ORGIA INSTITUTE OF TECHNOLOGY OFFICE OF CONTRACT ADMINISTRATION

PROJECT ADMINISTRATION DATA SHEET

☒ ORIGINAL ☐ REVISION NO. ______

oject No. A-3366 GTRI/ DATE 10/25/82

oject Director: M. N. Cohen Lab RAIL/AD

onsor: Naval Sea Systems Command, Washington, D. C.

pe Agreement: Contract N00024-82-K-5360

ward Period: From 9/22/82 To ~~3/22/82~~ 4/22/83 (Performance) ______ (Reports)

onsor Amount: Total Estimated: $ 77,100 Funded: $ 77,100

st Sharing Amount: $ None Cost Sharing No: N/A

tle: Phase II - IPAR Investigations

DMINISTRATIVE DATA OCA Contact William F. Brown Ext. 4820

Sponsor Technical Contact:	2) Sponsor Admin/Contractual Matters:
Mr. C. E. Jedrey, SEA-62R13	Mr. T. A. Bryant
Naval Sea Systems Command	ONR RR
Washington, D. C. 20362	214 O'Keefe Building
(202) 692-9760	Atlanta, GA 30332
	(404) 881-4213

efense Priority Rating: ______ Military Security Classification: Secret

(or) Company/Industrial Proprietary: ______

ESTRICTIONS

e Attached Govt. Supplemental Information Sheet for Additional Requirements.

avel: Foreign travel must have prior approval – Contact OCA in each case. Domestic travel requires sponsor approval where total will exceed greater of $500 or 125% of approved proposal budget category.

uipment: Title vests with None proposed or anticipated.

OMMENTS:

Note: DD Form 1423 referenced in contract is deleted per telecon discussion on 10/18/82 between W. F. Brown and Sponsor. Data requirements will be per our proposal RJ-RAD-1209.

OCT 1982
RECEIVED
Research Reports
Office

OPIES TO:

esearch Administrative Network
esearch Property Management
ccounting
ocurement/EES Supply Services

Research Security Services
Reports Coordinator (OCA)
GTRI
Library

Research Communications (2)
Project File
Other ______
Other ______

GEORGIA INSTITUTE OF TECHNOLOGY OFFICE OF CONTRACT ADMINISTRATION

SPONSORED PROJECT TERMINATION SHEET

Date July 12, 1983

Project Title: Phase II-IPAR Investigations

Project No: A-3366

Project Director: M. N. Cohen

Sponsor: Naval Sea Systems Command

Effective Termination Date: 4/22/83

Clearance of Accounting Charges: 4/22/83

Grant/Contract Closeout Actions Remaining:

- ☒ **Final Invoice and Closing Documents**
- ☐ Final Fiscal Report
- ☒ Final Report of Inventions
- ☒ Govt. Property Inventory & Related Certificate
- ☐ Classified Material Certificate
- ☐ Other ______

Assigned to: RAIL-DD (~~School~~/Laboratory)

COPIES TO:

Administrative Coordinator
Research Property Management
Accounting
Procurement/EES Supply Services

Research Security Services
[✓ Reports Coordinator (OCA)]
Legal Services (OCA)
Library

EES Public Relations (2)
Computer Input
Project File
Other Cohen

FORM OCA 10:781

ENGINEERING EXPERIMENT STATION
Georgia Institute of Technology
A Unit of the University System of Georgia
Atlanta, Georgia 30332

October 29, 1982

Naval Sea Systems Command
Code 62R13
Washington, D. C. 20362

Attention: Mr. Charles Jedrey

Subject: Monthly Contract Technical Status Report No. 1, "Phase 2 - IPAR Investigations," Contract No. N00024-82-K-5360, Georgia Tech Project A-3366, covering the period from September 22, 1982 through October 31, 1982

Gentlemen:

This status report summarizes program activities performed under the subject contract for the period September 22, 1982 through October 31, 1982.

TECHNICAL ACTIVITIES

Technical activities have included the IPAR system integration into the GT1 radar van, gathering and fabrication of the necessary ancillary equipment, and development of detailed local and Panama City data collection plans. The one-year layoff in funding resulted in partial dismantling of the IPAR system and reassignment of previously involved personnel. At this point, the system integration is 90% complete. A mixer-preamp was sent to the manufacturer for repair and is now on its way back. Two two-channel strip chart recorders are required for this field operation. One Georgia Tech recorder has been obtained, and if another is not available, the second will have to be rented. The personnel conducting the meansurement program are: Marvin N. Cohen, Project Director; Benjamin Perry, RF Engineer and Data Analyst; and Richard Folea, Co-op Student. Corner reflectors for use in the experiments are being constructed, and an existing pole and movable reflector will be refurbished for the multipath interference experiment.

The facilities manager at the Naval Coastal Systems Center (NCSC, Panama City, Florida), Niles Schuh, has been contacted and preliminary arrangements for the planned field operation are being made with him.

Mr. Eric Sjoberg is not yet back from London, but other Georgia Tech attendees report that his IPAR presentation at RADAR-82 was extremely well received.

FUTURE WORK

The local data collection will commence November 8th and end November 13th. The NCSC multipath interference and sea clutter data collection field exercise will commence November 22nd and continue to as late as December 5th, if necessary.

Respectfully submitted,

Marvin N. Cohen, Ph.D.
Project Director

APPROVED:

Robert N. Trebits, Ph.D.
Chief, Analysis Division

ENGINEERING EXPERIMENT STATION
Georgia Institute of Technology
A Unit of the University System of Georgia
Atlanta, Georgia 30332

January 11, 1983

Naval Sea Systems Command
Code 62R13
Washington, D. C. 20362

Attention: Mr. Charles Jedrey

Subject: Monthly Contract Technical Status Report No. 2, "Phase 2 - IPAR Investigations," Contract No. N00024-82-K-5360, Georgia Tech Project A-3366, covering the period from November 1, 1982 through November 30, 1982

Dear Mr. Jedrey:

This status report summarizes program activities performed under the subject contract for the period November 1, 1982 through November 30, 1982.

TECHNICAL ACTIVITIES

Data collection with the IPAR system was begun locally at Dobbins Air Force Base on November 23rd. A full matrix of data was collected on simple trihedral targets, trihedral targets in a (grass) multipath environment, and grass clutter. Two FM and two digital tapes of data were recorded. A preliminary analysis indicates that the data are correct and will be easily reduced when that phase of the program begins.

Although the local data collection was not yet completed, I decided to adhere to schedule and leave for Panama City, Florida for the sea measurements on November 29th. The local data collection will be resumed upon our return. As of November 30th, the system was set up, tested, and operational at a site overlooking the bay at NCSC in Panama City.

At the request of the organizers, we have submitted a paper describing IPAR and its potential for lowered probability of intercept (LPI) operation to MSAT'83 (sponsored by Microwave Systems and Technology, Inc.). The paper has been accepted and is scheduled for presentation on March 10th, 1983 at the Washington, D. C. Sheraton. The conference is being held Monday through Friday of that week. A copy of the paper is enclosed for your approval.

FUTURE WORK

The bay and sea measurements will commence December 1, 1982 and continue through December 15th. During this time we plan to collect a full matrix of multipath interference data over bay and sea, bay and sea backscatter data, and some signatures from complex targets of opportunity.

Respectrully submitted,

Marvin N. Cohen, Ph.D.
Project Director

Enclosure

APPROVED:

Robert N. Trebits, Ph.D.
Chief, Analysis Division

THE INTRAPULSE POLARIZATION AGILE RADAR

By: Dr. M. N. Cohen, Mr. E. E. Martin, Mr. E. S. Sjoberg
Georgia Institute of Technology
Engineering Experiment Station
Atlanta, Georgia USA

INTRODUCTION

The Intrapulse Polarization Agile Radar (IPAR) system transmits a pulse that is encoded by polarization modulation on a subpulse basis. This coding is then utilized to effect pulse compression of the received echo pulse. This method differs from the more classical approaches of pulse compression encoding on carrier phase or frequency in that the coding is contained in the relative phase between the horizontal and vertical polarization components of the transmit pulse. As a consequence of this novel approach, IPAR exhibits many unique characteristics including the ability to be implemented with a variety of RF waveforms such as single frequency, frequency agile, or noise carriers, Doppler-invariant pulse compression, and an intrinsic potential for discriminating stationary targets from surrounding clutter.

The current IPAR system has been implemented in an X-band radar designed and built at Georgia Tech. Traveling wave tubes amplify the independent horizontal and vertical polarization components of the RF energy which is then transmitted via a dual-mode feed as right or left circular polarization. The heart of the IPAR system is a high speed digital processor implemented in TTL and ECL technology. Binary codes up to 32 bits in length with bit rates up to 100 MHz may be generated and processed in real time.

This paper first presents a very brief summary of the utility of pulse compression in modern-day radar systems, followed by a discussion of the basic elements of polarization theory that are important for an understanding of the IPAR system's signal processing. Significantly more detailed discussions of each of these topics may be found in any of a number of standard texts (e.g., refs 1 and 2).

The paper proceeds with discussions of the theoretical basis for IPAR-type processing, the hardware configuration of the current IPAR system, some unique properties of IPAR waveforms that are of general interest, and the properties of IPAR waveforms that are most germane to low probability of intercept (LPI) operation. The paper ends with a description of ongoing efforts in this area as well as future research directions.

