AN APPLICATION OF THE HYPERBOLIC NAVIGATION RADIO SYSTEM FOR AUTOMATED POSITION AND CONTROL

by

Daniel S. Purdy

Thesis Submitted to the Faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

ELECTRICAL ENGINEERING

APPROVED:

Dr. Timothy Pratt, Chairman

Dr. Theodore Rappaport

Dr. Yvan Beliveau

August, 1989
Blacksburg, Virginia

AN APPLICATION OF THE

HYPERBOLIC NAVIGATION RADIO SYSTEM FOR

AUTOMATED POSITION AND CONTROL

by

Daniel S. Purdy

Timothy Pratt, Chairman

(ABSTRACT)

As automation in the construction site of the future becomes a reality, position location systems are necessary to provide real-time data to an operator. This thesis addresses problems associated with development of a real-time automated position location system using a method similar to hyperbolic navigation methods. The Automated Position and Control (APAC) project is a joint effort between the Civil and Electrical Engineering departments at Virginia Polytechnic and State University and Bechtel Eastern Power Corporation.

The discussion begins with a review of possible methods and answers the question of why the radio hyperbolic method is a promising candidate for providing centimeter accuracy results. The prototype system developed for experimental work is described. The design, analysis, and performance of a 10.7 MHz phase measurement system capable of 5 millisecond measurements is given. Receiver, transmitter, and antenna considerations necessary for accurate differential phase measurements are outlined. Methods of transmitting data and a phase reference to multiple remote receivers are presented. The problem of distributing the phase reference necessary for accurate differential phase measurements is discussed. Comments on providing data communications to a construction equipment operator location in a construction site are given. Included is a brief summary of experimental results from measurements of the prototype system. From the practical experience gained during the experimental prototype development, suggestions are made toward future development in this area of research.

ACKNOWLEDGEMENTS

Many sincere thanks go to Tim Pratt, my graduate advisor, for his patience, support, and thoughtfulness. His wisdom, advice and guidance were invaluable and very much appreciated.

Thanks also to both Yvan Beliveau for his ability to answer questions and provide insight, and to Ted Rappaport for being my committee member. You were all pleasant to work with.

The rest of the APAC team, supporting and helping me, for without you all, this project could not have been undertaken. A special thanks to , , and for all the great laughs during the long hours of this project.

And finally, I would like to thank Mother and Dad, Grandmother, my brother and the rest of my family, who have helped me out and kept me going on track. But last and most certainly most, thanks to my caring and patient wife , who has made numerous sacrifices to help me along, for with her help, this undertaking was possible.

TABLE OF CONTENTS

CHAPTER 4 PHASE MEASUREMENT SYSTEM

CHAPTER 5 HARDWARE SPECIFICATIONS AND CONSIDERATIONS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 THE NEED FOR POSITIONING SYSTEMS

A position location system can be defined as a system which can determine the coordinates of an object within a local volume or area. The ability to determine position has long been necessary for a wide range of applications. These applications include navigation, surveying, industrial control and construction automation.

There are currently several methods used for position location. A review of these methods is provided in Chapter 2. Some generalizations of these current methods can be made. Both high accuracy (order of centimeter) and speed (real-time) of measurements are generally difficult to achieve simultaneously. This is the primary motivation for the area of research presented in this thesis: the development of a real-time position location system which achieves centimeter accuracy.

1.2 AN INTRODUCTION TO APAC

The Automated Position and Control (APAC) project is a joint effort between the Civil and Electrical Engineering departments at Virginia Polytechnic Institute and State University (V.P.I. and S.U.) and Bechtel Eastern Power Corporation. APAC is funded by the National Science Foundation (N.S.F.) under grant number DMC-8717476. The project is in response to a construction automation initiative by the N.S.F.

The objective of the APAC project is to develop a position location system capable of centimeter accuracy on a real-time basis. A method using electromagnetic waves known as hyperbolic radio navigation is investigated in an experimental prototype system. The primary purpose of the project is to allow potential autonomous vehicle operation and better position data distribution to site personnel.

1.3 APPLICATION OF REAL TIME POSITION LOCATION SYSTEM: APAC

An accurate real-time position location system has numerous applications. Construction and earth moving techniques used presently require constant knowledge of position. A brief look into the construction site of the future reveals some exciting developments. The rapidly developing computer aided design (CAD) technologies can be integrated with Automated Positioning and Control (APAC).

One of the useful applications of APAC would be to enhance performance of earth moving equipment in a construction site, and to eventually automate the earth moving process. Other uses of APAC could be for remote control of equipment in an environment that is contaminated, hostile, or otherwise not safe for humans. The most basic system requirement of APAC is that it must provide real-time information.

1.4 THESIS SCOPE AND OBJECTIVES

The scope of this thesis is to provide a blueprint for future development of a hyperbolic navigational system. The APAC project and the subsystems developed specifically for collecting data for this research are discussed. A system approach is assumed in the description of the APAC system.

The objective of this thesis is to relay first-hand experience in the development of a prototype position location system. Insight is provided into the design and development of hardware for an APAC system. Included is information pertaining to the design and specification of phase measurement systems, receivers, transmitters, and a communication link for a hyperbolic position location system.

1.5 THESIS ORGANIZATION

Chapter 1 consists of an introduction to this thesis. The scope, objectives, and overview of the research project are provided.

Chapter 2 serves as a brief introduction to the position location systems used today. Following is a justification for the technique chosen for the APAC position location system. The theory of operation of the APAC system and basic definition of terms are presented.

Chapter 3 provides a description of the APAC prototype system developed for data collection purposes. A set of mathematical equations necessary to compute position from differential phase measurements are presented.

A 10.7 MHz phase detection system design and analysis is the topic of Chapter 4 which includes schematics, algorithms and software for the phase detection system. Sufficient detail is provided to serve as a guide if such a system is built in the future. The performance characteristics including accuracy, resolution and measurement time are also given.

Chapter 5 comments on position location hardware considerations and potential problems that must be overcome to realize a practical operating system. Some of the considerations included are equipment stability and calibration methods.

Chapter 6 outlines the communication links and methods necessary for the APAC system. There are two main areas discussed. The first area is specifically for the position location system. This includes establishment of a phase reference at, and data transfer to remote locations. The second area considers a typical CPU to CAD control application. A communications link necessary to provide an operator with interactive information on a construction site is specified.

Chapter 7 presents experimental results on measurements from the prototype position location system.

Chapter 8 summarizes and concludes the discussions. Some comments and suggestions are given towards areas of future research on position location systems.

CHAPTER 2

REVIEW OF THEORY ON POSITION SYSTEMS

2.1 SPECIFICATION OF POSITION LOCATION SYSTEM

A grader or bulldozer can become more productive if real-time position information about its environment is continuously provided. Current practice does not provide real-time position updates. A system of levels, transits, electronic distance measurement (EDM) devices and lines are the current methods of position information delivery to the equipment operator. This inherently produces lower than desirable productivity.

The immediate goal of the position location system used for APAC is to provide accurate real-time data to the equipment operator. The position location accuracy requirement is on the order of 1 centimeter. Position updates of about three per second constitute real-time operation for this application.

2.2 SOME POSITION LOCATION SYSTEMS CONSIDERED FOR APAC

There are several potential methods of position location worth considering for our application. Transits, tape measures, and electronic distance measurement (EDM) devices are all commonly used for position location. Yet these methods do not allow real-time position determination and require considerable user interaction. A reasonable assumption is that there are serious technical difficulties in converting these systems to real-time operation.

A common application of a position location system is found in navigation. One such navigation system is the Global Positioning System (GPS). GPS is a centralized satellite based coordinated measurement system [34]. In a demonstration of GPS performance [1], centimeter accuracy was achieved but several hours was required to resolve position to this accuracy. There have also been more recent developments in GPS. According to [15], real-time position location at desired accuracies are theoretically achievable. The APAC position location system is an alternate approach seeking similar accuracies on a localized system that is independent of GPS.

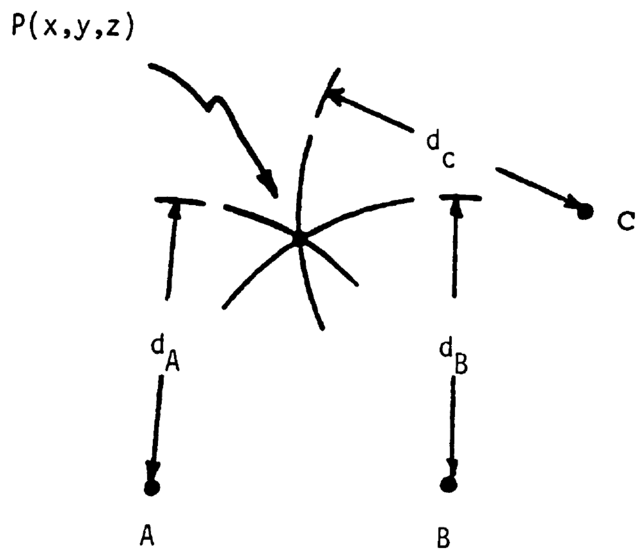
2.3 PULSE TIMING SYSTEMS FOR POSITION LOCATION

Most generally, if three distance measurements are made from three known points, triangulation can be used to determine position relative to the known points. This concept of triangulation is depicted in Figure 2.1. Triangulation requires determination of a distance. One can consider the problem of determining a distance electronically. Determination of a distance electronically must entail the use of a wave or pulse being launched through a medium (ie. atmosphere). The time to travel through the medium is then measured and converted to a distance assuming the speed of the wave is known.

Suppose that a pulse is launched for the purpose of measuring a distance. The pulse width requirement for 1 centimeter resolution is equal to $0.01m/3x10^{8}$ m/s, or about 33 ps. The bandwidth requirement for resolving this pulse about 15 GHz. It is obvious that for pulse operation, bandwidths associated with centimeter resolution are sufficiently large that other alternatives must be considered.

2.4 ACOUSTIC METHODS FOR POSITION LOCATION

When considering an acoustical wave some serious problems exist. The speed of sound in air is quite dependent on factors such as temperature, humidity, and weather, hence inducing errors unless continuously corrected. Furthermore, a typical measurement area would require a wave travel



NOTES: A,B,C are known position.

Distance from A,B,C to P(x,y,z) are known (d_A,d_B,d_C).

d_A,d_B,d_C each define a sphere about A,B,C respectively.

P(x,y,z) computed at interesction of spheres.

Figure 2.1 Concept of Triangulation in 3-D Space.

distances of several kilometers. The sound wave will have been attenuated far too much in that distance for practical operation.

2.5 JUSTIFICATION FOR HYPERBOLIC SYSTEM FOR POSITIONING

Methods using radio waves seem to be the most promising. One can consider a continuous wave transmitter and multiple receivers. Assume that the phase of the wave arriving at any receiver with respect to the other receivers can be accurately determined. Furthermore, assume that relative distances can be computed from measured phases. Some advantages of this system now become apparent.

Bandwidth of a continuous wave system can be reduced to a few kHz compared to GHz as in the pulse timing system described above. This implies that the minimum detectable signal is reduced compared to the pulse timing system. Transmitter power is reduced and hence increases the probability that a F.C.C. license can be granted. These reasons are a compelling case for development of the hyperbolic navigational system described in more detail in the following section.

2.6 GENERAL DESCRIPTION OF HYPERBOLIC NAVIGATION SYSTEMS

Hyperbolic navigational systems have been used to locate ships and aircraft globally for many years. To understand how a hyperbolic system works, consider two points (receivers and antennas) located as shown in Figure 2.2. Suppose that a transmitting beacon is placed somewhere between these two points. The differential distance between the transmitter and receiver results in constant phase lines which are hyperbolic in shape. Since the phase at the beacon is not known, the only phase information that can be measured is the differential phase between any two receivers. Hence, the system operates on differential quantities of phase and distance rather than absolute quantities.

From Figure 2.2, note that the baseline is defined as an imaginary, straight line drawn between the two receivers. Noting the periodicity on this baseline between the two receiving antennas, there are lines of constant phase separated by one half wavelength. The half wavelength separation of lines of constant phase is only on the baseline. Further away from the baseline this distance increases. The area between two adjacent hyperbolas with a differential phase of 0 degrees is defined as a "lane" [31].

An important result of these lanes is the ambiguities that occur. If the length of the baseline exceeds one half of a wavelength, ambiguities arise since the differential phase

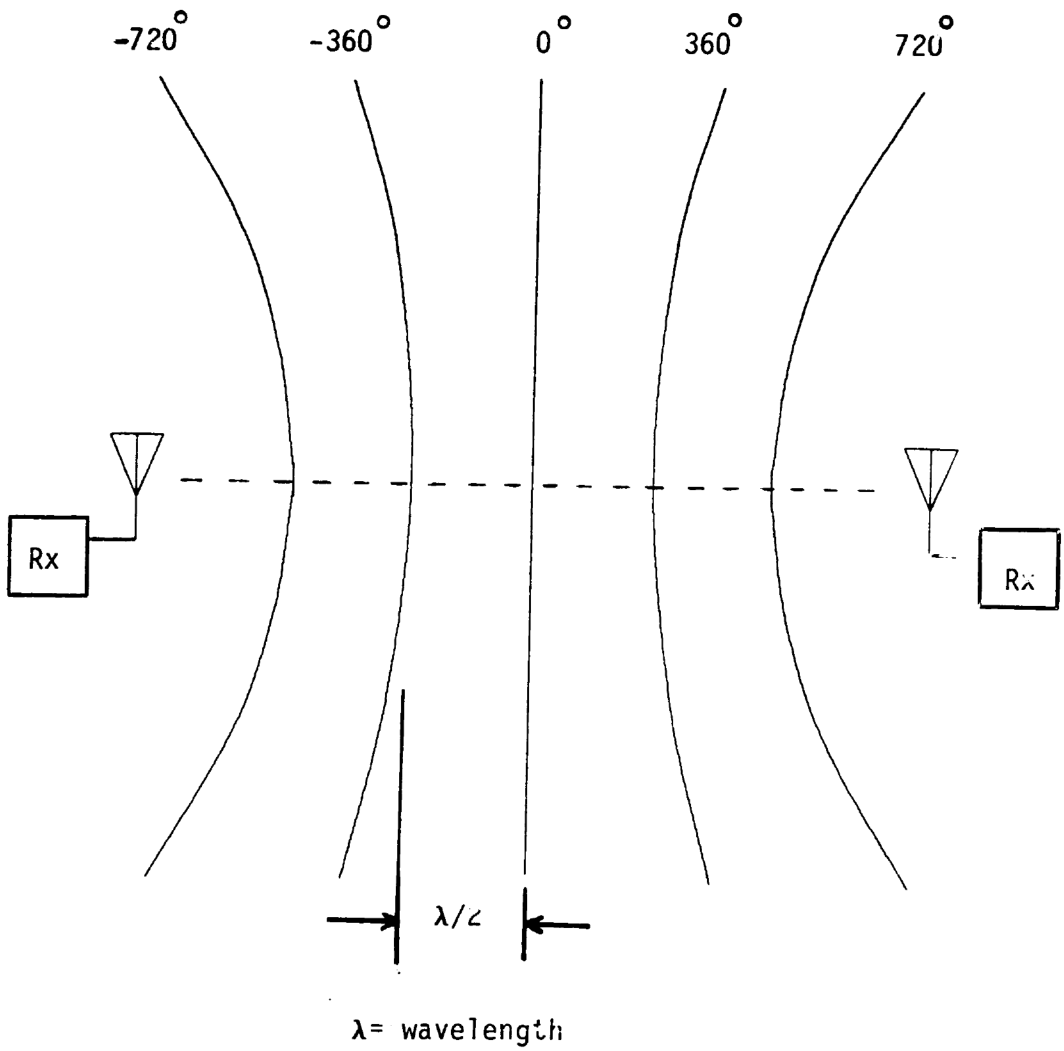


Figure 2.2 Hyperbolas of Constant Phase.

is periodic. Ambiguity resolution generally requires transmission of multiple wavelengths by the beacon and is a topic discussed in Chapter 3.

The hyperbolic system described above can be configured to solve for locations in two and three dimensions by utilizing three and four receivers respectively. The DECCA hyperbolic navigational system described by [31] uses multiple transmitters and one receiver located at the measurement point to determine differential phase for computing the location. Figure 2.3 depicts a two dimensional hyperbolic navigational system. The system developed for APAC differs from the DECCA system by employing a single transmitter and multiple receivers.

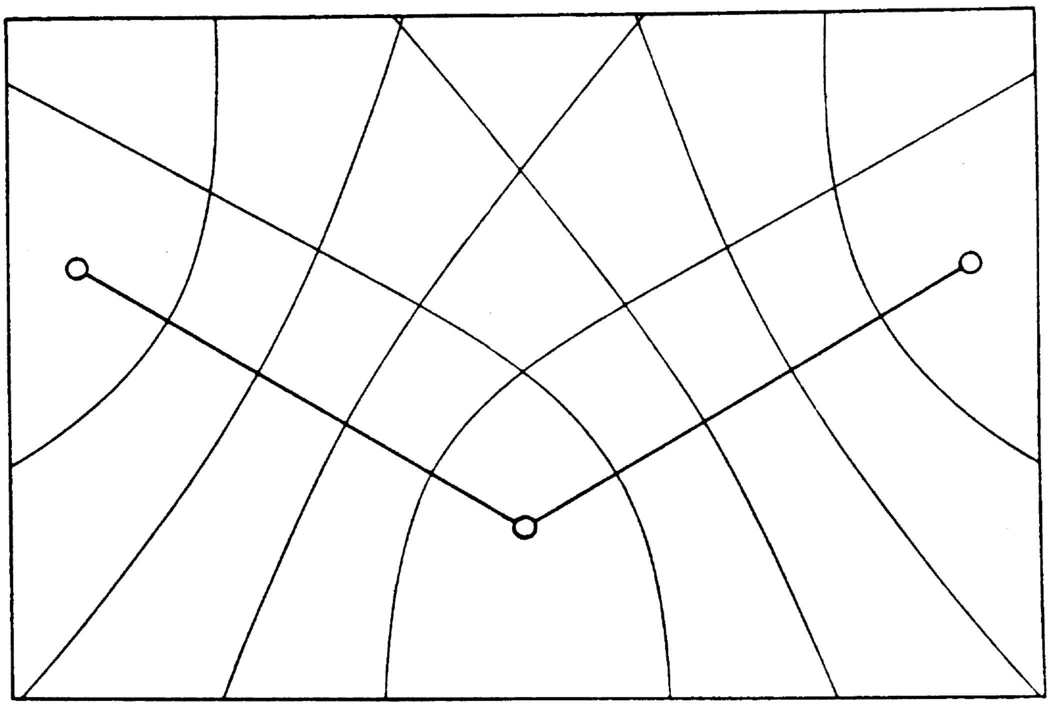


Figure 2.3 Hyperbolas of constant phase in a 2-D system.

Reprinted from [31]. The three points shown are receiver antenna locations and form hyperbolas of constant phase. Therefore, the system is capable of resolving positions in two dimensions.

CHAPTER 3

THE APAC PROTOTYPE POSITION LOCATION SYSTEM

3.1 GENERAL DESCRIPTION OF HARDWARE

A minimum of 4 receivers is necessary to determine positions in 3 dimensional space using the hyperbolic method. This most basic system was initially developed for the APAC prototype system.

The basic Hyperbolic system developed at V.P.I. and S.U. is shown in Figures 3.1 and 3.2 which depict both the electrical and physical layout. The CPU commands the beacon to generate and radiate an unmodulated signal at a given frequency. Each of the receivers are also commanded to receive at the same frequency. Supplying the same local oscillator to each of the receivers synchronizes the IF outputs. This IF signal is sent to the phase detectors to supply differential phase information to the CPU.

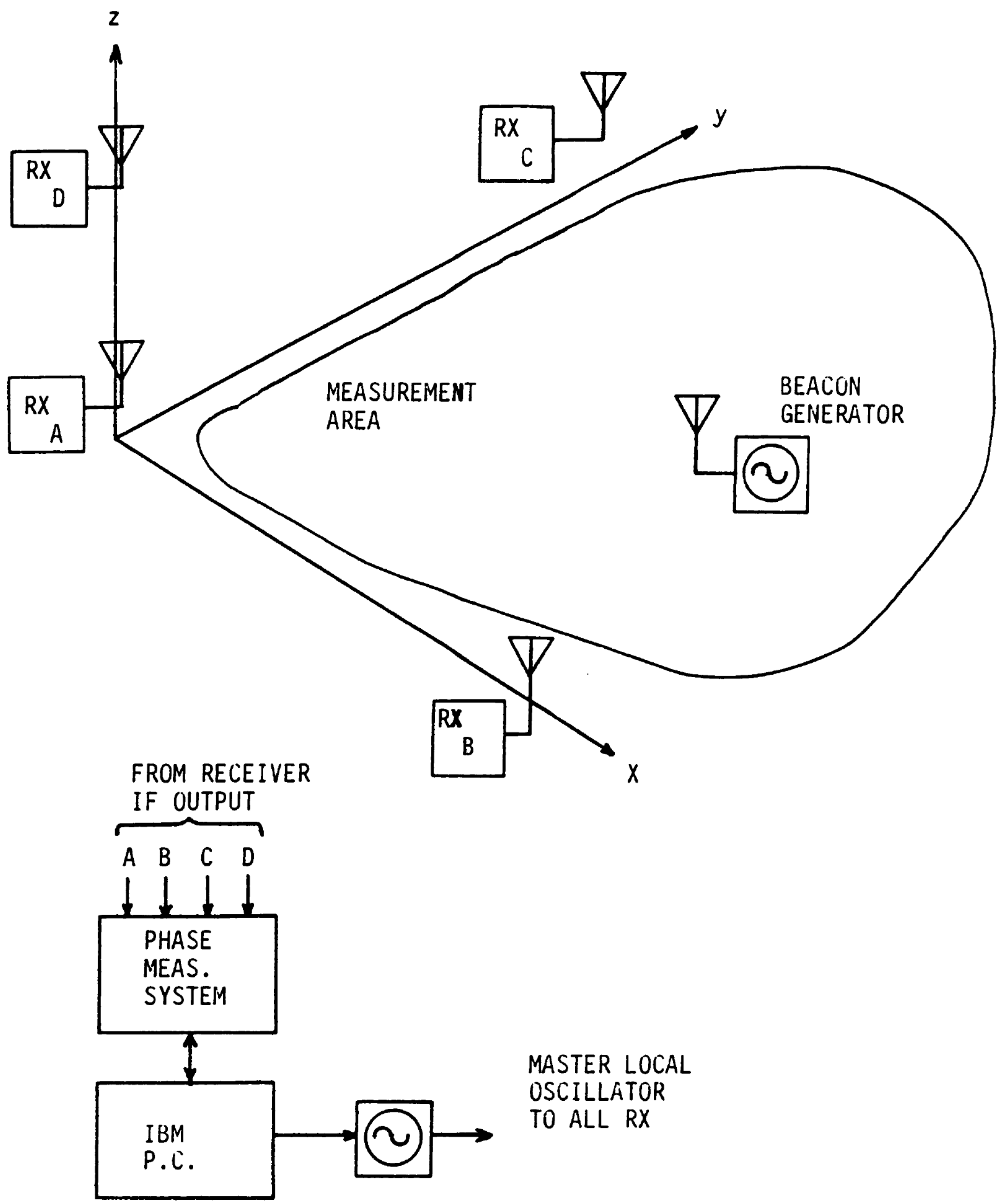


Figure 3.1 Physical Layout of the APAC Measurement Area.

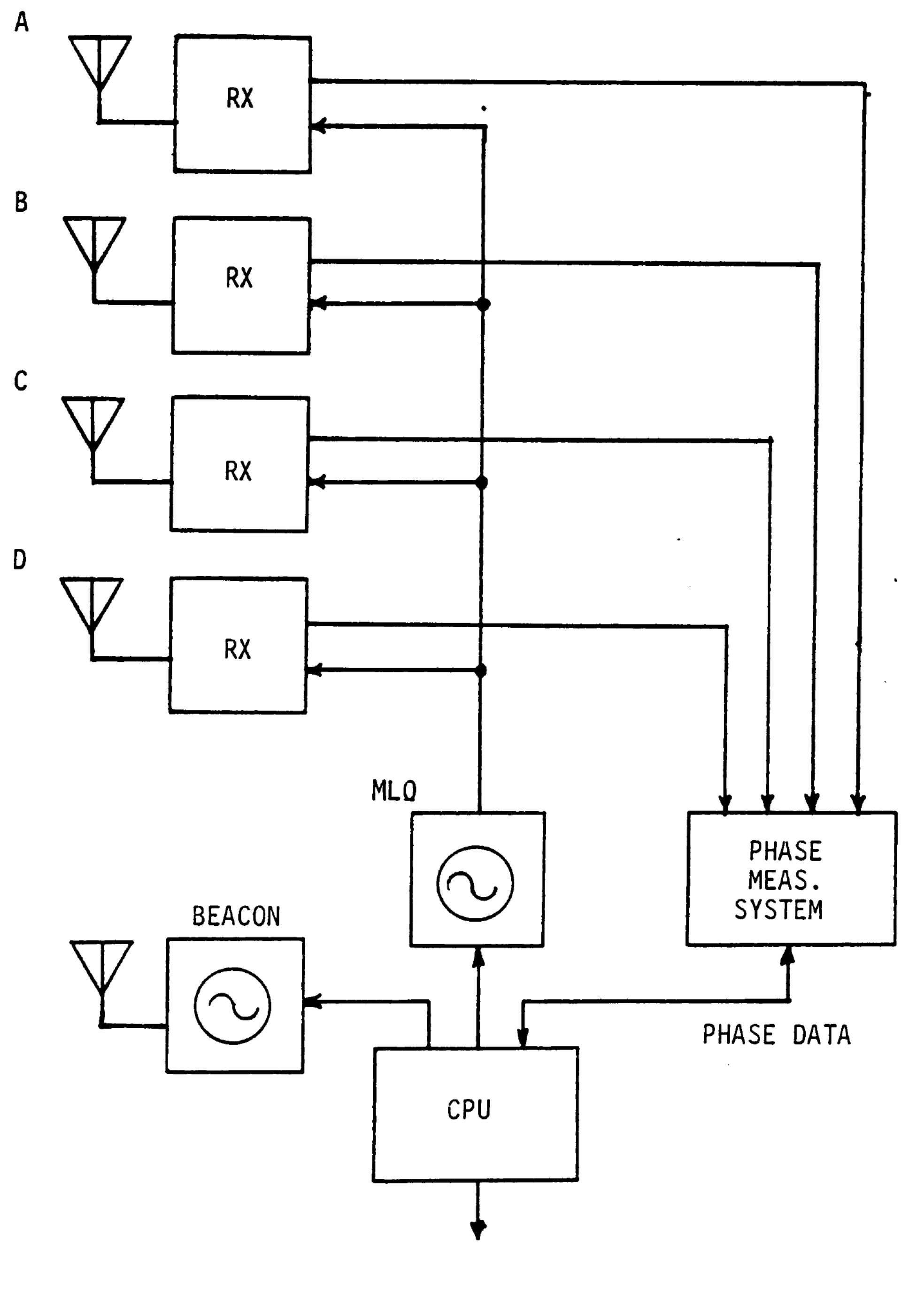


Figure 3.2 Position Location System Electrical Diagram.

One of the antennas (say, antenna A) is designated as a reference antenna. The signal applied to the phase detectors is then measured relative to antenna A and represents the electromagnetic field arriving at the phase center of each of the antennas. Since the receivers and phase measurement system are all at a fixed location, any differences in the cable lengths are calibrated out as a constant phase offset.

All of the receiver and beacon hardware used in the project was available commercially and required limited modification. The system operated in the UHF band at discrete frequencies between 400 and 450 MHz. Frequency switching was employed to measure differential phase at each frequency serially. In comparison to a parallel measurement system, frequency switching is more efficient in terms of hardware.

Internally, the local oscillator in the receivers was disconnected. A microprocessor controlled PLL was substituted and phase locked to the master local oscillator (MLO). The beacon generator was also microprocessor controlled. This maximizes the system flexibility and permits the frequency of operation to be remotely controlled over a wide range of frequencies.

It was necessary to develop a phase measurement system for rapid measurement of the differential phase signals at 10.7 MHz (see Chapter 4). Emitter Coupled Logic mixers were employed using in-phase and quadrature techniques that allowed phase measurements at approximately 50 millisecond intervals.

3.2 COMPUTATION OF LOCATION

To understand the computation of location from phase measurements, consider 4 known fixed points A,B,C and D as shown in Figure 3.3. Furthermore assume that they do not exist in the same plane. Let P(x,y,z) be an arbitrarily unknown point to be located in the measurement area. Defining vectors $\vec{a},\vec{b},\vec{d},\vec{c}$ and $\vec{r}$ as shown, the differential difference between A and B can be represented as:

$$\Delta r_{ab} = \left|\vec{r} - \vec{a}\right| - \left|\vec{r} - \vec{b}\right|$$

For the case of A located on the origin (eg. $\vec{a} = \vec{0}$) and considering C and D also we have:

$$\Delta r_{ab} = \left|\vec{r}\right| - \left|\vec{r} - \vec{b}\right| \tag{3.1}$$

$$\Delta r_{ac} = \left|\vec{r}\right| - \left|\vec{r} - \vec{c}\right| \tag{3.2}$$

$$\Delta r_{ad} = \left|\vec{r}\right| - \left|\vec{r} - \vec{d}\right| \tag{3.3}$$

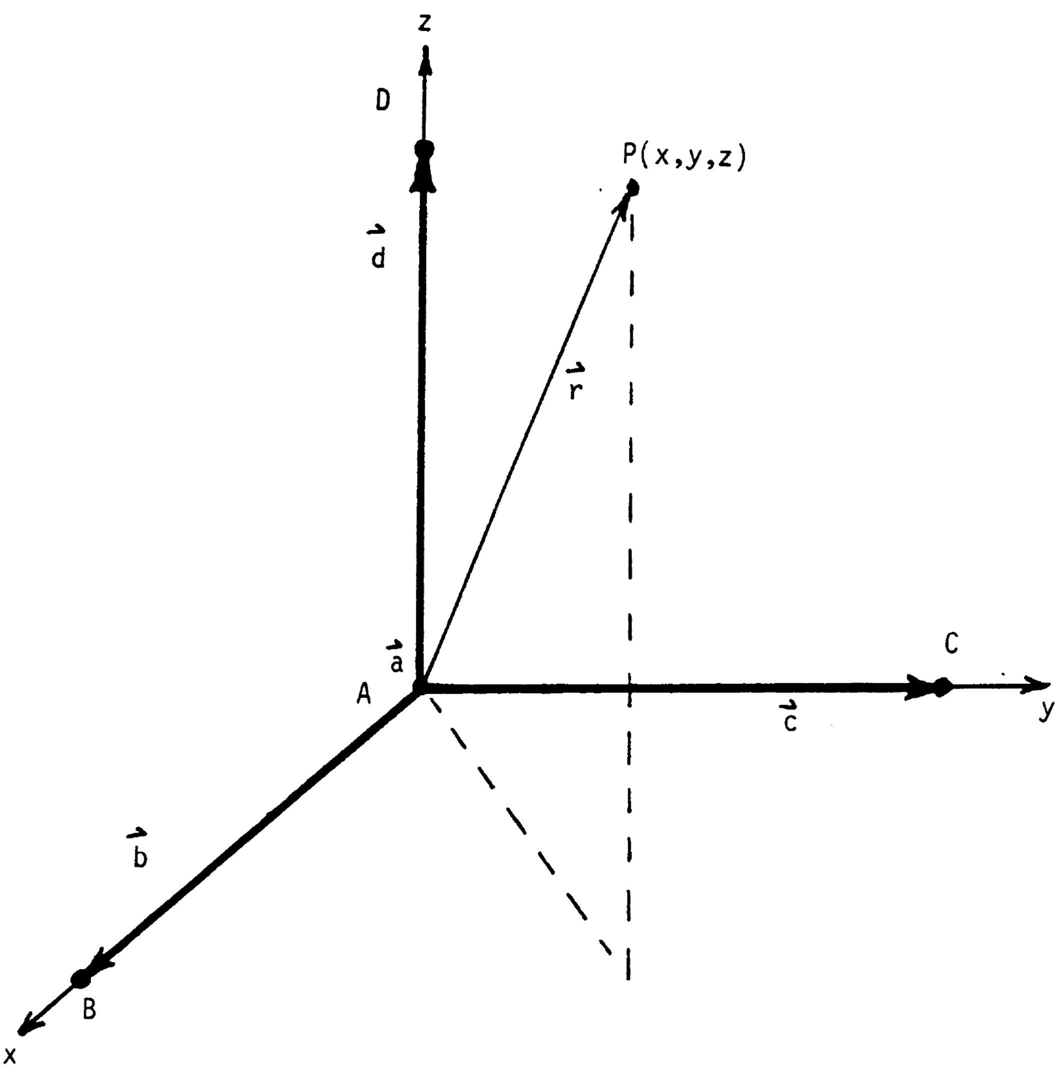


Figure 3.3 Points in the Measurement Area.

These differential distances can be equated to the corresponding phase measurements as follows:

$$\Delta r_{ab} = \Phi_{ab} \lambda/360 \text{ degrees} \tag{3.4}$$

$$\Delta r_{ac} = \Phi_{ac} \lambda/360 \text{ degrees} \tag{3.5}$$

$$\Delta r_{ad} = \Phi_{ad} \lambda/360 \text{ degrees} \tag{3.6}$$

where: Φ_{ab} = phase at B with respect to A (degrees)

Φ_{ac} = phase at C with respect to A (degrees)

Φ_{ad} = phase at D with respect to A (degrees)

Since A,B,C and D represent the four receiving antennas and the beacon antenna locations, it follows that there are three equations and three unknowns, so the equations must be solved simultaneously. Transmitting at a single frequency restricts the unambiguous measurable locations to roughly within a cube shaped volume with sides not exceeding approximately a half wavelength. The significance of the half wavelength is that it is the maximum distance on the baseline in which the phase is unambiguous.

If one assumes some phase shift resolution (due to measurement resolution) and some phase error (due to say, multipath) it becomes obvious that higher frequencies are required to resolve finer position. Conversely, lower frequencies are required to resolve coarse position, because of the phase ambiguity which occurs at the shorter wavelength.

