];
       }
    amp_est = (max - min) / 2.0;
    off_est = (max + min) / 2.0;

    w = 2.0 * PI * freq;
```

```
/*-------------------------------------------------------------------*/
/*  Estimate parameters using the three parameter least              */
/*  squared fit technique                                            */
/*-------------------------------------------------------------------*/

    sum_yn = ZERO;                        /* Initialize sums         */
    sum_an = ZERO;
    sum_bn = ZERO;

    sum_abn = ZERO;
    sum_aan = ZERO;
    sum_bbn = ZERO;

    sum_yan = ZERO;
    sum_ybn = ZERO;
    sum_yyn = ZERO;

/*  Compute nine sums required for the estimates                    */

    for (i = 0; i < np; i++)
       {
       alpha = cos(w * x_dat[i]);
       beta = sin(w * x_dat[i]);

       sum_yn = sum_yn + y_dat[i];
       sum_an = sum_an + alpha;
       sum_bn = sum_bn + beta;

       sum_abn = sum_abn + alpha * beta;
       sum_aan = sum_aan + alpha * alpha;
       sum_bbn = sum_bbn + beta * beta;

       sum_yan = sum_yan + y_dat[i] * alpha;
       sum_ybn = sum_ybn + y_dat[i] * beta;
       sum_yyn = sum_yyn + y_dat[i] * y_dat[i];
       }

/*  Compute the following parameters using the sums                */
/*  calculated above                                               */

    ybar = (sum_yn / ((double)np));
    alpha_bar = (sum_an / ((double)np));
    beta_bar = (sum_bn / ((double)np));

    a_n = (((sum_yan - ybar * sum_an) / (sum_abn - beta_bar *
            sum_an)) - ((sum_ybn - ybar * sum_bn) / (sum_bbn
            - beta_bar * sum_bn)));

    a_d = (((sum_aan - alpha_bar * sum_an) / (sum_abn -
           beta_bar * sum_an)) - ((sum_abn - alpha_bar *
           sum_bn) / (sum_bbn - beta_bar * sum_bn)));
```

```
    b_n = (((sum_yan - ybar * sum_an) / (sum_aan - alpha_bar
            * sum_an)) - ((sum_ybn - ybar * sum_bn) / (sum_abn
            - alpha_bar * sum_bn)));

    b_d = (((sum_abn - beta_bar * sum_an) / (sum_aan -
            alpha_bar * sum_an)) - ((sum_bbn - beta_bar *
            sum_bn) / (sum_abn - alpha_bar * sum_bn)));

    a = a_n / a_d;
    b = b_n / b_d;
    c = (ybar - (a * alpha_bar) - (b * beta_bar));
    *amplitude = sqrt(a * a + b * b);
    *theta = atan( 1.0 * a/b);
    *theta = *theta * 180.0 / PI;
    if (b < 0.0) *theta = *theta + 180.0;
    if (b > 0.0 && a < 0.0) *theta = *theta + 360.0;
    *dc = c;
/*------------------------------------------------------------------*/
/*  Normal termination                                              */
/*------------------------------------------------------------------*/
    return (NORMAL);
}
```

```
/******************************************************************
 *
 *   SOURCE FILE:      atfm.h
 *
 *
 *
 *   DESCRIPTION:      Header files for transfer function
 *                     measurements using cross correlation and
 *                     three parameter least squared fit algorithm
 *
 *
 *   DOCUMENTATION
 *   FILES:            None
 *
 *
 *   AUTHOR:           T.R.Ramesh
 *
 *
 *   DATE CREATED:     22 March 1989              Version 1.00
 *
 *
******************************************************************/

#include <stdio.h>
#include <math.h>
#include 'complex.h'                    /* Header file for complex     */
                                        /* functions                   */

#define  MAX          1000              /* Maximum number of points    */
#define  PI           3.141592653       /* Constant PI                 */
#define  DFT          1                 /* Discrete Fourier Transform  */
#define  IDFT         -1                /* Inverse Discrete Fourier    */
                                        /* Transform                   */
#define  LSF          1                 /* Least squared method        */
#define  CORR         2                 /* Cross correlation method    */
#define  ONE          1.0
#define  ZERO         0.0

/*---------------------------------------------------------------*/
/*  Error codes for functions used in xfer_fn routine            */
/*---------------------------------------------------------------*/
#define  NORMAL       0                 /* Normal return               */
#define  ERR_FFT      1                 /* Return error code for fft() */
#define  ERR_INDFT    2                 /* Error in input voltage DFT  */
#define  ERR_OUTDFT   3                 /* Error in output voltage DFT */
#define  ERR_PRIDFT   4                 /* Error in product IDFT       */
```

```
/*************************************************************************
 *
 *  SOURCE FILE:      atfm.d
 *
 *
 *
 *  DESCRIPTION:      Provides details regarding file formats
 *                    for xfer_fn routines
 *
 *
 *  DOCUMENTATION
 *  FILES:            None
 *
 *
 *  AUTHOR:           T.R.Ramesh
 *
 *
 *  DATE CREATED:     22 April 1989                Version 1.00
 *
 *
*************************************************************************/

Transfer function measurements of linear sytems are accomplished
using two algorithms:

1. Three parameter least squared fit algorithm
2. Cross correlation of input and output signals

The input data file for both the methods must be of the
following format:

     (i)   Number of frequencies   (integer)

     For every frequency,

     (i)   Number of points        (integer)
                                   (must be an integer power of 2
                                    for cross correlation method)
     (ii)  Frequency               (double)
     (iii) Sampling frequency      (double)
     (iv)  Input samples           (double)
     (v)   Output samples          (double)

The output files created are of the following format:

     (i)   Number of points        (integer)

     For every frequency

     (ii)  Frequency (double), Gain/Phase  (double)
```

AUTOMATED TRANSFER FUNCTION MEASUREMENTS

by

TIRUVOOR RADHAKRISHNA RAMESH

B.E., University of Mysore, India, 1985

AN ABSTRACT OF A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Electrical and Computer Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1989

ABSTRACT

This thesis discusses hardware and software techniques for automating transfer function measurements of linear sytems.

The hardware approach is using computer controlled lock-in systems. The lock-in system is designed using a precision IC Multiplier AD534. Software techniques considered are the three parameter (known frequency) least squares fit algorithm and the cross correlation of input and output signals of the linear system

The hardware approach provides phase accuracy dependent on the resolution of the phase shifter. The amplitude response is a function of the phase response due to the nature of the measurement process.

Software techniques thoroughly rely on the simultaneous sampling of input and output signals. The three parameter method is faster and provides better accuracy than the cross correlation method for a given number of samples per cycle.