PULSE COMPRESSION

Historically, pulse compression was first achieved utilizing linear frequency modulated (LFM) waveforms. More recently, as a result of the ascendancy of digital signal processing, biphase coding of the carrier RF as well as discretely stepped inter- and intra-pulse frequency modulation have come to the fore.

A pulse compression radar codes a transmit pulse of duration T to attain a bandwidth of $1/\tau$, where $T >> \tau$. By matched filter processing of the echo pulse, the received signal is compressed to a width τ, which achieves a range resolution of $\frac{1}{2}c\tau$. A pulse compression system thus achieves the resolution of a pulsed radar that utilizes a pulse of duration τ while in actuality transmitting a pulse of duration T. The primary applications of pulse compression in modern-day radars include: (1) signal-to-noise ratio enhancement due to the compression gain (T/τ), (2) clutter reduction due to the resulting decrease in range cell size, (3) lowered probability of intercept (LPI) operation, again due to the pulse compression gain, and (4) realization of increased waveform flexibility for the implementation of multi-mode radars (e.g., detect, track, identify).

POLARIZATION

Radar signals are polarized in that the E-field modulation is constrained to a particular plane in 3-dimensional space or the plane of modulation can be varied in a particular manner. For example, two orthogonal linear polarizations are vertically (V) polarized and horizontally (H) polarized electromagnetic waves, where the plane of polarization is defined as that perpendicular to which no E-field modulation can be detected. Electromagnetic waves with elliptical polarizations are generated by transmitting sinusoidal H and V polarized waves simultaneously which differ in amplitude and/or phase. If the amplitudes of the H and V sinusoids are equal in amplitude and are out of phase by 90^{o}, then the polarization is said to be circular. If, in addition, the H component lags the V component by 90^{o}, the resultant wave's plane of polarization varies according to the right hand rule in the direction of propagation, and the wave is said to be right circularly (RC) polarized. When the H component leads the V component by 90^{o}, the wave is left circularly (LC) polarized. RC and LC are orthogonal polarizations.

The circular polarization of a reflected wave is affected by the nature of the reflecting surface. In particular, the circular polarization of a wave reflected from an odd-bounce scatterer is opposite in sense from the circular polarization of the incident wave (e.g., RC incident yields LC reflected and vice versa). As can be deduced from the odd-bounce case, even-bounce reflectors leave circular polarizations unchanged (e.g., RC incident results in RC reflected). The phenomenon underlying these effects is that on a single bounce the relative phase between the H and V components of the wave remains unchanged (since both are flipped by 180^{o} in phase) while the direction of the propagation is reversed. The net result is a change in handedness of the electromagnetic wave, just as one would observe a change in handedness of the rotation of a transparent clock when it is viewed from front and back.

PULSE COMPRESSION ON POLARIZATION MODULATION

The IPAR system is capable of switching between right and left circular polarization on an intrapulse basis at rates up to 100 MHz. Utilizing separate H and V ports on receive, IPAR downconverts the two channels separately and accurately measures the phase between them. Let ϕ_{HV} represent the phase difference between the H and V channels (that is, $\phi_{HV} = \phi_H - \phi_V$), and recall that a signal is RC polarized if $\phi_{HV} = +90^\circ$ and LC polarized if $\phi_{HV} = -90^\circ$. For convenience, let $S\phi = \sin\phi$. Then $S\phi_{HV} = 1$ for RC signals and $S\phi_{HV} = -1$ for LC signals, thus giving a natural correspondence between binary codes and intrapulse polarization modulation. IPAR utilizes this correspondence to construct well-behaved polarization codes from the class of well-known, well-behaved binary codes.

Figure 1(a) depicts a natural choice for an IPAR transmit waveform. According to the correspondence described above, the chosen code represents a 13 bit Barker code and yields a pulse compression ratio of 13 to 1. Figures 1(b) and 1(c) represent the expected returns from an idealized flat plate (or any other odd-bounce scatterer) and an idealized dihedral (or any other even-bounce scatterer), respectively.

The IPAR processor computes, on a subpulse basis, $S\phi_{HV}$ of any received echo waveform and passes the results of that computation through a filter matched to the coding on the transmit waveform. Figure 2(a) represents the output from the matched filter compressor given the return, as depicted in Figure 1(b) from an odd-bounce reflector. Note that the time resolution is $\tau/2$ rather than $T/2$ (a 13-to-1 improvement) and that the peak signal voltage is T/τ or 13 times the nominal uncompressed level. Figure 2(b) represents the output in response to an even-bounce reflector. Note that its power and resolution characteristics are precisely the same as those of 2(a) although there is a change of sign in voltage.

If the reflecting surface is more complex than an idealized simple scatterer, then so will the receive waveform and, thus, the output from the matched filter be more complex, especially so if time averaging or a spread spectrum carrier is used to decorrelate the target in polarization. These effects are described in more detail elsewhere (ref 3).

IPAR IMPLEMENTATION

IPAR was designed and built as a demonstration and data collection system. The goals were first to establish that the IPAR technique would work and second to collect data to quantify IPAR's strengths and weaknesses relative to classical radar techniques.

Some of the design goals were:

(1) achieve a 100 MHz real time correlation and compression rate;
(2) correlate and compress using noncoherent radar techniques and a wide bandwidth noise RF carrier;
(3) permit growth to computer control and data analysis;
(4) incorporate the ability to range gate and replay correlation data for quick, on-site observation and analysis.

Figure 3 shows a block diagram of the resulting system. Functionally, the system may be viewed as two subsystems: the RF subsystem and the IPAR digital processor subsystem. The basic system operation is described in the following paragraphs.

The IPAR system includes a two-channel master oscillator power amplifier (MOPA) radar transmitter. The two channels are connected to the vertical and horizontal ports of a dual-polarized antenna feed. Whenever a 90° phase relationship exists between the two channels, the resultant transmission is circularly polarized. By switching the phase relationship to a negative 90°, the opposite sense circular polarization is transmitted. By switching in accordance with an advantageous coding sequence, a pulse is transmitted which is coded right and left circular.

The phase shift for one channel referenced to the other is achieved at a low power level due to the speed limitations of high powered phase shifters. For this reason, the IPAR system was built with two high power amplifier channels. The phase shift of one channel is accomplished at low power using a mixer, which, for the devices used, could conceivably modulate the polarization at rates in excess of 2 GHz. The mixer (i.e., phase shifter or phase modulator) has the characteristic that a high level signal (i.e., +1) at the IF port causes zero phase shift from the input to output, whereas a low level signal (i.e., -1) causes a phase shift of 180°. With a permanent 90° phase shift added to the 0°, 180° phase shift combinations, the result is -90° or 90°. Hence, the required phase relationships are generated for the right and left circular polarization-coded transmissions, which may be at a high data rate independent of carrier frequency.

In fact, two RF sources in addition to a narrowband 9.35 GHz carrier have been incorporated in the current system. An HP 624C signal generator may be used to impress a 60 Hz sinusoidal frequency spread of 30 to 40 MHz on the transmit pulse to provide a frequency agile mode of operation. In addition, a noise source with a spectral width of 140 MHz may also be used to modulate the RF carrier (ref 4).

As in transmission, two receive channels are employed. The signals in the two receive channels are translated down in frequency, retaining the relative phase between the two signals, and applied to a phase detector. The polarity and amplitude of the resultant bipolar video signal will be similar to the coding impressed on the transmission (and stored in the correlator) provided that the return is of high fidelity and not noise-like. The high speed correlator samples the returning signal at a maximum rate of 100 MHz. Future VHSIC and gallium arsenide technology advances could increase this rate considerably. Once the video signal has been quantized and input to the correlator, the remainder of the operation proceeds like conventional, biphase modulated, pulse compression techniques.

The prototype IPAR digital processor was designed to provide a flexible, complete, and self-contained radar signal correlation processor. It may be operated in a stand-alone mode, under front panel control, or connected to and controlled by a minicomputer or microcomputer. The processor is capable of correlating received signals with the stored reference code in real time at rates up to 100 MHz. The operator has freedom to independently determine the transmitted code, subpulse width, PRF, and range gate setting to achieve his objectives.

Figure 4 presents a simplified block diagram of the digital processor and its interfaces to the external world. The slower TTL processor (6 MHz) may be considered the host or control portion of the unit, which interfaces with the front panel and the control computer (when provided) and determines the operational mode of the processor. The high speed ECL processor interfaces with the radar generating the transmit and coding signals, receives and quantizes the video from the radar, performs correlation processing in real time and stores data from the range gate position. A list of basic capabilities is presented in Table 1.

TABLE 1. IPAR DIGITAL PROCESSOR CAPABILITIES

Subpulse Length:	10 - 160 ns (100 - 6.25 MHz)
PRF:	500 - 8,000 pps plus manual
Range Gate:	0 - 15 km, 0.75 m maximum accuracy
Codes:	Any binary code up to 32 bits long
Correlation Processor:	True correlation for 32 bit code length Pseudo-correlation for 1 - 31 bit code lengths
Digital Threshold:	Adjustable
Self Test/Calibration:	Built-in

IPAR CHARACTERISTICS

Classical forms of pulse compression are quite senstive to Doppler shift. Linear FM and its discrete approximations exhibit this sensitivity in the form of range-Doppler coupling (see, for example, ref 1) which makes it impossible to distinguish short range, high velocity targets from distant, stationary ones on a single pulse basis. Binary phase codes evince this sensitivity in the form of a severe loss-in-processing gain for targets whose radial velocities are relatively high. However, since IPAR processes only the relative phase between the two polarization components of the received pulse, IPAR processing should prove insensitive to Doppler shifts of the receive signal.