For the basic developmental system the base frequency of operation chosen was 450 MHz. This corresponds to a wavelength of 0.67 meters. Obviously, using a single frequency such as this results in a very small unambiguous measurement area with about 0.33 meters along the baseline. To resolve larger measurement areas an additional frequency of 400 MHz is transmitted. Using a technique referred to as phase and frequency differencing, subtraction of the two frequencies yields 50 MHz and the phase differences at this lower frequency can be obtained [30]. A course position estimate at a wavelength of 6 meters results, and the first order of ambiguity resolution is achieved. The first order of ambiguity resolution is used to locate the beacon in one of the lanes from which fine positioning is computed using the higher frequency-phase data.

The procedure of ambiguity resolution described above generalizes to n^{th} orders by measuring at frequencies of 430 MHz, 440 MHz, 445 MHz, 448 MHz, 449.5 MHz etc, such that each lower difference frequency is used to locate the beacon in a larger lane. The difference frequency wavelength required to resolve the ambiguity determines the size of each frequency step. Recall that from Section 2.6, position is unambiguous in lengths of one half wavelength (each lane).

CHAPTER 4

PHASE MEASUREMENT SYSTEM

4.1 MOTIVATION FOR THE PHASE MEASUREMENT SYSTEM DEVELOPMENT

The APAC system accomplishes position measurements by first measuring differential phases of signals arriving at the corresponding antennas. Measurements at the RF frequency of 450 MHz are not practical, thus it is necessary to mix down to some IF frequency, chosen to be 10.7 MHz, where hardware is more readily available. A significant amount of time has been invested in developing the 10.7 MHz phase measurement system. This chapter documents the design and theory of operation for reference in future work.

4.2 COMPARISON OF METHODS FOR PHASE MEASUREMENTS

4.2.1 PHASE DETECTOR METHODS

There are several acceptable and widely used methods of phase detection. Three Phase Detectors (PD) were evaluated for merit in this application.

4.2.1.1 ANALOG MULTIPLIER WITH SINUSOIDAL OUTPUT

A PD comprised of a multiplier, limiter, band-pass filter and low-pass filter is depicted in Figure 4.1.a. According to [14] the limiter and bandpass filter are essential in most practical designs. The typical characteristic output response is shown in Figure 4.1.b.

An application using the sinusoidal output analog version is shown in the block diagram of Figure 4.2. Assuming ideal limiters, the inputs to the multipliers have constant amplitude A volts. Upon analysis, we see that the output of the in-phase multiplier is $A^2 \sin(w_o t)\sin(w_o t+\Phi) = A^2/2\{\sin(2w_o t + \text{constants}) + \sin(\Phi)\}$. The low pass filter removes the $2w_o$ term leaving only the $\sin(\Phi)$ DC term. Likewise, considering the 90 degree offset, the quadrature output is $A^2/2\sin(\Phi + 90°)$. If we were to perform an inverse sine function on only one of the outputs, it is obvious that ambiguities would arise in the 360 degree period. The responses of the in-phase and quadrature outputs are shown in Figure 4.3. The phase between the applied signals can then be computed by noting:

$$\frac{V(\text{in-phase})}{V(\text{quad})} = \frac{A^2/2\sin(\Phi)}{A^2/2\sin(\Phi + 90°)} \qquad (4.1)$$

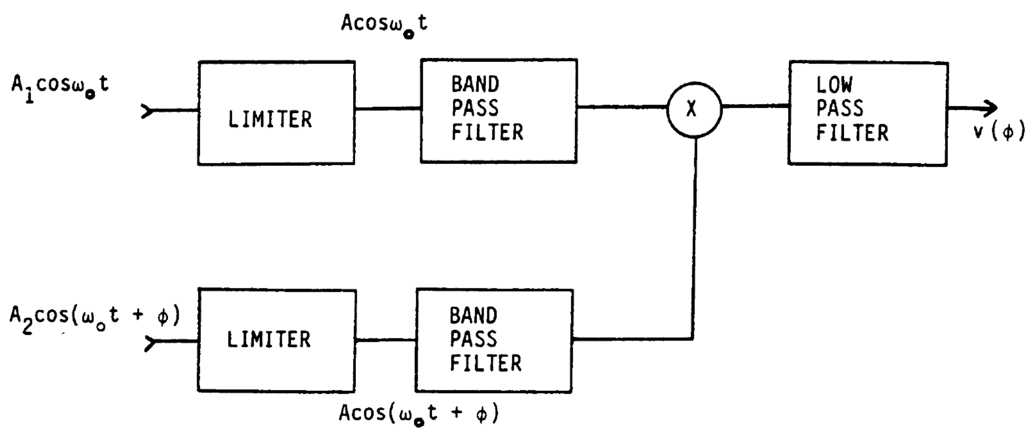


Figure 4.1.a Analog Multiplier PD.

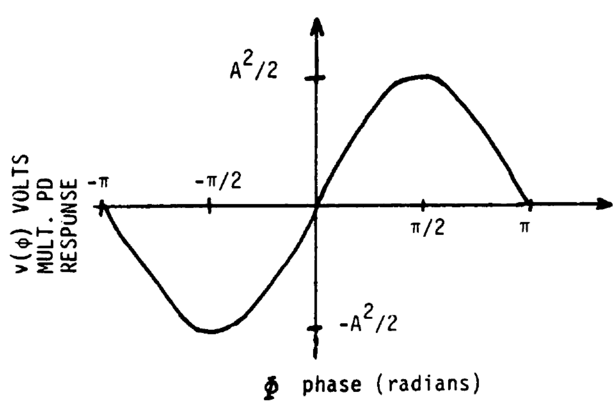


Figure 4.1.b Response of Multiplier PD.

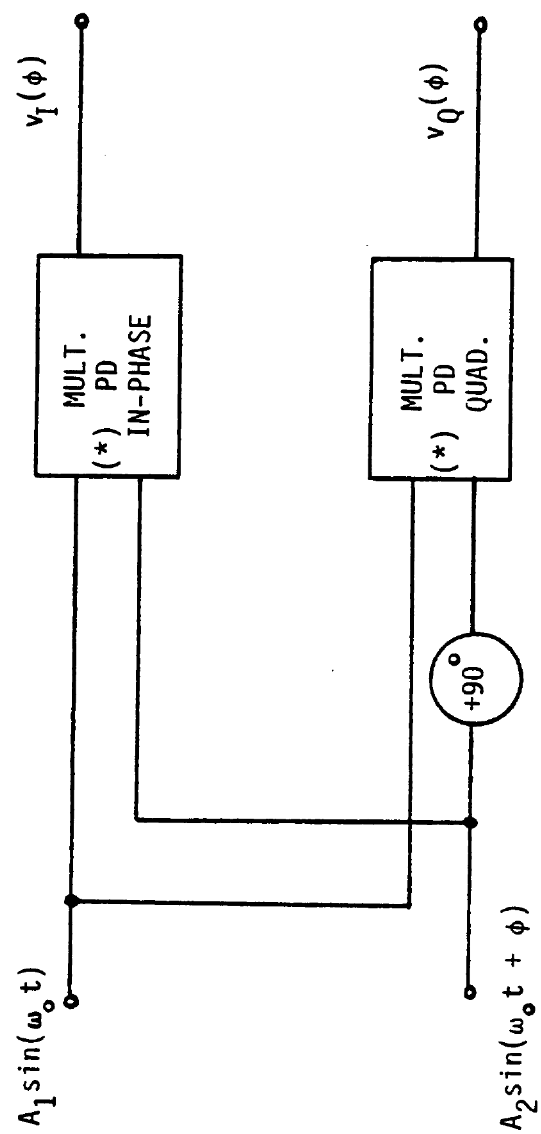


Figure 4.2 Phase Measurement Using the Analog Multiplier.

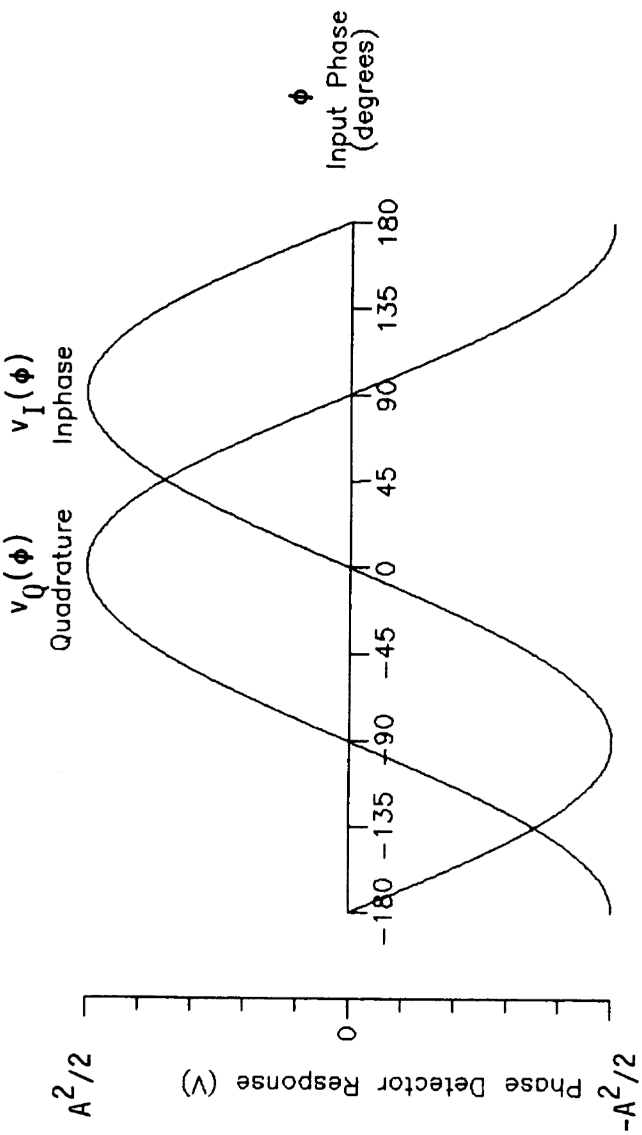


Figure 4.3 Response of Phase Measurent System Shown in Figure 4.2 Circuit.

$$= \tan(\Phi)$$

and computing the inverse tangent. The phase can also be found by using the sign of each of the outputs to resolve the quadrant and computing the inverse sine function. More detail on the validity of each computational method will follow in section 4.4.

4.2.1.2 EXCLUSIVE-OR WITH TRIANGLE OUTPUT

The response of a phase detector which is linear on a 180 degree interval is the equivalent of an EXCLUSIVE-OR circuit and is shown in Figure 4.4.a. The characteristic response is depicted in Figure 4.4.b. The discussion for resolving phase ambiguities in the section above also applies. The same output response is obtained using the four diode detector arrangement discussed in reference [2].

4.2.1.3 FLIP-FLOP WITH SAWTOOTH OUTPUT

Another phase detector method utilizes a J-K flip-flop (bi-stable multivibrator). A block diagram and the sawtooth response are shown in Figure 4.5.a and Figure 4.5.b respectively [2]. The input voltage is limited and differentiated. The resulting positive going pulse is applied to the flip-flop. The output is time averaged and is equal to

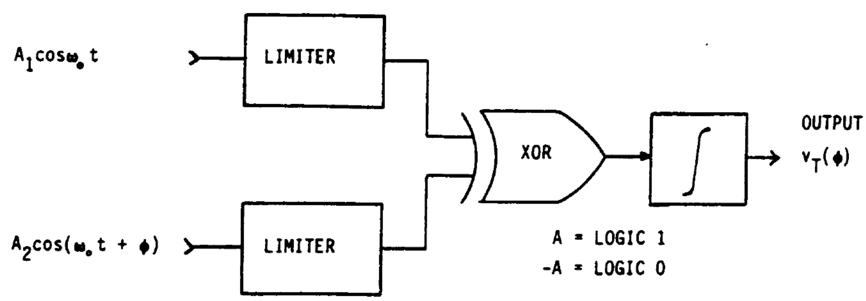


Figure 4.4.a Equivalent Circuit For Triangular Output PD.

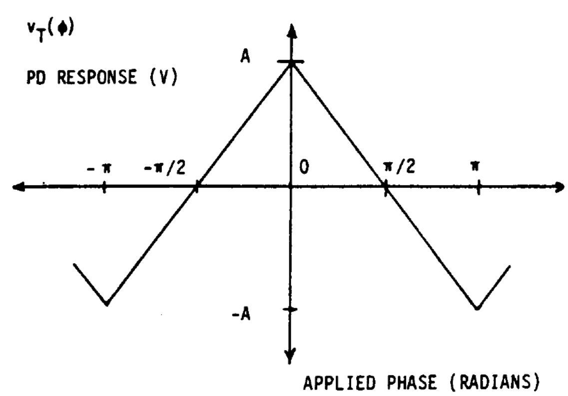


Figure 4.4.b Triangular PD Response.

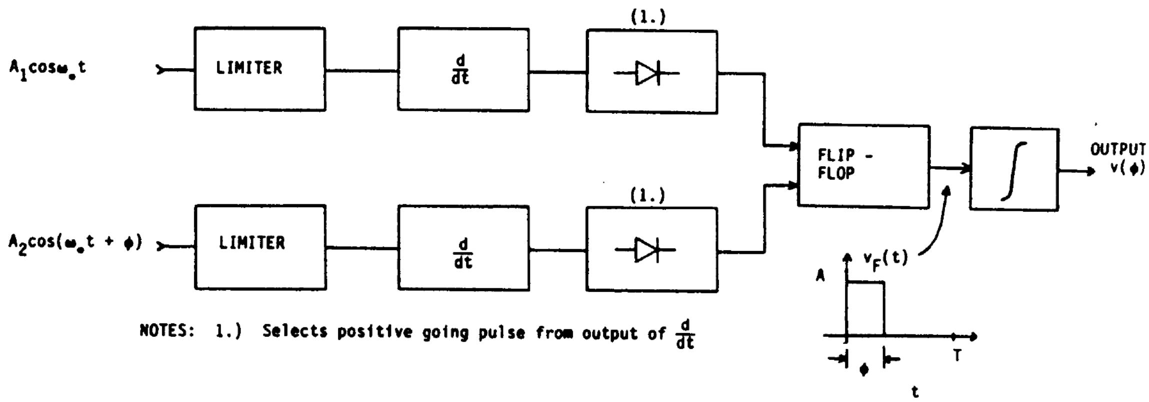


Figure 4.5.a Diagram of Flip-Flop PD.

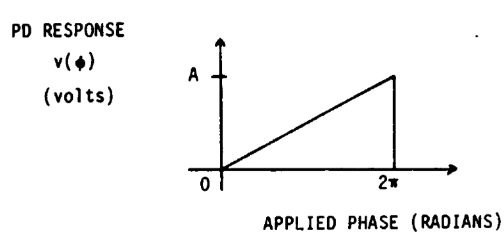


Figure 4.5.b Sawtooth Response of Flip-Flop PD.

$$v(\Phi) = \frac{1}{T_o} \int_0^{T_o} v_F(t)\,dt = \frac{A}{2\pi}\,\Phi \qquad (4.2)$$

where: A is the magnitude of the flip-flop logical 1 output.

4.2.1.4 OTHER PHASE DETECTORS

A rectangular PD response is shown in Figure 4.6 but is of little use in this application due to the inability to resolve phases less than 180 degrees.

Another method would require mixing down to a second IF frequency and sampling the signal of lower frequency. Digital signaling processing techniques could then be used to compute the phase. This more elaborate method is perfectly valid for determining phase but is significantly more demanding in terms of hardware requirements and computing time.

4.2.2 NOISE CONSIDERATIONS

The phase detector responses shown in Figure 4.1.b, 4.4.b and 4.5.b are ideal and do not account for noise. In order to determine which type of phase detector should be developed, performance in the presence of noise must be investigated.

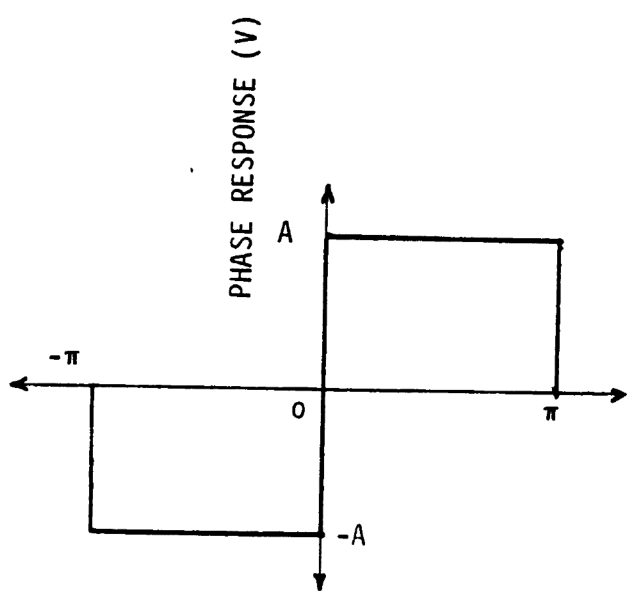


Figure 4.6 Rectangular PD Response.

Results from [25] in Figure 4.7.a through 4.7.c show PD responses with added noise. From careful observation, it can be concluded that with a lower SNR(input) all PDs under consideration tend towards a sinusoidal response. However, more importantly, when the SNR(input) equals 10, the sinusoidal PD response very closely resembles the ideal noiseless case. Both the triangular and the sawtooth PD responses exhibit serious degradation at the same SNR(input). This fact is quite obvious since in the case of the sawtooth PD, significant jitter (due to noise) near +/- 180 degrees will adversely affect the PD performance.

4.2.3 JUSTIFICATION OF DESIGN CHOICE

A practical APAC system will suffer from multipath phase errors of between 5 and 25 degrees [11]. Hence, it is reasonable to assume phase measurements within about 5 degrees as an acceptable design goal. Based on the accuracy requirements, the decision was made to develop and use the analog PD using in-phase and quadrature techniques. This choice is further supported by the noise considerations reviewed in Section 4.2.2, especially since the APAC environment may be noisy.

4.3 HARDWARE FOR THE MEASUREMENT SYSTEM

A schematic of the phase detector circuit system is shown in Figure 4.8. It should be noted that this is only half of

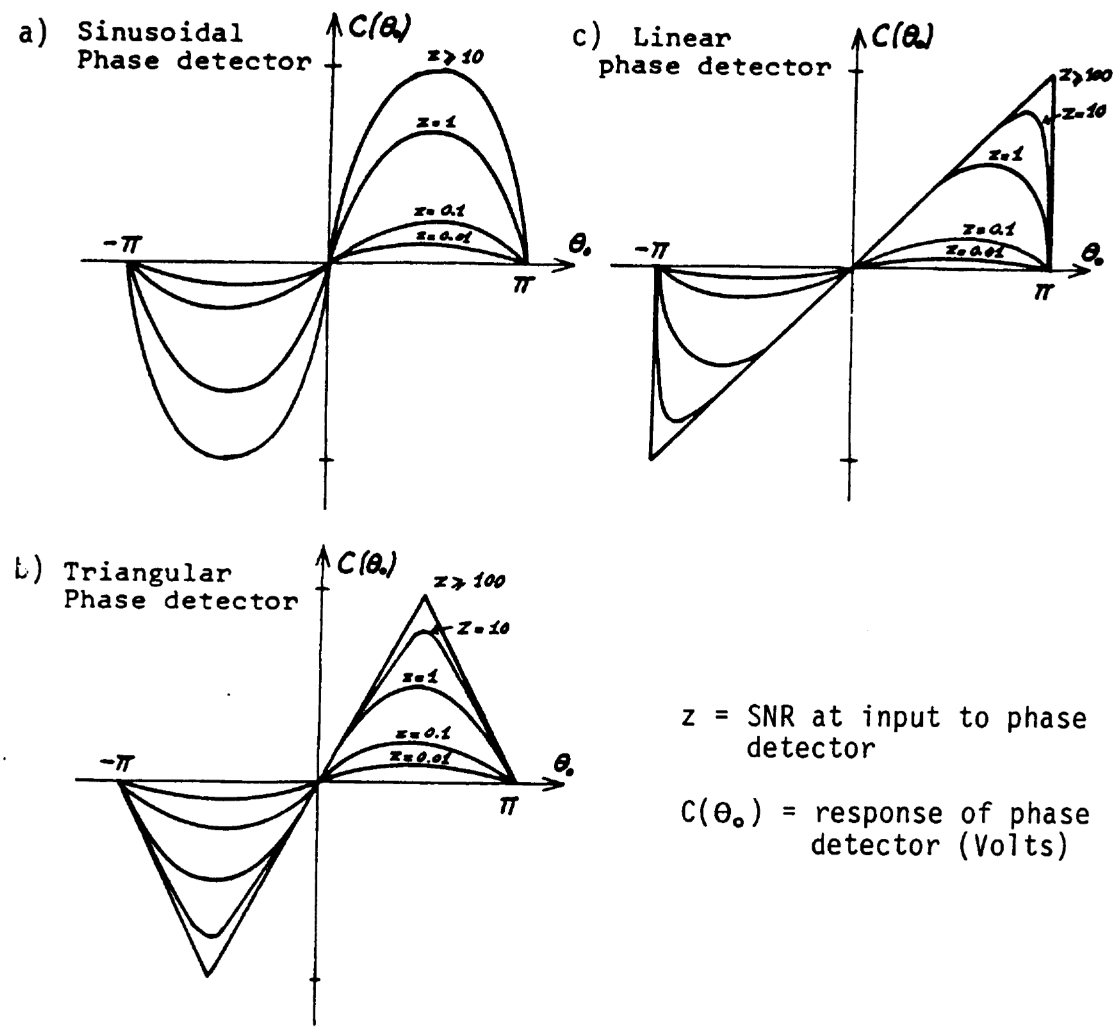


Figure 4.7 Response characteristics in the presence of noise, reprinted from [25].

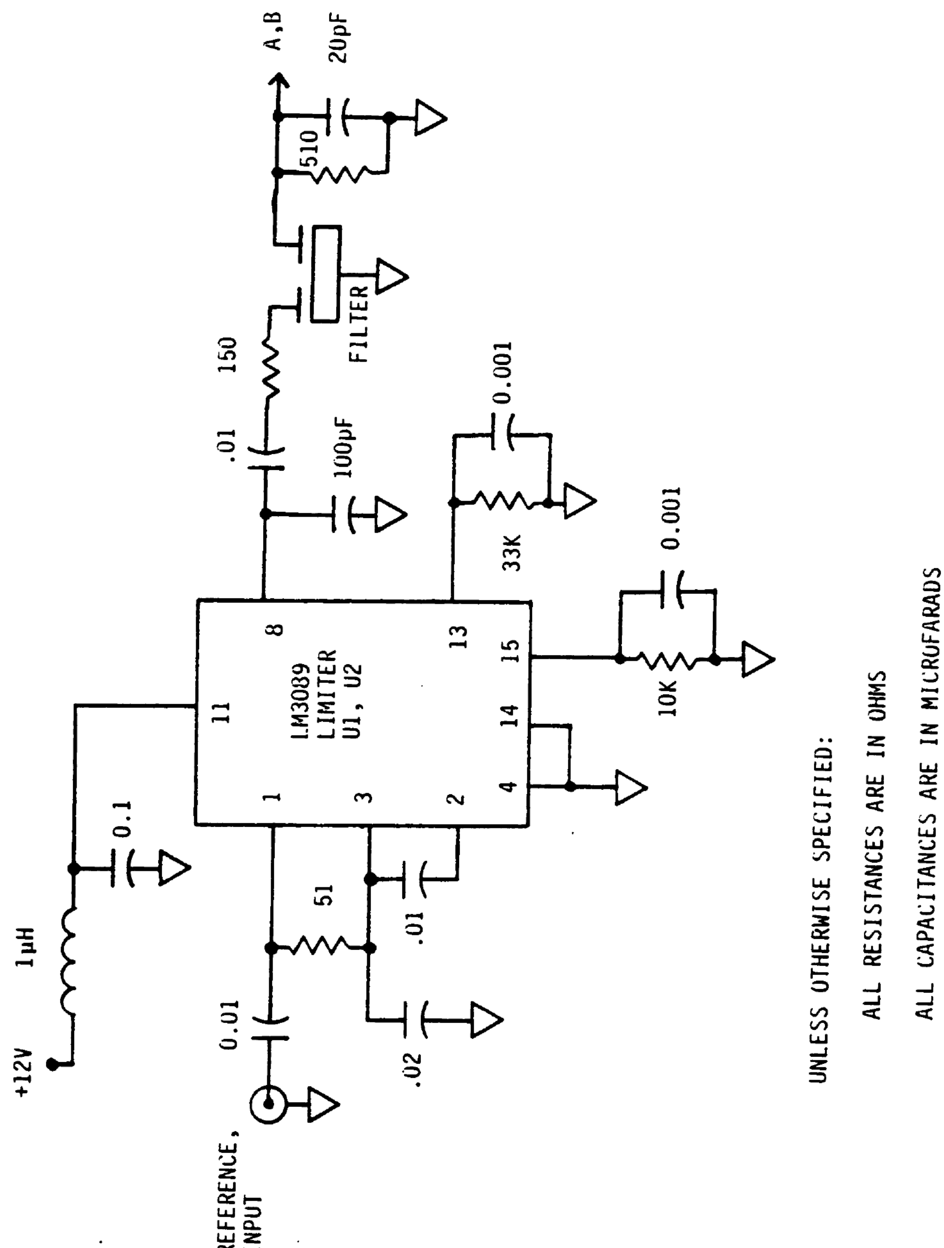


Figure 4.8 Phase Detector Circuit.

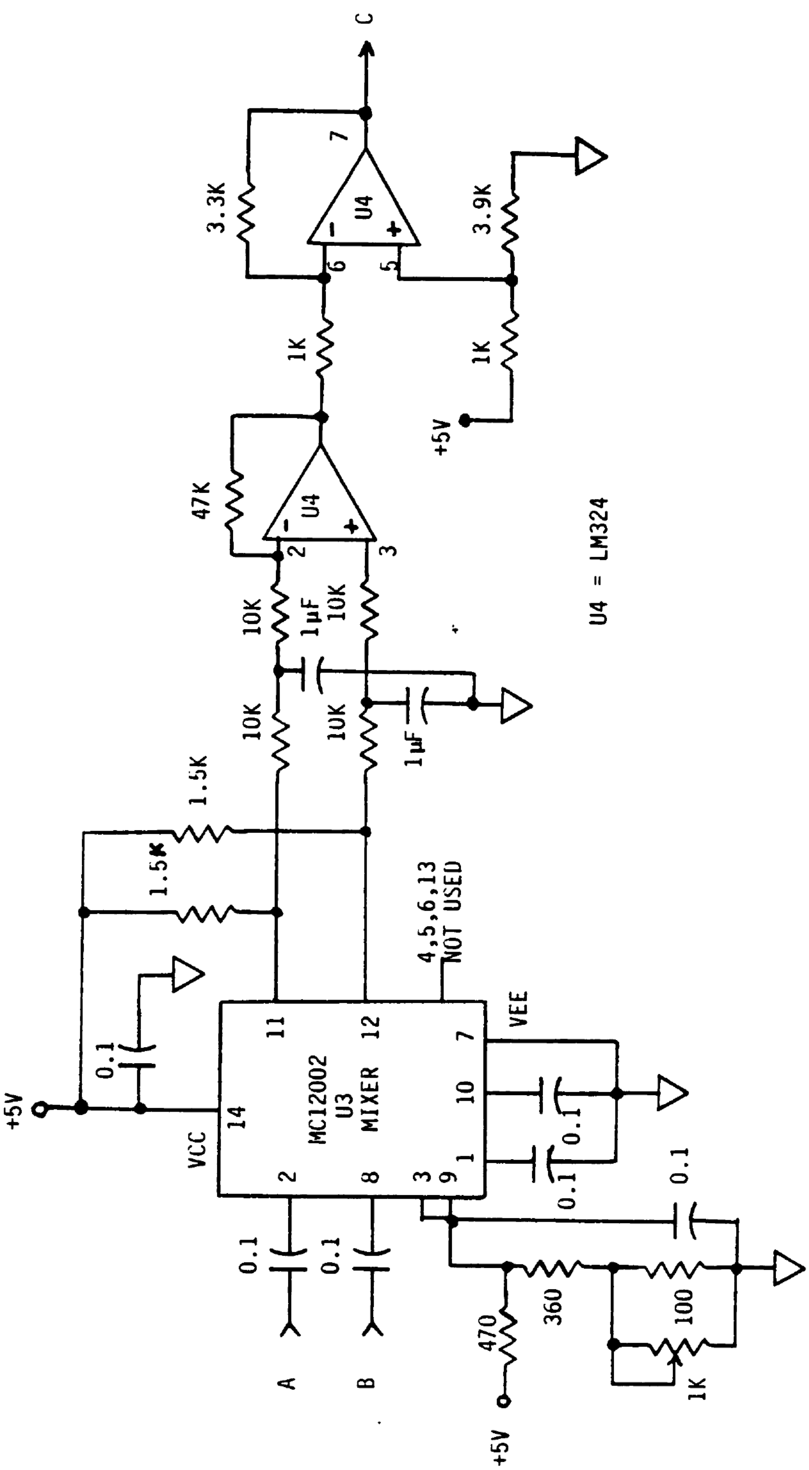


Figure 4.8 Phase Detector Circuit (continued).

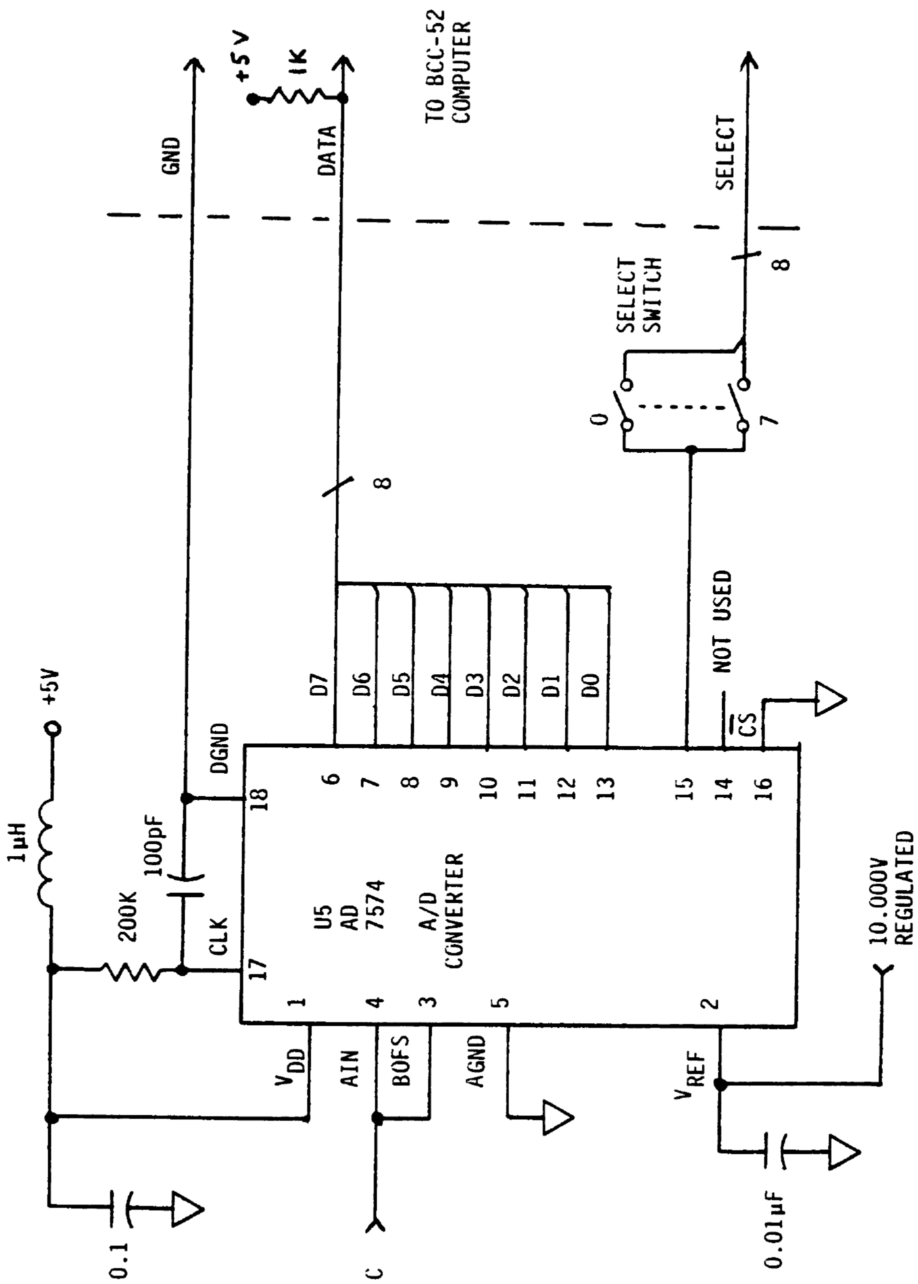


Figure 4.8 Phase Detector Circuit (continued).

a phase measurement channel, and represents either an in-phase or a quadrature circuit. Two circuits are needed along with a 90 degree phase shift for a measurement channel, as shown in Figure 4.9. The description of one circuit follows.

The purpose of the LM3089 is to provide gain and limiting. One possible alternative to the LM3089 could have been a pair of back to back diodes, however this alternative does not provide gain. Experimentally the LM3089 was found to have a constant output of 0.29 Volts for input ranges between 5 and 500 mV. Additionally, the LM3089 contains a logarithmic amplifier which could be used to sense signal strength if necessary in some future applications. The LM3089 operates well at 10.7 MHz which is the IF frequency used in the APAC system.

Following the limiter circuit is a ceramic filter with a 3 dB bandwidth of 60 kHz. This bandwidth is sufficiently wide that it should not be the limiting factor in the system. A different filter with bandwidth of a few KHz is assumed to be placed prior to the phase measurement system to minimize noise power into the phase detector. Chapter 5 discusses the system bandwidth.

The Motorola MC12002 is an Emitter Coupled Logic (ECL)

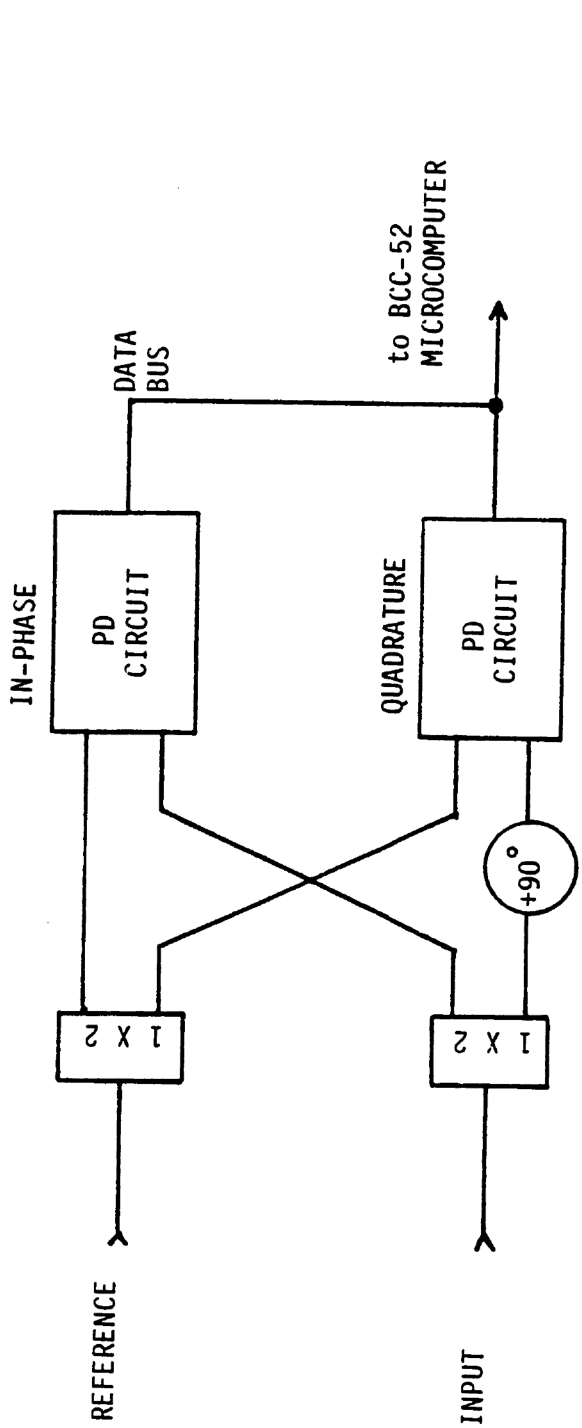


Figure 4.9 Configuration of Phase Detector Circuit Cards.

mixer, chosen for its favorable high frequency characteristics. A bias circuit provides a DC voltage level on pin 3 and 9 of the MCL12002. This is adjusted for a bias voltage of approximately 2.45 volts to establish a (2x10.7)MHz signal on the output, pins 11 and 12. The (2x10.7)MHz signal is an indication that the MC12002 is biased such that it resembles the multiplier described in Section 4.2.1.1. This adjustment criterion was experimentally developed and found to guarantee a sinusoidal DC response depending on the phase of the applied signal.

The RC network on pins 11 and 12 comprises a low pass filter and provides integration after the phase detector. The purpose of the RC network is to pass the DC component of the phase detector output which contains the phase information. Since it also acts like a low-pass filter, there is a cut-off frequency equal to $1/(2\pi RC)$. Therefore it will have the effect of filtering out rapid changes in phase, or likewise, time averaging phase jitter in the system. One of the system design considerations is to ensure that there is sufficient time allowed between frequency switches in the receiver system. The RC network feeds directly into a differential amplifier circuit.

Following the differential amplifier is an operational amplifier configured as a level shifter with gain. The function of this stage is to provide the correct input levels for the analog to digital (A/D) converter. Consequently, a voltage on the output of the differential amplifier ranging from 2.4 to 4.8 Volts is shifted to a voltage ranging between 1 and 9 Volts. The choice of 1 through 9 Volts allows some variation in circuit component values for an A/D converter with an operating range of 0 to 10 Volts, resulting in increased resolution compared to using the 2.4 to 4.8 Volt output of the differential amplifier.

The A/D converter is an 8 bit Analog Devices AD7574 and is addressable by a DIP switch connected to the data bus of the controller. The controller software is capable of supporting 8 PD circuits (4 measurement channels) depending on the DIP switch configuration. Table 4.1 shows the DIP switch settings used in the phase measurement system.

Although the choice of a controller is somewhat arbitrary, it was decided to purchase a manufactured and proven microcomputer rather than investing development time for a design exclusively for this project. The BCC-52 is a low cost microprocessor controller manufactured by Micromint Corporation and is programable both in Basic and assembly

Table 4.1 Dip Switch Configuration for PD Card.

CARD PAIR	CIRCUIT	
	IN-PHASE	QUADRATURE
0	0	1
1	2	3
2	4	5
3	6	7

Indicated switches are closed, all others are open.

languages.

4.4 COMPUTATION OF PHASE

4.4.1 ALGORITHM

As mentioned in Section 4.2, there are two methods for computing the phase from in-phase and quadrature data. The most obvious is simply using the inverse tangent function. However, careful observation will prove otherwise.

Due to quantization error, measurement accuracy becomes poor around +/- 90 degrees, where the slope of the phase detector response is near zero. This problem is overcome by switching between the in-phase and quadrature phase detectors so that only the high slope regions are used, as illustrated in Figure 4.10. As the phase Φ of the applied input signal changes, the microcomputer selects the phase detector operating in the higher slope region which results in the most resolution. Regions X and Y of the in-phase circuit and regions Q and P of the quadrature circuit are selected in software and are the usable regions for computation of phase. The discussion in the remainder of this subsection describes how the phase Φ is determined.

Measurements are made of the output phase detector circuitry by the A/D converter and are represented digitally ranging between 0 and 255. $V_i(\Phi)$ and $V_q(\Phi)$ are

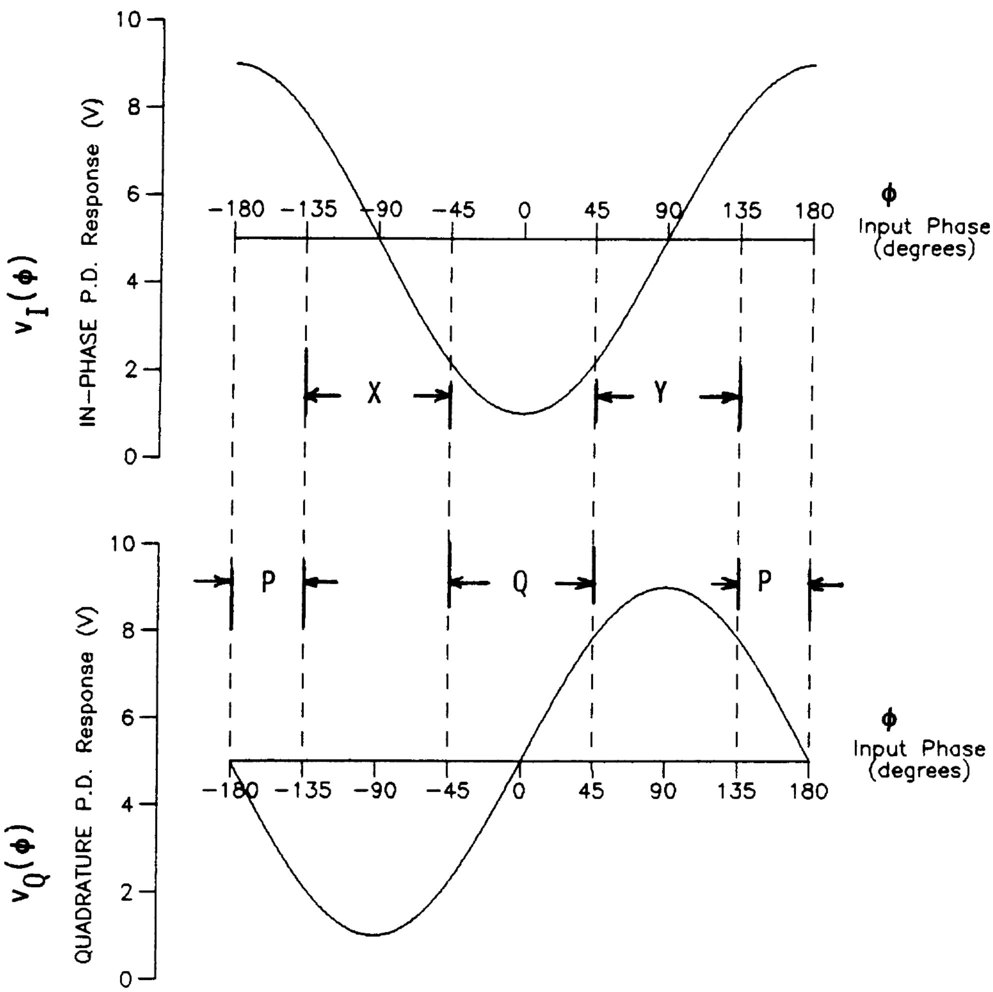


Figure 4.10 Response of In-phase and Quadrature PDs.

defined as the in-phase and quadrature voltage levels respectively. The measured values of $V_i(\Phi)$ and $V_q(\Phi)$ are normalized as follows:

$$V_{in}(\Phi) = 256 \frac{V_i(\Phi) - V_{imin}}{V_{imax}} - V_{imin} \qquad (4.3)$$

$$V_{qn}(\Phi) = 256 \frac{V_q(\Phi) - V_{qmin}}{V_{qmax}} - V_{qmin} \qquad (4.4)$$

Where V_{imin}, V_{imax}, V_{qmin} and V_{qmax} are the calibration values define in Section 4.4.2. The inverse sine of $V_{qn}(\Phi)$ and $V_{in}(\Phi)$ are computed from a look-up table in software.

Phase offsets defined in Section 4.4.2 are added as a calibration correction factor. See Section 4.4.2 for a definition of the calibration correction factors. The phase Φ is computed from the normalized values $V_{qn}(\Phi)$ and $V_{in}(\Phi)$ as per Table 4.2 and ranges from -180 degrees to +180 degrees. A listing of the software is provided in Appendix A.

4.4.2 CALIBRATION AND PHASE OFFSETS

In practice, no two circuits are ever identical nor do they possess ideal phase responses. Component tolerances, cable lengths and other factors contribute to variations. The

Table 4.2 Case statements for computation of phase, Φ, from in-phase and quadrature data.

DEFINITIONS:

$$\Phi_i = \sin^{-1}(V_{in}(\Phi))$$

$$\Phi_q = \sin^{-1}(V_{qn}(\Phi))$$

COMPUTATION:

IF $\Phi_i < -45°$	THEN	$\Phi = \Phi_q$
IF $\Phi_i > 45°$	THEN IF $\Phi_q > 0°$ IF $\Phi_q < 0°$	 THEN $\Phi = 180° - \Phi_q$ THEN $\Phi = -180° - \Phi_q$
IF $\Phi_q < -45°$	THEN	$\Phi = -90° - \Phi_i$
IF $\Phi_q > 45°$	THEN	$\Phi = 90° + \Phi_i$

sinusoidal response of typical PDs used in the system are depicted in Figure 4.11. V_{qmax} and V_{qmin} are the minimum and maximum values respectively of the quadrature channel response. Likewise, V_{imin} and V_{imax} are the minimum values of the in-phase channel response. These minimum and maximum data are used for computations given in equations (4.3) and (4.4).

A normalized zero crossing can be defined as midway between both the minimum and maximum described values above. For an ideal phase detector response, the quadrature response would intersect the normalized zero crossing at 0 and 180 degrees. Similarly, the ideal in-phase response would intersect the normalized zero crossing at +/- 90 degrees.

In practice, the responses of the phase detectors are such that the intersections are not at the same voltages as in the ideal case, as described above. The corresponding amount of phase shift occurring between the practical and the ideal case defines the calibration phase offset shown in Figure 4.11.

The software has provision for storing minimum, maximum, and phase offset calibration values. The calibration values are stored in battery backed RAM. The memory locations are listed in Table 4.3. For example, considering card pair 1,

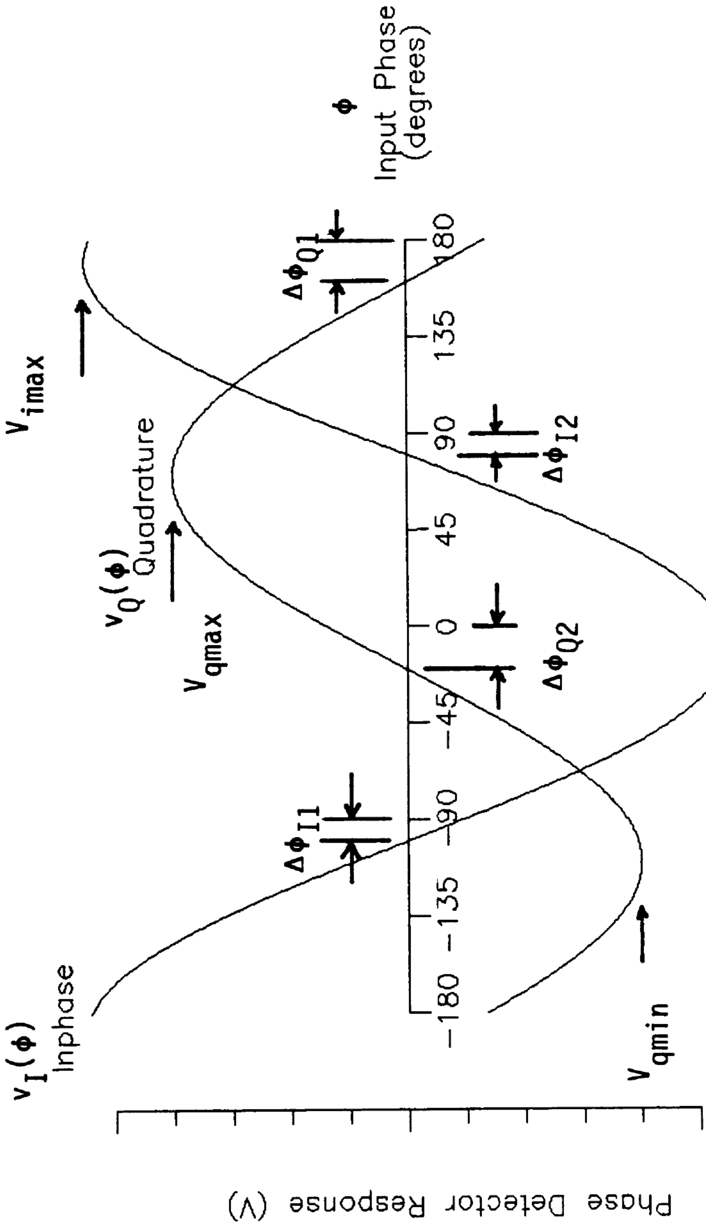


Figure 4.11 Typical PD responses with calibration definitions.

Table 4.3 Memory location of calibration data.

Minimum and maximum calibration values are integers ranging from 0 to 255, eg., absolute values measured by A/D converter:

CARD PAIR	MEMORY LOCATIONS			
	V_{imin}	V_{imax}	V_{qmin}	V_{qmax}
0	4000	4001	4002	4003
1	4004	4005	4006	4007
2	4008	4009	400A	400B
3	400C	400D	400E	400F

Phase offset calibration values are in units of 0.1 degrees and stored as hi-byte,low byte pair. All phase calibration data is stored as a two byte signed integer pair.

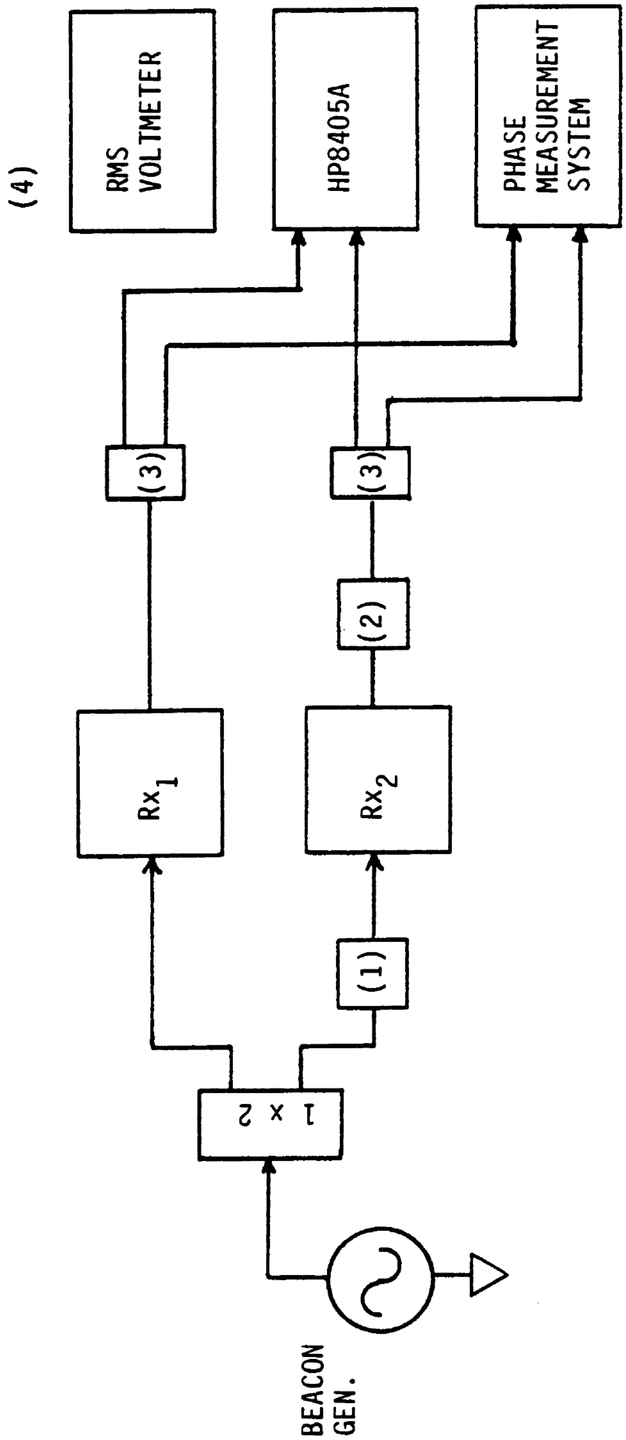
CARD PAIR	MEMORY LOCATIONS			
	$\Delta\Phi_{I1}$	$\Delta\Phi_{I2}$	$\Delta\Phi_{Q1}$	$\Delta\Phi_{Q2}$
0	4011,4010	4013,4012	4015,4014	4017,4016
1	4019,4018	402B,402A	402D,402C	402F,402E
2	4021,4020	4023,4022	4025,4024	4027,4026
3	4029,4028	402B,402A	402D,402C	402F,402E

a typical value of V_{qmax} is 226, and a typical value of V_{qmin} is 24. These data are entered in memory locations 4005 and 4004 respectively. A typical value of $\Delta\Phi_{Q1}$ is 5 degrees. The data is stored in units of 0.1 degrees in memory locations 4019 and 4018 as a two byte signed integer. Likewise, a typical value of $\Delta\Phi_{Q2}$ is -3 degrees, and is stored as a two bit signed integer in memory location 4021 and 4020. Utility software provided in Appendix A permits one to enter minimum and maximum calibration data as integers of 0 to 255. Similarly, phase offsets are entered in units of 0.1 degrees using the utility software in Appendix A.

4.5 PERFORMANCE AND CONCLUSIONS

4.5.1 ACCURACY

To evaluate the performance of the phase measurement system, equipment was configured per Figure 4.12. The Hewlett Packard 8405A Vector Voltmeter was used to measure phase at 5 degree increments from -180 to 180 degrees of input phase. A comparison of the Vector Voltmeter and the phase measurement system results are shown in Figure 4.13. Under the equal amplitude and frequency conditions of this application, the accuracy of the Hewlett Packard 8405A is quoted at 1.5 degrees. A transmission line stretcher was used to vary the phase of the applied signal. At each increment, five phase measurements were taken and averaged.



NOTES:
(1) Transmission line stretcher
(2) Variable attenuator
(3) 50 ohm T
(4) For signal amplitude measurement of phase meas. system inputs

Figure 4.12 Equipment Set-up for Performance Measurements.

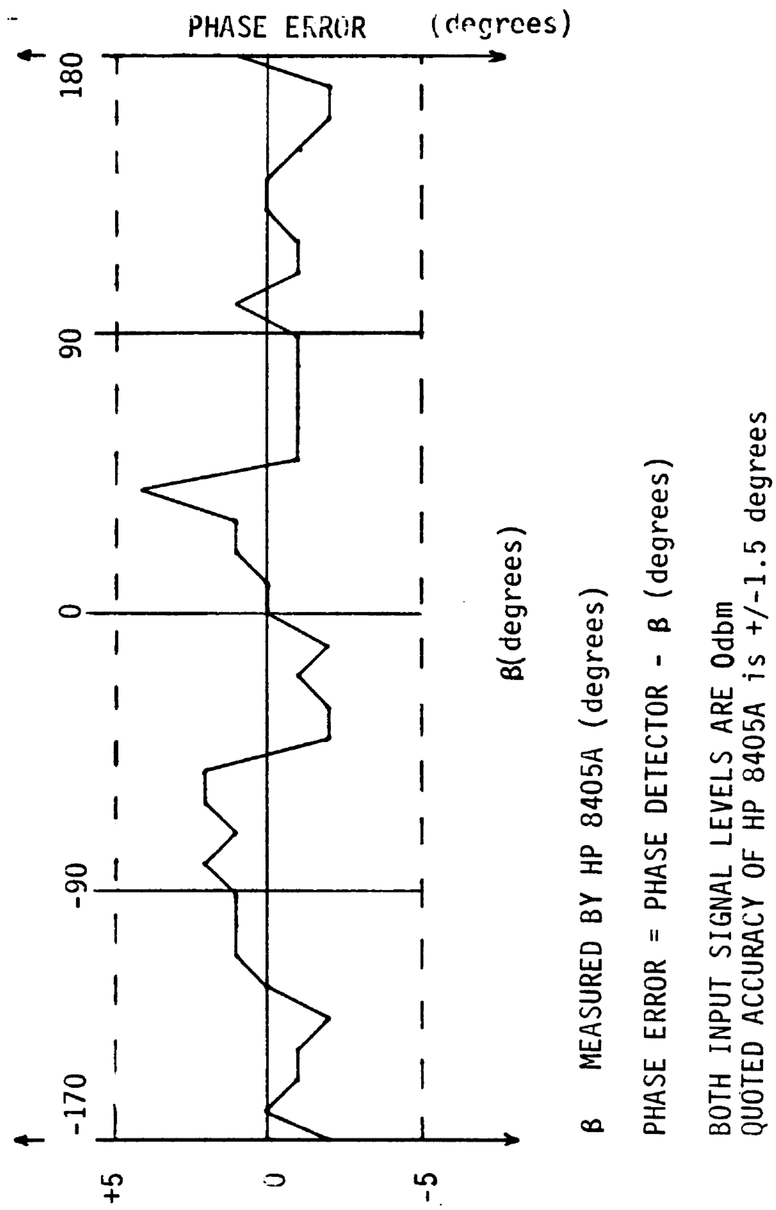


Figure 4.13 Plot of phase errors with constant, equal input amplitudes.

Most deviations in measurements were within 1 degree with an RC time constant of 10 milliseconds.

Several comments can be made about observations of Figure 4.13. Notice that there is a 4 degree phase error at β equal to 45 degrees. Additionally, there is an abrupt change in phase of 4 degrees at β equal to -45 degrees. From Section 4.4 it was stated that switching between the in-phase and quadrature channels occurs at +/-45 degrees of input phase Φ. It can be concluded that since calibration occurs at 0, -90 and 90 degrees, the abrupt changes in phase errors are due to switching between in-phase and quadrature channels.

Another set of measurements was made to determine the effect of input amplitude variations on phase measurement accuracy (see Figure 4.12). One input was set to a constant amplitude while the second input amplitude was varied with an attenuator. Measurements using a Hewlett Packard 8405A Vector Voltmeter were compared with the output of the phase measurement system. The quoted accuracy of the Hewlett Packard 8405A under unequal input level conditions is +/- 5 degrees. Measured data is shown in the graph of Figure 4.14 and indicates typically less than 5 degree change over a 20 dB variation of input level. By the Friis transmission equation [9] loss varies by $1/r^2$, and 20 dB would

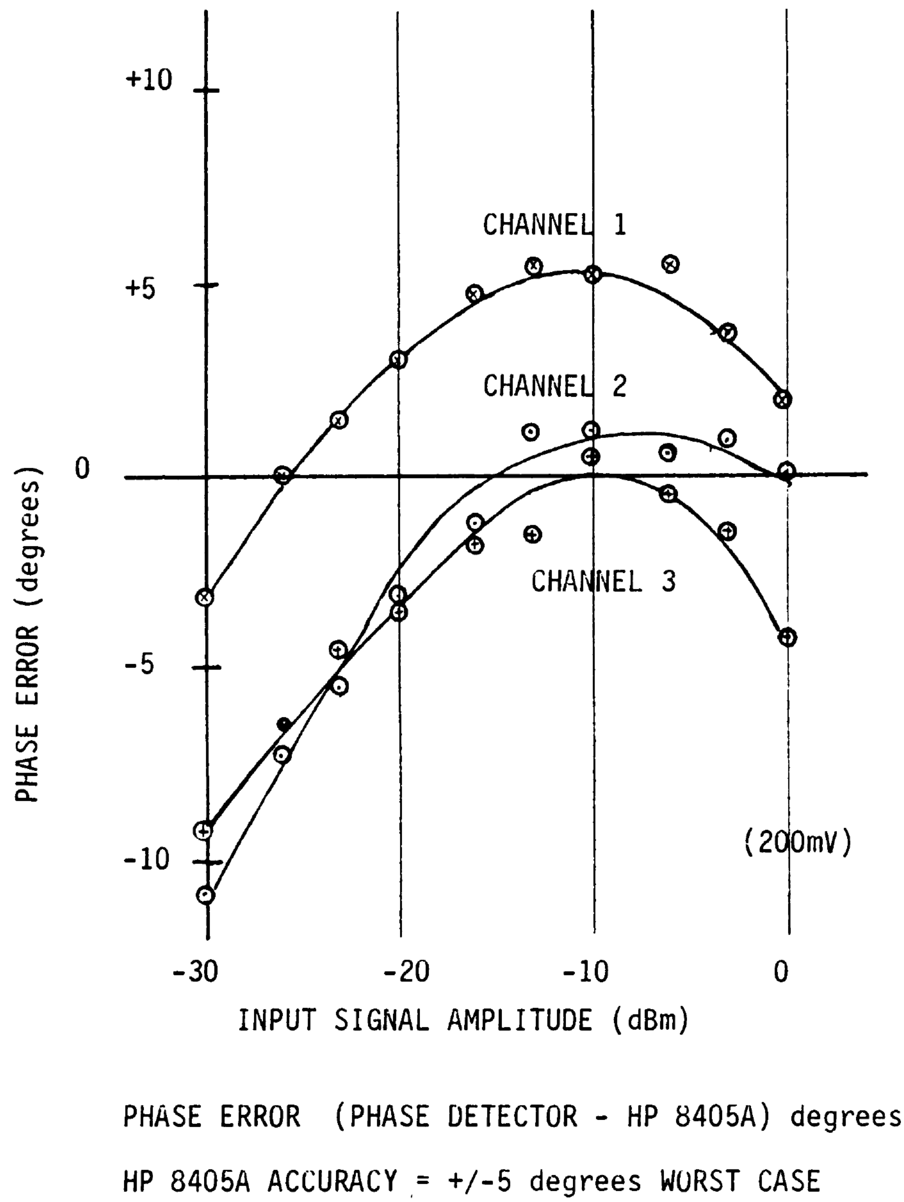


Figure 4.14 Phase measurement system errors for varied input amplitude.

correspond to movement of the beacon from, say, 10 to 100 meters. Both of these measurements show that the phase measurement system is indeed sufficiently accurate for the intended purposes.

4.5.2 RESOLUTION

In order to determine the resolution of the phase detector, one can consider the quadrature channel mixer output shown in Figure 4.10. Resolution is a function of V_{qmax}, V_{qmin}, A/D resolution, and to a lesser degree, the phase of the applied signals. Since an 8 bit A/D is used with an output voltage between 1 and 9 Volts, the number of quantization levels used between V_{qmax} and V_{qmin} is approximately

$$\frac{9V - 1V}{10V} 2^8 = 205 \text{ quantization levels.}$$

Thus V_{qmax} is at a quantization level of about 230. V_{qmin} is a quantization level of 26. V_{qmax} and V_{qmin} correspond to +1 and -1 respectively of the sine function.

Also from Figure 4.10 we see that the worst case resolution occurs at +/- 45 degrees from the normalized zero crossing:

$$\text{resolution(worst case)} = 45 - \sin^{-1}(.7071 - 2/205) = 0.8°.$$

The best case resolution occurs at the normalized zero crossing:

$$\text{resolution(best case)} = 0 - \sin^{-1}(0 - 2/205) = 0.6°.$$

A similar result applies to the in-phase channel in terms of resolution.

Assuming a 1 degree phase measurement resolution, the position resolution can be calculated. Recall from Section 2.6 that on the baseline, a 360 degree differential phase corresponds to a distance of one half wavelength. At 450 MHz, assuming 1 degree in phase measurement resolution, the position resolution on the baseline is equal to one half wavelength (0.33m) times 1/360 or about 0.1 centimeter. Away from the baseline, resolution in position changes according to the shape of the lane described in Section 2.6. As the width of the lane increases away from the baseline, the resolution in position becomes more coarse (ie. a larger measurable increment). For 1 degree phase measurement resolution, the APAC system design goal of centimeter resolution is achieved.

4.5.3 MEASUREMENT TIME

The rate at which the measurements can be made is established by two factors, the computing time of the microcomputer, and the settling time required by the low pass filter discussed in Section 4.3. The computation time using a version of assembly software is slightly less than 1 millisecond. Since the phase measurement system responds in less than 1 millisecond after a measurement request, settling time must be externally controlled by the host computer (or CPU).

Measurements were initially made with an RC time constant of 1 millisecond. At least five time constants are needed between measurements to approach within 99 percent of steady state voltage. The signals from the developmental APAC receiver were noisy and contained phase noise and jitter observable on the oscilloscope. Thus, the RC time constant was increased to provide integration after the phase detector to ensure consistent phase measurements.

After some experimenting, the RC time constant was changed to 10 millisecond, that is 10 kilohm and 1 microFarad, with a settling time of 50 milliseconds. The reason for this longer measurement time was to satisfy the system performance in terms of providing consistent measurements in the presence of noise. However, the phase measurement system by itself is capable of more rapid measurement.

4.5.4 AUTOMATED CALIBRATION

Another improvement that can be easily implemented in the phase measurement system is an automated calibration routine. The associated hardware for this modification is shown in Figure 4.15. A description of operation is as follows. The controller closes S_a and S_b and opens S_0 through S_3 during normal operation. During calibration, S_a and S_b open, while only one of S_0 through S_3 is closed in sequence. This provides a series of 10.7 MHz signals with 90 degree phase increments to calibrate the system in accordance with Section 4.4 criteria. The 90 degree phase shift increments are provided by either a 1/4 wavelength cable or a commercially available 90 degree phase shift.

4.5.5 TEMPERATURE STABILIZATION

It is important to note the temperature dependence of semiconductor device characteristics [8]. As the temperature of the phase measurement system changes, temperature dependence of the semiconductors result in level shifts in the multiplier circuitry. This was experimentally investigated by heating the circuit board and observing the PD output levels. Although no quantitative data is currently available, it is believed that a 10-20 degree Celsius temperature change results in sufficient degradation that the system becomes unusable. In anticipation of this problem in a field environment,

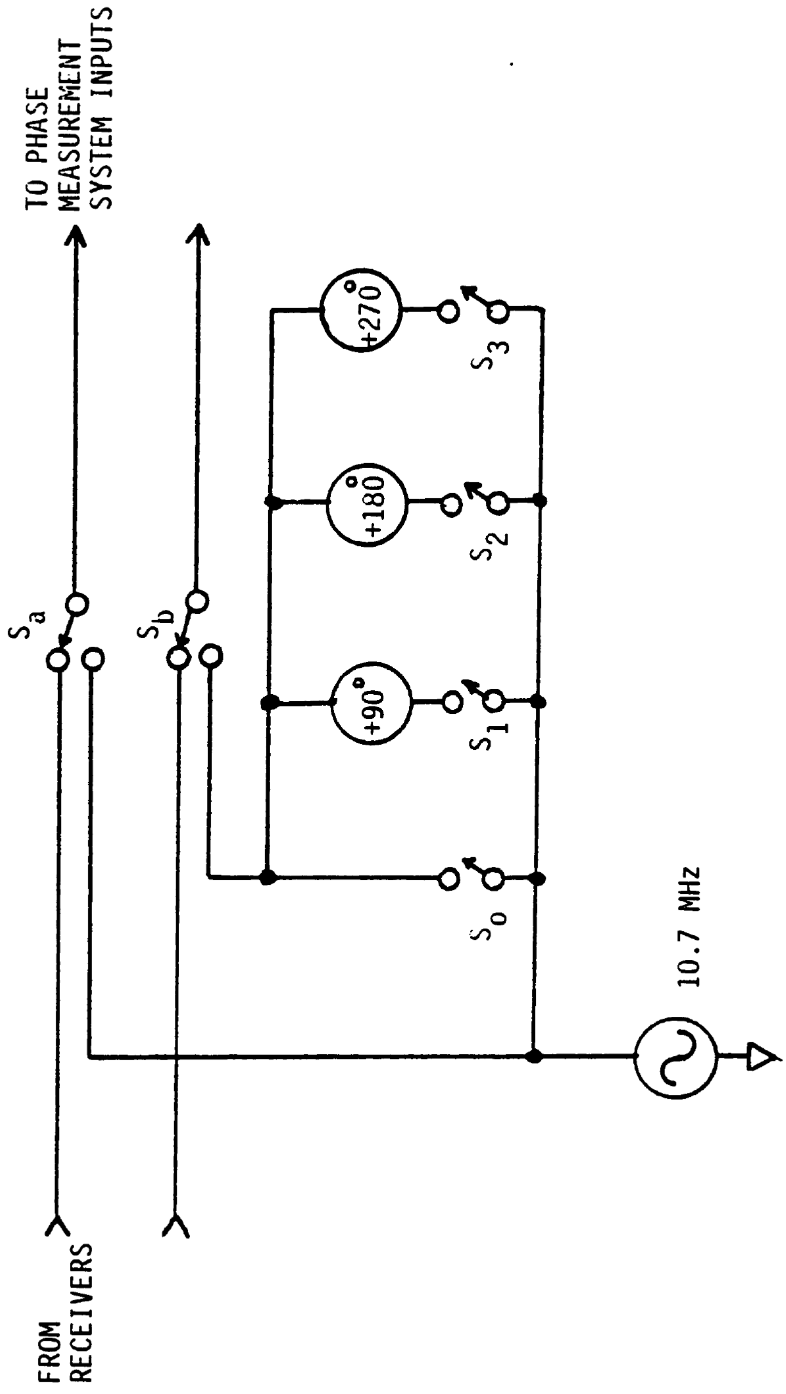


Figure 4.15 Automated calibration for the phase measurement system.

precautions were taken by installing the phase detector circuits in an insulated box equipped with a thermostat and heater for temperature stabilization.