Returns from reflectors that preserve IPAR's polarization coding or simply rotate it 180^0 compress fully and steadily. Returns from reflectors that do not preserve IPAR's coding compress poorly and sporadically. Assuming a spread spectrum carrier and/or pulse-to-pulse integration to decorrelate complex (in polarization) targets, it follows that the more complex a target is on a subpulse basis, the more poorly it will compress.

Finally, assuming that at sufficiently fine resolutions man-made targets are less complex in polarization than many types of clutter (sea clutter, grass, and trees for examples), it follows that IPAR processing should provide inherent clutter suppression and stationary target-to-clutter discrimination capabilities. These capabilities have, in fact, been documented both through theoretical analyses (ref 5) and qualitative observations (ref 4).

Additional characteristics and potential applications of IPAR and IPAR-like systems include reduction of multipath interference, target characterization in terms of polarization-signed range profiles for identification purposes, and potential for LPI operation.

LPI CHARACTERISTICS

Lowered probability of intercept (LPI) operation results from many factors including system beamwidth, dwell time, and PRF; operational scenario and strategy; waveform characteristics; and the intercept receiver. In addition, the problem contains so many elements of one-upsmanship that a reasonable measure-of-goodness for an LPI technique might be the ratio of cost and effort required to implement the technique to the cost and effort required to defeat the technique. Intrapulse polarization modulation, being a new, unique, and extremely flexible technique, embodies various characteristics which may prove valuable as a basis for an LPI compatible waveform.

The three features of IPAR-type orocessing that seem most important to its worth as a quiet radar technique are: (1) compatibility with various RF carriers including noise, (2) pulse compression processing, and (3) polarization diversity.

IPAR's pulse compression is indeoendent of the RF carrier and may be implemented in a carrier-diverse system where the carrier may be made to vary radically on a pulse-to-pulse or as-necessary basis. In particular, IPAR has been demonstrated with a wideband noise carrier as well as narrowband and frequency swept sources. Thus all of these carriers are available in devising a strategy to defeat an unintended listener.

The pulse compression gain achieved by IPAR is the same as that achieved by the classical forms of pulse compression. Thus, IPAR achieves the same signal-to-noise advantage that more standard pulse compression radars achieve over a "dumb" intercept receiver.

The inherent polarization diversity in IPAR-type processing leads to various results depending on the assumed polarization characteristics of the intercept receiver. Under most assumptions, the intercept receiver experiences 3 dB more loss against IPAR than against a more convential, fixed polarization radar, and under the most severe assumption (that the intercept receiver is H and V polarized and can process the RC and LC polarization), IPAR comes out no worse than a conventional radar.

CURRENT ACTIVITIES AND FUTURE DIRECTIONS

The current IPAR system has been field tested. During these field tests many of its theoretically-predicted properties were qualitatively verified (refs 3 and 5). Among these properties were pulse compression on narrowband, frequency swept, and wideband noise RF carriers, and the suppression of tree clutter returns with respect to simple targets when the frequency swept and noise source carriers were employed.

At this writing the IPAR system is being utilized to collect reflectivity data on simple targets, multipath over the sea, and sea clutter. These data will be reduced and analyzed to provide proof-of-concept and to quantify IPAR's performance under various scenarios.

In addition, an advanced IPAR system is being designed and built which includes several significant advances over the current system:

(1) embedded microprocessor controller;
(2) 6 bit A/D conversion of the phase video at 100 MHz;
(3) 64 bit code length and true correlation for all codes;
(4) signal integration capability;
(5) Pseudo real time processing of 256 range bins;
(6) compatibility with 500 MHz operation.

The embedded microprocessor controller will allow pulse-to-pulse code agility as well as provide a signal integration capability and an automatic digital tape recording facility. By employing 6 bit A/D converters, small scatterers separated by more than one subpulse length from a larger scatterer will be resolved from the larger reflector, which will enable true target polarization profiling. The code length will be expanded to 64 bits, and a true correlation will be formed for all possible codes including those with embedded zeros. The extended code length and correlation capability is obtained at the expense of real time operation. Instead, 256 range cells will be sampled and processed during each pulse repetition interval.

The IPAR system has evolved from a concept to a demonstrated working radar system. When the full potential of the IPAR concept has been investigated, it may prove to be a significant advance in radar and communications technology.

ACKNOWLEDGEMENTS

The results presented herein represent the culmination of four years of work at the Georgia Institute of Technology by the authors and others. In particular, Mr. Bobby C. Appling, currently with Parks-Jaggers Aerospace Company, Orlando, Florida, was responsible for the inception of this program. The authors wish to acknowledge their debt and thanks to him as well as to Mr. Charles Jedrey, Naval Sea Systems Command, U. S. Navy, for his continuous funding of this work.

REFERENCES

1. M. Skolnik, Introduction to Radar Systems, McGraw-Hill, Inc., New York, New York, 1980.

2. F. Nathanson, Radar Design and Principles, McGraw-Hill, New York, New York, 1969.

3. M. N. Cohen and E. S. Sjoberg, "Intrapulse Polarization Agile Radar," RADAR '82, London, England, October, 1982.

4. B. C. Appling, E. S. Sjoberg, E. E. Martin, "Intrapulse Polarization Agile Radar," SCEEE Contract No. N00024-78-C-5338, SCEEE-NAVSEA/79-2, GIT Final Report, July, 1981.

5. M. N. Cohen, E. S. Sjoberg, E. E. Martin, "Intrapulse Polarization Agile Report Development Program," SCEEE Contract No. SCEEE/81-2, Prime Contract No. NAVSEA N00024-78-C-5338, GIT Final Report, December, 1981.

ENGINEERING EXPERIMENT STATION
Georgia Institute of Technology
A Unit of the University System of Georgia
Atlanta, Georgia 30332

January 11, 1983

Naval Sea Systems Command
Code 62R13
Washington, D. C. 20362

Attention: Mr. Charles Jedrey

Subject: Monthly Contract Technical Status Report No. 3, "Phase 2 - IPAR Investigations," Contract No. N00024-82-K-5360, Georgia Tech Project A-3366, covering the period from December 1, 1982 through December 31, 1982

Dear Mr. Jedrey:

This status report summarizes program activities performed under the subject contract for the period December 1, 1982 through December 31, 1982.

TECHNICAL ACTIVITIES

A full matrix of multipath (simple reflectors) over bay and sea; bay clutter and sea clutter data of opportunity, and some signatures of passing vessels, the Proline (the boat provided for sea support by NCSC), and Stage Two were recorded. The digital data tapes generated were sent back to Georgia Tech on a daily basis for verification of data format and reasonableness of the data. All preliminary indications are that we have recorded a good, useful set of data.

Due to the vagaries of weather and minor system problems that needed attending, the collection program lasted through December 21. The crew and equipment arrived safely home early the morning of the 22nd. The immediate results of our efforts are 13 FM and 15 digital tapes of IPAR data upon which to base our analyses.

The remainder of the work month (December 22-24) was spent putting equipment and paperwork in order.

Enclosed for your approval is a draft copy of our paper, "IPAR as a Target Identification Radar," that will be submitted for presentation on February 9, 1983 at CISC '83 in Monterey, California.

FUTURE WORK

A two pronged effort will begin January 3, 1983. Misters Benjamin Perry, Brett Freeman, and Michael Baden will work at completing the local measurements interrupted by the excursion to NCSC. The data to be collected include IPAR backscatter from complex targets configured from combinations of dihedrals and trihedrals, tree clutter, and targets in tree clutter, as well as signatures from some targets of opportunity. The effort should take a week or two depending on factors such as weather. At the same time, Misters Michael Shannon and Richard Folea will begin developing the architecture of and software for the construction of a complete, efficient data base consisting of all the data collected on this program. I, as Project Director, will be deeply involved in both efforts, providing direction, guidance, and coordination.

Respectrully submitted,

Marvin N. Cohen, Ph.D.
Project Director

APPROVED:

Robert N. Trebits, Ph.D.
Chief, Analysis Division

"IPAR AS A TARGET IDENTIFICATION RADAR"

By: Dr. M. N. Cohen and Mr. E. S. Sjoberg

Georgia Institute of Technology
Engineering Experiment Station
Atlanta, Georgia 30332 USA

"IPAR AS A TARGET IDENTIFICATION RADAR"

By: Dr. M. N. Cohen and Mr. E. S. Sjoberg

Georgia Institute of Technology
Engineering Experiment Station
Atlanta, Georgia 30332 USA

ABSTRACT

The Intrapulse Polarization Agile Radar (IPAR) system transmits a pulse that is encoded by polarization modulation on a subpulse basis. The coding is then utilized to effect pulse compression of the received echo pulse. This method differs from the more conventional approaches of pulse compression encoding on carrier phase or frequency in that the coding is contained in the relative phase between the horizontal and vertical polarization components of the transmit pulse. As a consequence of this novel approach, IPAR exhibits many unique characteristics including the ability to be implemented with a variety of RF waveforms such as single frequency, frequency agile, or noise carriers, Doppler-invariant pulse compression, and an intrinsic potential for discriminating stationary targets from surrounding clutter.

The current IPAR system has been implemented in an X-band radar designed and built at Georgia Tech. Traveling wave tubes amplify the independent horizontal and vertical polarization components of the RF energy which is then transmitted via a dual-mode feed as right or left circular polarization. The heart of the IPAR system is a high speed digital processor implemented in TTL and ECL technology. Binary codes up to 32 bits in length with bit rates up to 100 MHz may be generated and processed in real time. The system has been successfully demonstrated, and has been utilized for a comprehensive data collection program. The resulting data are currently being reduced and analyzed to investigate and quantify some of IPAR's predicted properties.

INTRODUCTION

The exposition that follows begins with a very brief review of the elemental concepts necessary for a thorough understanding of the IPAR system's signal processing as applied to target identification. The utility of pulse compression in modern-day radar systems, the basic, germane elements of polarization theory, and some fundamentals of target identification terminology and concepts are presented. Since there discusions are of necessity quite brief, references for more complete descriptions are given at each heading.

The paper continues with a discussion of the IPAR technique for pulse compression on polarization modulation. The properties of IPAR and IPAR-like waveforms are then presented in the context of their applicability to target identification. Target-to-clutter enhancement, target/clutter discrimination, and high resolution operation for target characterization are discussed. This is followed by a description of other IPAR characteristics that may prove useful for various operational scenarios and tasks.

A description of the actual hardware implementation in the currently operational Georgia Tech system is given, and the paper closes with a summary of on-going and planned, future work in this area at Georgia Tech. The recently completed IPAR data collection program, plans for reduction and analysis of these data, and progress on the configuration of an advanced development model are described.

PULSE COMPRESSION(1,2)

Historically, pulse compression was first achieved utilizing linear frequency modulated (LFM) waveforms. More recently, as a result of the ascendancy of digital signal processing, biphase coding of the carrier RF as well as discretely stepped inter- and intra-pulse frequency modulation have come to the fore.

A pulse compression radar codes a transmit pulse of duration T to attain a bandwidth of $1/\tau$, where $T >> \tau$. By matched filter processing of the echo pulse, the received signal is compressed to a width τ, which achieves a range resolution of $\frac{1}{2}c\tau$. A pulse compression system thus achieves the resolution of a pulsed radar that utilizes a pulse of duration τ while in actuality transmitting a pulse of duration T.

The primary applications of pulse compression in modern-day radars include: (1) signal-to-noise ratio enhancement due to the compression gain (T/τ), (2) clutter reduction due to the resulting decrease in range cell size, (3) lowered probability of intercept (LPI) operation, again due to the pulse compression gain, and (4) realization of increased waveform flexibility for the implementation of multi-mode radars (e.g., detect, track, identify).

POLARIZATION(3)

Radar signals are polarized in that the E-field modulation is constrained to a particular plane in 3-dimensional space or the plane of modulation can be varied in a particular manner. For example, two orthogonal linear polarizations are vertically (V) polarized and horizontally (H) polarized electromagnetic waves, where the plane of polarization is defined as that perpendicular to which no E-field modulation can be detected. Electromagnetic waves with elliptical polarizations are generated by simultaneously transmitting sinusoidal H and V polarized waves which differ in amplitude and/or phase. If the H and V sinusoids are equal in amplitude and are out of phase by 90^{0}, then the polarization is said to be circular. If the H component is made to lag the V component by 90^{0}, the resultant wave's plane of polarization varies according to the right hand rule in the direction of propagation, and the wave is said to be right circularly (RC) polarized. When the H component leads the V component by 90^{0}, the wave is left circularly (LC) polarized. RC and LC are orthogonal polarizations.

The polarization of a reflected wave is affected by the nature of the reflecting surface. In particular, the circular polarization of a wave reflected from a conducting odd-bounce scatterer is opposite in sense from the circular polarization of the incident wave (e.g., RC incident yields LC reflected and vice versa). As can be deduced from the odd-bounce case, even-bounce reflectors leave circular polarizations unchanged (e.g., RC incident results in RC reflected). The phenomenon underlying these effects is that on a single bounce the relative phase between the H and V components of the wave remains unchanged (since both are flipped by 180^{0}) while the direction of the propagation is reversed. The net result is a change in handedness of the electromagnetic wave, just as one would observe a change in handedness of the rotation of a transparent clock when it is viewed from front and back.

TARGET IDENTIFICATION(4)

With the recent advances in millimeter wave radar and other sensor technologies and the tremendous improvements in digital signal processing capabilities, automatic target identification has become a feasible method for force-effectiveness multiplication, and has thus received a great deal of attention.

The task of automatic target identification may be conceptualized as a three step process: detection, discrimination, and classification. Detection refers to distinguishing target returns from noise. This stage of target identification may be achieved through the use of classical techniques and careful system design. It does not require new technology to be successfully addressed.

Target discrimination refers to the task of distinguishing potential target returns from clutter returns. It is generally assumed that this function will be performed in a search mode. To perform this task successfully it is necessary to examine each range-azimuth bin in the field of view and to make a decision as to whether each does or does not contain a potential target of interest. Since in most applications, especially air-to-ground or ground-to-ground, one cannot assume a high target-to-clutter ratio, these decisions must be based on some characteristics that distinguish targets from clutter as opposed to mere amplitude differences in the respective returns. This situation presents three primary problems: (1) to find characteristics that distinguish certain man-made objects from as many naturally ocurring objects as possible, (2) to discover the waveforms that are most responsive to these characteristics, and (3) to overcome the processing limitations imposed by implementing a possibly-sophisticated algorithm for each of a very large number of range-azimuth bins.

Although no completely satisfactory method has yet been devised for the discrimination of stationary ground targets from clutter on the basis of radar backscatter, there seems to be a consensus within the community as to the approaches necessary for the solution of each of these problems. For properly chosen range-azimuth bin sizes, it appears that many types of clutter (e.g., trees, grass, sea,...) consist of a larger number of major scattering centers than do many manmade objects of interest (e.g., tanks, trucks, naval vessels,...). Furthermore, it is believed in some circles, that potential targets of interest, as a rule, may exhibit certain physical characteristics that can be distinguished from clutter on the basis of the polarization of the backscattered return. Thus reflector complexity and polarization sensitivity have been pursued as characteristics upon which target discrimination may be based.

Polarization and frequency agile waveforms hold great promise for displaying these backscatter characteristics. Frequency agility may be utilized to decorrelate the more complex returns on a pulse-to-pulse or intrapulse basis or to imprint a signature on the reflected waveform that is representative of physical characteristics of the reflector. Polarization diversity may be used to take advantage of the differences in the polarization matrices of targets and clutter. Furthermore, frequency and polarization diversity may be combined to enhance the ability to extract the information that may be garnered from each separately.

Based on considerations of processing constraints, the target and clutter characteristics thought most promising for discrimination, and the nature of the signal processing algorithms required to extract these characteristics, the range resolution of a system performing a discrimination task ought to be on the order of the range extent of the targets of interest. However, it is generally accepted that accurate target classification may require very fine range resolution. An obvious approach to designing a system that can address both discrimination and classification is to design a multimode system that utilizes different waveforms for each task transmitting a relatively long pulse for discrimination, then switching to a short, high resolution pulse for classification. An alternative is to utilize a relatively long transmit pulse modulated by a high bandwidth pulse compression code. Discrimination and classification processing may then be done in parallel on a single pulse basis where the discrimination algorithm is applied to the uncompressed echo and acts as a detector that activates the classifier. Thus pulse compression processing and the implementation of target classification algorithms may be restricted to only those range-azimuth bins that have been designated as potential threats by the discriminator. Note that this approach may be applied to azimuthal resolution in synthetic aperture radar (SAR) systems as well.

PULSE COMPRESSION ON POLARIZATION MODULATION

The IPAR system is capable of switching between right and left circular polarization on an intrapulse basis at rates of up to 100 MHz. Utilizing separate H and V ports on receive, IPAR downconverts the two channels separately and accurately measures the relative phase between them. Let ϕ_{HV} represent the phase difference between the H and V channels (that is, $\phi_{HV} = \phi_H - \phi_V$) ,and recall that a signal is RC polarized if $\phi_{HV} = +90^o$ and LC polarized if $\phi_{HV} = -90^o$. For convenience, let $S\phi = \sin\phi$. Then $S\phi_{HV} = 1$ for RC signals and $S\phi_{HV} = -1$ for LC signals, thus giving a natural correspondence between binary codes and intrapulse polarization modulation. IPAR utilizes this correspondence to construct well-behaved polarization codes from the class of well-known, well-behaved binary codes.