CHAPTER 5

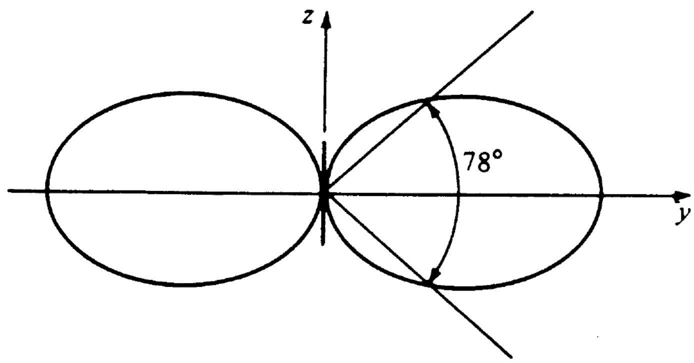
HARDWARE SPECIFICATIONS AND CONSIDERATIONS

5.1 ANTENNAS FOR APAC

5.1.1 HALF WAVE DIPOLE

The antennas used for both the beacon and the receiver during the experimental APAC system measurements were half wave dipoles. The antenna radiation pattern is shown in Figure 5.1 and the maximum gain is 1.64 [9]. The half wave dipole was chosen because it has omni-directional characteristics in the x-y (horizontal) plane. Due to the x-y plane symmetry the phase center of the half wave dipole is assumed to be located directly in the center of the two elements. This may not necessarily be the case as discussed in the following section.

For grazing angles between 5 and 50 degrees, the reflection coefficient of the ground for vertical polarization is less than that of horizontal polarization at a frequency of 400



ELECTRIC FIELD RADIATION PATTERN FOR HALF-WAVE DIPOLE. z IS IN VERTICAL DIRECTION. HALF POWER BEAMWIDTH IS 78 DEGREES.

Figure 5.1 Diplole antenna radiation pattern. Reprinted from [9].

MHz [9]. Therefore, vertical polarization is chosen because it is expected to reduce position errors due to multipath.

5.1.2 ANTENNA FEED CABLE

During the APAC test measurements using the half wave dipole antennas, phase errors of roughly 60 degrees were induced when the beacon (transmitting) antenna was rotated in the horizontal plane. A 60 degree error is indeed a catastrophic error and would cause serious lane errors in the APAC system, thus rendering the system useless for accurate positioning.

Many texts on antennas show radiation patterns of dipole antennas but neglect the effect of the feed cable. For the case of receiving antennas, the voltage appearing on the antenna terminals is a resultant of the vector sum of the waves arriving at the antenna elements. The location of the feed cable and balun matching transformer with respect to the dipole perturbs the phase center as depicted in Figure 5.2. Although the exact cause of this phase perturbation is not known, it would be reasonable to speculate that it is due to the reflections from the matching transformer in the near field of the dipole. By reciprocity, a similar statement can be made of phase center perturbation for the beacon (transmitter) antenna.

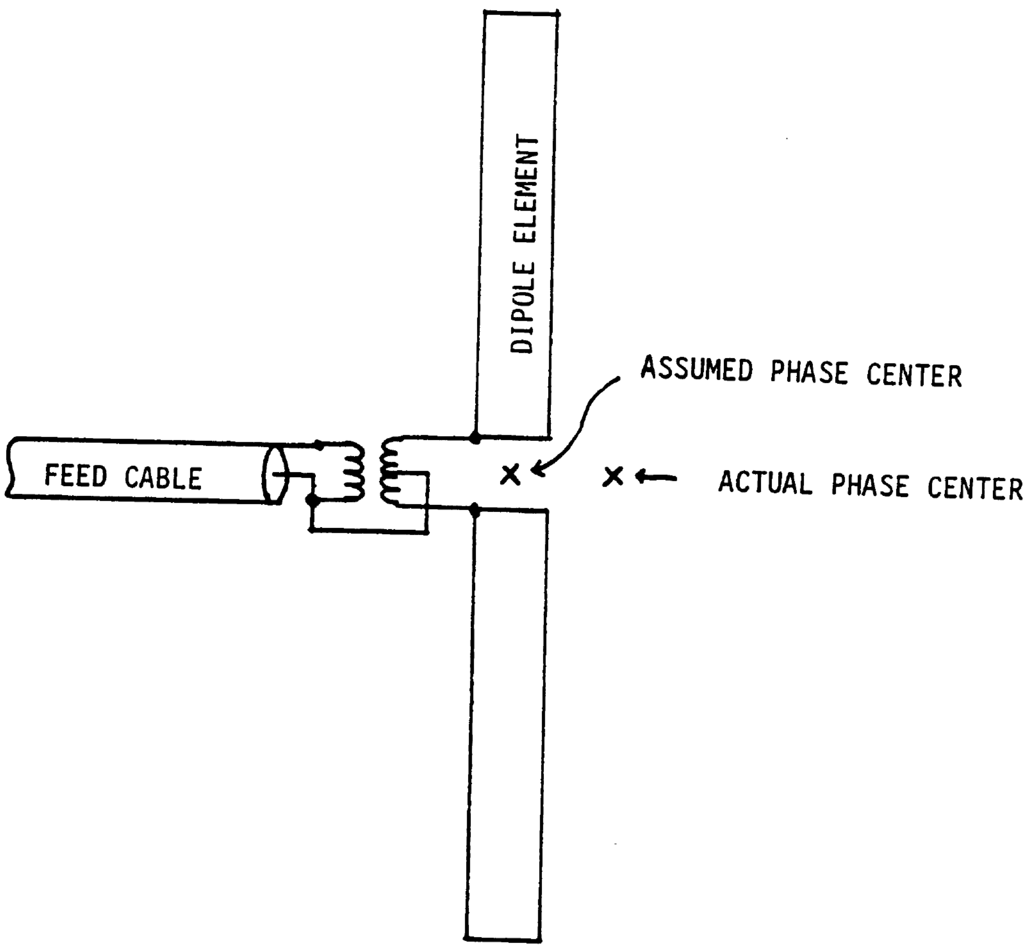


Figure 5.2 Phase perturbation due to antenna feed.

This is not a serious problem in the case of the receiver antennas since the receivers are in a fixed position. The antennas can be oriented with the feed cable away from the measurement site and need not be moved. However, the transmitting beacon case is more difficult to contend with since the beacon can be located anywhere within the measurement site and radiates to receiving antennas located outside the site. The feed cable cannot be directed in such a way as to not interfere with the antenna pattern for one of the antennas located at the four corners of the site.

With some thought, noting that the polarization is vertical, it is likely that extending the feed cable horizontally for several wavelengths should help the problem. A horizontal two-wire transmission line can be used to remove the balun from the near field. This, however, is one issue that has yet to be quantified and addressed in the APAC system.

5.1.3 COMMENTS ON THE USE OF A YAGI ANTENNA

On an operating APAC measurement site, reflections from the ground and other objects will occur. From these reflections phase errors, and therefore position errors, will result. Consider the use of a higher gain antenna such as the Yagi-Uda array with gain of typically around 10 [32]. It would seem initially that if the receiver antenna were significantly more directive (compared to the half wave dipole), phase errors could be reduced.

In practice, use of the Yagi antenna presents some problems for the APAC system. The phase center cannot be assumed to be at a single point as in the dipole. Furthermore, it is uncertain as to whether or not there is frequency dependence of the phase center. Experimental testing is required to determine the phase center of the Yagi antenna.

5.1.4 COMPARISON OF YAGI AND DIPOLE MEASUREMENTS

Some experimentation was done to compare the performance of the APAC location system with Yagi antennas (commercially available UHF television Yagi antennas) and with the dipoles. Twelve measurements at identical locations on a 4.2 x 4.2 meter site were performed with both dipole and Yagi antennas. Data from the Yagi antenna measurements showed that 75 percent of the position location measurements were within 10 centimeters. The remaining 25 percent measurements had lane errors.

When compared to measurements with dipoles: 58 percent of the results were significantly improved, 17 percent showed no significant improvement, and 25 percent showed increased errors. Keeping in mind that this was a very limited sample size, it cannot be concluded that Yagi antennas will solve the multipath problems in the APAC system. Other factors such as antenna gain, phase center, etc could conceivably contribute to this improvement. However, from these limited data, it can be concluded that some improvement was observed. This should warrant further investigation on a larger test area when a larger number of points can be tried.

5.2 RECEIVERS

This section specifies performance requirements for an APAC receiver and comments on the receivers used for test measurements.

5.2.1 BANDWIDTH REQUIREMENT

The factors that determine bandwidth are the phase measurement time and the frequency stability of the system. The phase measurement time depends on propagation time (over the links and measurement area), and the phase detector time constant. Since the round trip propagation delay on a 2 km measurement site is less than 20 microseconds, the delay can be neglected. According to the discussion in Section 4.5.3, phase measurements were demonstrated with an RC time constant (at the PD output) of 1 millisecond. Allowing for five RC time constants, it should be possible to obtain measurements at a rate of one every 5 milliseconds. This 5 millisecond rate can be assumed for bandwidth computation, provided that there is minimal jitter in the receiver system such that a longer RC time constant is not required to provide consistent measurements.

The frequency switching techniques described in Section 3.2 require that a burst of RF is transmitted by the beacon. According to [7] the minimum bandwidth B required for receiving a pulse width of duration t is given by: $B \geq 1/t$. For a burst of duration 5 milliseconds, the minimum bandwidth is 200 Hz.

Two practical considerations must be investigated to specify the bandwidth. First, beacon and MLO frequency stability are important. They are topics discussed in Section 5.3.1. It is shown that realistic frequency stability is +/- 1.5 kHz for each oscillator (over time periods assumed longer than the time between system calibrations). The worst case frequency drift from both the beacon and the MLO will be +/-3.0 kHz.

Second, the filter phase response is another important factor to consider. Figure 5.3 [20] shows the response of a filter with n poles. Group delay (the derivative of phase

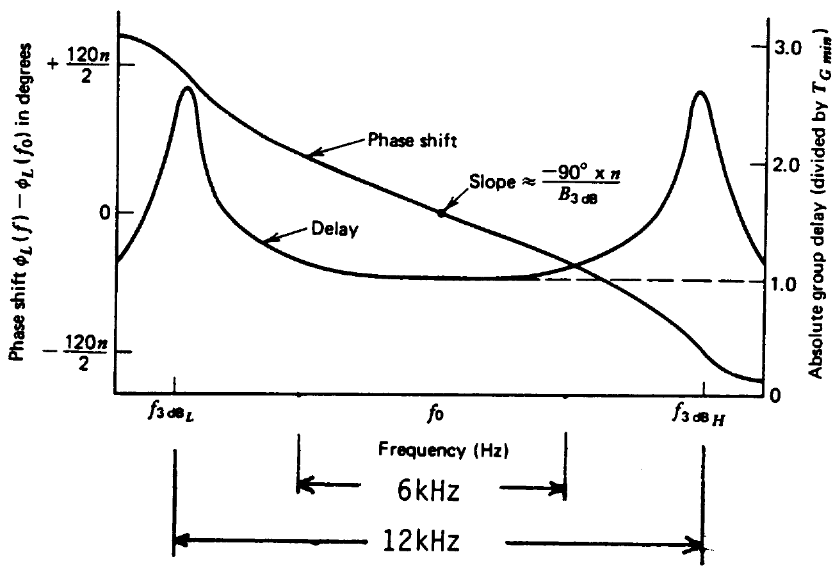


Figure 5.3 Group delay (T_G) and phase (Φ) frequency response. Reprinted from [20].

with respect to angular frequency) is a maximum near the 3dB frequency points. At frequencies near these group delay maxima, phase errors can be induced into the system by frequency fluctuations. To minimize this effect, the system must operate near the center of the filter passband and not near the 3dB points. In other words, the bandwidth of the receiver must be sufficiently large that it guarantees that the frequency of operation is within the region of constant group delay. Using the +/- 3.0 kHz frequency requirement, a bandwidth of +/- 6.0 or 12.0 kHz is specified. Figure 5.3 illustrates this justification.

5.2.2 RECEIVER NOISE CONSIDERATIONS

The basic APAC receiver system is shown in Figure 5.4.a. The required phase detector SNR(input) was specified at greater than 10 dB in Section 4.2.2. Remember that the discussion of phase detector noise indicates that a 10 dB SNR(input) provides an adequate phase measurement for these purposes. Assuming a 3 dB margin, the SNR for the receiver system output is specified at 13 dB.

The cable loss in Figure 5.4.a is that expected in a 50 ft. length of RG-58 between the low noise amplifier (located on top of a 30 foot antenna tower) and the receiver. The purpose of placing the LNA close to the antenna is to increase the SNR of the receiver system.

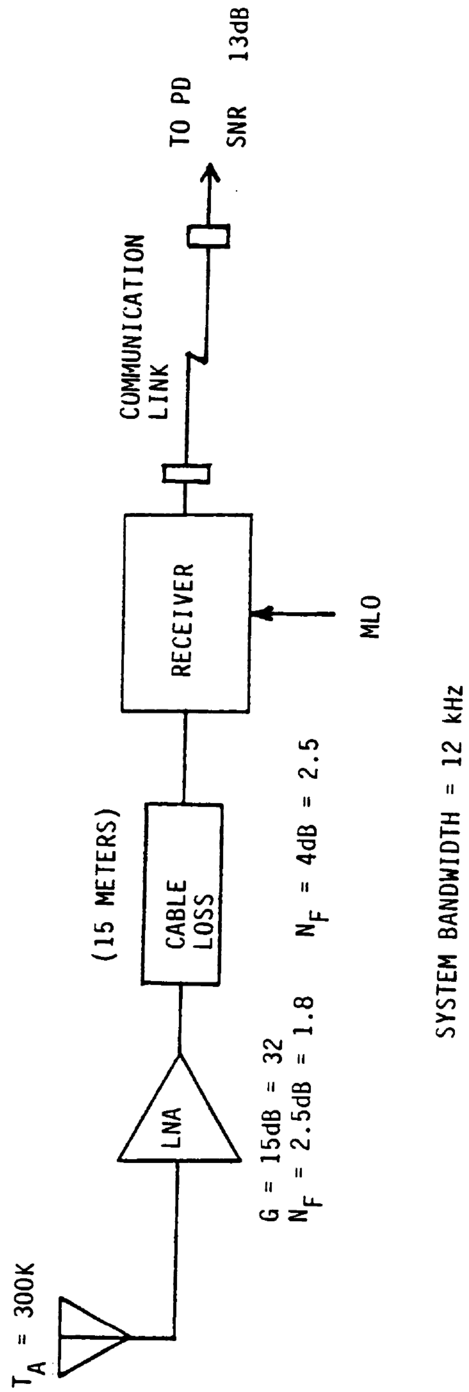


Figure 5.4.a Receiver System for APAC.

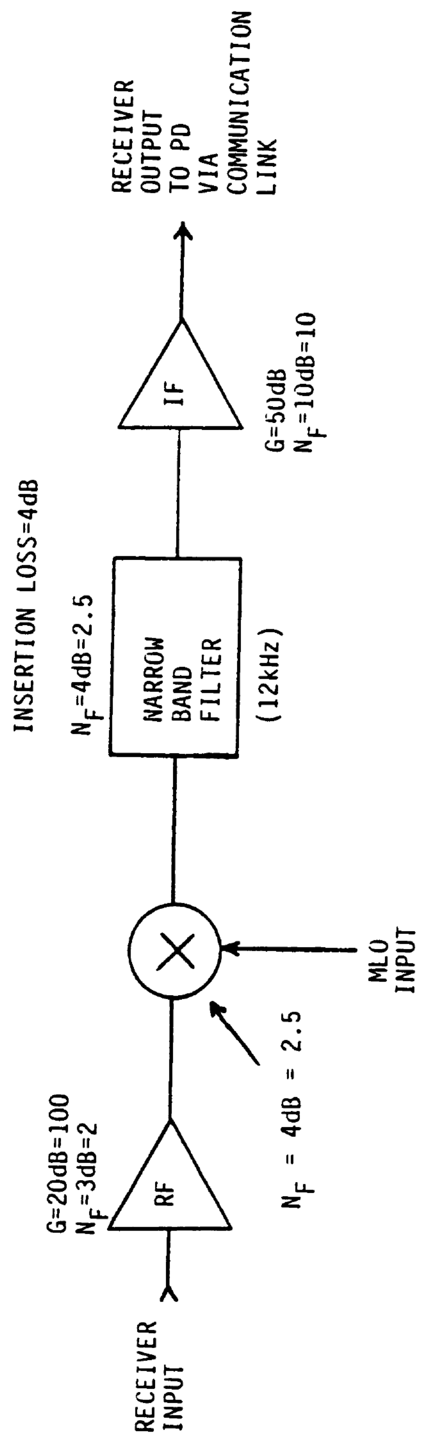


Figure 5.4.b Receiver Block Diagram.

Figure 5.4.b shows a block diagram of the receiver. Typical values of noise figure have been assumed for illustrative purposes. Calculation of the overall noise figure referred to the antenna input is shown in Appendix B [21]. The output of the receiver is applied to the phase measurement system through a link. This link is the topic of discussion in section 6.1. In practice, it consists of anything from a short cable to an optical laser link. Calculation of the minimum signal required at the receiver antenna terminals depends on the noise figure of the link. For example, in the case of a lossless link, the minimum signal to the receiving antenna terminals is equal to -147 dBw for a SNR of 13 dB at the phase detector.

5.2.3 RECEIVER OUTPUT LEVEL CONTROL

A practical operating APAC system consists of blocks (eg. components) that are easily configured and setup on a measurement site. For this reason, it would be preferable to have receivers that are interchangeable among locations. Furthermore, from a system point of view, it is preferable to have an IF output that connects to a standard link (or phase measurement system) input and provides a signal with a desired amplitude. Two methods of providing a signal with a defined, constant amplitude IF output level are Automatic Gain Control (AGC) and limiting. The AGC must be derived from the strength of the carrier since there is no modulation or envelope to detect. A typical limiter would be the LM3089 IC amplifier, which also provides a voltage output proportional to the logarithm of the input signal power. The importance of deriving a voltage proportional to applied signal is seen in the following section.

5.2.4 PHASE VARIATIONS DUE TO AMPLITUDE LEVEL CHANGES

Consider the transistors in the receiver circuitry. According to [23] internal capacitances exist between the device terminals. These parasitic capacitances vary as a function of the applied signal voltage. At the frequency of operation, (about 450 MHz) the capacitance changes with signal level result in variations in the phase shift through the receiver. Remember that as the beacon position changes, the signal amplitude changes, causing phase changes in each receiver and consequently degrading system performance. Measurements must be made assuming constant phase shift through the system at a particular frequency, unless the received signal level is measured and a compensation factor introduced into the calculations.

Analytically, this change in phase shift with respect to voltage is difficult to calculate in the receiver system. However, it can be determined experimentally. For example, the LM3089 (limiting integrated circuit) was determined to have a 6 degree change in phase shift over a 10 dB range of input level. The receiver characteristics can be measured and characterized for phase shift as a function of the input signal level.

Once this characteristic curve has been determined, one of two methods can be employed to correct for phase changes with signal level. These two methods are analog and software correction. Figure 5.5.a. shows an analog phase correction block diagram. A voltage is derived from the IF AGC or logarithmic amplifier (as in the LM3089), integrated, and applied to a variable phase shifter. The characteristic of the phase shifter must be such the it cancels any signal amplitude dependence of the receiver phase shift. Since the characteristic curve may not necessarily be linear, some scaling and manipulation of the signal strength voltage must be done to match the integrator output with the voltage required on the phase shifter. This seriously complicates the calibration.

Another method, which is preferable to the previous method, is shown in Figure 5.5.b. and employs an analog to digital (A/D) converter to measure the signal level voltage. The digital output from the A/D converter can be used directly for phase correction by a computer. A look-up table can be stored in the computer containing a list of phase

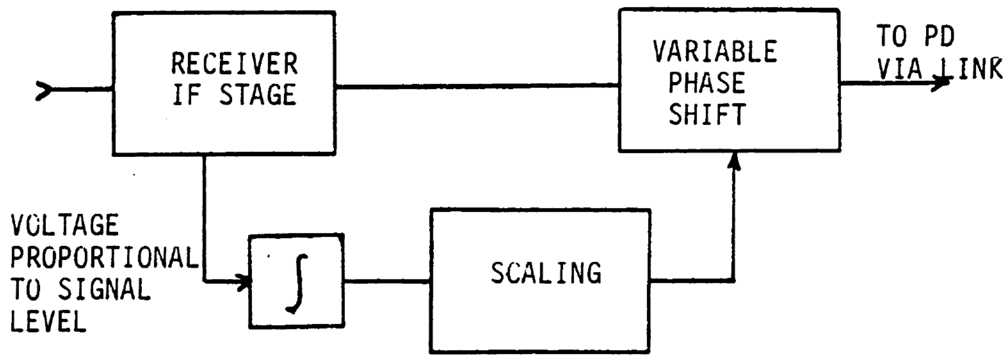


Figure 5.5.a Analog method of receiver phase correction.

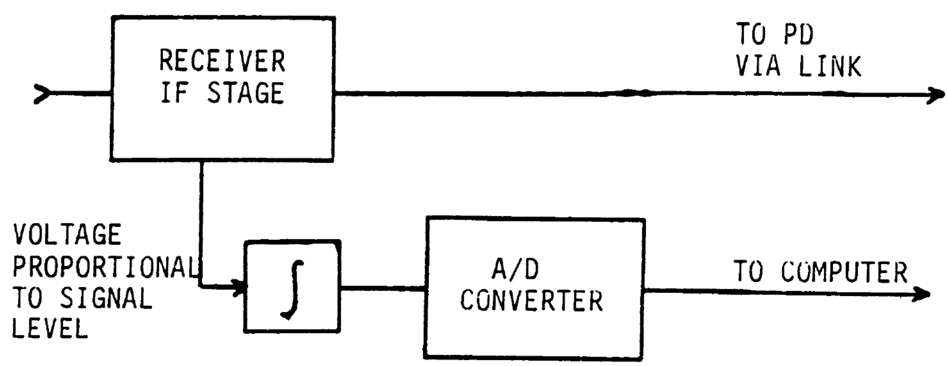


Figure 5.5.b Receiver phase correction using software.

correction values with signal level. This is a more straight forward approach since it does not require difficult alignment as in the analog version.

5.3 BEACON AND MLO REQUIREMENTS

5.3.1 FREQUENCY STABILITY

Both the beacon and Master Local Oscillator (MLO) must satisfy frequency stability requirements. Consider the block diagrams shown in Figure 5.6. A frequency standard, nominally at 10 MHz, is multiplied (via PLL frequency synthesis techniques) to the beacon or MLO frequency. The frequency stability requirement of this standard is dependent on the bandwidth of the receiver system.

An Oven Controlled Crystal Oscillator (OCCO) provides adequate stability characteristics to satisfy the design criteria for this receiver. Some typical performance characteristics are given by [35]. For the series CO-200 OCCO, long term frequency stability is 1.0E-6/year of the operating frequency. Noting that this frequency drift is multiplied by a factor of about 45, the beacon or MLO drift is equal to about 500 Hz/year. A long term frequency stability for each of the beacons and MLO of +/- 1.5 kHz will ensure that the receiver maintains the IF signal in the central portion of the narrow IF filter bandwidth.

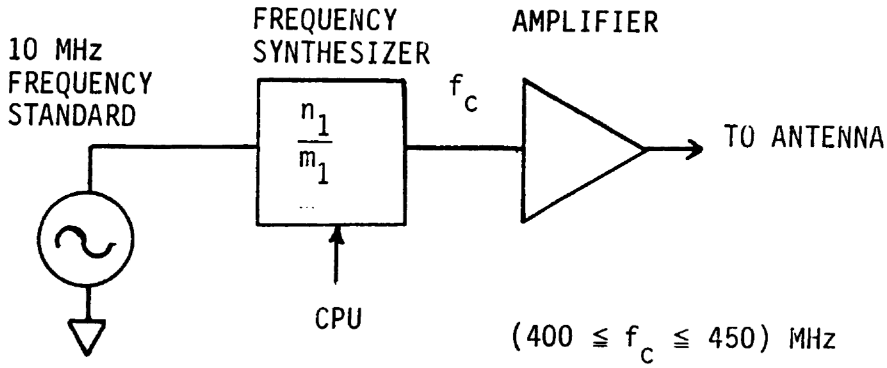


Figure 5.6.a Beacon frequency generator.

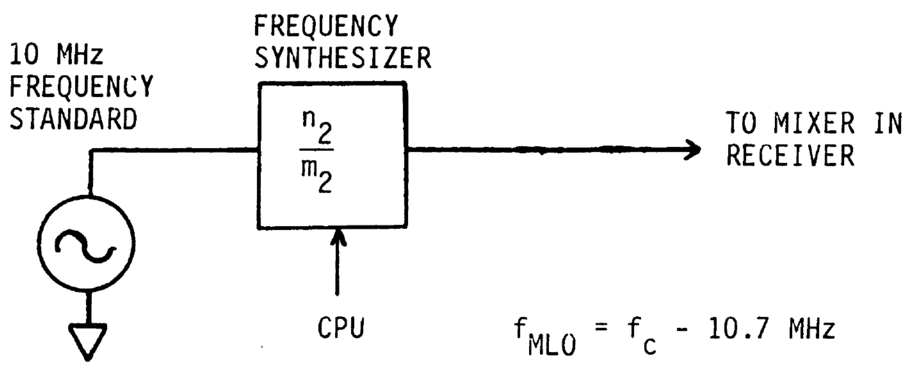


Figure 5.6.b Master Local Oscillator (MLO).

The short term drift of this OCCO standard is +/- 3E-11/second, which translates to under 0.02 Hz /second at the beacon and MLO frequency. From Figure 5.3 in section 5.2.1, the slope of the phase response near the center frequency is approximated by

$$\text{slope} = \frac{-\,90^\circ \times n}{\text{bandwidth(3dB)}}; \qquad (5.1)$$

where: n = number of resonators

For a 2 resonator filter and 12 kHz bandwidth, the change in phase over a 10 second calibration period is

$$\Delta\Phi(10s) = \frac{-90^\circ \times 2 \times 10s \times 2}{12{,}000 \text{ Hz}} \; 0.02 \text{ Hz/s} = 0.006^\circ$$

Phase shift due to short and long term frequency drift will not be adversely affected by the filter phase response.

5.3.2 BEACON POWER OUTPUT REQUIREMENTS

The minimum signal requirement at the receiver was computed in Section 5.2.2 and is equal to -117 dBm. From the Friis transmission equation the power transmitter requirement can be computed:

$$P_{rx} = \frac{G^2 \lambda^2}{(4\pi R)^2} \times P_{tx} \times \text{losses} \qquad (5.2)$$

where: λ = wavelength
P_{tx} = transmitter power
P_{rx} = receiver power
G = gain of antennas (dipole)
R = distance between transmitter and receiver

Assuming a 10 dB system loss, antenna gain of 1.6, R = 2 km, and a wavelength of 0.66m, a transmitter output power of -20 dBm is necessary. This demonstrates that the APAC system is a low power system and is a good candidate for FCC approval.

A few comments on the performance of the APAC prototype position location system are now appropriate. There is a discrepancy of several orders of magnitude between the predicted transmitter power and the power required for the prototype system. Several factors could conceivably contribute to this.

The actual noise figure of the commercially available receivers was not known and is probably considerably worse than the noise figure assumed in Section 5.5.2. In addition, the LNA (Figure 5.4.a) was located on the receiver side of the cable loss, thus increasing the system noise figure. Other factors such as matching losses, connectors, and increased bandwidth can result in added noise power. Another important factor is the link between the receiver output and the phase measurement system (Figure 5.4.a). The prototype system used cable (lossy) for this link between the receiver output and the phase measurement system. A final system version placing the phase measurement system at the remote site will improve noise figure. Chapter 6 addresses this.

5.4 CALIBRATION

5.4.1 ATMOSPHERIC

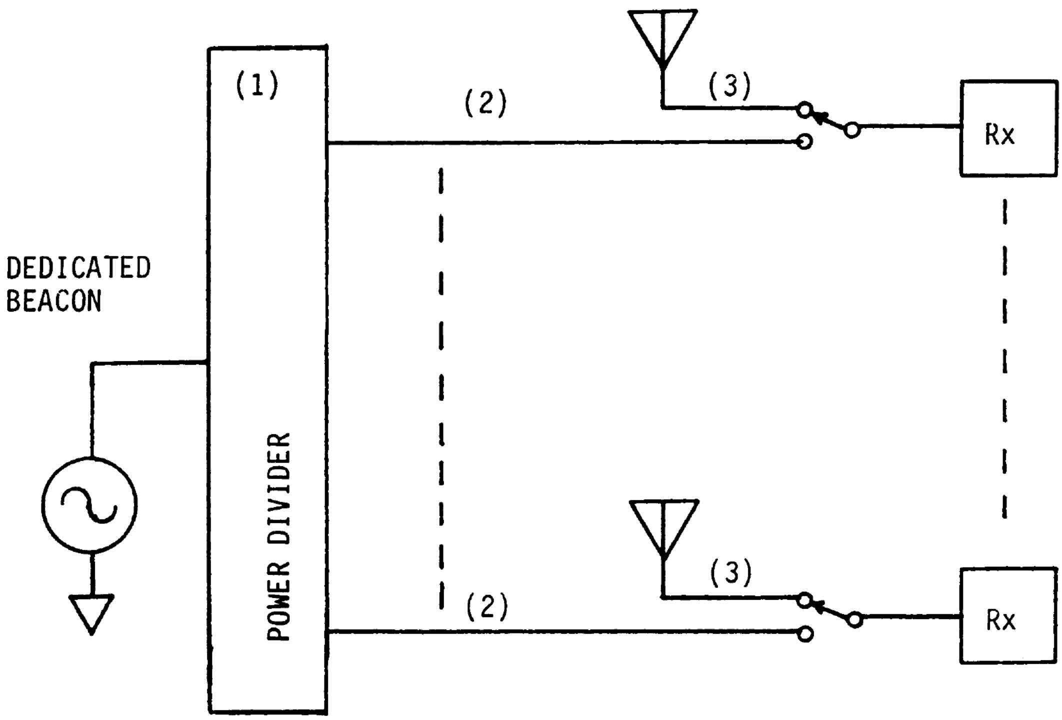
Calibration is necessary to account for any differences in phase shift among the receivers, communication links and associated cables. Atmospheric calibration is implemented by placing the beacon transmitter in a known location and measuring the differential phases at each of the receivers. The expected value of phase for the known beacon position is computed using equations from Section 3.2. A constant phase offset is then computed as the difference between the measured phase and the expected phase. This calibration offset is then added to each phase measurement during normal operation in the measurement area. It is important to note that this phase offset is only valid at the particular frequency for which it was measured. It is necessary to calibrate at all frequencies of system operation.

The advantage of atmospheric calibration is that hardware is simplified. A dedicated beacon can be placed at a fixed known location in the measurement area and repeatedly activated via computer, for calibration purposes.

5.4.2 CABLE

A method of calibration using cables is shown in Figure 5.7. The cables are trimmed to identical lengths with the use of a time domain reflectometer. Therefore, the differential phases of the RF signal arriving at the receiver inputs are zero. The same effect would result if the beacon were placed at the phase center of a measurement area of a multipath free environment. The phase offsets are determined independently of multipath errors. This is one clear advantage that cable calibration offers over atmospheric calibration.

In practice, there are several disadvantages to the cable calibration. For example, the system can be calibrated using RG-58 cable with attenuation at 450 MHz of approximately 8dB/100 ft. or 0.26 dB/meter. In a large measurement area with one dimension of about 2000 meters, cable loss from the dedicated beacon generator to the receiver would result in a loss of about 520 dB, but line amplifiers can overcome this loss.



NOTES:

1) Output of power divider phase tracked.