Figure 1(a) depicts a natural choice for an IPAR transmit waveform. According to the correspondence described above, the chosen code represents a 13 bit Barker code and yields a pulse compression ratio of 13 to 1. Figures 1(b) and 1(c) represent the expected

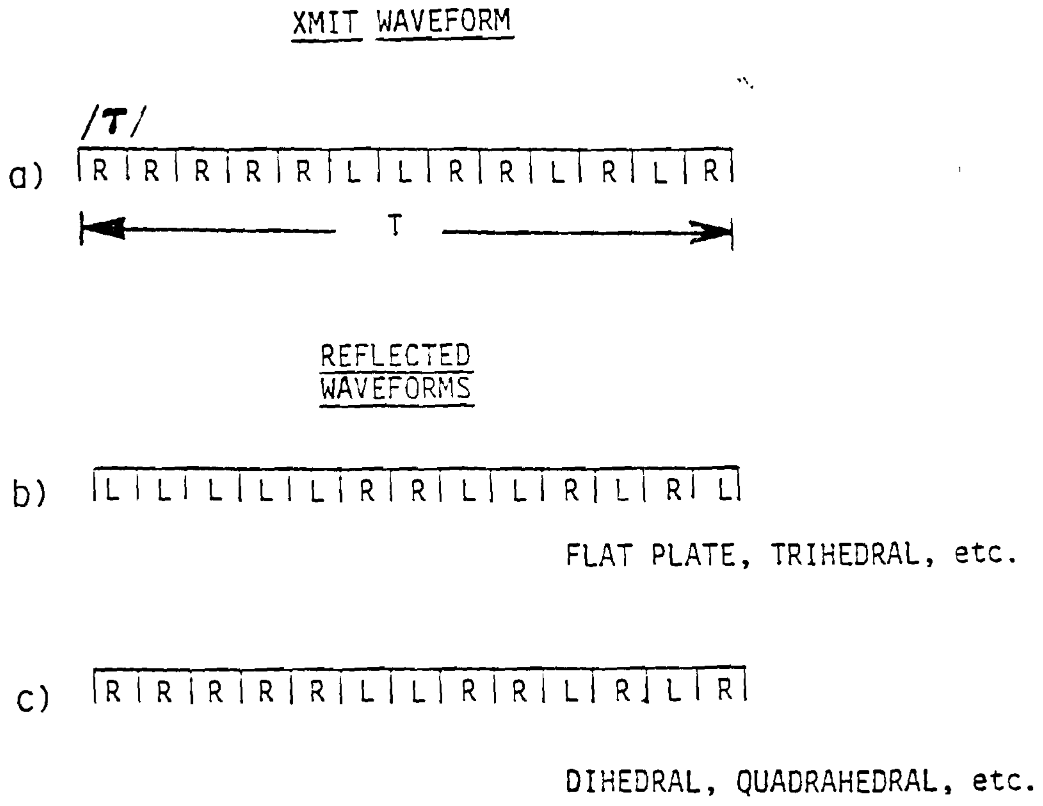

XMIT WAVEFORM
/τ/
a) R R R R R L L R R L R L R
T
REFLECTED WAVEFORMS
b) L L L L L R R L L R L R L
FLAT PLATE, TRIHEDRAL, etc.
c) R R R R R L L R R L R L R
DIHEDRAL, QUADRAHEDRAL, etc.

Figure 1. IPAR Transmit and Receive Waveforms

returns from an idealized conducting flat plate (or any other odd-bounce scatterer) and an idealized dihedral (or any other even-bounce scatterer), respectively.

The IPAR processor computes, on a subpulse basis, $S\phi_{HV}$ of the received echo waveform and passes the results of that computation through a filter matched to the coding on the transmit waveform. Figure 2(a) represents the output from the matched filter compressor given the return, as depicted in Figure 1(b), from an odd-bounce reflector. Note that the time resolution is $\tau / 2$ rather than T/2 (a 13-to-1 improvement) and that the peak signal voltage is T/τ or 13 times the nominal uncompressed level. Figure 2(b) represents the output in response to an even-bounce reflector. Note that its power and resolution characteristics are precisely the same as those of 2(a) although there is a change of sign in voltage.

If the reflecting surface is more complex than a simple scatterer, then the receive waveform and, thus, the output from the matched filter will also be more complex, especially so if time averaging or a spread spectrum carrier is used to decorrelate the target in polarization. These effects are described in more detail elsewhere.(5)

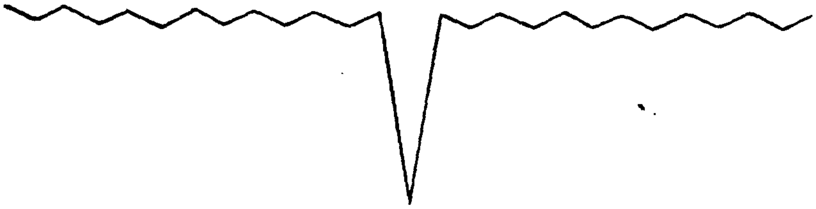
TARGET IDENTIFICATION CHARACTERISTICS OF IPAR

IPAR waveforms embody many characteristics that are particularly germane to solving the problem of automatically identifying stationary ground targets with a radar sensor. Among these system characteristics are polarization diversity, frequency diversity, pulse compression processing, and extreme flexibility. These characteristics allow for system implementation in such a way that target-to-clutter discrimination, target/clutter enhancement, target characterization, and multimode, single-pass operation may be incorporated in a target identification radar configured around its central concept of intrapulse polarimetric pulse compression coding.

The polarization modulation inherent in IPAR processing holds much promise for the discrimination between stationary tactical targets and the clutter backgrounds they are often found in. It has been observed that while the polarimetric phase of the radar backscatter from simple targets (dihedrals and trihedrals) are bipolar, steady, and predictable, the phase of various clutter backscatter (grass, tree, and sea clutter) is spread, albeit in a predictable and recognizable way. The spread has been noted both on a pulse to pulse basis when the narrowband and frequency agile carrier are employed with IPAR as well as within a single pulse when the carrier is modulated by the noise source. Automatic target/clutter discrimination based on these observed phenomena will be attempted on recorded data during the next phase of Georgia Tech's IPAR program.

Due to the same phenomena described above, the returns from simple targets correlate to a higher level and more steadily than do the returns from clutter. This characteristic may prove useful for target-to-clutter enhancement in a target classification mode by relative suppression of the within-cell clutter accompanying a target.

Since IPAR processing is carrier independent, the carrier waveform may be optimized for particular tasks. It is likely that different target classes and clutter environments will require different levels of spectrum spreading to optimally display and utilize the physical differences upon which characterizations may be based. Since IPAR may be implemented with any of a narrowband, frequency swept, or noise modulated carrier, it is compatible with the optimization of these parameters.



a) Pure odd-bounce reflection

b) Pure even-bounce reflection

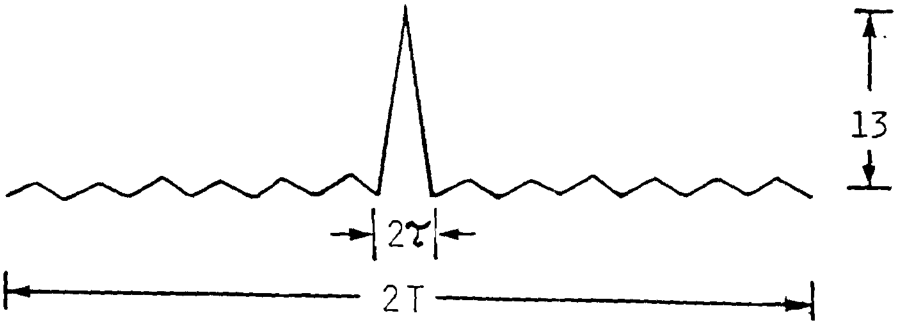


Figure 2. IPAR Compressed Waveforms

IPAR's high range resolution is achieved through pulse compression processing. Thus it may be utilized for single-pass, parallel processing for both the discrimination and classification tasks of a target identification radar. Furthermore, range profiles generated by IPAR include polarimetric information about the dominant scatterers in each range bin in the sense that predominantly even-bounce scatterers produce positive and predominantly odd-bounce scatterers produce negative correlations. This additional information may prove useful for target characterization and, thus, target classification.

Polarimetric pulse compression as implemented in IPAR is invariant to Doppler shift because both receive channels are frequency shifted on identical amount by a moving target, and the coding is contained in the relative phase between these channels. Thus, the need for Doppler or motion compensation processing when viewing stationary and moving targets from a moving platform is obviated. Furthermore, IPAR may be implemented coherently so that it would be compatible with a multimode stationary and moving target identification system that would take advantage of Doppler processing for the identification of moving targets. Additional properties of IPAR that may prove benefical in a tactical military environment include resistance to multipath interference and a potential for lowered probability of intercept (LPI) operation.(6)

IPAR IMPLEMENTATION

IPAR was designed and built as a demonstration and data collection system. The goals were first to verify that the IPAR technique works and second to collect data to quantify IPAR's strengths and weaknesses relative to classical radar techniques.

Figure 3 shows a block diagram of the IPAR system. Functionally, the system may be viewed as two subsystems: the RF subsystem and the IPAR digital processor subsystem. The basic system operation is described in the following paragraphs.

The IPAR system includes a two channel master oscillator power amplifier (MOPA) radar transmitter. The two channels are connected to the vertical and horizontal ports of a dual-polarized antenna feed. The 90° phase shifts of one channel with respect to the other that produce the RC and LC transmit polarizations are generated at a low power level due to the speed limitations of high powered phase shifters. Thus, the system utilizes two high power amplifier channels as well.

The mixer (i.e., phase shifter) that controls the phasing of the H and V channels is capable of modulation rates up to 2 GHz. It has the characteristic that a high level signal (i.e., +1) at the IF port causes zero phase shift from the input to output, whereas a low level signal (i.e., -1) causes a phase shift of 180°. With a permanent 90° phase shift added to the 0°, 180° phase shift combinations, the result is -90° or 90° relative phase between the H and V components of the transmitted wave.