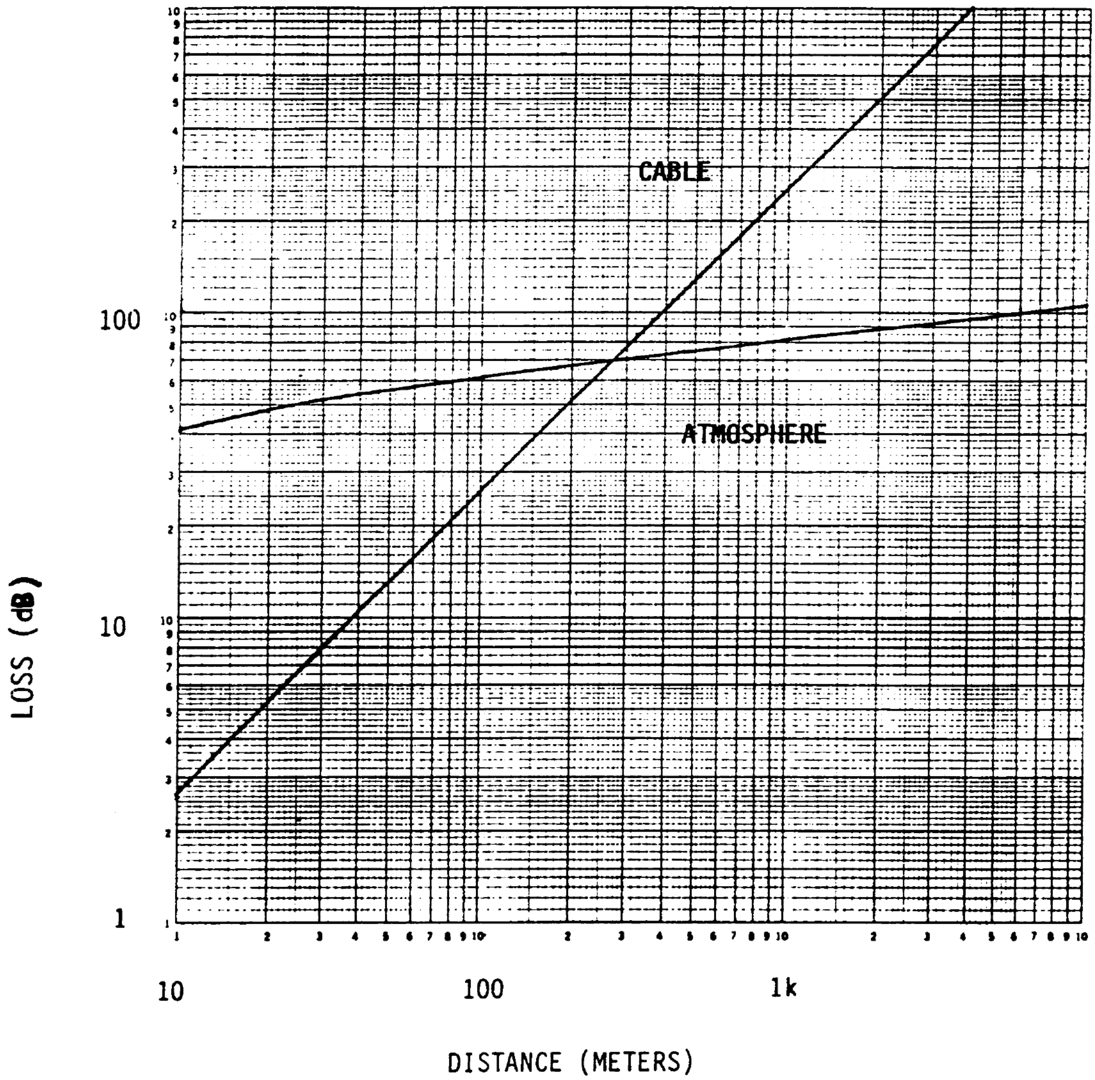
2) Equal length RG-58 cable.

3) Identical antennas, feed cables and switch in terms of phase shift.

Figure 5.7 Calibration through cable.

The receiver input power requirements from Section 5.1 are -117 dBm. To overcome this loss would require one of two approaches. The first would be transmitting into the cable 403 dBm-an unreasonably large amount of power. Alternatively, amplifiers can be inserted at intervals along the calibration cable. A requirement of these amplifiers is phase tracking since the phase must not be altered among different receivers. Furthermore, power must be distributed to the amplifiers at points along the cable length of 2 km.

Atmospheric losses are due to the Friis transmission equation and cable losses are 0.26 dB/meter. Figure 5.8 shows both losses as a function of distance for a frequency of 450 MHz. There is a cross-over point at about 300 meters in which both cable and atmospheric losses are equal. At a further distance, the cable calibration becomes more difficult to implement in hardware. This result suggests that on larger measurement areas, if cable calibration must be used to achieve desired accuracies, the measurement area should be divided up into smaller measurement cells. The smaller measurement cells can more realistically be calibrated using cable methods.



CABLE LOSS = 0.26b dB/m using RG-58 cable at 450 MHz

ATMOSPHERE LOSS = 22dB + 20 log (distance) at 450 MHz

Figure 5.8 Plot of cable and atmospheric losses for calibration.

CHAPTER 6

COMMUNICATIONS LINKS FOR THE APAC SYSTEM

6.1 METHODS OF DATA AND PHASE TRANSFER FOR THE POSITIONING SYSTEM

Figure 6.1 shows a receiver at a common control location and a remote receiver, located nominally up to 2 km from the control location. The most basic system described earlier in Chapter 3 is somewhat simplified in that it does not specify how differential phase measurements are achieved at each of the remote receiver locations. Recall that measurement of differential phase requires establishing a signal of known phase at each receiver site at a remote location. Suppose that a master local oscillator (MLO) is to be distributed to each of these locations and used as the phase stable reference. In addition, data transmission is required to control the remote location. This section outlines three possible methods of transferring the phase reference and data.

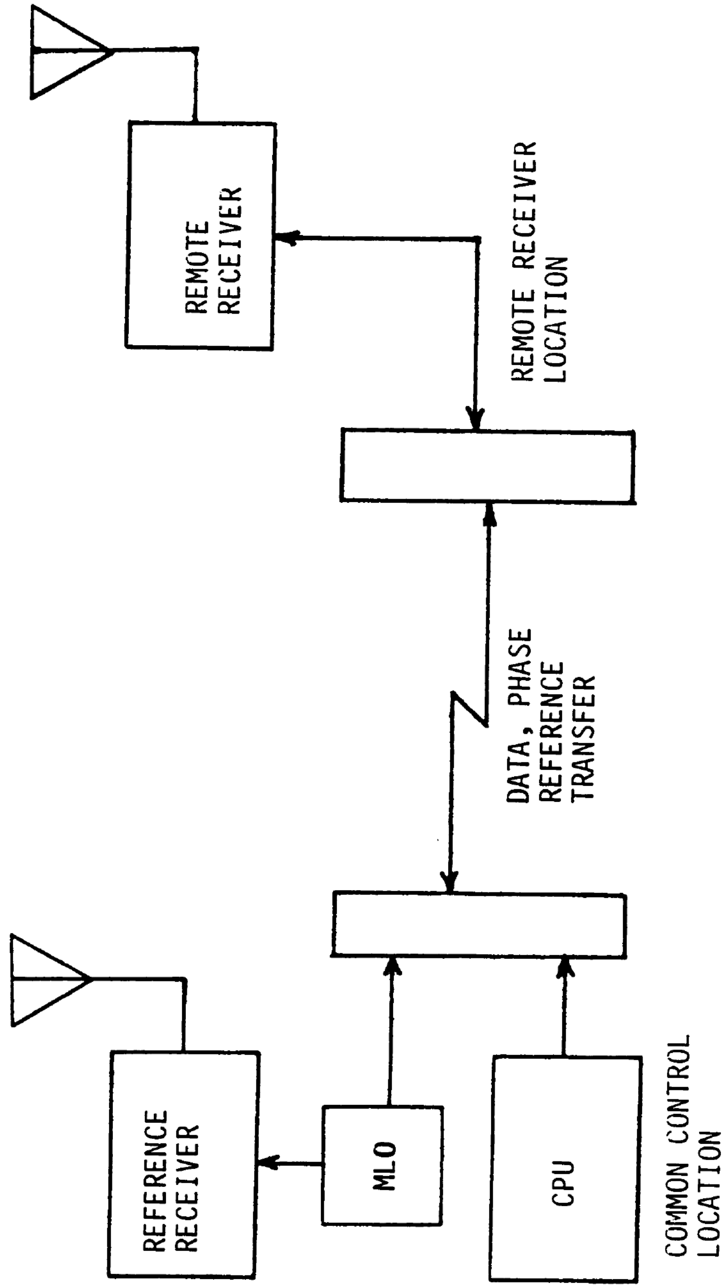


Figure 6.1 Data and phase reference transfer.

Although this discussion is for a pair of receivers it can be generalized to multiple receivers.

6.1.1 METHOD 1

Figure 6.2.a depicts the first method for the phase standard under consideration. The master local oscillator (MLO) is at a constant frequency specified in Chapter 5. Directly after the MLO is a frequency synthesizer programmable by the CPU to some frequency depending of the beacon carrier frequency. Following the power divider this multiplied signal is distributed to the remote receiver location though a communications link.

The communication link introduces some phase error Φ_{e1} into the system for signals arriving at each remote location. The communication link also introduces some phase error Φ_{e2} for signals returning from the remote location to the common control location. At present, assume that Φ_{e1}, Φ_{e2} are constants; in Section 6.2 it is assumed otherwise.

The multiplier just prior to the mixer is a nonlinear device which generates harmonics, one of which is the required local oscillator frequency. This multiplier is optional, and allows a lower frequency to be sent through the communications link so that cable losses are reduced. For example, if a transmission line such as RG-58 cable

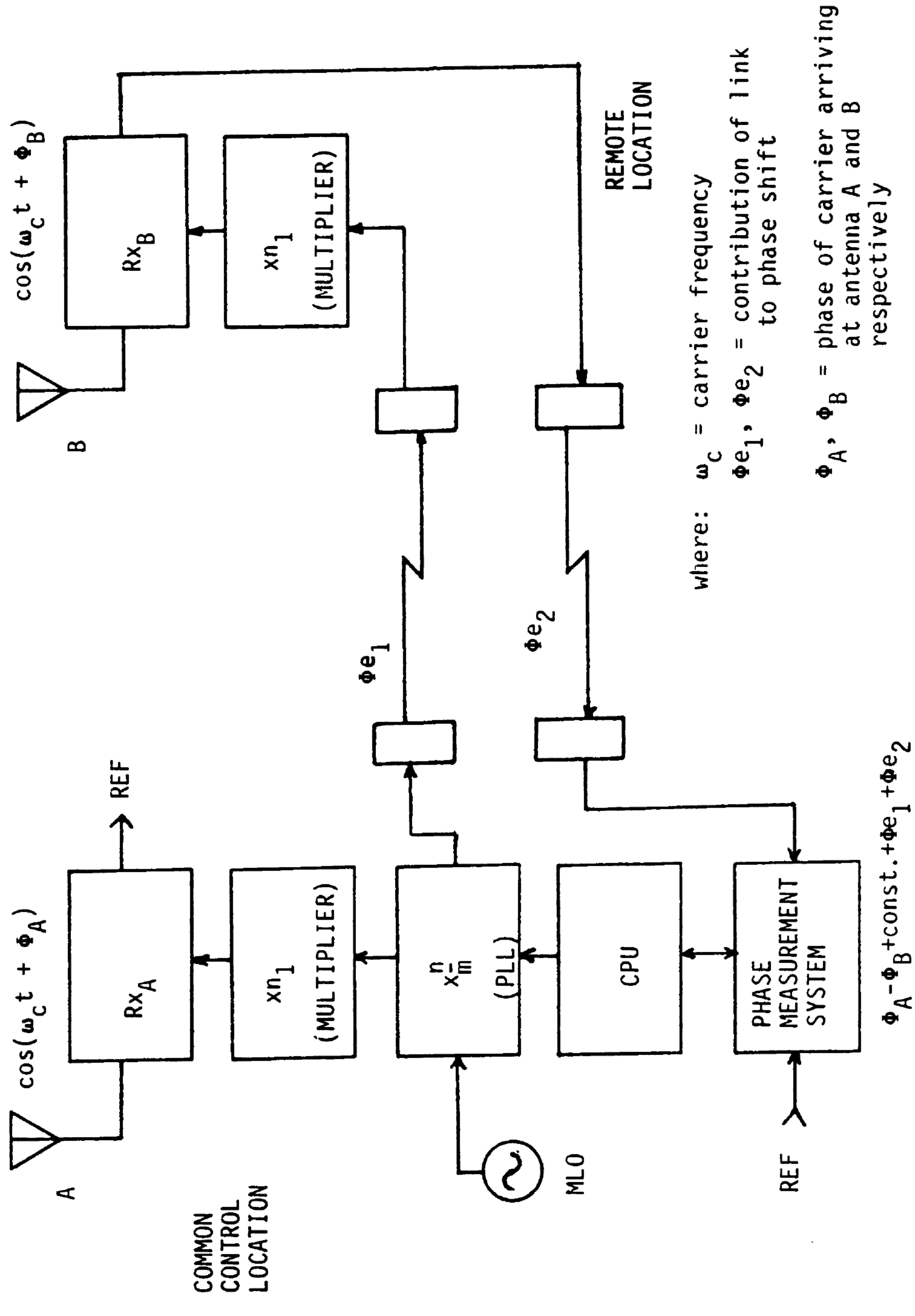


Figure 6.2.a Method 1 block diagram.

were to be used, losses are lower at the lower frequencies. The loss through a 100 meter length of RG-58 cable is about 26 db at 450 MHz compared to a loss of 13 dB at 150 MHz.

The receiver output IF signal (10.7 MHz) is returned to the control location for phase comparison with other receivers.

6.1.2 METHOD 2

The next method (see Figure 6.2.b) is similar to the first except that the programable PLL is placed at each receiver site. This option has the advantage of somewhat simplifying the communications link since operation is at the single MLO frequency of say, 10.0 MHz. The 10.7 MHz signal is returned for comparison at the reference location, as in the first method.

Successful operation of this system assumes that all of the PLLs (at each remote location) remain locked and do not slip cycles during measurements. The APAC prototype position location system suffered from this problem. It was finally decided to use method 1 since the inherent problem with PLL drift had to be eliminated.

6.1.3 METHOD 3

The third method under consideration is shown in Figure

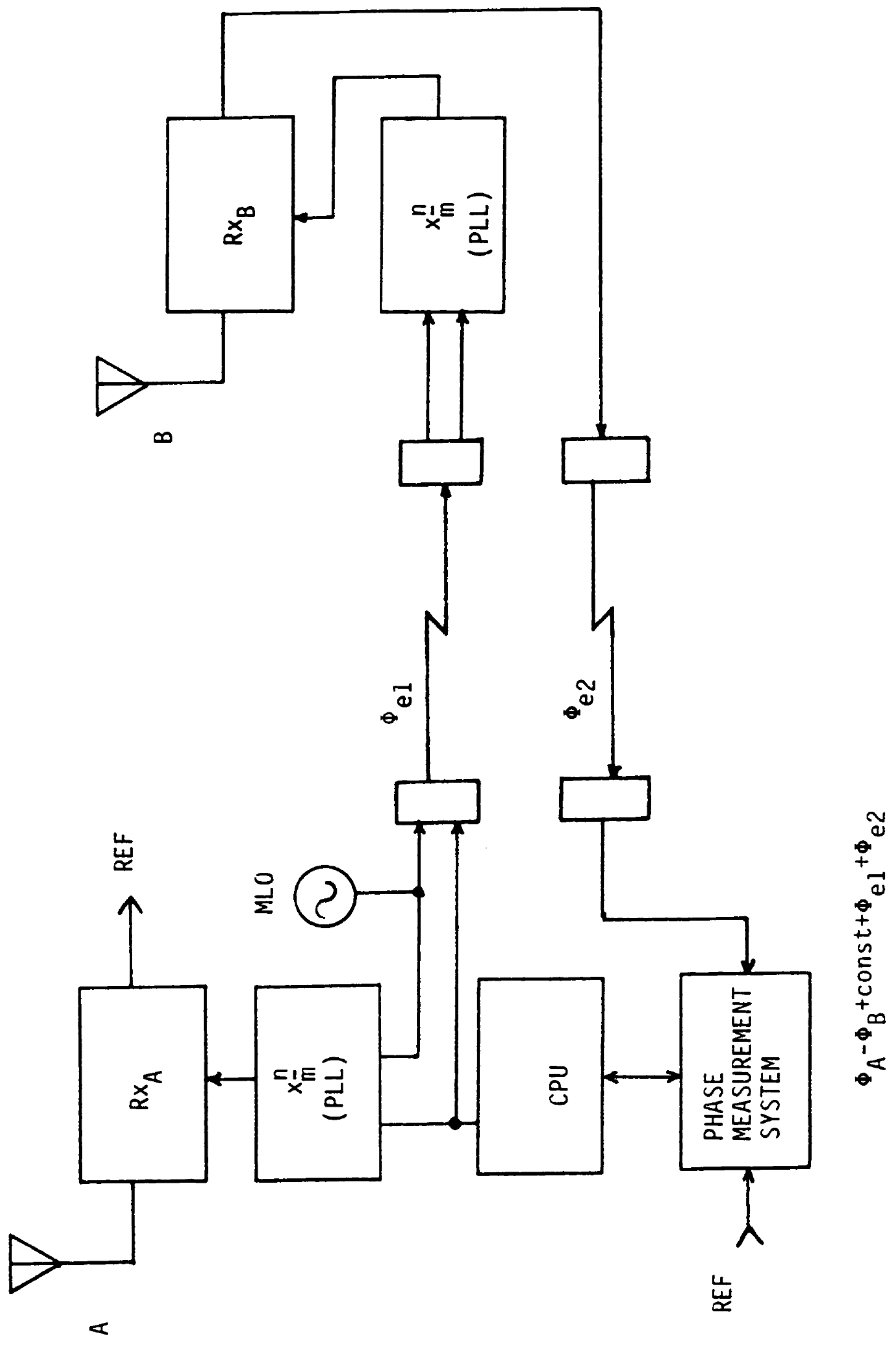


Figure 6.2.b Method 2 block diagram.

6.2.c. It is implemented by transmitting a 10.7 MHz reference to the remote location and also placing the phase measurement system at the remote location. In addition, the MLO signal is also distributed in accordance with the technique described in method 1. This offers the advantages of method 1 since the PLLs are not required to remain precisely locked at multiple remote locations.

The communications link from the control location to the remote location is somewhat complicated since it must send 10.7 MHz and the variable MLO frequency. The return link however is somewhat simplified and must only send data to the reference site. Section 6.3 specifies the best method to implement in a final system.

6.2 COMMUNICATION LINKS USED FOR THE PHASE REFERENCES

Section 6.1 discussed system block diagrams for establishing phase references at remote locations, each utilizing a generically labeled communication link. In preceding discussions, the phase shift (Φ_{e1} and Φ_{e2} from Figures 6.2.a, 6.2.b, and 6.2.c) of this link has been assumed to be a constant and subtracted out of the system as a calibration offset. This assumption is valid if cables are used between the remote and the control locations. If the link is atmospheric, this assumption generally is no longer valid. The link must be adequately analyzed since

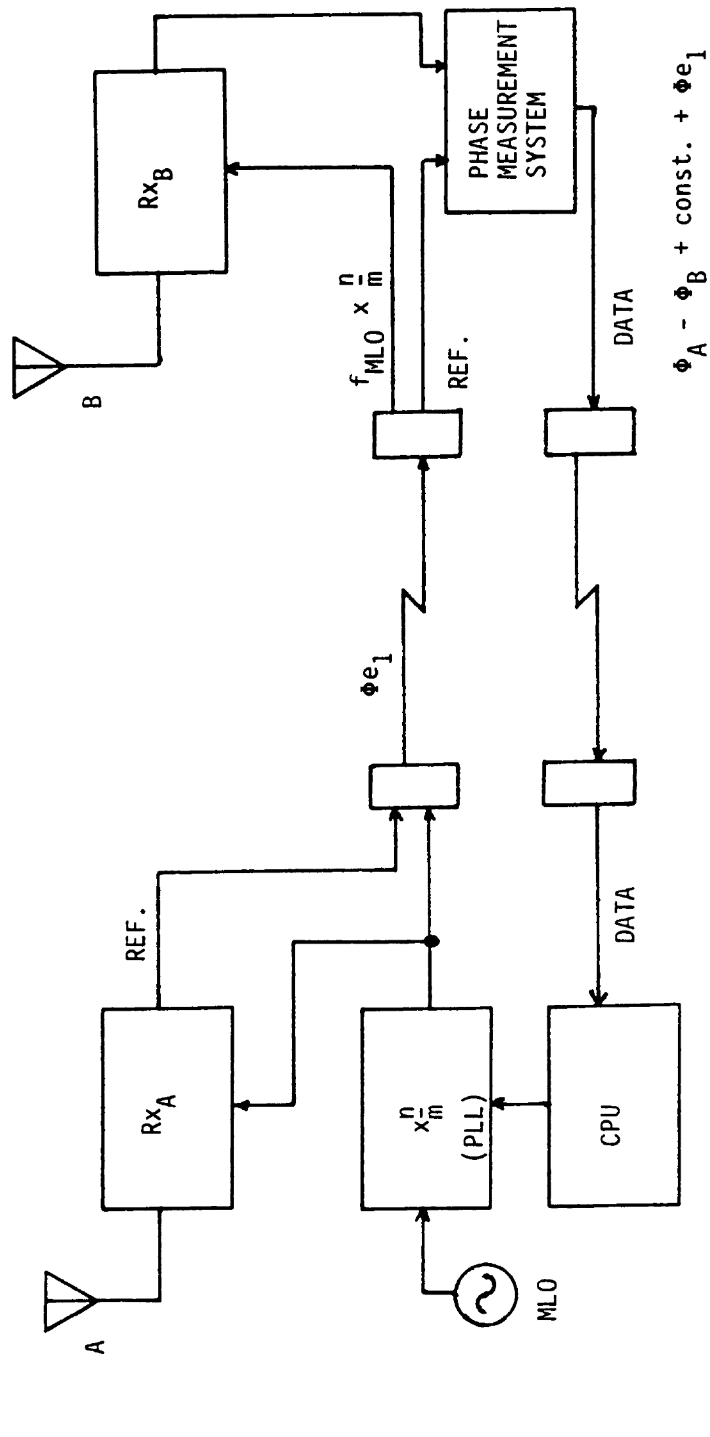


Figure 6.2.c Method 3 block diagram.

Φ_{e1} and Φ_{e2} actually vary through time as a function of the environment around the APAC system. This section addresses the implementation of both RF and optical links for this purpose.

6.2.1 RADIO FREQUENCY LINKS

Figure 6.3 shows two antennas mounted at fixed positions separated by a nominal distance of 2 km. These two antennas are the transmitting and receiving locations of the communications link which takes the phase reference signal from the reference (control) location to the remote receivers. From a propagation perspective, two factors will affect the phase of the signal arriving at the link receiver. The first factor is multipath due to reflections. The second factor is change in the propagation path length due to atmospheric conditions. Both of these factors must be taken into account to ensure stability of the phase reference.

Between the transmitting and receiving antennas is the APAC measurement area. Inside the measurement area are moving metal objects (such as construction equipment, etc.) which could affect the multipath. As these reflecting objects move, phase errors in the MLO input to the remote receivers will result. In order to satisfy the requirement of constant phase at each receiver, the transmitted beam must

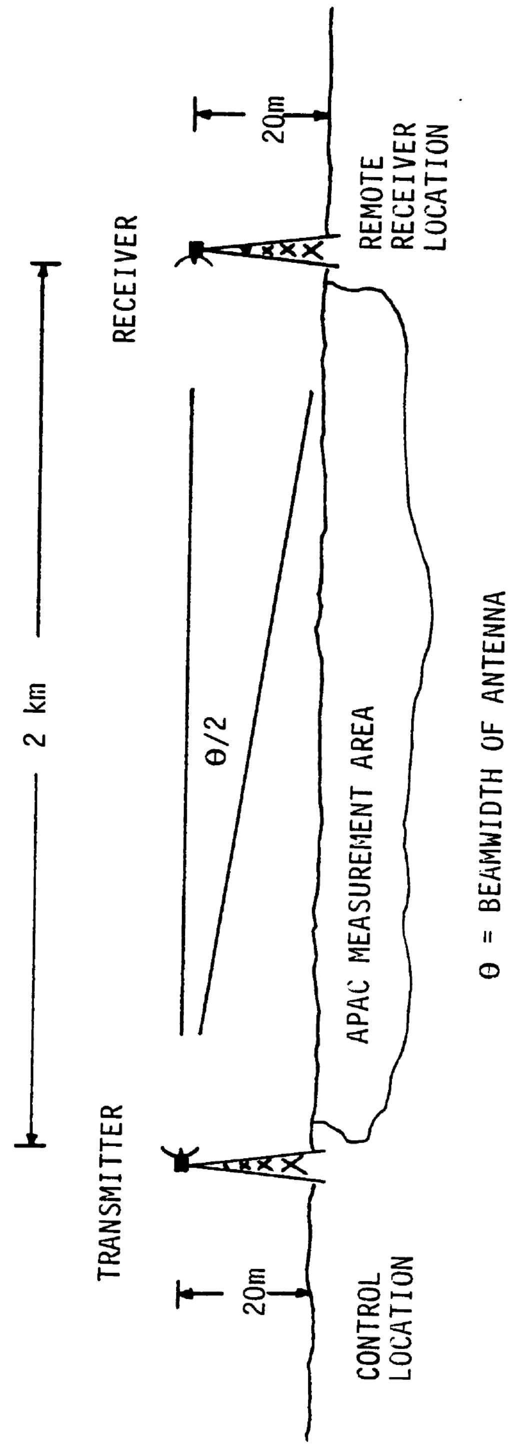


Figure 6.3 Physical location of radio link used for phase reference.

be directed away from the measurement area and towards the communication link receiver antenna.

To estimate the size of the antenna for this requirement, assume that the communication link is over flat earth, with the antenna on a tower of 20 meters height. The beamwidth required to satisfy this is equal to 1.2 degrees. From an approximation given in [27]:

$$D = 75\,\lambda/\theta \tag{6.1}$$

where: θ = antenna beamwidth
λ = wavelength

At HF (30 MHz) and UHF (450 Mhz) the antenna diameter requirements are 600 and 40 meters respectively. These antenna sizes are not practical for this application.

At 30 GHz the antenna diameter is 0.6 meters and is more realistic than at UHF or VHF frequencies. The first requirement for reduction of multipath reflections from the ground and moving equipment has now been satisfied. However, one more factor must be considered. The second factor, variations in path length due to variations in the atmospheric index of refraction, must be investigated.

A model that approximates the index of refraction, n, of air is given by [3] and is based on experimental data:

$$N(\text{laser}) = \frac{77.6}{T} p \qquad (6.2)$$

$$N(\text{uWave}) = \frac{77.6}{T} \left(p + \frac{4810e}{T}\right) \qquad (6.3)$$

where: $N = (n - 1)\text{x}10^6$ = deviation from unity
T = temperature (degrees Celsius)
p = total atmospheric pressure (millibar)
e = partial pressure content of water vapor pressure (millibar)

Observing equation (6.2) and (6.3) some generalizations can now be made. At microwave frequencies, the index of refraction depends more on water vapor content than at optical frequencies. At laser frequencies, the optical path length of laser light is almost independent of water vapor content. Because of the spatial random variations of water vapor content, the laser system offers superior phase stability compared to a microwave link in this application. From this generalization, it can be concluded that optical methods are preferable and should be employed for the phase reference. When a microwave or optical link is used, the 450 MHz phase reference signal must be modulated onto the microwave or optical carrier.

6.2.2 LASER COMMUNICATION LINKS

6.2.2.1 LASER SYSTEM OVERVIEW

In the previous section it was concluded that when compared to microwave frequencies, laser frequencies offer better phase stability in terms of propagation through the atmosphere. Other factors must also be investigated. Figure 6.4 [4] shows atmospheric absorption as a function of wavelength for frequencies obtainable from ruby lasers. The variations in absorption are the result of molecular resonant lines. Table 6.1 [4] shows the absorption line free coherence bandwidth for ruby, GaAs, and CO_2 laser wavelengths for atmospheric propagation. It is obvious that ample bandwidth exists for transmission of a 450 MHz modulation for the phase reference.

6.2.2.2 SOME BASIC ASSUMPTIONS

Numerous experiments have been conducted demonstrating laser communications over lengths of 4, 24 and 36 km [16,24]. The path length for the proposed APAC system would be limited to about 2 km. Furthermore, the critical specification to satisfy in the APAC application is the change in phase of the modulation with respect to time, not amplitude deviations. For these reasons, this analysis assumes the effects of attenuation and fading are negligible. Also, the electronics supporting the system will be neglected from this analysis. Excellent

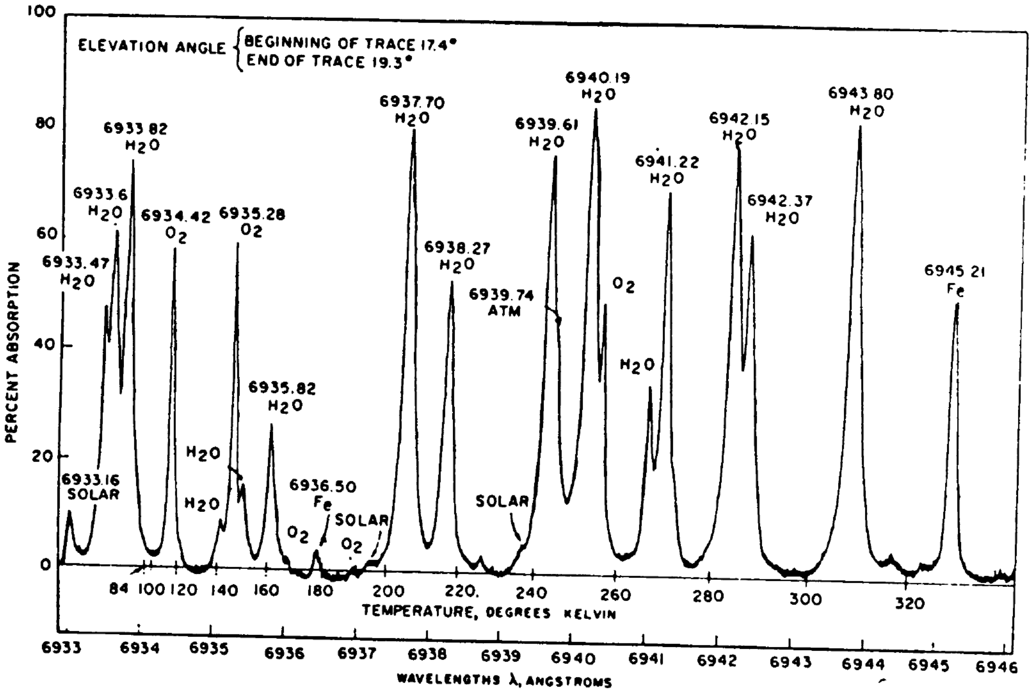


Figure 6.4 Atmospheric absorption lines for ruby laser wavelengths, reprinted from [4].

Table 6.1 Absorption line free coherence bandwidth [4].

wavelength (micrometers)	bandwidth (GHz)	type laser
0.6943	25	ruby
0.8446	168	GaAs
10.5912	≥ 2	CO_2

discussions of these additional parameters are provided in references [26,10,18]. Commercial optical equipment is readily available that is capable of 450 MHz transmission over 2 km distances.

6.2.2.3 LASER COMMUNICATION SYSTEM DESCRIPTION

A basic system block diagram of an optical link is shown in Figure 6.5. Suppose that a master local oscillator (pulse train) is generated for distribution to each of the remote receiver locations. The MLO clock pulse is applied to the optical link transmitter, modulating the laser carrier. The form of modulation is on-off-keying (OOK) and is at the rate of the MLO.

The laser beam propagates through the atmosphere, introducing some signal attenuation. In addition, the atmosphere affects the phase of the carrier and distorts the shape of the pulse. At the optical link receiver, the received signal consisting of a pulse train is recovered and applied to a band pass filter which recovers the fundamental frequency. This fundamental frequency is then applied to the receiver MLO input and is the phase reference necessary for the differential phase measurements.

Of concern is the amount of degradation occurring in the

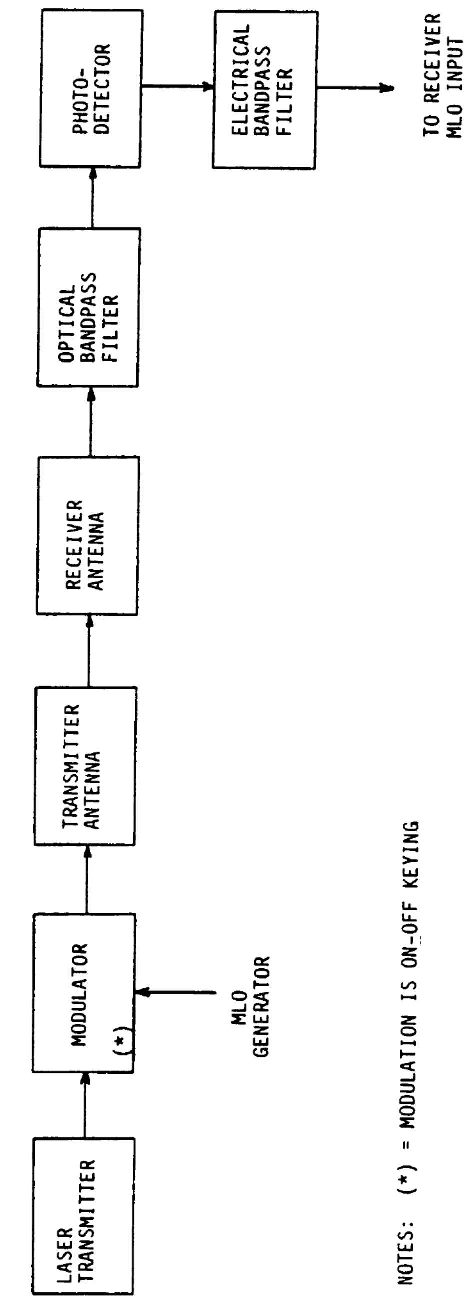


Figure 6.5 Laser communication system block diagram used for phase reference.

relative phase of the recovered 450 MHz signal. The analysis in the remainder of this section addresses the problem of determining to what extent the atmosphere affects the relative phase of the MLO input signal when using an optical link to establish the phase reference.

6.2.2.4 ATMOSPHERIC EFFECTS ON OPTICAL PROPAGATION

The atmosphere is a dynamic and randomly changing environment. The heating effects of the sun, different temperatures and layers of air, wind and other factors influence the characteristics of the atmosphere. Figure 6.6 demonstrates how the atmosphere affects laser light propagation. As the beam propagates it encounters 'blobs' of atmosphere. Inhomogeneous 'blobs' are the result of wind and temperature related disturbances. The turbulence in the path of the beam results in beam steering, scintillation, beam spreading and other degradations. Unlike free space propagation, the atmosphere will alter the phase of the carrier and thus the modulated signal. Atmospheric effects must be investigated to predict performance of the laser communications system.

The structure constant, C_n, described by [33] is a measure of the effect of the atmosphere on laser light propagation in a turbulent atmosphere. C_n^2 is proportional to variance in the index of refraction of the atmosphere. Figure 6.7

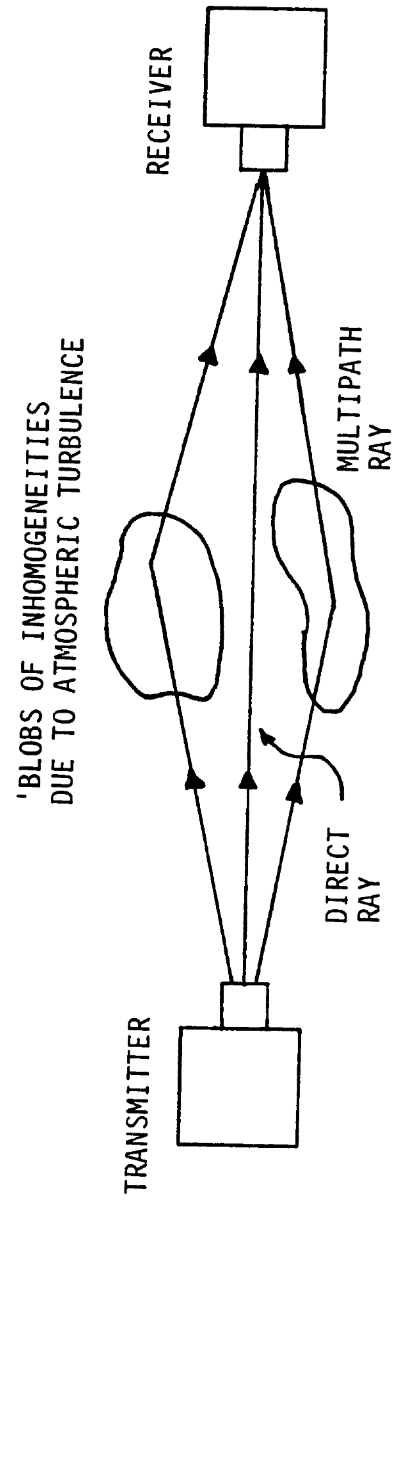


Figure 6.6 Optical propagation in the presence of turbulence.

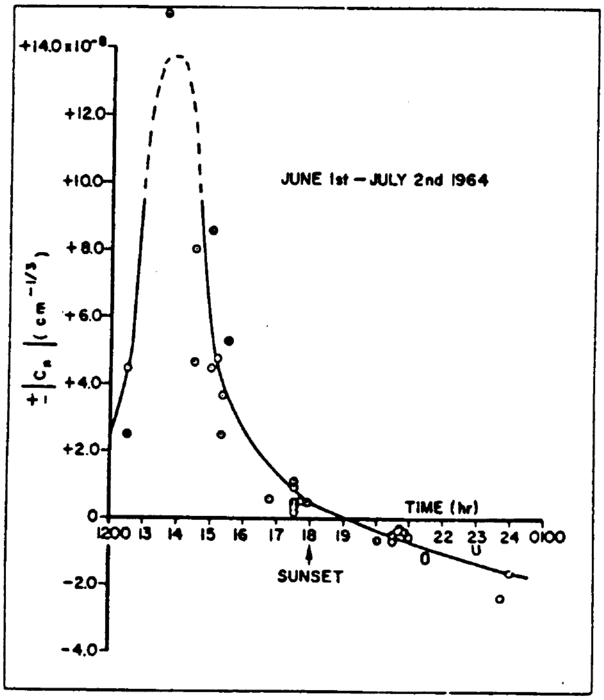


Figure 6.7 Diurnal changes in the structure constant, C_n, reprinted from [16].

shows experimental results of data collected by [16] and characterizes diurnal changes in C_n . The interpretation of C_n is that it represents effects of turbulence on laser light propagation assuming that the atmosphere is frozen in time [33]. The time period for which this frozen-in model remains valid depends the on characteristics of the turbulence, such as wind velocity, temperature, and other factors.

It is necessary that the index of refraction remain constant over a time period equal to the time between calibrations of the APAC system. As shown by [33], the phase of the carrier at the receiver is Gaussian distributed within the time period for which the frozen-in model is valid. As time progresses (outside the calibration period) the local mean of the characteristics of the atmosphere could be expected to vary. At this new increment of time, there is a new mean value of index of refraction expected to be independent of the previous time period.

A model given by Brookner [3] expresses the atmospheric index of refraction , n, as follows:

$$n = n_o + \Delta n \tag{6.4}$$

where: n_o = average atmospheric index of refraction
Δn = deviation of index of refraction from the average value

Unlike equations (6.2) and 6.3), equation (6.4) accounts for atmospheric turbulence. According to Brookner, the mean square deviation in index of refraction $<\Delta n^2>$ is given by:

$$<\Delta n^2> = \frac{1}{2} C_n^2 L_o^{2/3} \qquad (6.5)$$

where L_o is defined as the outer scale of turbulence and is equal to the largest inhomogeneity for which the blob size holds. The value of L_o depends on height, h, above the ground and is equal to:

$$L_o = \sqrt{4h} \quad (m) \qquad (6.6)$$

The RMS value of the variations in phase of the received optical signal depends on the RMS change in index of refraction. Therefore, it can be stated the RMS phase fluctuation is equal to

$$\Delta\Phi(RMS) = \frac{2\pi}{\lambda} L \, [<\Delta n^2>]^{1/2}$$

$$= \frac{2\pi}{\lambda} L \sqrt{\frac{1}{2} C_n^2 L_o^{2/3}} \qquad (6.7)$$

where: λ = wavelength of the modulated signal
L = physical length of propagation

The value of C_n in strong turbulence is given by [26] and is equal to $5 \times 10^{-7}\ m^{-1/3}$. For the APAC application, L is equal to 2000 meters and the modulation wavelength is equal to 0.67 meters. If the height of the optical receiver and transmitter are 20 meters, the value of L_0 is about 9 meters. These conditions result in an upper bound in the RMS phase variation of the 450 MHz modulation equal to 0.25 degrees. This amount is negligible for the APAC system.

Also using the frozen-in model, [4] gives a predicted RMS frequency spread of less than 150 Hz for laser propagation along a path of 1 km. Another interesting note is that increasing the size of the receiver aperture tends to decrease the RMS frequency spread. According to [13] it is reasonable to estimate that a time shift in the carrier (laser) will approximate the time shift of the modulated signal. In the APAC system, the carrier frequency is about 1×10^{14} Hz and the modulation is at about 450 MHz. Therefore, it can be stated that the RMS frequency spread (at the laser frequency) of 150 Hz should have negligible effect on the phase of the modulation.

The above discussions seem to indicate that the laser system should perform excellently for this application. It is asserted that the above discussions pertain to clear weather conditions in the presence of atmospheric turbulence. Dispersion is the topic of the following section.

6.2.2.5 ATMOSPHERIC DISPERSION

Atmospheric dispersion imposes limits on the rate at which a pulse train can be transmitted using an atmospheric optical link. For the APAC system, assuming a 50 percent duty cycle for pulses at 450 MHz rate, the corresponding pulse width is 1111 picoseconds. During clear weather conditions, (i.e., absence of fog, cloud etc) the dispersion will result in pulse widening of only about 0.1 picoseconds [5]. This is a negligible amount compared to the modulation pulse width.

As one would expect, clouds and fog result in increased dispersion. Bucher and Lerner [6] performed experiments using a physical path length of 6 km though cloud cover. An optical communication system was used to investigate the effect of propagation delay by transmitting 30 nanosecond pulses. The received pulses were on the order of 1 - 10 microseconds in duration when cloud cover was in the path. Theory developed by [17] is in agreement with these results.

Liu and Yen [22] were able to derive equations from the parabolic wave equation assuming forward scattering in a turbulent atmosphere. They computed the mean arrival time and mean square pulse width for an electromagnetic wave traveling through the atmosphere. Although a solution to these equations is outside the scope of this paper, an example can be stated for qualitative purposes. Assuming a wavelength of 0.7 nm, beam diameter of 5 cm, path length of 5 km, and a cloud with water density of 0.06 g/m^3, there will be a pulse broadening of approximately 10 nanoseconds. This pulse broadening results in overlapping of successive pulses such that all phase information of the 450 MHz modulated signal is lost.

Considering the pulse width for the 450 MHz system of 1111 picoseconds, it is obvious from both experimental data and theory that cloud and fog cover severely degrades the system performance beyond use. Under clear weather conditions, optical methods offer the best performance when compared to the other methods discussed.

6.3 PROPOSED COMMUNICATION LINK FOR POSITIONING SYSTEM

6.3.1 LINK FOR REMOTE LOCATION

An APAC system communication link, where the use of cables is impractical, is now specified. From Section 6.2, the laser link was determined to be the superior method for transmitting a phase reference. Section 6.1 presented several methods of transferring a phase reference and data between the control and the remote receiver locations. A comparison of these methods is necessary to determine which method is superior in terms of cost effectiveness and reliability.

Both the MLO and the IF signal must be transmitted between the control and remote locations during system operation. Methods 1 and 2 require that these two signal travel in opposite directions. This implies two laser transmitters and receivers for each remote location. Method 3 places the phase measurement system at the remote location. From section 6.2, there is ample bandwidth to transmit both the MLO and the IF signal on the same channel. This method uses only one laser transmitter and receiver. Data controlling the phase measurements system can also be sent on the same channel.

Phase data from the phase measurement system at the remote location is returned to the control site for processing by the CPU. Phase information ranges from -180.0 to +180.0 degrees. Using standard ASCII data format, 2 bytes or 16 bits contains all of this information. A commercially available radio modem operating at 19.2 kBaud is capable of supporting the 2 bytes per millisecond requirement.

6.3.2 LINK FOR BEACON LOCATION

The APAC system requires frequency switching. The beacon must perform switching between about 10 different frequencies for ambiguity resolution. The most efficient way of accomplishing this seems to be to transmit the frequency data during system start-up or calibration. Each frequency is then assigned a frequency number, eg. f1,f2,...,f10. During operation of the system, it is only necessary to switch to each of these pre-programmed frequencies. In this manner, only one byte (8 bits) is required during the 1 millisecond period. A commercially available radio modem operating at 9.6 kBaud will support the 1 byte per millisecond requirement.

6.4 CAD TO OPERATOR INTERFACE COMMUNICATION LINK

The ultimate purpose of the APAC system is to enhance the performance of construction equipment in real-time operations. The productivity of an equipment operator is increased when data about the environment is continuously available to the operator. This section considers the feasibility of providing data to an operator of the construction equipment in real-time.

From Section 6.1, it is noted that the CPU is physically placed at a common control location. The CAD system is also at the same location and directly connects to the CPU for position data. The operator is performing tasks somewhere in the measurement area, perhaps as much as 2 km from the control location.

For the purposes of this study, a commercially available computer system is assumed to be available and to provide the capability for any numerical calculations which are required. The SUN model 4/260 computer is currently being used as a development tool for APAC in the Civil Engineering Department at V.P.I. and S.U. The SUN computer utilizes a Raster Technologies 'GX4000' video driver to display high resolution graphics on a monitor. A '9 dial control box' provides the operator a means of interacting with the CAD system. The remainder of this section considers the problem of placing the '9 dial control box' and the display monitor at an operator location away from the SUN computer. Hence, the need for a communications link.

6.4.1 '9 DIAL CONTROL BOX'

According to the Raster Technologies GX4000 9 Dial Control Box User Guide, data is sent from the 'GX4000' to the '9 dial control box' at a rate of 30 packets per second. Each packet consists of 4 bytes. Therefore, 120 bytes are sent each second. A commercially available 1200 Baud digital radio modem should support this requirement.

6.4.2 RASTER TECHNOLOGY HIGH RESOLUTION GRAPHICS DISPLAY MONITOR

Data transfer also occurs between the 'GX4000' and the display monitor. According to the Raster Technology User Guide, the display contains 1280 columns x 1024 lines which form 1,320,720 pixels. The refresh rate of the display is specified at 60 Hz. Therefore, this would imply that data must transfer at a rate of corresponding to 78,643,200 pixels per second. Assuming that the information is to be transmitted digitally, each pixel would be sampled and quantified by an eight bit or one byte digital representation. The result is 629,145,600 bits per second. The RF bandwidth requirement for this data rate is in excess of 600 MHz.

Transmission of signals with 600 MHz bandwidth requires special consideration for this application. The frequency of a carrier would have to be at least 10 GHz, or perhaps in the millimeter wave region. It is the opinion of the author that this is probably an inefficient approach in terms of hardware costs. Several alternative approaches are suggested next.

6.4.3 DATA COMPRESSION FOR DISPLAY MONITOR

A paper written by [19] summarizes methods of data compression including zero memory methods, predictive coding, transfer coding, and hybrid coding. Table 6.2 lists methods of data compression, typical data rates and the

Table 6.2 Methods and performance of image data compression, reprinted from [19].

Method	Typical Average Rates Bits/Pixel	Comments
<u>Zero Memory Methods</u>		Simple to implement
PCM	6-8	
Contrast Quantization	4-5	
Pseudorandom Noise - Quantization	4-5	
Line Interlace	4	
Dot Interlace	2-4	
<u>Predictive Coding</u>		
Delta Modulation	1	Performance poorer than DPCM over sample data for improvement.
Intraframe DPCM	2-3	Predictive methods are generally simple to implement, are sensitive to data statistics. Adaptive techniques improve performance substantially. Channel error effects are commulative and severely degrade image quality.
Intraframe Adaptive DPCM	1-2	
Interframe Conditional - Replenishment	1-2	
Interframe DPCM	1-1.5	
Interframe Adaptive DPCM	0.5-1	
<u>Transform Coding</u>		
Intraframe	1-1.5	Achieve high performance, small sensitivity to fluctuation in data statistics, channel and quantization errors distributed over the image block. Easy to provide channel protection. Hardware complexity is high.
Intraframe Adaptive	0.5-1	
Interframe	0.5-1	
Interframe Adaptive	0.1-0.5	
<u>Hybrid Coding</u>		
Intraframe	1-2	Achieve performance close to transform coding at moderate rates (.5 to 1 bit/pixel). Complexity lies midway between transform coding and DPCM.
Intraframe Adaptive	0.5-1.5	
Interframe	0.5-1	
Interframe Adaptive	0.25-0.5	

corresponding performance. Note that it should be possible to compress 8 bits per pixel to around 0.5 bits per pixel.

The sixteen-fold reduction in data rates results in around 39 Mbit/s. A T1 data frame is a standard data structure and conveys 1.544 Mbit/s [7]. A data link supporting this requirement would need to convey 25 T1 frame structures.

It is asserted that the process of data compression degrades the quality of the image. The question arises as to what is acceptable for the construction equipment application. This is quite a subjective question and should be addressed in future work.

6.4.4 OTHER ALTERNATE APPROACHES

Consider a different approach to the problem of providing data display to the equipment operator. Suppose that the contents of the entire monitor were held in memory on board the equipment location. The memory contents itself would probably be in the form of compressed data [19]. It is only necessary to transmit the information necessary to update the display. A reasonable assumption is that this data could be conveyed in somewhere between one and five T1 data structures.

Another entirely different approach is to place the CAD system (eg. SUN computer) on the construction equipment. The position data would then be transferred from the CPU at the common control location to the SUN computer at the construction equipment. Real-time position data is only a small fraction of the display data.

It is difficult to answer the questions of how data could be provided to the equipment operator for the general case. Human factors, environment, level of automation, and type of automation are all variables that affect what the final system will look like in the future. This is an excellent area for future research work.

CHAPTER 7

EXPERIMENTAL RESULTS OF THE APAC SYSTEM

7.1 TESTS ON A 4.2 x 4.2 x 2 METER SITE

Measurements were made using the position location system described in Section 3.1. The minimum number of receivers, that is four, for a three dimensional test were used. A 4.2 x 4.2 x 2 meter test corral was constructed to facilitate the measurements. A series of strings and lines were used to determine the actual beacon location for comparison to the measured position. A set of fifty measurements was made over dry earth at randomly selected locations within the test corral.

The floor of the test corral was then flooded with water. The same set of randomly selected locations were re-measured for comparison to the dry earth case. Multipath errors due to increased reflections of water are expected to result in increased errors in position.

Figure 7.1 shows results of both the dry earth and over water measurements [12]. For the dry earth case, the position errors are less than 15 centimeters 50 percent of the time. As expected, errors in the measurements above water were notably increased.

Results from above measurements were used by Feuerstein to develop a computer simulator model and show that the phase errors in the measurement system are Gaussian distributed [11]. Figure 7.2 shows experimental results and the results from the computer simulation. Figure 7.3 shows the distribution of phase errors.

A thorough analysis of the APAC system test results will be the research dissertation topic soon to be written by Feuerstein. Methods including (but not limited to) redundant receivers and multiple antennas are expected to net substantial improvement in accuracy. This however, remains to be demonstrated in practice.

7.2 TESTS ON A 80 x 100 METER SITE

Work is currently underway at the time of this publication to evaluate performance of the position location system on a site larger than the one described in Section 7.1.

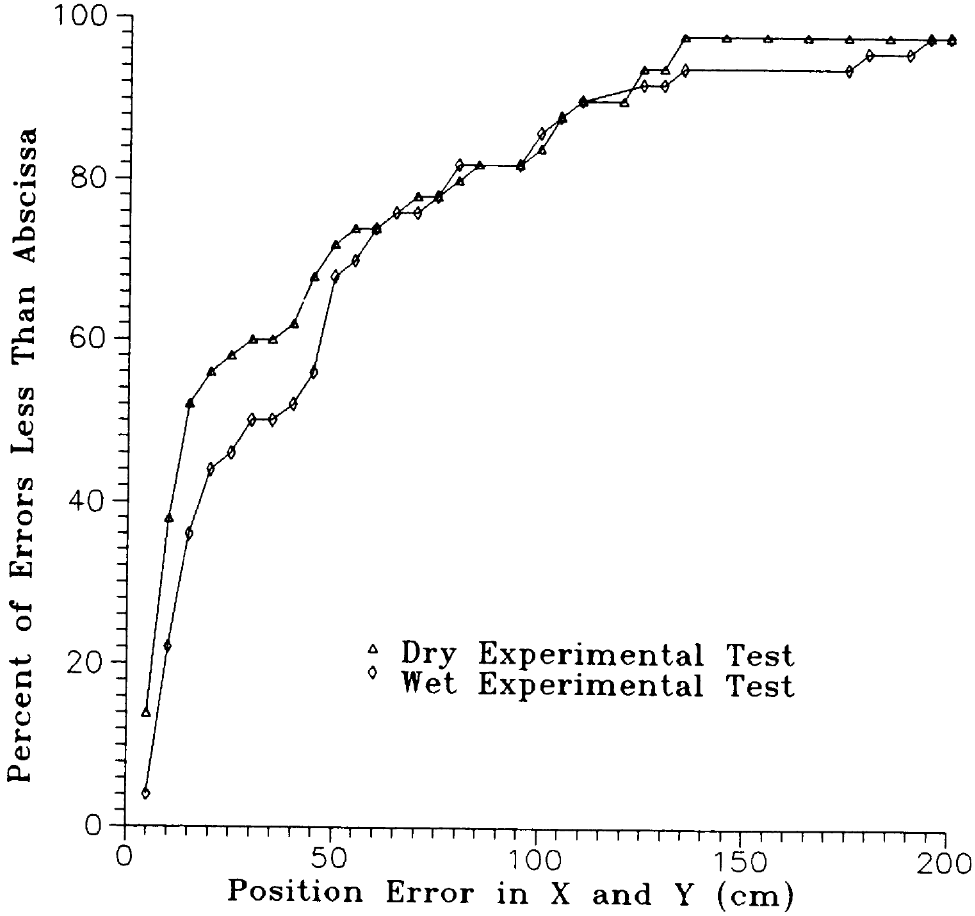


Figure 7.1 APAC test data from measurements in a 4.2 x 4.2 meter area [12].

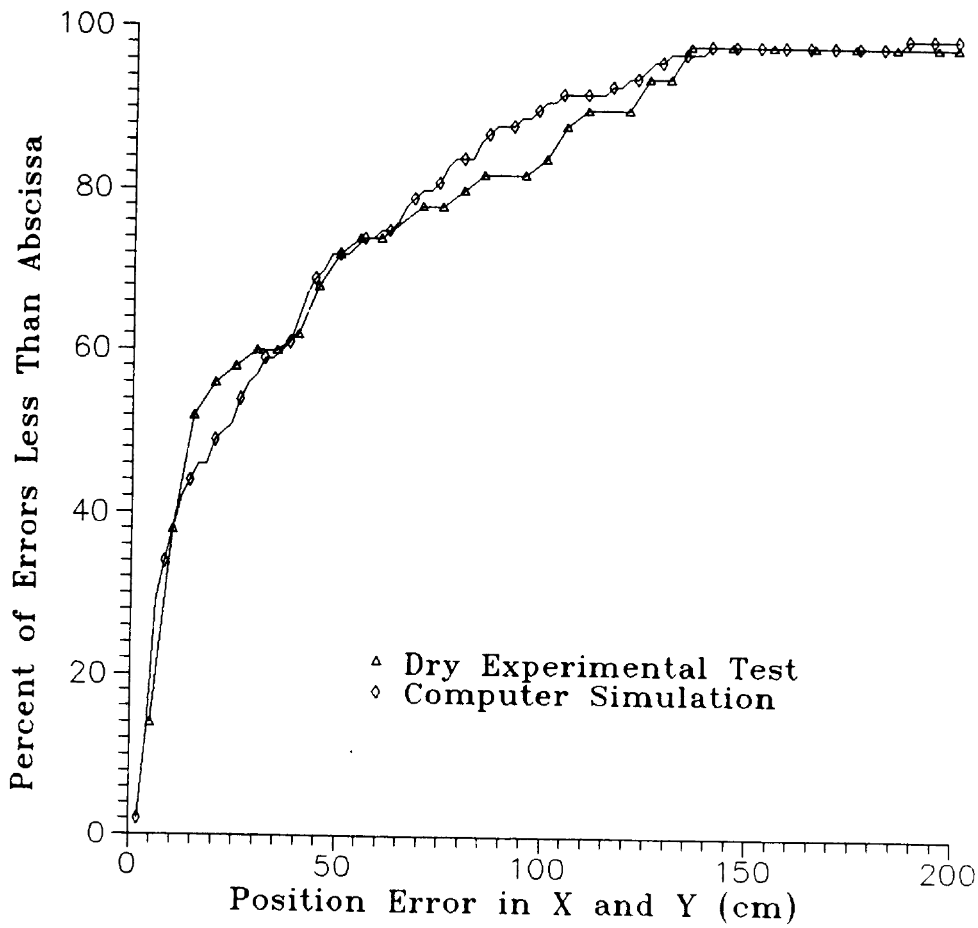


Figure 7.2 Comparison of simulator and test data [12].

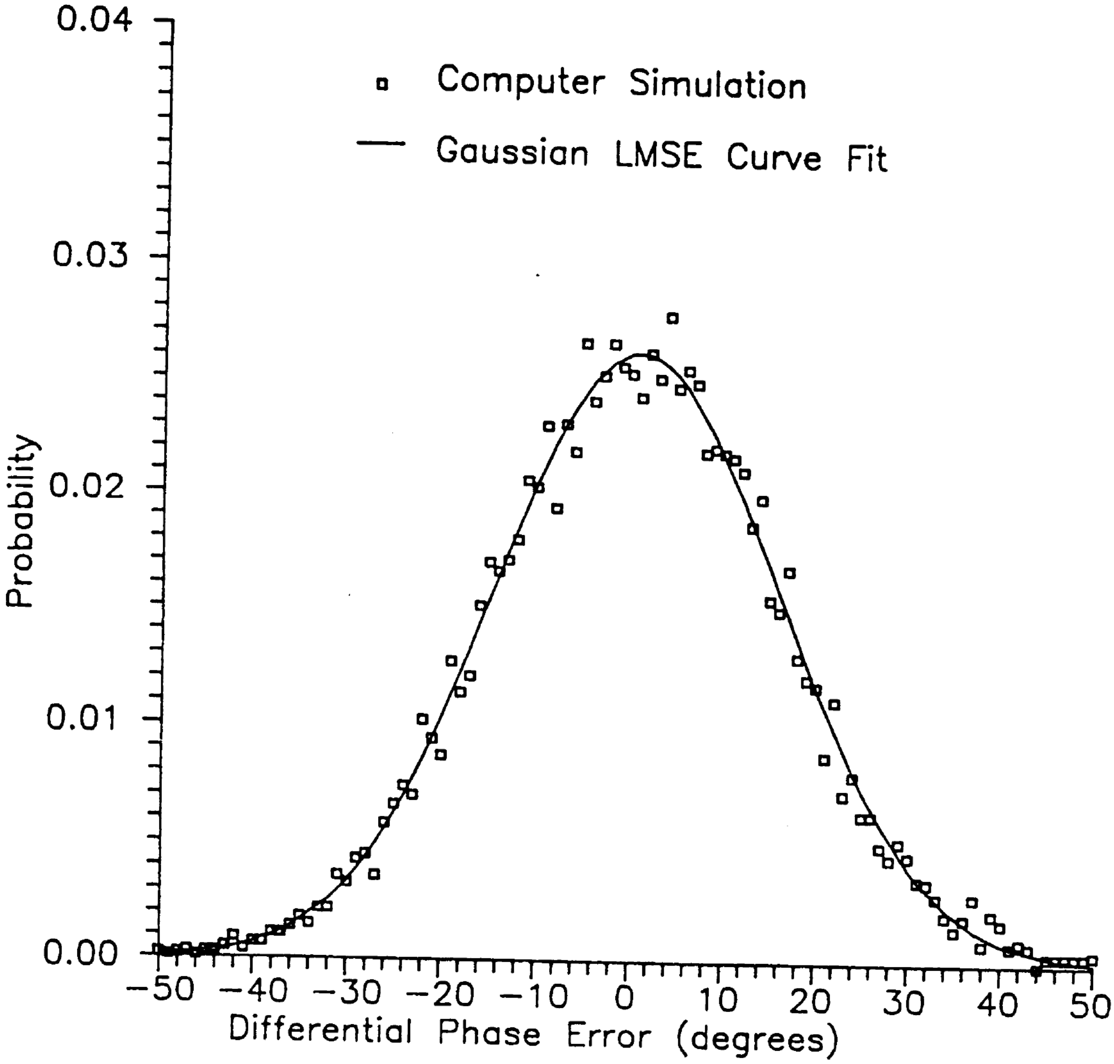


Figure 7.3 Distribution of phase errors from computer simulation data, reprinted from [11].

Hardware progress has been hampered by technical problems and a lightning storm. Results are as yet inconclusive but will be published in the future when available.

CHAPTER 8

CONCLUSIONS

The necessity for development of a real-time position location system capable of centimeter accuracy has been outlined. The purpose of APAC is to fulfill the vital requirement of providing data to an equipment operator on a construction site to enhance performance and increase productivity.

Several methods for position location have been evaluated for merit in a real-time, centimeter accuracy system. Current distance measurements methods require user interaction and cannot satisfy the real-time requirement. Pulse timing systems require large bandwidths, hence more power than continuous wave systems. The hyperbolic navigation system seems to offer advantages over other methods and is a likely candidate for satisfying the real-time, centimeter accuracy specification. Basic hyperbolic navigation terms have been defined. The system operates on measurements of differential phases.

A general description of the hardware of the APAC prototype system has been provided. The system operates at frequencies between 400 and 450 MHz and employs frequency switching. The technique of frequency and phase differencing is used to resolve lane ambiguities.

The design and analysis of a 10.7 MHz phase measurement system has been provided in detail. Of the phase detectors presented, sinusoidal response obtained with an analog multiplier offers superior performance in the presence of noise. Phase measurements accurate to within about 5 degrees have been demonstrated at 5 ms intervals. The resolution and accuracy of the phase measurement system are sufficient for use in APAC. Temperature stabilization and automated calibration have been recommended for use in a field environment.

Dipole antennas were used for the prototype system measurements. A problem exists with the phase center perturbation of the beacon antenna. It is believed that this problem can be resolved but a solution remains to be tested and evaluated. Future work should also include additional data collection with Yagi antennas at the receiver locations.

Bandwidth and receiver phase shift correction are important receiver specifications and have been outlined. The receiver hardware considerations and specifications outlined are not generally satisfied in commercially available equipment. An important area of future work would be the design and development of receivers which adhere to these specifications.

For the beacon and master local oscillator, Oven Controlled Crystal Oscillators (OCCO) are necessary to provide frequency stability. The beacon need not transmit more than 0 dBm of power on a 2 km measurement site, based on optimal receiver design.

There are two methods of calibration, atmospheric and cable. The atmospheric calibration has the advantage of being simplified in terms of hardware. Cable calibration on a small site is probably superior to atmospheric calibration but suffers from unacceptable cable loss on a larger site.

Communication links are necessary to perform differential phase measurements between multiple receiver locations. In complexity, this link can vary somewhere between a cable and a laser communication link. Three methods or schemes of transferring phase and data between a common control location and a remote receiver location have been presented. An argument was provided for which method is probably the most efficient.

Radio waves (micro-waves and below) are inferior to optical links for transmitting a phase reference to remote locations at 2 km range. A laser system which can be used for transmitting a phase reference was described. An analysis of the RMS change in atmospheric index of refraction and spectral widening seem to indicate that the phase of a 450 MHz modulation should not be altered within a calibration period specified at 10 seconds. The dispersive effects of atmosphere are negligible in clear weather. In the presence of clouds, fog and rain, the atmosphere becomes more dispersive and does not permit one to transmit and receive an optical pulse train with a period of 2 ns.

Position error data from the prototype APAC system on a small scale site (4 x 4 meter) has been collected and is presented. Position errors were within 15 centimeters, 50 percent of the time. Analysis of this data and suggestions on improvement of errors is an area of future work. At the time of this publication, the prototype system was in the process of being upgraded to a larger site (100 x 80 meters). Hardware requirements necessary for reliable, accurate and consistent differential phase measurements are problems that must be overcome to continue development.

APPENDIX A

SOFTWARE FOR PHASE MEASUREMENT SYSTEM

The following computer programs were written by Andrew Dornbusch for the APAC phase measurement system. The programs were developed for use on an IBM compatible personal computer and on a BCC-52 microcomputer available from Micromint Inc., Massachusetts.