Two RF sources in addition to a narrowband 9.35 GHz carrier have been incorporated in the current system. An HP 624C signal generator may be used to impress a 60 Hz sinusoidal frequency spread of 30 to 40 MHz on the transmit pulse to provide a frequency agile mode of operation. In addition, a noise source with a spectral width of 140 MHz may also be used to modulate the RF carrier.(7)

As in transmission, two receive channels are employed. The signals in the two receive channels are translated down in frequency, without changing the relative phase between the two signals, and they are applied to a phase detector. The polarity and

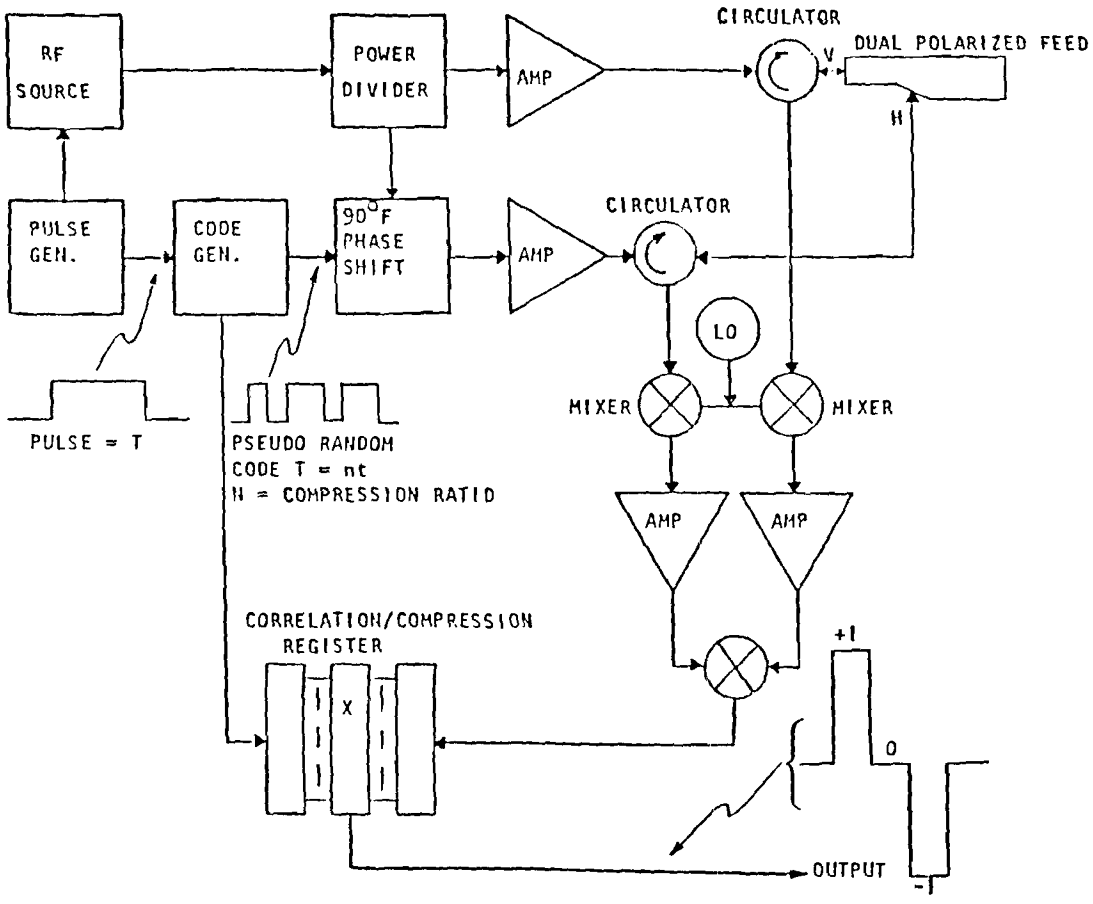


Figure 3. IPAR Block Diagram

amplitude of the resultant bipolar video signal is similar to the coding impressed on the transmission (and stored in the correlator), provided that the return is of high fidelity and not noise-like. The high speed correlator samples the returning signal at a maximum rate of 100 MHz. Once the video signal is quantized and input to the correlator, the remainder of the operation proceeds in a manner similar to conventional, biphase-modulated pulse compression techniques.

The IPAR digital processor was designed to provide a flexible, complete, and self-contained radar signal correlation processor. It may be operated in a stand-alone mode, under front panel control, or connected to and controlled by a minicomputer or microcomputer. The processor is capable of correlating received signals against the stored reference code in real time at rates up to 100 MHz. The operator has freedom to independently determine the transmitted code, subpulse width, PRF, and range gate setting.

Figure 4 presents a simplified block diagram of the digital processor and its interfaces to the external world. The slower TTL processor (6 MHz) may be considered the host or control portion of the unit. It interfaces with the front panel and the control computer (when provided), and it determines the operational mode of the processor. The high speed ECL processor interfaces with the radar by generating the transmit and coding signals, receiving and quantizing the video from the radar, performing correlation processing in real time, and storing data from the range gate position. A list of basic capabilities is presented in Table 1.

TABLE 1. IPAR DIGITAL PROCESSOR CAPABILITIES

Subpulse Length:	10 - 160 ns (100 - 6.25 MHz)
PRF:	500 - 8,000 pps plus manual
Range Gate:	0 - 15 km, 0.75 m maximum accuracy
Codes:	Any binary code up to 32 bits long
Correlation Processor:	True correlation for 32 bit code length Pseudo-correlation for 1 - 31 bit code lengths
Digital Threshold:	Adjustable
Self Test/Calibration:	Built-in

CURRENT ACTIVITIES AND FUTURE DIRECTIONS

The current IPAR system has been field tested. During these field tests many of its theoretically-predicted properties were qualitatively verified.(3, 5) Among these properties were pulse compression on narrowband, frequency swept, and wideband noise RF carriers, and the suppression of tree clutter returns with respect to simple targets when the frequency swept and noise source carriers were employed. The IPAR system has subsequently been utilized to collect reflectivity data on simple targets, multipath

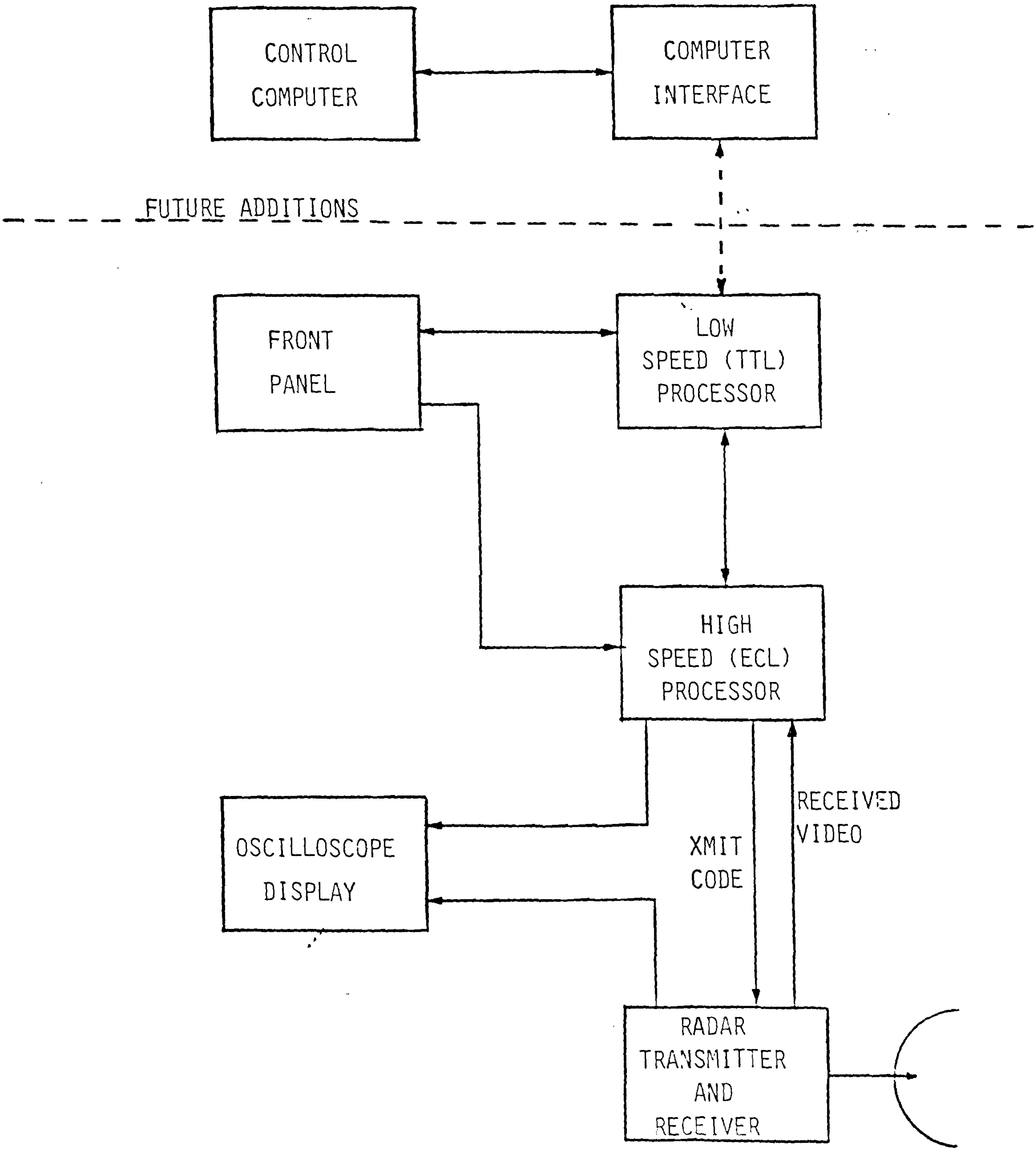


FIGURE 4. IPAR DIGITAL PROCESSOR BLOCK DIAGRAM

over the sea, and sea clutter. These data will soon be reduced and analyzed on the current program to provide proof-of-concept and to quantify IPAR's performance under various scenarios.