A.1 MAIN PHASE DETECTOR ROUTINE

The main phase detector routine is the program the runs in the BCC-52 microcomputer. The BCC-52 microcomputer accesses the phase detector circuits (A/D converter) and performs computations that provide phase data corresponding to the phase off the applied signal. The program is written in 8031 Assembly language and follows:

```
; FILE          FINAL.51
;               APAC phase measurment system
;
;               Version 2.34
;               Written by Andrew Dornbusch
;               3/26/89

CAL_ARRY EQU 4000H                 ; starting at the 7K boundary (7168-7183)
P_OFF    EQU 4010H                 ; 32 bytes (8 boards, 2 per, 2 bytes per)
         ; NOTE: P_OFF is stored low-byte, hi; all others are hi-byte, lo
INPHASE  EQU 4030H                 ; two bytes   (high byte-low)
QUAD     EQU 4032H                 ; two bytes
INDEX    EQU 4034H
PHI      EQU 4038H
CARD0    EQU 403AH
CARD1    EQU 403BH
TEMP     EQU 403CH
DATARRY  EQU 4040H
CR       EQU  0DH
LF       EQU  0AH
ESC      EQU  1BH


        ORG  0000H
START:
         AJMP INIT

; Initialize system for boot-up
; must also init serial port, timers, etc.
         ORG  0100H
INIT:
         clr  EA
         mov  SP, #30H             ; set stack pointer
         mov  IE, #00H             ; disable all interrupts

         mov  SCON, #52h           ; set serial port to 4800/8/N/1
         mov  TMOD, #20h
         mov  TH1, #0fah
         setb TR1

         lcall init8255

; Transmit startup message out serial port
         mov  DPTR, #STRING
         clr  A
         movc A, @A+DPTR
X_STRING:
         jnb  TI, $
         clr  TI
         mov  SBUF, A
         inc  DPTR
         clr  A
         movc A, @A+DPTR
         cjne A, #ESC, X_STRING


; Read data from boards specified and return via argument stack
; Wait until character received from serial port.
GO:
         jnb  RI, $                ; wait until command received from host
```

(A.1) program listing.

```
        subb A, #41H              ; use characters 'a' on
        jc   GO

        clr  C                    ; 'a'-'d', take full phase measurement
        subb A, #04H              ; 65-68
        jc   MEASURE

        clr  C                    ; 'e'-'h', take raw A/D measurement
        subb A, #04H              ; 69-72
        jnc  CONT
        ljmp RAW
CONT:
        clr  C
        subb A, #02h
        jc   GO

        clr  C
        subb A, #01H              ; 'k', do phase system calibration
        jnc  FOOBAR               ; 75
        ljmp CAL
FOOBAR:
        subb A, #08h              ; 'l'-'s', transmit system cal values
        jnc  FOOTWO               ; 76-83
        ljmp TXCAL
FOOTWO:
        subb A, #08h              ; 't'-..., tranmit phase offset values
        jnc  FOOTHREE             ; 84-91
        ljmp TXPOFF
FOOTHREE:
        subb A, #08h              ; 92-99, receive calibration values
        jnc  FOOFOUR
        ljmp RXCAL
FOOFOUR:
        subb A, #08h              ; 100-107, receive phase offset values
        jnc  FOOFIVE
        ljmp RXPOFF
FOOFIVE:
        subb A, #01h              ; 108, set relays for testing
        jnc  GO
        ljmp TESTCAL


; take phase measurement and return phase in degrees*10
MEASURE:
        ADD  A, #04H              ; restore value
        MOV  DPTR, #INDEX         ; measurement requested
        MOVX @DPTR, A
        LCALL READING             ; take measurements

        MOV  DPTR, #INDEX
        MOVX A, @DPTR
        LCALL CALCPQ              ; calculate inphase and quadrature

        LCALL MAKEPHI             ; calculate phi from inphase and quad
```

(A.1) program listing.

```
; RAW  takes a raw data measurement for testing/calibrational purposes
RAW:
        ADD  A, #04H
        LCALL READING

; transmit R4:R3, and R6:R5
        MOV  A, #00                 ; transmit zero value preceeding raw
        JNB  TI, $                  ;  measurement value
        CLR  TI
        MOV  SBUF, A

        MOV  A, R3
        JNB  TI, $
        CLR  TI
        MOV  SBUF, A

        MOV  A, #00
        JNB  TI, $
        CLR  TI
        MOV  SBUF, A

        MOV  A, R5
        JNB  TI, $
        CLR  TI
        MOV  SBUF, A

        LJMP GO


; CAL performs an auto calibration of the system, using the external
; switching and phase-shifting equipment controlled by the 82C55 port C.
CAL:
        lcall init8255
        mov  DPTR, #TEMP            ; degrees offset
        mov  A, #00H
        movx @DPTR, A

OUTER:
        mov  R7, #00H               ; counter (0-3)
LOOPBACK:
        mov  A, R7
        rl   A                      ; mult. by four
        rl   A
        mov  B, A
        mov  DPTR, #TEMP            ; add in degree offset
        movx A, @DPTR
        add  A, B
        lcall SETCAL                ; set relays to proper position

        mov  A, R7
        lcall READING               ; read board-pair

; calculate addreses for read values in memory array
        mov  DPTR, #TEMP            ; get degrees offset
        movx A, @DPTR
        rl   A                      ; mult. by eight
        rl   A
        rl   A
        mov  R2, A
        mov  DPTR, #DATARRY         ; get pointer to data array
        mov  A, R7                  ; get board-pair number
```

(A.1) program listing.

```
; and store read values in memory
        MOV  A, R3              ; inphase value goes in lower byte
        MOVX @DPTR, A
        INC  DPTR
        MOV  A, R5              ; quadrature goes in next
        MOVX @DPTR, A

; loop back again for next pair of boards
        MOV  A, R7
        INC  A
        MOV  R7, A
        CLR  C
        SUBB A, #04H            ; four board pairs
        JC   LOOPBACK

; loop back again for measurements at next phase increment
        MOV  DPTR, #TEMP
        MOVX A, @DPTR
        INC  A
        MOVX @DPTR, A
        CLR  C
        SUBB A, #04H            ; four phase positions to measure
        JNC  ENDCAL1
        LJMP OUTER

; reset relays for normal use
ENDCAL1:
        MOV   A, #10H
        LCALL SETCAL


; copy new calibration data into calibration array

; do even boards first
        MOV  R0, #04H           ; count (4->0)
        MOV  R1, #00H           ; DATARRY address offset
        MOV  R2, #00H           ; CAL_ARRY address offset
MOVLOOP1:
        MOV  DPTR, #DATARRY     ; get pointer to data storage array
        MOV  A, R1              ; add offset
        ADD  A, DPL
        MOV  DPL, A
        MOVX A, @DPTR           ; fetch minimum value for channel 2*R0
        MOV  R3, A              ; and save
        MOV  A, #10H            ; add offset for maximum value,
        ADD  A, DPL
        MOV  DPL, A
        MOVX A, @DPTR           ; fetch
        MOV  R4, A              ; and save
        MOV  DPTR, #CAL_ARRY    ; get pointer to calibration array
        MOV  A, R2              ; add offset
        ADD  A, DPL
        MOV  DPL, A
        MOV  A, R3
        dec  A                  ; FUDGE FACTOR
        MOVX @DPTR, A           ; store minimum value
        INC  DPTR
        MOV  A, R4              ; and the max in the next consecutive loc.
        inc  A                  ; FUDGE FACTOR
        inc  A
        MOVX @DPTR, A
```

(A.1) program listing.

```
        MOV  R2, A

        djnz R0, MOVLOOP1        ; loop back

; and now do odd boards
        MOV  R0, #04H            ; count (4->0)
        MOV  R1, #09H            ; DATARRY address offset
        MOV  R2, #02H            ; CAL_ARRY address offset
MOVLOOP2:
        MOV  DPTR, #DATARRY      ; get pointer for data storage array
        MOV  A, R1               ; add offset
        ADD  A, DPL
        MOV  DPL, A
        MOVX A, @DPTR            ; fetch minimum value for channel 2*R0+1
        MOV  R3, A               ; and save
        MOV  A, #10H             ; add offset for maximum value,
        ADD  A, DPL
        MOV  DPL, A
        MOVX A, @DPTR            ; fetch,
        MOV  R4, A               ; and save
        MOV  DPTR, #CAL_ARRY     ; get pointer to calibration array
        MOV  A, R2               ; add offset
        ADD  A, DPL
        MOV  DPL, A
        MOV  A, R3
        dec  A                   ; FUDGE FACTOR
        MOVX @DPTR, A            ; store minimum value,
        INC  DPTR
        MOV  A, R4               ; and then the max in the next loc
        inc  A                   ; FUDGE FACTOR
        inc  A
        MOVX @DPTR, A
        INC  R1                  ; move DATARRY pointer two positions to next
        INC  R1                  ;  minimum value
        MOV  A, #04H
        ADD  A, R2               ; add step to calibration array offset
        MOV  R2, A

        djnz R0, MOVLOOP2        ; and loop back


; Now let's go thru and calculate what the inphase and quadrature measurements
; would be if we use the data we just took.  We will then invert the sign of
; the inphase and quadrature values and store them as the phase offsets.
        mov  DPTR, #TEMP         ; initialize loop count
        mov  A, #00H
        movx @DPTR, A
PQLOOP:
        mov  DPTR, #TEMP         ; get loop count   ( 0 <= temp <= 7)
        movx A, @DPTR
        jnb  ACC.2, LTFOUR       ; test if less than four
        anl  A, #03h             ; if >= 4, mask bit 2 and add 8
        add  A, #08h
LTFOUR:
        rl   A                   ; multiply by two
        inc  A                   ; move pointer to quad value
        xrl  A, #10h             ; TEST- reverse quad phase offsets
        mov  DPTR, #DATARRY      ; and add to base
        add  A, DPL
        mov  DPL, A
        movx A, @DPTR            ; get quadrature value
```

(A.1) program listing.

```
        mov  DPL, A
        movx A, @DPTR           ; get inphase value
        mov  R3, A              ; and save
        mov  DPTR, #TEMP
        movx A, @DPTR
        anl  A, #03h
        mov  DPTR, #INDEX
        movx @DPTR, A
        lcall CALCPQ            ; pass board pair in A, P&Q in R3, R5
                                ; returned in INPHASE and QUAD

; invert sign of returned value preparatory to being stored in p_off array
        mov  DPTR, #INPHASE
        movx A, @DPTR           ; get inphase measurement
        mov  R2, A
        inc  DPTR
        movx A, @DPTR
        mov  R3, A
        mov  A, #00H            ; and invert sign
        clr  C
        subb A, R3
        mov  R3, A
        mov  A, #00H
        subb A, R2
        mov  R2, A

        mov  DPTR, #QUAD        ; get quadrature measurement
        movx A, @DPTR
        mov  R4, A
        inc  DPTR
        movx A, @DPTR
        mov  R5, A
        mov  A, #00H            ; and invert sign
        clr  C
        subb A, R5
        mov  R5, A
        mov  A, #00H
        subb A, R4
        mov  R4, A

; and store phase offsets in p_off array
        mov  DPTR, #TEMP        ; get loop count
        movx A, @DPTR
        mov  R0, A
        anl  A, #03H            ; mask out bit three
        rl   A                  ; multiply by eight (eight bytes allocated
        rl   A                  ;  per board-pair
        rl   A
        xch  A, R0
        mov  C, ACC.2           ; get bit three (offset 1 vs. offset 2)
        xch  A, R0
        mov  ACC.1, C           ; and move to bit 1
        xrl  A, #02H            ; toggle bit 1
        mov  DPTR, #P_OFF       ; get pointer to P_OFF data array
        add  A, DPL
        mov  DPL, A

        mov  A, R3              ; store low byte of inphase
        movx @DPTR, A
        inc  DPTR
        mov  A, R2              ; and high byte (stored lo-hi)
        movx @DPTR, A
```

(A.1) program listing.

```
        movx @DPTR, A
        inc  DPTR
        mov  A, R4              ; and high byte
        movx @DPTR, A

        mov  DPTR, #TEMP        ; increment loop counter
        movx A, @DPTR
        inc  A
        movx @DPTR, A
        clr  C
        subb A, #07H
        jnc  ENDPQLP
        ljmp PQLOOP

ENDPQLP:

; transmit 'OK'
        MOV  A, #79
        JNB  TI, $
        CLR  TI
        MOV  SBUF, A

        MOV  A, #75
        JNB  TI, $
        CLR  TI
        MOV  SBUF, A

        LJMP GO


; transmits four bytes of data from the cal array (one column)
TXCAL:
        add  A, #08h
        mov  DPTR, #DATARRY
        add  A, DPL
        mov  DPL, A
        movx A, @DPTR
        jnb  TI, $
        clr  TI
        mov  SBUF, A

        mov  A, #08h
        add  A, DPL
        mov  DPL, A
        movx A, @DPTR
        jnb  TI, $
        clr  TI
        mov  SBUF, A

        mov  A, #08h
        add  A, DPL
        mov  DPL, A
        movx A, @DPTR
        jnb  TI, $
        clr  TI
        mov  SBUF, A

        mov  A, #08h
        add  A, DPL
        mov  DPL, A
        movx A, @DPTR
```

(A.1) program listing.

```
        ljmp GO


; Transmits four bytes of data from the phase offset array (one board-pair)
; Since data is stored lo-hi in memory, byte reversal must be done when
; transmitting.  The data sent is the two phase offsets for the requested
; board. (not board-pair)
TXPOFF:
        add  A, #08h            ; readjust A to positive value
        anl  A, #07h
        rl   A                  ; multiply by four
        rl   A
        mov  DPTR, #P_OFF       ; get pointer to phase offset array
        add  A, DPL
        mov  DPL, A
        movx A, @DPTR           ; get first byte (high byte)
        mov  R0, A              ; and save
        inc  DPTR
        movx A, @DPTR           ; get second byte (low byte)
        jnb  TI, $              ; and transmit
        clr  TI
        mov  SBUF, A

        mov  A, R0
        jnb  TI, $              ; transmit first byte
        clr  TI
        mov  SBUF, A

        inc  DPTR
        movx A, @DPTR           ; get third byte (high byte)
        mov  R0, A              ; and save
        inc  DPTR
        movx A, @DPTR           ; get fourth byte (low byte)
        jnb  TI, $              ; and transmit
        clr  TI
        mov  SBUF, A

        mov  A, R0
        jnb  TI, $              ; transmit third byte
        clr  TI
        mov  SBUF, A

        ljmp GO


; receives and stores two bytes of data from the host to the cal array
; (one column)
RXCAL:
        add  A, #08h
        clr  C
        rlc  A                  ; mult by two (cal data in pairs)
        mov  DPTR, #CAL_ARRY
        add  A, DPL
        mov  DPL, A

        jnb  RI, $
        clr  RI
        mov  A, SBUF
        movx @DPTR, A
        inc  DPTR
```

(A.1) program listing.

```
        movx @DPTR, A

        ljmp GO


; Receives four bytes of data from the phase offset array.
; Since data is stored lo-hi in memory, byte reversal must be done when
; receiving.  The data received is the two phase offsets for the requested
; board. (not board-pair)
RXPOFF:
        add  A, #08h
        anl  A, #07h
        rl   A
        rl   A
        mov  DPTR, #P_OFF       ; get pointer to phase offset array
        add  A, DPL
        mov  DPL, A
        jnb  RI, $
        clr  RI
        mov  A, SBUF
        mov  R0, A
        jnb  RI, $
        clr  RI
        mov  A, SBUF
        movx @DPTR, A
        inc  DPTR
        mov  A, R0
        movx @DPTR, A
        inc  DPTR

        jnb  RI, $
        clr  RI
        mov  A, SBUF
        mov  R0, A
        jnb  RI, $
        clr  RI
        mov  A, SBUF
        movx @DPTR, A
        inc  DPTR
        mov  A, R0
        movx @DPTR, A
        inc  DPTR

        ljmp GO


; sets the cal relays to the requested position
TESTCAL:
        jnb  RI, $
        clr  RI
        mov  A, SBUF
        lcall SETCAL

        mov  A, #79
        jnb  TI, $
        clr  TI
        mov  SBUF, A

        mov  A, #75
        jnb  TI, $
```

(A.1) program listing.

```
;****************
;* Subroutines *
;****************


; INIT8255 initializes the 8255 for proper input & output
INIT8255:
        mov  DPTR, #0C803H       ; set 8255 for B-out, A-in
        mov  A, #90H
        movx @DPTR, A
        mov  DPTR, #0C801H       ; de-select all phase detector cards
        mov  A, #0FFH
        movx @DPTR, A
        mov  DPTR, #0C802H       ; set cal mechanism to normal operation
        mov  A, #0F0H
        movx @DPTR, A
        ret


; READING takes a set of four measurements of the board-pair pointed to
; by A.  The measurements are returned in R3, R5; R3 contains the inphase
; measurement, R5 the quadrature measurement.
READING:
        MOV  DPTR, #CONVERT
        MOVC A, @A+DPTR
        MOV  R1, A

        MOV  R3, #00H            ; zero out the total inphase
        MOV  R4, #00H
        MOV  R5, #00H            ; zero out the total quadrature
        MOV  R6, #00H
        MOV  R2, #04H            ; average four readings
INLOOP:
        MOV  DPTR, #0C801H       ; address of select lines
        MOV  A, R1               ; get mask for select lines
        MOVX @DPTR, A            ; start conversion for inphase card
        MOV  A, #0FFH
        MOVX @DPTR, A
        MUL  AB                  ; wait around 12uS
        MUL  AB
        MUL  AB
        MUL  AB

        MOV  A, R1               ; get mask for inphase card
        CLR  C
        RL   A                   ; shift left for quadrature card
        MOVX @DPTR, A            ; and start conversion for quadrature card
        MOV  A, #0FFH
        MOVX @DPTR, A
        MUL  AB                  ; Wait around 12uS
        MUL  AB
        MUL  AB
        MUL  AB

        MOV  A, R1               ; get mask for inphase card
        MOVX @DPTR, A            ; and select for read
        MUL  AB                  ; wait 8uS for bus to stabalize
        MUL  AB
```

(A.1) program listing.

```
            MOV  R3, A
            JNC  NO_CARRYI
            INC  R4
NO_CARRYI:

            MOV  DPTR, #0C801H       ; deselect inphase detector card
            MOV  A, R1
            CLR  C
            RL   A
            MOVX @DPTR, A            ; and select quadrature card for read

            MUL  AB                  ; wait 8uS
            MUL  AB
            MOV  DPTR, #0C800H       ; point to port A
            MOVX A, @DPTR            ; get A/D  value

            ADD  A, R5
            MOV  R5, A
            JNC  NO_CARRYQ
            INC  R6
NO_CARRYQ:

            MOV  DPTR, #0C801H       ; address of select lines
            MOV  A, #0FFH
            MOVX @DPTR, A            ; and deselect everything

            MOV  A, R2
            CLR  C
            SUBB A, #01H             ; decrement count
            MOV  R2, A
            JNZ  INLOOP

            MOV  A, R6               ; shift right two bits
            RRC  A
            MOV  R6, A
            MOV  A, R5
            RRC  A
            MOV  R5, A
            MOV  A, R6
            RRC  A
            MOV  R6, A
            MOV  A, R5
            RRC  A
            MOV  R5, A

            MOV  A, R4
            RRC  A
            MOV  R4, A
            MOV  A, R3
            RRC  A
            MOV  R3, A
            MOV  A, R4
            RRC  A
            MOV  R4, A
            MOV  A, R3
            RRC  A
            MOV  R3, A
            RET
; end of READING
```

(A.1) program listing.

```
;         04   output sent to channel 0
;         08   output sent to channel 0
;         0c   output sent to channel 0
;                   +
;         00   output at   0 degrees offset
;         01   output at  90 degrees offset
;         02   output at 180 degrees offset
;         03   output at 270 degrees offset
;
;         10   normal operation
SETCAL:
          MOV  DPTR, #CALTAB        ; address of translate table
          MOVC A, @A+DPTR           ; get bit pattern for state
          MOV  DPTR, #0C802H        ; address of 82C55
          MOVX @DPTR, A             ; and set outputs

; let's wait about 30ms for the relays to fire
          MOV  DPTR, #00000H
WAIT:
          INC  DPTR
          MOV  A, DPL
          JNZ  WAIT
          MOV  A, DPH
          JNZ  WAIT

          RET
; end SETCAL



; CLIP clips the read values to the range stored in the table of calibration
; data.  Clip is passed the value to be clipped in R3, and the memory offset
; in the calibration table in A.  The clipped value is returned in A.
CLIP:
; clip values to stored maximums and minimums
          MOV  DPTR, #CAL_ARRY      ; address of calibration data
          ADD  A, DPL               ; point to needed minimum value
          MOV  DPL, A
          MOVX A, @DPTR             ; get minimum value
          CLR  C
          SUBB A, R3                ; and subtract read value
          JNC  SMALL                ; too small
          MOV  A, R3                ; restore read value
          LJMP SKIPS
SMALL:
          MOVX A, @DPTR             ; get minimum value
SKIPS:
          MOV  R3, A                ; update current read value


          INC  DPTR                 ; point to maximum value
          MOVX A, @DPTR             ; get maximum value
          CLR  C
          SUBB A, R3                ; subtract read value
          JC   LARGE                ; too big
          MOV  A, R3                ; restore value
          LJMP SKIPI
LARGE:
          MOVX A, @DPTR
SKIPI:
          RET
```

(A.1) program listing.

```
; This subroutine will normalize the board reading pointed to by R1 according
; to this formula:
;
;       scaled = (256 * read_value) / factor
;
; The value to be normalized is pointed to by DPTR, and the result is stored
; in the same location.
NORMAL:
          MOVX A, @DPTR              ; get read value
          MOV  R0, A
          PUSH DPH
          PUSH DPL
          MOV  DPTR, #CAL_ARRY       ; get pointer to calibration data
          MOV  A, R1                 ; get offset
          CLR  C
          RLC  A                     ; multiply by two
          ADD  A, DPL                ; and add offset
          MOV  DPL, A
          MOVX A, @DPTR              ; get minimum value
          MOV  R2, A                 ; and save
          MOV  A, R0                 ; get back measured value
          CLR  C
          SUBB A, R2                 ; and subtract minimum value (scale down)
          MOV  R0, A                 ; save
          INC  DPTR
          MOVX A, @DPTR              ; get maximum value
          CLR  C
          SUBB A, R2                 ; and find difference
          MOV  R2, A

          MOV  R6, #07H
          MOV  R1, #00H

          CLR  C
          MOV  A, R2
          RRC  A                     ; divide factor by two
          MOV  R2, A
          MOV  A, #00H               ; and 'multiply' both by 256
          RRC  A
          MOV  R3, A

DIV_LOOP:
          CLR  C
          MOV  A, R4                 ; multiply result by two
          RLC  A
          MOV  R4, A

          CLR  C
          MOV  A, R1                 ; subtract factor from value
          SUBB A, R3
          MOV  R5, A                 ; save intermediate value
          MOV  A, R0
          SUBB A, R2
          JC   ENDIFDIV              ; too big, retain previous value
          MOV  R0, A                 ; small enough, update value and scale
          MOV  A, R5                 ; low byte of value
          MOV  R1, A
          MOV  A, R4                 ; incrementing scale
          INC  A
          MOV  R4, A
ENDIFDIV:
```

(A.1) program listing.

```
        MOV  A, R3
        RRC  A
        MOV  R3, A

        MOV  A, R6              ; get loop value
        JZ   END_DIV            ; end of the line
        DEC  A
        MOV  R6, A
        LJMP DIV_LOOP

END_DIV:
        POP  DPL
        POP  DPH
        MOV  A, R4
        MOVX @DPTR, A
        RET
; end NORMAL


; The function ASIN returns the inverse sine of the supplied value.
; This is returned in R2:R0.  The value is passed in A.
; It must be passed a value from 0-255 scaled as 128*n+128.
; That is, -1 = 0; 0 = 128; +1 = 255.  This may be adjusted later
; to scale the range to +/- 0.707 for increased accuracy.
; ASIN returns the inverse sine value in the format of a 16-bit signed
; integer.  The returned value is in degrees*10.
ASIN:
        MOV  DPTR, #LOOKUP      ; pointer to lookup table (most sig. half)
        MOV  R0, A              ; save
        MOVC A, @A+DPTR         ; get sin-1 value (most significant half)
        MOV  R2, A

        MOV  A, R0
        MOV  DPTR, #LOOKUP+256  ; pointer to lookup table (least sig. half)
        MOVC A, @A+DPTR         ; get sin-1 value (least significant half)
        MOV  R0, A

        RET
; end ASIN


; CALCPQ is passed the inphase and quadrature measured values in R3
; and R5.  They are clipped, normalized, and the inverse sine is calculated.
; The inphase and quadrature values are passed back stored in memory
CALCPQ:
; save R3 and R5
        MOV  DPTR, #CARD0
        MOV  A, R3
        MOVX @DPTR, A
        MOV  DPTR, #CARD1
        MOV  A, R5
        MOVX @DPTR, A

; note that R3 is still preserved from above
        MOV  DPTR, #INDEX       ; set pointer
        MOVX A, @DPTR           ; to the clipping value
```

(A.1) program listing.

```
; save clipped value in INPHASE
        MOV  DPTR, #INPHASE
        MOVX @DPTR, A

; retrieve CARD1 to R3
        MOV  DPTR, #CARD1
        MOVX A, @DPTR
        MOV  R3, A

        MOV  DPTR, #INDEX           ; set pointer
        MOVX A, @DPTR               ; to point to the clipping array
        MOV  B, #04H
        MUL  AB
        ADD  A, #02H
        LCALL CLIP

; save clipped value in QUAD
        MOV  DPTR, #QUAD
        MOVX @DPTR, A


; normalize values
        MOV  DPTR, #INDEX
        MOVX A, @DPTR               ; get number of board pair to normalize
        CLR  C
        RLC  A                      ; multiply by two
        MOV  R1, A
        MOV  DPTR, #INPHASE         ; point DPTR to value to be normalized
        LCALL NORMAL                ; and normalize

        MOV  DPTR, #INDEX
        MOVX A, @DPTR               ; get number of board pair to normalize
        CLR  C
        RLC  A                      ; multiply by two
        INC  A                      ; and increment for quadrature
        MOV  R1, A
        MOV  DPTR, #QUAD            ; point DPTR to value to be normalized
        LCALL NORMAL                ; and normalize


; calculate inverse sines of inphase and quadrature
; value passed to ASIN in A, returned in R0:R2
        MOV  DPTR, #INPHASE
        MOVX A, @DPTR               ; get normalized data values
        LCALL ASIN
        MOV  DPTR, #INPHASE
        MOV  A, R2                  ; and save asin values
        MOVX @DPTR, A
        INC  DPTR
        MOV  A, R0
        MOVX @DPTR, A

        MOV  DPTR, #QUAD
        MOVX A, @DPTR
        LCALL ASIN
        MOV  DPTR, #QUAD
        MOV  A, R2
        MOVX @DPTR, A
        INC  DPTR
        MOV  A, R0
        MOVX @DPTR, A
        RET
```

(A.1) program listing.

```
; CHECK01 receives a 16-bit signed number in R0:R1 and returns
; a zero if the number is less than -450, a 1 if it is between
; -450 and 450, and a 2 if it is greater than 450
CHECK01:
        MOV  A, R0
        CLR  C
        RLC  A
        JC   CHKNEG
        CLR  C
        MOV  A, R0
        SUBB A, #01H
        JC   CHKSML
        SUBB A, #01H
        JNC  ISBIG
        MOV  A, R1
        CLR  C
        SUBB A, #0C2H             ; (450-256)
        JC   CHKSML               ; too small
ISBIG:
        MOV  A, #02H
        RET

CHKNEG:
        MOV  A, R0
        ADD  A, #01H
        JC   CHKSML
        ADD  A, #01H
        JNC  ISSMALL
        MOV  A, R1
        ADD  A, #3DH
        JC   CHKBIG
ISSMALL:
        MOV  A, #00H
        RET

CHKSML:
CHKBIG:
        MOV  A, #01H
        RET
; end CHECK01


; now calculate phi from inphase and quadrature values
MAKEPHI:
; add in delta-phi values
        MOV  DPTR, #INPHASE       ; inphase is stored in R0:R1
        MOVX A, @DPTR
        MOV  R0, A
        INC  DPTR
        MOVX A, @DPTR
        MOV  R1, A
        MOV  DPTR, #INDEX         ; check if this is needed
        MOVX A, @DPTR
        MOV  B, #08H
        MUL  AB
        MOV  R2, A
        MOV  DPTR, #QUAD          ; get pointer to quadrature high byte
        LCALL ADDOFF
        MOV  DPTR, #INPHASE       ; save updated inphase values
```

(A.1) program listing.

```
            MOVX @DPTR, A

            MOV  DPTR, #QUAD
            MOVX A, @DPTR
            MOV  R0, A
            INC  DPTR
            MOVX A, @DPTR
            MOV  R1, A
            MOV  DPTR, #INDEX
            MOVX A, @DPTR
            MOV  B, #08H
            MUL  AB
            ADD  A, #04H
            MOV  R2, A
            MOV  DPTR, #INPHASE
            LCALL ADDOFF
            MOV  DPTR, #QUAD
            MOV  A, R0
            MOVX @DPTR, A
            INC  DPTR
            MOV  A, R1
            MOVX @DPTR, A


SKIPOFF:
            MOV  DPTR, #INPHASE
            MOVX A, @DPTR
            MOV  R0, A
            INC  DPTR
            MOVX A, @DPTR
            MOV  R1, A
            LCALL CHECK01
            RL   A
            MOV  DPTR, #JMPTBL1
            JMP  @A+DPTR

; inphase is not >45 degrees
NOTINPH:
            MOV  DPTR, #QUAD
            MOVX A, @DPTR
            MOV  R0, A
            INC  DPTR
            MOVX A, @DPTR
            MOV  R1, A
            LCALL CHECK01
            RL   A
            MOV  DPTR, #JMPTBL2
            JMP  @A+DPTR

; since inphase and quad fell through all of the above tests, let's now
; determine which has the greater magnitude and use it.
NOTQUAD:
            MOV  DPTR, #INPHASE       ; Get inphase and quadrature
            MOVX A, @DPTR             ;  read in and stored in R0:R1,
            MOV  R0, A                ;  and R2:R3 respectively.
            INC  DPTR                 ;  This is done so that calculation
            MOVX A, @DPTR             ;  of the absolute values will not
            MOV  R1, A                ;  disturb what's stored in memory
            INC  DPTR
            MOVX A, @DPTR
```

(A.1) program listing.

```
         MOV  A, R0                  ; Get high byte of inphase
         CLR  C
         RLC  A
         JNC  ISPOS1
         MOV  A, R1                  ; Get low byte of inphase
         CPL  A
         ADD  A, #01H
         MOV  R1, A
         MOV  A, R0                  ; Get high byte of inphase
         CPL  A
         ADDC A, #00H
         MOV  R0, A
ISPOS1:
         MOV  A, R2                  ; Get high byte of quad
         CLR  C
         RLC  A
         JNC  ISPOS2
         MOV  A, R3                  ; Get low byte of quad
         CPL  A
         ADD  A, #01H
         MOV  R3, A
         MOV  A, R2                  ; Get high byte of quad
         CPL  A
         ADDC A, #00H
         MOV  R2, A
ISPOS2:
         MOV  A, R2
         CLR  C
         SUBB A, R0
         JZ   EQUAL
         JC   INBIG
QUADBIG:
         MOV  DPTR, #QUAD
         MOVX A, @DPTR
         XRL  A, #40H                ; this should increase the magnitude
                                     ;  irregardless of sign
         MOVX @DPTR, A
         LJMP SKIPOFF

INBIG:
         MOV  DPTR, #INPHASE
         MOVX A, @DPTR
         XRL  A, #40H                ; this should increase the magnitude
                                     ;  irregardless of sign
         MOVX @DPTR, A
         LJMP SKIPOFF

EQUAL:
         MOV  A, R3
         CLR  C
         SUBB A, R1
         JC   INBIG
         LJMP QUADBIG

; case 1:   inphase <= -45 degrees   --> phi = quad
CASE1:
         MOV  DPTR, #QUAD            ; calculate phi = quad
         MOVX A, @DPTR
         MOV  R3, A
         INC  DPTR
```

(A.1) program listing.

```
        MOV  A, R3
        MOVX @DPTR, A
        INC  DPTR
        MOV  A, R1
        MOVX @DPTR, A

        LCALL OUTPUT
        RET

; case 2: inphase >= 45 degrees      --> phi= 1800-quad or phi=-1800-quad
CASE2:
        MOV  DPTR, #QUAD         ; calculate phi = 1800-quad...
        MOVX A, @DPTR
        CLR  C
        RLC  A
        JC   C2MINUS
        MOV  R0, #07H            ; hi byte
        MOV  R1, #08H            ; and low
        AJMP C2SKIP
C2MINUS:
        MOV  R0, #0F8H           ; hi byte
        MOV  R1, #0F8H           ; and low
C2SKIP:
        MOV  DPTR, #QUAD+1       ; must do low byte first
        MOVX A, @DPTR
        XCH  A, R1
        CLR  C
        SUBB A, R1
        MOV  R1, A
        MOV  DPTR, #QUAD         ; and now do the high byte with carry
        MOVX A, @DPTR
        XCH  A, R0
        SUBB A, R0
        MOV  R3, A


        MOV  DPTR, #PHI          ; store phi in mem
        MOV  A, R3
        MOVX @DPTR, A
        INC  DPTR
        MOV  A, R1
        MOVX @DPTR, A

        LCALL OUTPUT
        RET

; case 3: quad <= -45 degrees        --> phi = -900 - inphase
CASE3:
        MOV  DPTR, #INPHASE+1    ; must do low byte first
        MOVX A, @DPTR
        MOV  R1, A
        MOV  A, #7CH
        CLR  C
        SUBB A, R1
        MOV  R1, A
        MOV  DPTR, #INPHASE      ; and now do the high byte with carry
        MOVX A, @DPTR
        MOV  R3, A
        MOV  A, #0FCH
        SUBB A, R3
        MOV  R3, A
```

(A.1) program listing.

```
        INC   DPTR
        MOV   A, R1
        MOVX @DPTR, A

        LCALL OUTPUT
        RET

; case 4: quad >= 45 degrees        --> phi = inphase + 900
CASE4:
        MOV   DPTR, #INPHASE+1
        MOVX A, @DPTR
        ADD   A, #84H
        MOV   R1, A
        MOV   DPTR, #INPHASE
        MOVX A, @DPTR
        ADDC A, #03H
        MOV   R3, A

        MOV   DPTR, #PHI            ; store phi in mem
        MOV   A, R3
        MOVX @DPTR, A
        INC   DPTR
        MOV   A, R1
        MOVX @DPTR, A

        LCALL OUTPUT
        RET
; end MAKEPHI



; addoff expects the 16-bit value in R0:R1, the 8-bit offset in R2,
; and DPTR pointing to the appropriate value to be tested
ADDOFF:
        MOVX A, @DPTR               ; get high byte of current value pointed to
        CLR   C
        RLC   A                     ; and check its sign
        JNC   PLUS1
        MOV   A, #02                ; if it's negative, use other P_OFF
        ADD   A, R2
        MOV   R2, A
PLUS1:
        MOV   A, R2
        MOV   DPTR, #P_OFF          ; add in pointer to P_OFF
        ADD   A, DPL                ; we will assume table does not cross page
        MOV   DPL, A

        MOVX A, @DPTR               ; get P_OFF low byte
        ADD   A, R1                 ; add to inphase low byte
        MOV   R1, A
        INC   DPTR                  ; move pointer to P_OFF high byte
        MOVX A, @DPTR               ; get P_OFF high byte
        ADDC A, R0                  ; and add to inphase high byte with carry
        MOV   R0, A
        RET
; end ADDOFF
```

(A.1) program listing.

```
        MOV  A, R3
        JNB  TI, $
        CLR  TI
        MOV  SBUF, A

        MOV  A, R1
        JNB  TI, $
        CLR  TI
        MOV  SBUF, A

        RET
; end OUTPUT



; Data constants

; Boot-up message sent out serial port
STRING: DB 'APAC Phase measurement system', CR, LF
        DB 'Version 2.34', CR, LF
        DB '3/26/89, A. Dornbusch', CR, LF
        DB ESC


; Bit masking values for taking A/D measurements
CONVERT: DB 0FEH, 0FBH, 0EFH, 0BFH

TRANSLATE: DB 0FEH, 0FDH, 0FBH, 0F7H, 0EFH, 0DFH, 0BFH, 07FH


; Bit values for setting relays when doing system calibration
CALTAB:     DB 071H, 072H, 074H, 078H
            DB 0B1H, 0B2H, 0B4H, 0B8H
            DB 0D1H, 0D2H, 0D4H, 0D8H
            DB 0E1H, 0E2H, 0E4H, 0E8H, 0F0H


; Jump tables for calculate phi function
JMPTBL1: AJMP CASE1
         AJMP NOTINPH
         AJMP CASE2

JMPTBL2: AJMP CASE3
         AJMP NOTQUAD
         AJMP CASE4


; lookup table for inverse sine function
LOOKUP:
; high byte of sin-1 function
      DB 0FCH, 0FCH, 0FCH, 0FCH, 0FDH, 0FDH, 0FDH, 0FDH
      DB 0FDH, 0FDH, 0FDH, 0FDH, 0FDH, 0FDH, 0FDH, 0FDH
      DB 0FDH, 0FDH, 0FDH, 0FDH, 0FDH, 0FDH, 0FDH, 0FDH
      DB 0FDH, 0FDH, 0FDH, 0FDH, 0FEH, 0FEH, 0FEH, 0FEH
      DB 0FEH, 0FEH, 0FEH, 0FEH, 0FEH, 0FEH, 0FEH, 0FEH
      DB 0FEH, 0FEH, 0FEH, 0FEH, 0FEH, 0FEH, 0FEH, 0FEH
      DB 0FEH, 0FEH, 0FEH, 0FEH, 0FEH, 0FEH, 0FEH, 0FEH
      DB 0FEH, 0FEH, 0FEH, 0FEH, 0FEH, 0FEH, 0FEH, 0FEH
      DB 0FEH, 0FEH, 0FEH, 0FEH, 0FEH, 0FEH, 0FEH, 0FEH
      DB 0FEH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH
```

(A.1) program listing.

```
        DB 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH
        DB 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH
        DB 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H
        DB 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H
        DB 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H
        DB 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H
        DB 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H
        DB 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H
        DB 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H
        DB 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H
        DB 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H
        DB 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H
        DB 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H
        DB 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H
        DB 001H, 001H, 001H, 001H, 002H, 002H, 002H, 002H
        DB 002H, 002H, 002H, 002H, 002H, 002H, 002H, 002H
        DB 002H, 002H, 002H, 002H, 002H, 002H, 002H, 002H
        DB 002H, 002H, 002H, 002H, 002H, 003H, 003H, 003H


;LOOKUP+256
; low byte of sin-1 function
        DB 07EH, 0C6H, 0E3H, 0FAH, 00DH, 01EH, 02EH, 03CH
        DB 049H, 056H, 062H, 06DH, 078H, 082H, 08CH, 096H
        DB 09FH, 0A8H, 0B1H, 0BAH, 0C2H, 0CAH, 0D2H, 0DAH
        DB 0E2H, 0EAH, 0F1H, 0F8H, 000H, 007H, 00EH, 015H
        DB 01BH, 022H, 029H, 02FH, 036H, 03CH, 042H, 049H
        DB 04FH, 055H, 05BH, 061H, 067H, 06DH, 073H, 079H
        DB 07EH, 084H, 08AH, 08FH, 095H, 09AH, 0A0H, 0A5H
        DB 0ABH, 0B0H, 0B6H, 0BBH, 0C0H, 0C5H, 0CBH, 0D0H
        DB 0D5H, 0DAH, 0DFH, 0E4H, 0E9H, 0EEH, 0F4H, 0F9H
        DB 0FDH, 002H, 007H, 00CH, 011H, 016H, 01BH, 020H
        DB 025H, 029H, 02EH, 033H, 038H, 03DH, 041H, 046H
        DB 04BH, 04FH, 054H, 059H, 05DH, 062H, 067H, 06BH
        DB 070H, 075H, 079H, 07EH, 082H, 087H, 08BH, 090H
        DB 095H, 099H, 09EH, 0A2H, 0A7H, 0ABH, 0B0H, 0B4H
        DB 0B9H, 0BDH, 0C2H, 0C6H, 0CBH, 0CFH, 0D4H, 0D8H
        DB 0DDH, 0E1H, 0E6H, 0EAH, 0EFH, 0F3H, 0F8H, 0FCH
        DB 000H, 004H, 008H, 00DH, 011H, 016H, 01AH, 01FH
        DB 023H, 028H, 02CH, 031H, 035H, 03AH, 03EH, 043H
        DB 047H, 04CH, 050H, 055H, 059H, 05EH, 062H, 067H
        DB 06BH, 070H, 075H, 079H, 07EH, 082H, 087H, 08BH
        DB 090H, 095H, 099H, 09EH, 0A3H, 0A7H, 0ACH, 0B1H
        DB 0B5H, 0BAH, 0BFH, 0C3H, 0C8H, 0CDH, 0D2H, 0D7H
        DB 0DBH, 0E0H, 0E5H, 0EAH, 0EFH, 0F4H, 0F9H, 0FEH
        DB 003H, 007H, 00CH, 012H, 017H, 01CH, 021H, 026H
        DB 02BH, 030H, 035H, 03BH, 040H, 045H, 04AH, 050H
        DB 055H, 05BH, 060H, 066H, 06BH, 071H, 076H, 07CH
        DB 082H, 087H, 08DH, 093H, 099H, 09FH, 0A5H, 0ABH
        DB 0B1H, 0B7H, 0BEH, 0C4H, 0CAH, 0D1H, 0D7H, 0DEH
        DB 0E5H, 0EBH, 0F2H, 0F9H, 000H, 008H, 00FH, 016H
        DB 01EH, 026H, 02EH, 036H, 03EH, 046H, 04FH, 058H
        DB 061H, 06AH, 074H, 07EH, 088H, 093H, 09EH, 0AAH
        DB 0B7H, 0C4H, 0D2H, 0E2H, 0F3H, 006H, 01DH, 03AH


        END
```

(A.1) program listing.

A.2 SERIAL PORT COMMUNICATION ROUTINE FOR PHASE MEASUREMENT

Communications are necessary for the BCC-52 to transmit phase data to an IBM compatible personal computer via the serial port. This program is written in 8088 Assembly language and operates on an IBM compatible computer. The program listing follows:

```
; This routine will communicate with the BCC-52C to request a phase
; measurement of a specified pair.  It is passed a single byte value,
; which it in turn passes directly to the BCC-52C.  Said value is unchecked
; and unmodified.  Then the routine will wait for a response from the BCC.
; The two byte response is converted into a 16-bit signed integer, and returned
; to the calling routine (in AX).  If this routine does not receive a response
; from the BCC within approx. 1/2 second, it will set an error flag and return.

uart_base EQU 2f8h        ; address of COM1
xmit      EQU 0           ; transmit register offset
rcv       EQU 0           ; receive register offset
lsb_off   EQU 0           ; clock divide LSB
msb_off   EQU 1           ; clock divide MSB
fmt_off   EQU 3           ; UART format register
control   EQU 4           ; UART control register
stat      EQU 5           ; UART status register
format    EQU 03h         ; 8 bits, 1 stop bit, parity none
div_msb   EQU 00h         ; 4800 baud
div_lsb   EQU 18h
uart_cntl EQU 0Bh         ; set DTR and RTS true just to be safe

CR        EQU     0dh      ; carrage return
LF        EQU     0ah      ;line feed

onstack  STRUC
oldbp    DW  ?            ; old BP
retaddr  DW  ?            ; return address
seraddr  DW  ?            ; serial port address
board    DW  ?            ; board pair to be requested
onstack  ENDS


code     SEGMENT PUBLIC
         ASSUME CS:code

         PUBLIC measure, get_second


measure  PROC  NEAR
         push  BP
         mov   BP, SP

         push  BX
         push  CX
         push  DX

         mov   DX, [bp+04h]
         add   DX, stat             ; address of status port
         mov   CX, 0ffffh           ; wait .48 sec until error
xmitloop:
         dec   CX                   ; increment waiting count
         jz    error                ; it's been too long, error
         in    AL, DX               ; get status byte
         and   AL, 20h              ; mask for xmit holding reg status
         jz    xmitloop             ; loop until holding reg empty

         mov   DX, [bp+04h]
         add   DX, xmit             ; address of transmit buffer
         mov   AX, [bp+06h]         ; get parameter to be sent
         out   DX, AL               ; and send it
```

(A.2) program listing.

```
rcvloop1:
        dec     CX              ; decrement waiting count
        jz      error           ; it's been too long, error
        in      AL, DX          ; get status byte
        and     AL, 01h         ; mask for rcv holding register
        jz      rcvloop1

        mov     DX, [bp+04h]
        add     DX, rcv         ; address of receive register
        in      AL, DX          ; get byte received from serial line
        mov     BX, AX          ; save for later


        mov     DX, [bp+04h]
        add     DX, stat        ; address of status port
        mov     CX, 0ffffh      ; wait .48 sec until error
rcvloop2:
        dec     CX              ; increment waiting count
        jz      error           ; it's been too long, error
        in      AL, DX          ; get status byte
        and     AL, 01h         ; mask for rcv holding register
        jz      rcvloop2

        mov     DX, [bp+04h]
        add     DX, rcv         ; address of receive register
        in      AL, DX          ; got second byte from serial line

        mov     AH, BL          ; and retrieve other byte

        pop     DX
        pop     CX
        pop     BX
        pop     BP

        ret     4

error:
        mov     AX, 8000h       ; flag for error condition

        pop     DX
        pop     CX
        pop     BX
        pop     BP

        ret     4

measure ENDP


get_second PROC NEAR
        push    BP
        mov     BP, SP

        push    BX
        push    CX
        push    DX

        mov     DX, [bp+04h]
        add     DX, stat        ; address of status port
        mov     CX, 0ffffh      ; wait .48 sec until error
```

(A.2) program listing.

```
            and     AL, 01h              ; mask for rcv holding register
            jz      rcvloop3

            mov     DX, [bp+04h]
            add     DX, rcv              ; address of receive register
            in      AL, DX               ; get byte received from serial line
            mov     BX, AX               ; save for later


            mov     DX, [bp+04h]
            add     DX, stat             ; address of status port
            mov     CX, 0ffffh           ; wait .48 sec until error
rcvloop4:
            dec     CX                   ; increment waiting count
            jz      error2               ; it's been too long, error
            in      AL, DX               ; get status byte
            and     AL, 01h              ; mask for rcv holding register
            jz      rcvloop4

            mov     DX, [bp+04h]
            add     DX, rcv              ; address of receive register
            in      AL, DX               ; got second byte from serial line

            mov     AH, BL               ; and retrieve other byte

            pop     DX
            pop     CX
            pop     BX
            pop     BP

            ret     2


error2:
            mov     AX, 8000h            ; flag for error condition

            pop     DX
            pop     CX
            pop     BX
            pop     BP

            ret     2

get_second ENDP

code        ENDS

            END
```

(A.2) program listing.

A.3 MINIMUM AND MAXIMUM VALUE CALIBRATION ROUTINE

This program is used to take raw A/D converter data and display it on the screen of the IBM compatible computer. The program is written in Turbo Pascal 4.0 and the listing follows:

```
program test(input,output);
{ take repeated raw measurements and display on screen }

{$L nocoma.obj}

const
   UART_BASE = $3f8;     { address of COM1 }
   XMIT      = 0;        { transmit register offset }
   RCV       = 0;        { receive register offset  }
   LSB_OFF   = 0;        { clock divide LSB }
   MSB_OFF   = 1;        { clock divide MSB }
   FMT_OFF   = 3;        { UART format register }
   CONTROL   = 4;        { UART control register }
   STAT      = 5;        { UART status register }
   FORMAT    = $03;      { 8 bits, 1 stop bit, parity none }
   DIV_MSB   = $00;      { 4800 baud                       }
   DIV_LSB   = $18;
   UART_CNTL = $0B;      { set DTR and RTS true just to be safe }
   count     = 800;
   step      = 8;


var
   board_pair     :  byte;
   data           :  integer;
   input          :  byte;
   loop           :  integer;
   inphase        :  integer;
   quad           :  integer;
   inmax, inmin   :  integer;
   qumax, qumin   :  integer;

function measure(board_pair : integer; seraddr : integer)
          : integer; external;

function get_second(seraddr : integer)
          : integer; external;

function get_status : byte;
   begin { function get_status }
      get_status := port[UART_BASE+STAT];
   end;  { function get_status }

procedure set_up_serial;
   begin { procedure set_up_serial }
      port[UART_BASE+FMT_OFF] := FORMAT + $80;
      port[UART_BASE+LSB_OFF] := DIV_LSB;
      port[UART_BASE+MSB_OFF] := DIV_MSB;
      port[UART_BASE+FMT_OFF] := FORMAT;
      port[UART_BASE+CONTROL] := UART_CNTL;
      while (get_status mod 2) <> 0 do
         data := port[UART_BASE+RCV];
   end; { procedure set_up_serial }


begin
   set_up_serial;
   inmax := 0;
   inmin := 255;
   qumax := 0;
   qumin := 255;
```

(A.3) program listing.

```
            quad := get_second(UART_BASE);
            if (inphase > inmax) then inmax := inphase;
            if (inphase < inmin) then inmin := inphase;
            if (quad > qumax) then qumax := quad;
            if (quad < qumin) then qumin := quad;

            writeln(loop:4, '-  inphase: ', inphase:6, '  quad: ', quad:6);
         end;
         writeln('inphase-   min: ',inmin:4,'   max: ',inmax:4);
         writeln('quad    -   min: ',qumin:4,'   max: ',qumax:4);
   end.
```

(A.3) program listing.

A.4 PHASE OFFSET CALIBRATION ROUTINE

This program is used to get raw phase data from the in-phase and quadrature phase detector channels and display it on the screen. The program is written in Turbo Pascal 4.0 and the listing follows:

```
program test(input,output);
{ take repeated phase measurements and display on screen }

{$L nocoma.obj}

const
   UART_BASE = $3f8;     { address of COM1 }
   XMIT      = 0;        { transmit register offset }
   RCV       = 0;        { receive register offset  }
   LSB_OFF   = 0;        { clock divide LSB }
   MSB_OFF   = 1;        { clock divide MSB }
   FMT_OFF   = 3;        { UART format register }
   CONTROL   = 4;        { UART control register }
   STAT      = 5;        { UART status register }
   FORMAT    = $03;      { 8 bits, 1 stop bit, parity none }
   DIV_MSB   = $00;      { 4800 baud                       }
   DIV_LSB   = $18;
   UART_CNTL = $0B;      { set DTR and RTS true just to be safe }
   count     = 2048;
   step      = 8;


var
   board_pair     :  byte;
   data           :  integer;
   input          :  byte;
   loop           :  integer;
   inphase        :  integer;
   quad           :  integer;
   inmax, inmin   :  integer;
   qumax, qumin   :  integer;

procedure set_up_serial;
   begin { procedure set_up_serial }
      port[UART_BASE+FMT_OFF] := FORMAT + $80;
      port[UART_BASE+LSB_OFF] := DIV_LSB;
      port[UART_BASE+MSB_OFF] := DIV_MSB;
      port[UART_BASE+FMT_OFF] := FORMAT;
      port[UART_BASE+CONTROL] := UART_CNTL;
      while (get_status mod 2) <> 0 do
         data := port[UART_BASE+RCV];
   end; { procedure set_up_serial }


function measure(board_pair : integer; seraddr : integer)
          : integer; external;

function get_second(seraddr : integer)
          : integer; external;

function get_status : byte;
   begin { function get_status }
      get_status := port[UART_BASE+STAT];
   end;  { function get_status }


begin
   set_up_serial;
   inmax := 0;
   inmin := 255;
```

(A.4) program listing.

```
    qumax := 0;
    qumin := 255;

    for loop := 0 to count do
       begin
          inphase := measure(65+0, UART_BASE);

          writeln(loop:4, '-  phi: ', inphase/10:6:1);
       end;
end.
```

(A.4) program listing.

A.5 CALIBRATION DATA TRANSMISSION ROUTINE

Calibration data gathered from (A.3) and (A.4) is entered in the IBM compatible computer and transmitted to the BCC-52 microcomputer using this program. Note that (A.6) is used in conjunction with (A.5). The program is written in Turbo Pascal 4.0 and the listing follows:

```
program test(input,output);

{$L nocoma.obj}
{$L txdata.obj}

const
   UART_BASE = $3f8;     { address of COM1 }
   XMIT      = 0;        { transmit register offset }
   RCV       = 0;        { receive register offset  }
   LSB_OFF   = 0;        { clock divide LSB }
   MSB_OFF   = 1;        { clock divide MSB }
   FMT_OFF   = 3;        { UART format register }
   CONTROL   = 4;        { UART control register }
   STAT      = 5;        { UART status register }
   FORMAT    = $03;      { 8 bits, 1 stop bit, parity none }
   DIV_MSB   = $00;      { 4800 baud                       }
   DIV_LSB   = $18;
   UART_CNTL = $0B;      { set DTR and RTS true just to be safe }
   count     = 800;
   step      = 8;


var
   board_pair      :  byte;
   loop, j         :  word;
   inphase         :  word;
   quad            :  word;
   data            :  word;
   data0, data1, data2, data3, data4, data5, data6, data7 : integer;
   calarry         :  array[0..31] of integer;
   p_off           :  array[0..31] of integer;


function measure(board_pair : integer; seraddr : integer)
         : integer; external;

function get_second(seraddr : integer)
         : integer; external;

function txcal(board : integer; data : word;  seraddr : integer)
         : integer; external;
{ NOTE: cal data is sent as  (max*256)+min }

function txpoff(board, data1, data2, seraddr : integer)
         : integer; external;

function get_status : byte;
   begin { function get_status }
      get_status := port[UART_BASE+STAT];
   end;  { function get_status }

procedure set_up_serial;
   begin { procedure set_up_serial }
      port[UART_BASE+FMT_OFF] := FORMAT + $80;
      port[UART_BASE+LSB_OFF] := DIV_LSB;
      port[UART_BASE+MSB_OFF] := DIV_MSB;
      port[UART_BASE+FMT_OFF] := FORMAT;
      port[UART_BASE+CONTROL] := UART_CNTL;
      while (port[UART_BASE+STAT] mod 2) <> 0 do
         data := port[UART_BASE+RCV];
   end; { procedure set_up_serial }
```

(A.5) program listing.

```
        in_val1, in_val2 : byte;
        status : integer;

    begin
        while ((get_status shr 5) mod 2) = 0 do ;
        port[UART_BASE+XMIT] := board_pair;

        while (get_status mod 2) = 0 do ;
        in_val1 := port[UART_BASE+RCV];
        while (get_status mod 2) = 0 do ;
        in_val2 := port[UART_BASE+RCV];

        measure2 := in_val1*256 + in_val2;
    end;


begin
    set_up_serial;

{ min/max data transmitted as Vmax*256+Vmin }
    data0 := txcal(92, 49453, UART_BASE);
    data1 := txcal(93, 48160, UART_BASE);
    data2 := txcal(94, 47397, UART_BASE);
    data3 := txcal(95, 41002, UART_BASE);
    data4 := txcal(96, 50723, UART_BASE);
    data5 := txcal(97, 53274, UART_BASE);
    data6 := txcal(98, 45857, UART_BASE);
    data7 := txcal(99, 45597, UART_BASE);
    writeln('data = ',data0, data1, data2, data3, data4, data5, data6, data7);


{ INPHASE: first value is +90 degrees, second is -90 degrees }
{ QUAD:    first value is 180 degrees, second is   0 degrees }

    data0 := txpoff(100, 0, 0, UART_BASE);
    data1 := txpoff(101, 0, 0, UART_BASE);

    data2 := txpoff(102, 0, 80, UART_BASE);
    data3 := txpoff(103, 50, 130, UART_BASE);

    data4 := txpoff(104, -160, 160, UART_BASE);
    data5 := txpoff(105, 100, 50, UART_BASE);

    data6 := txpoff(106, -80, 130, UART_BASE);
    data7 := txpoff(107, 100, -50, UART_BASE);
    writeln('data = ',data0, data1, data2, data3, data4, data5, data6, data7);

    for loop := 0 to 7 do
        begin
            inphase := measure(76+loop, UART_BASE);
            quad := get_second(UART_BASE);
            calarry[loop] := inphase div 256;
            calarry[loop+8] := inphase - 256*calarry[loop];
            calarry[loop+16] := quad div 256;
            calarry[loop+24] := quad - 256*calarry[loop+16];

            inphase := measure(84+loop, UART_BASE);
            quad    := get_second(UART_BASE);
            p_off[loop*2]   := inphase;
            p_off[loop*2+1] := quad;
        end;
```

(A.5) program listing.

```
      writeln;
      for loop := 0 to 7 do
         write(p_off[loop*2+1]/10:7:1);
      writeln;
   end.
```

(A.5) program listing.

A.6 SERIAL PORT COMMUNICATIONS FOR CALIBRATION DATA

The function of this program is to provide communications between the IBM compatible Personal computer and the BCC-52 microcomputer. This program is written in 8088 Assembly language and is used in conjunction with the Pascal program provided in (A.6). The listing follows:

```
; FILE        TXDATA.ASM
;
; procedures txcal, txpoff
;             transmit cal data,
;             transmit phase offset data
;
;             Written by Andrew Dornbusch
;             3/6/89
;

xmit      EQU 0          ; transmit register offset
rcv       EQU 0          ; receive register offset
lsb_off   EQU 0          ; clock divide LSB
msb_off   EQU 1          ; clock divide MSB
fmt_off   EQU 3          ; UART format register
control   EQU 4          ; UART control register
stat      EQU 5          ; UART status register
format    EQU 03h        ; 8 bits, 1 stop bit, parity none
div_msb   EQU 00h        ; 4800 baud
div_lsb   EQU 18h
uart_cntl EQU 0Bh        ; set DTR and RTS true just to be safe

CR        EQU     0dh    ; carrage return
LF        EQU     0ah    ;line feed


code      SEGMENT PUBLIC
          ASSUME CS:code

          PUBLIC txcal, txpoff

; The procedure txcal causes the BCC to set a pair of calibration values.
; It is passed a command value, and one two-byte integer, which are passed
; as three bytes to the BCC.

txcal   PROC  NEAR
        push  BP
        mov   BP, SP

        push  BX
        push  CX
        push  DX

        mov   DX, [bp+04h]
        add   DX, stat              ; address of status port
        mov   CX, 0ffffh            ; wait .48 sec until error
xmitloop1:
        dec   CX                    ; increment waiting count
        jz    errorcal              ; it's been too long, error
        in    AL, DX                ; get status byte
        and   AL, 20h               ; mask for xmit holding reg status
        jz    xmitloop1             ; loop until holding reg empty

        mov   DX, [bp+04h]
        add   DX, xmit              ; address of transmit buffer
        mov   AX, [bp+08h]          ; get parameter to be sent
        out   DX, AL                ; and send it

        mov   DX, [bp+04h]
        add   DX, stat              ; address of status port
        mov   CX, 0ffffh            ; wait .48 sec until error
```

(A.6) program listing.

```
        and     AL, 20h             ; mask for xmit holding reg status
        jz      xmitloop2           ; loop until holding reg empty

        mov     DX, [bp+04h]
        add     DX, xmit            ; address of transmit buffer
        mov     AX, [bp+06h]        ; get parameter to be sent
        out     DX, AL              ; and send it

        mov     DX, [bp+04h]
        add     DX, stat            ; address of status port
        mov     CX, 0ffffh          ; wait .48 sec until error
xmitloop3:
        dec     CX                  ; increment waiting count
        jz      errorcal            ; it's been too long, error
        in      AL, DX              ; get status byte
        and     AL, 20h             ; mask for xmit holding reg status
        jz      xmitloop3           ; loop until holding reg empty

        mov     DX, [bp+04h]
        add     DX, xmit            ; address of transmit buffer
        mov     AX, [bp+07h]        ; get parameter to be sent
        out     DX, AL              ; and send it

        mov     AX, 0

        pop     DX
        pop     CX
        pop     BX
        pop     BP

        ret     4

errorcal:
        mov     AX, 0ffffh          ; flag for error condition

        pop     DX
        pop     CX
        pop     BX
        pop     BP

        ret     4

txcal   ENDP


; The procedure txpoff causes the BCC to set a pair of phase offsets.
; It is passed a command value, and two two-byte integers, which are passed
; as five bytes to the BCC.
txpoff  PROC NEAR
        push    BP
        mov     BP, SP

        push    BX
        push    CX
        push    DX

        mov     DX, [bp+04h]
        add     DX, stat            ; address of status port
        mov     CX, 0ffffh          ; wait .48 sec until error
xmitloop5:
        dec     CX                  ; increment waiting count
```

(A.6) program listing.

```
        and    AL, 20h           ; mask for xmit holding reg status
        jz     xmitloop5         ; loop until holding reg empty

        mov    DX, [bp+04h]
        add    DX, xmit          ; address of transmit buffer
        mov    AX, [bp+0Ah]      ; get parameter to be sent
        out    DX, AL            ; and send it


        mov    DX, [bp+04h]
        add    DX, stat          ; address of status port
        mov    CX, 0ffffh        ; wait .48 sec until error
xmitloop6:
        dec    CX                ; increment waiting count
        jz     errorpoff         ; it's been too long, error
        in     AL, DX            ; get status byte
        and    AL, 20h           ; mask for xmit holding reg status
        jz     xmitloop6         ; loop until holding reg empty

        mov    DX, [bp+04h]
        add    DX, xmit          ; address of transmit buffer
        mov    AX, [bp+09h]      ; get parameter to be sent
        out    DX, AL            ; and send it

        mov    DX, [bp+04h]
        add    DX, stat          ; address of status port
        mov    CX, 0ffffh        ; wait .48 sec until error
xmitloop7:
        dec    CX                ; increment waiting count
        jz     errorpoff         ; it's been too long, error
        in     AL, DX            ; get status byte
        and    AL, 20h           ; mask for xmit holding reg status
        jz     xmitloop7         ; loop until holding reg empty

        mov    DX, [bp+04h]
        add    DX, xmit          ; address of transmit buffer
        mov    AX, [bp+08h]      ; get parameter to be sent
        out    DX, AL            ; and send it


        mov    DX, [bp+04h]
        add    DX, stat          ; address of status port
        mov    CX, 0ffffh        ; wait .48 sec until error
xmitloop8:
        dec    CX                ; increment waiting count
        jz     errorpoff         ; it's been too long, error
        in     AL, DX            ; get status byte
        and    AL, 20h           ; mask for xmit holding reg status
        jz     xmitloop8         ; loop until holding reg empty

        mov    DX, [bp+04h]
        add    DX, xmit          ; address of transmit buffer
        mov    AX, [bp+07h]      ; get parameter to be sent
        out    DX, AL            ; and send it

        mov    DX, [bp+04h]
        add    DX, stat          ; address of status port
        mov    CX, 0ffffh        ; wait .48 sec until error
xmitloop9:
        dec    CX                ; increment waiting count
        jz     errorpoff          ; it's been too long, error
        in     AL, DX            ; get status byte
```

(A.6) program listing.

```
        add     DX, xmit            ; address of transmit buffer
        mov     AX, [bp+06h]        ; get parameter to be sent
        out     DX, AL              ; and send it

        mov     AX, 0

        pop     DX
        pop     CX
        pop     BX
        pop     BP

        ret     6

errorpoff:
        mov     AX, 0ffffh          ; flag for error condition

        pop     DX
        pop     CX
        pop     BX
        pop     BP

        ret     4

txpoff  ENDP

code    ENDS
        END
```

(A.6) program listing.

APPENDIX B

COMPUTATION OF NOISE FIGURE

The receiver noise figure (NF) is now calculated [21]. Calcutaions are made using receiver parameters given in Figures 5.4.a and 5.4.b. In the calculations, several assumptions being are made. Each of the stages are assumed to be perfectly matched. The passive, lossy devices are at the standard temperature of 290 K. It is asserted that the these are ideal conditions and in reality, the performance will be slightly lower.

Noise figure (NF) referred to the input is given by [21]:

$$NF_{1,n} = NF_1 + \frac{NF_2 - 1}{G_1} + \cdots + \frac{NF_{(n-1)} - 1}{G_1 G_2 \cdots G_{n-1}}$$

where G_1 is the gain of the first stage and so on.

Computing the noise figure of the receiver system referred to the input we have:

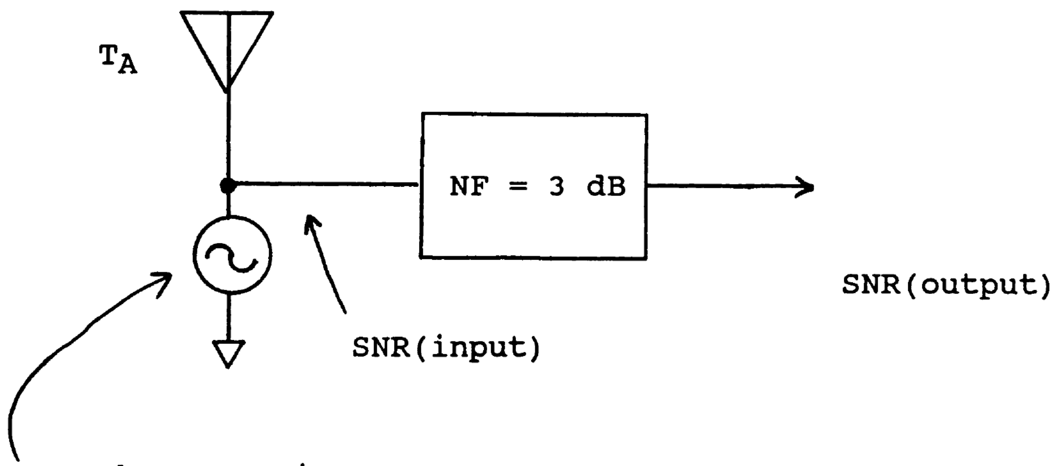
$$NF = 1.8 + \frac{2.5 - 1}{32} + \frac{2 - 1}{32 \times .4} + \frac{2.5 - 1}{32 \times .4 \times 100} +$$

$$\frac{2.5 - 1}{32 \times .4^2 \times 100} + \frac{10 - 1}{32 \times .4^3 \times 100} + \frac{NF(link) - 1}{205 \times G(link)}$$

$$NF = 2.0 + \frac{NF(link)}{205 \times G(link)}$$

<u>For the case of a lossless link</u>:

NF = 2.0 or 3 dB
B = Bandwidth = 12kHz
SNR(output) required = 13 dB
K = Boltzman's constant = 1.38E-23



$N = kT_AB$ = noise power
= -163 dB

SNR(input) = SNR(output) + NF = 13 + 3 = 16 dB

S = N + SNR(input)
= -163 dBw + 16 dB
= -147 dBw

where S is the signal power on the receiver system antenna terminals for a SNR(output) of 13 dB. The SNR(output) is also the SNR applied to the phase measurement system input.

REFERENCES:

[1] Bertiger,W.I., Lichten,S.M., Katsigris,E.C., "A Demonstration of Sub-Meter GPS Orbit and High Precision User Positioning", **IEEE Position Location and Navigation Symposium**, 1988.

[2] Blanchard, A., Phase-Lock Loops, John Wiley and Sons, New York, NY, 1976.

[3] Brookner, E., Radar Technology, Chapter 15, Artech House Inc., Dedham Massachusetts, 1977.

[4] Brookner, E., "Atmospheric Propagation and Communication Channel Model for Laser Wavelengths", **IEEE Transactions on Communications Technology**, Vol. Com-18, pp. 396-416, August 1970.

[5] Brookner, E., "Limit imposed by atmospheric dispersion on the minimum laser pulse width that can be transmitted undistorted", **Proc. IEEE**, June 1969.

[6] Bucher, E.A., and Lerner, R.M., "Experiments on light pulse communications and propagation through the atmospheric clouds", **Appl. Opt.**, Vol. 12, No. 10, 2401-2414 (1973).

[7] Carlson, A.B., Communications Systems: An Introduction to Signals and Noise in Electronic Communications, McGraw-Hill, New York, New York, 1986.

[8] Clarke, K.K., Hess, D.T., Communications Circuits: Analysis and Design, Addison-Wesley Publishing Company, Reading Massachusetts, 1971.

[9] Collin, R.E., Antennas and Radio Wave Propagation, McGraw Hill Book Company, 1985.

[10] Davis, J.I., "Consideration of Atmospheric Turbulence in Laser System Design", **Appl. Opt.**, Vol. 5, No. 1, January 1966.

[11] Feuerstein,M.J, Beliveau,Y.J., Rappaport,T.S., and Pratt,T., "The distribution of phase errors in a UHF position location system", to appear in **Electronic Letters**.

[12] Feuerstein,M.J., private communication, Aug. 17, 1989.

[13] Gardner, C.S., Plonus, M.A., "optical Pulses in Atmospheric Turbulence", **Journal of the Optical Society of America**, Vol. 64, No. 1, Jan. 1974.

[14] Gardner, F.M., Phaselock Techniques, John Wiley and Sons, New York, NY, 1979.

[15] Goade, C., A.S.C.E. Construction Congress Presentation, San Francisco, CA. March, 1989.

[16] Goldstein, I., Miles, P.A., and Chabot, A., "Heterodyne measurements of light propagation through atmospheric turbulence", **Proc. IEEE**, Vol. 53, No. 9, Sept. 1965.

[17] Ito, S. and Furutsu, K., "Theory of light pulse propagation through thick clouds", **J. Opt. Soc. Am.**, Vol 70, No. 4, April 1980.

[18] Iyer,R.S., "Arrival Times of Satellite-Broadened Laser Pulses", **IEEE Transactions On Aerospace And Electronic Systems**, Sept., 1976.

[19] Jain, A. K., "Image Data Compression: A Review", **Proceedings of the IEEE**, Vol. 69, No. 3, March 1973.

[20] Johnson, R.A., Mechanical Filters In Electronics, John Wiley and Sons, New York, New York, 1983.

[21] Krauss, H.L., Bostian, C.W., Raab, F.H., Solid State Radio Engineering. John Wiley and Sons, New York, New York, 1980.

[22] Lui, C.H., and Yen, K.C., "Propagation of pulse beams through turbulence, cloud, rain or fog", **J. Opt. Soc. Am.** Vol. 67, No. 9, Sept. 1977.

[23] Millan, J., Microelectronics: Digital and Analog Circuits and Systems, McGraw-Hill, New York, New York, 1979.

[24] Nussmeier, T.A., Goodwin, F.E., and Zavin, J.E., "A 10.6 micrometer Terrestrial Communication Link", **IEEE J. Quant. Electron.**, Vol. QE-10, pt. 2, pp. 230-235, Feb. 1974.

[25] Pouzet, A.H., "Characteristics of Phase Detectors in Presence of Noise", **Proceedings on International Telemetry Conference**, 1972.

[26] Pratt, W.K., Laser communication systems, John Wiley and Sons, New York,NY, 1969.

[27] Pratt, T., and Bostian, C.W., Satellite Communications, John Wiley and Sons, New York,NY, 1986.

[30] Skolnik, Merrill I. Introduction to Radar Systems, McGraw Hill, 1980.

[31] Sonnenberg, G.J., Radar and Electronic Navigation, Newnes Butterworth, London, 1978.

[32] Stutzman, W.L. and Thiele, G.A., Antenna Theory and Design, John Wiley and Sons, New York, New York., 1981.

[33] Tatarski, V.I., **Wave Propagation in a Random Medium**, Magraw-Hill, New York, 1961.

[34] Wells, David et. al., **Guide to G.P.S. Positioning**, Canadian GPS associates, New Brunswick, Canada, 1986.

[35] "Crystal Oscillators, 1987", Published by Vectron Laboratories, Inc. Norwalk, CT., 1987.

The vita has been removed from the scanned document