In addition, an advanced IPAR system is being designed and built which includes several significant advances over the current system:

(1) embedded microprocessor controller;
(2) 6 bit A/D conversion of the phase video at 100 MHz;
(3) 64 bit code length and true correlation for all codes;
(4) signal integration capability;
(5) Pseudo real time processing of 256 range bins;
(6) compatibility with 500 MHz operation.

The embedded microprocessor controller will allow pulse-to-pulse code agility and an automatic digital tape recording facility. By employing 6 bit A/D converters, small scatterers separated by more than one subpulse length from a larger scatterer will be resolved from the larger reflector, which will enable true target polarization profiling. The code length will be expanded to 64 bits, and a true correlation will be formed for all possible codes including those with embedded zeros. The extended code length and correlation capability is obtained at the expense of real time operation. Instead, 256 range cells will be sampled and processed during each pulse repetition interval.

The IPAR system has evolved from a concept to a demonstrated working radar system. When the full potential of the IPAR concept has been investigated, it may prove to be a significant advance in radar technology.

REFERENCES

1. M. Skolnik, Introduction to Radar Systems, McGraw-Hill, Inc., New York, New York, 1980.

2. F. Nathanson, Radar Design and Principles, McGraw-Hill, New York, New York, 1969.

3. M. N. Cohen, E. S. Sjoberg, E. E. Martin, "Intrapulse Polarization Agile Report Development Program," SCEEE Contract No. SCEEE/81-2, Prime Contract No. NAVSEA N00024-78-C-5338, GIT Final Report, December, 1981.

4. J. D. Echard and N. F. Ezquerra, "Classification of Military Targets by Radar (U)," 25th Annual Tri-Service Radar Symposium, September 1979, paper CONFIDENTIAL, proceedings, SECRET.

5. M. N. Cohen and E. S. Sjoberg, "Intrapulse Polarization Agile Radar," RADAR '82, London, England, October, 1982.

6. M. N. Cohen, E. E. Martin, E. S. Sjoberg, "The Intrapulse Polarization Agile Radar," (to appear) MSAT '83, Washington, D. C., March 1983.

7. B. C. Appling, E. S. Sjoberg, E. E. Martin, "Intrapulse Polarization Agile Radar," SCEEE Contract No. N00024-78-C-5338, SCEEE-NAVSEA/79-2, GIT Final Report, July, 1981.

SUMMARY

"IPAR AS A TARGET IDENTIFICATION RADAR"

M. N. Cohen and E. S. Sjoberg

Georgia Institute of Technology
Engineering Experiment Station
Atlanta, Georgia 30332

The Intrapulse Polarization Agile Radar (IPAR) system developed by the Georgia Institute of Technology Engineering Experiment Station in conjunction with the United States Navy Sea Systems Command (NAVSEA) employs a new and unique form of pulse compression. Rather than encoding the RF carrier with either phase or frequency modulation, IPAR utilizes polarization modulation to imprint the pulse compression coding on the carrier. By compressing on the relative phase between the received horizontal and vertical polarization components of the radar return, IPAR achieves pulse compression that is Doppler invariant. In addition, this compression technique inherently provides target-to-clutter ratio improvement that is enhanced by its compatibility with either a narrowband, frequency agile, or a wideband noise carrier.

The IPAR system has been implemented in an X-band radar designed and constructed at Georgia Tech. Traveling wave tubes amplify the independent horizontal and vertical components of the RF energy which is then transmitted as right or left circular polarization. The heart of the IPAR system is a high speed digital processor implemented in TTL and ECL technology. Binary polarization codes up to 32 bits in length with bit rates up to 100 MHz may be generated and processed in real time to generate signatures with a five foot range resolution. An advanced design exists that will be compatible with one foot range resolution.

After detection, the next step in any noncooperative target identification process is discrimination: the separation of potential target returns from clutter returns. Since the processing gain of the IPAR technique is inversely related to the polarization complexity of the set of scatterers in a range bin, discrimination is enhanced for many types of targets and clutter.

The final stage in target identification is recognition: the assignment of a potential target to one of a number of prespecified target classes based on the observable features of the target. The IPAR system provides range profiles that are signed according to the odd/even-bounce characteristics of the reflector. The signs of the returns in a profile may prove to be valuable features for the recognition process.

Operationally, IPAR need not be coherent and its pulse compression processing is Doppler invariant. These characteristis allow for a relatively inexpensive system that could be used for the identification of both fixed and moving targets. Furthermore, the capability of independent polarization and frequency diversity give IPAR unique potential in the area of lowered probability of incercept (LPI) operation.

A-3366

ENGINEERING EXPERIMENT STATION
Georgia Institute of Technology
A Unit of the University System of Georgia
Atlanta, Georgia 30332

February 2, 1983

Naval Sea Systems Command
Code 62R13
Washington, D. C. 20362

Attention: Mr. Charles Jedrey

Subject: Monthly Contract Technical Status Report No. ~~3~~ 4, "Phase 2 - IPAR Investigations," Contract No. N00024-82-K-5360, Georgia Tech Project A-3366, covering the period from January 1, 1983 through January 31, 1983

Dear Sir:

This status report summarizes program activities performed under the subject contract for the period January 1, 1983 through January 31, 1983.

TECHNICAL ACTIVITIES

After the Panama City field operation, Georgia Tech reestablished the local test site at Dobbins Air Force Base and began the completion of the data collection program. At this site the (1) resolution, (2) complex scatterer, (3) tree/grass clutter, and (4) simple targets-in-clutter experiments will be conducted. With these measurements the collection portion of the program will be complete.

At this point, the resolution experiments have been completed and the complex scatterer experiments are near completion. The program is somewhat behind schedule but will be completed within budget and schedule (assuming the no-cost extension discussed).

The first look at a simple, extended target (two trihedrals spaced by three resolution cells in range) during the resolution experiments did not precisely fit with what was expected. These results could be explained away phenomenologically, but further analysis suggests a new interpretation. In IPAR (as in any other form of pulse compression) compressing on phase-only information leads to non-linearities in range when viewing extended targets. The status of the current IPAR program as well as all ancillary efforts, in light of these new findings, may be summarized as follows:

Pulse compression on polarization modulation using phase-only information results in a non-linear "mixing" of range information for extended targets. This effect is independent of the complexity of the reflectors in the individual range resolution cells that comprise the extended target. If phase-only information were used for any other pulse compression process (phase or frequency coding), the net effect would be the same; i.e., the process would be non-linear in range for extended targets.

Adding (or, actually, retaining) amplitude information dispenses with these nonlinearities in IPAR (again, just as in any other form of pulse compression). Although the current IPAR configuration, which is being utilized for this program's data collection operation, processes only phase to generate the real-time video correlation function and the recorded digital data, both amplitude and phase data are recorded on FM tape. These FM data will be available for analysis during the proposed add-on program.

The efforts toward reduction and analysis of the digital data have begun with Mr. Michael Shannon taking charge of establishing the data base and analysis software on Tech's SEL computer. The data are being stored on disk, and an interactive analysis system is being developed. The result will be an efficient, flexible procedure for reducing and analyzing the data. Mr. Michael Baden and Mr. Brett Freemon have direct responsibility for software development and data handling, respectively.

FUTURE WORK

The data collection will be completed during the third week of February. Concommitantly, data analysis will continue through the end of February. If necessary, the first week in March will be utilized for continuing the analyses.

Dr. Marvin N. Cohen will present "IPAR As A Target Identification Radar" at CISC'83 on February 9th, and he and Mr. Ben Perry will begin preparation of the program's final report during the final week of February.

Respectfully submitted,

Marvin N. Cohen, Ph.D.
Project Director

APPROVED:

Robert N. Trebits, Ph.D.
Chief, Analysis Division

A-336

ENGINEERING EXPERIMENT STATION
Georgia Institute of Technology
A Unit of the University System of Georgia
Atlanta, Georgia 30332

March 1, 1983

Naval Sea Systems Command
Code 62R13
Washington, D. C. 20362

Attention: Mr. Charles Jedrey

Subject: Monthly Contract Technical Status Report No. ~~4~~ 5, "Phase 2 - IPAR Investigations," Contract No. N00024-82-K-5360, Georgia Tech Project A-3366, covering the period from February 1, 1983 through February 28, 1983

Dear Sir:

This status report summarizes program activities performed under the subject contract for the period February 1, 1983 through February 28, 1983.

TECHNICAL ACTIVITIES

The collection program was fully completed as of February 24th, with the recording of signatures from a pickup truck as a target of opportunity. We had hoped to record data on a U.S. Army armored personnel carrier, but the only one available could not be made to run.

The data assimilation and analysis software development tasks are 90% complete, each well into the testing and integration stages. Analysis of the recorded digital data will begin the second week in March and continue through April 1st.

The presentation of "IPAR as a Target Identification Radar" at CISC '83 seemed very well received. Many attendees showed significant interest, and many proferred encouragement for continuation of the work.

FUTURE WORK

Data analysis will begin the second week of March and continue through the end of the month. Final report preparation will begin the last week in March.

The paper "The Intrapulse Polarization Agile Radar" will be presented by Dr. Marvin N. Cohen at MSAT '83 in Washington, D. C. on Thursday, March 10th. The thrust of the talk will be IPAR's potential in the area of lowered probability of intercept (LPI) operation.

Respectfully submitted,

Marvin N. Cohen, Ph.D.
Project Director

APPROVED:

Robert N. Trebits, Ph.D.
Chief, Analysis Division

ENGINEERING EXPERIMENT STATION
Georgia Institute of Technology
A Unit of the University System of Georgia
Atlanta, Georgia 30332

April 5, 1983

Naval Sea Systems Command
Code 62R13
Washington, D. C. 20362

Attention: Mr. Charles Jedrey

Subject: Monthly Contract Technical Status Report No. ~~5~~ 6, "Phase 2 - IPAR Investigations," Contract No. N00024-82-K-5360, Georgia Tech Project A-3366, covering the period from March 1, 1983 through March 31, 1983

Dear Sir:

This status report summarizes program activities performed under the subject contract for the period March 1, 1983 through March 31, 1983.

The technical efforts on this program were completed and concluded. Four hundred and eighty (480) runs of digital data were loaded to disk and processed through an analysis program. Amplitude documentation for each of these runs was generated from the accompanying FM data. The result is a full set of documented results for (1) the simple and complex scatterer analyses, (2) multipath-over-bay analyses, and (3) targets in non-extended clutter analyses. Reduction of these analyses results and interpretation of the same will be undertaken as the first tasks on the proposed follow-on program.

Dr. Cohen presented "The Intrapulse Polarization Agile Radar" at the Microwave Systems and Technology (MSAT '83) conference in Washington, D.C. The paper concentrated on the potential for LPI operation of the fielded IPAR system. The talk was well received and generated requests for reprints and/or past final reports from the following individuals:

King Lear, ESL, Inc. (TRW)
William Bridge, MITRE Corporation
R. J. Keeler - National Center for Atmospheric Research
Harry Urkowitz, RCA Corporation
Bill Feiden, Hughes Aircraft Company

With the approval of our Technical Monitor, Mr. Charles Jedrey, we will forward reprints of the MSAT talk and/or the last final technical report to each of these individuals.

Mr. Charles Jedrey and Mr. Irv Olin were presented an extensive project review on March 31st at Georgia Tech by the major contributors (Dr. Marvin N. Cohen, Mr. Ben Perry, Mr. Michael Shannon) to the technical efforts on this program.

Respectfully submitted,

Marvin N. Cohen, Ph.D.
Project Director

APPROVED:

Robert N. Trebits, Ph.D.
Chief, Analysis Division

[illegible]

Final Technical Report
EES/GIT Project A-3366

"IPAR INVESTIGATIONS – PHASE II"

By

Marvin N. Cohen, Benjamin Perry,
Michael Shannon, Richard V. Folea, and John M. Baden

Prepared for

DEPARTMENT OF THE NAVY
Sea Systems Command (NAVSEA)
Washington, D.C. 20362

Under

NAVSEA Contract No. N00024-82-K-5360

May 1983

GEORGIA INSTITUTE OF TECHNOLOGY
A Unit of the University System of Georgia
Engineering Experiment Station
Atlanta, Georgia 30332

1983

1885

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (*When Data Entered*)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (*and Subtitle*) IPAR INVESTIGATIONS - PHASE II		5. TYPE OF REPORT & PERIOD COVERED Final Technical Report 6. PERFORMING ORG. REPORT NUMBER GIT/EES A-3366 FTR
7. AUTHOR(*s*) Marvin N. Cohen, Benjamin Perry, Michael Shannon, Richard V. Folea, and John M. Baden		8. CONTRACT OR GRANT NUMBER(*s*) N00024-82-K-5360
9. PERFORMING ORGANIZATION NAME AND ADDRESS Georgia Institute of Technology Engineering Experiment Station Atlanta, Georgia 30332		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS United States Navy Sea Systems Command (NAVSEA) Washington, D.C. 20362		12. REPORT DATE May 1983 13. NUMBER OF PAGES 138 + i thru iv
14. MONITORING AGENCY NAME & ADDRESS(*if different from Controlling Office*)		15. SECURITY CLASS. (*of this report*) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (*of this Report*)

17. DISTRIBUTION STATEMENT (*of the abstract entered in Block 20, if different from Report*)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (*Continue on reverse side if necessary and identify by block number*)

Clutter Data
Clutter Suppression
Multipath Data
Polarimetric Data
Polarimetric Processing
Polarization
Pulse Compression
Radar
Target Enhancement

20. ABSTRACT (*Continue on reverse side if necessary and identify by block number*)

The IPAR system has been utilized to collect target, clutter, and multipath-interference radar backscatter data in various environments. The target data include backscatter from assorted trihedrals and dihedrals taken separately and in various combinations and from complex, extended targets. Sea, bay, grass, and tree clutter backscatter data were also collected, as well as multipath-interference data over sea and relatively calm bay waters.

(continued)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (*When Data Entered*)

20. Abstract (continued)

During the collection program, the capabilities and characteristics of the IPAR system were fully exploited in that each of the three system rf carriers (narrowband, frequency agile, and noise), various range resolution cell sizes, and various code lengths were implemented for many of the experiments conducted. The resulting data base should, therefore, prove adequate for quantification and understanding of many aspects of IPAR processing.

The resulting data base consists of radar backscatter data from over 1,300 separate IPAR system runs. Approximately two-thirds of the digitally-recorded data have been downloaded from 22 digital tapes generated in the field to a computer disk file, which allows for random access to the data and, therefore, efficient data analysis. An analysis routine was developed and applied to the data to yield statistical measures of the pulse-compression performance of the IPAR system for each of the runs recorded on disk.

TABLE OF CONTENTS

Section	Title	Page
1	INTRODUCTION	1
1.1	Program Overview	1
1.2	Program History	1
1.3	Program Objectives	3
1.4	Program Description	3
2	TEST SITES	7
2.1	Naval Coastal Systems Center Site	7
2.2	Panama City Beach Site	7
2.3	Dobbins Air Force Base Site	7
2.4	Georgia Tech Site	11
3	SYSTEM DESCRIPTIONS	15
3.1	The IPAR Radar System	15
3.1.1	Transmitter	15
3.1.2	Receiver	21
3.1.3	IPAR Processor	21
3.2	The Data Collection System	24
3.3	Ancillary Equipment	29
4	EXPERIMENTAL PROCEDURES	37
4.1	Radar Calibration	37
4.1.1	Power Measurements	37
4.1.1.1	RF Source Power Measurements	37
4.1.1.2	Transmitter Power Measurements	38
4.1.2	Transmit Signal Phase Adjustment	39
4.1.3	Receiver Amplitude Calibration	40
4.1.4	Receiver Phase Verification	41
4.1.4.1	Receiver Phase Shifter Adjustment	41
4.1.4.2	Phase Detector Verification	42
4.1.5	Calibration Verification	42
4.2	Data Collection	43
4.2.1	Mode A: Digital and Analog	44
4.2.2	Mode B: Analog Only	46
5	THE EXPERIMENTS	47
5.1	Compression on Simple Targets	47
5.2	Multipath Interference over Bay	47
5.3	Multipath Interference over Sea	48
5.4	Bay and Sea Clutter	48
5.5	Simple Targets in Bay and Sea Clutter	49
5.6	Resolution	49
5.7	Complex Targets	50
5.8	Tree and Grass Clutter	50

TABLE OF CONTENTS
(Continued)

Section	Title	Page
5.9	Targets in Tree Clutter	51
5.10	Targets of Opportunity	51
6	DATA SUMMARY	53
6.1	Summary	53
6.2	Breakdown According to Analysis Objectives	53
7	DATA REDUCTION AND ANALYSIS PROCEDURES	63
7.1	Digital Tape-to-Disk Procedure	63
7.2	Digital Master Analysis Program	64
7.3	FM Signal-to-Noise Documentation	67
8	PROGRAM RESULTS	71
8.1	Data Collection, Reduction, and Analysis	71
8.2	Modeling and Theoretical Analysis	71
8.3	Delineation of Practical Applications	76
9	SUMMARY AND RECOMMENDATIONS	79
9.1	Summary	79
9.2	Recommendations for Continued Research	79
	REFERENCES	81
Appendix A	TAPE-TO-DISK TRANSFER ROUTINE	83
Appendix B	DATA ANALYSIS ROUTINE	115

LIST OF FIGURES

Figure	Title	Page
1	St. Andrews Bay field site geometry	8
2	Panama City Beach field site geometry	9
3	Panama City field site locations	10
4	Dobbins Air Force Base field site geometry	12
5	Georgia Tech Research Facility field site geometry	13
6	Transmitter block diagram	16
7	Receiver block diagram	22
8	Phase detection network	23
9	Phase detector response function	25
10	IPAR digital correlator output	26
11	IPAR processor front panel	28
12	Recording and monitoring equipment block diagram	30
13	Multipath pole in St. Andrews Bay	31
14	Trihedral reflector geometry and backscatter pattern	32
15	Dihedral reflector geometry and backscatter pattern	35
16	FM data reduction system block diagram	68
17	Examples of receiver response function and target amplitude distribution	69
18	Pulse-compressed output	73
19	Response from an amplitude-dependent pulse compression system	75
20	Response from an amplitude-independent pulse compression system	77

LIST OF TABLES

Table	Title	Page
1	Field Site and Data Descriptors	14
2	Radar System Parameters	17
3	Radar Transmitter Sources	18
4	Traveling Wave Tube Amplifier Characteristics	20
5	IPAR Processor Operational Parameters	27
6	Trihedral and Dihedral Corner Reflector Characteristics	33
7	Data Collection Variables	45
8	Summary of Data Collected	54
9	Data Summary According to Analysis Objectives	59
10	Complex Targets Experiments	62
11	Sample Digital Analysis Output	65

SECTION 1
INTRODUCTION

1.1 PROGRAM OVERVIEW

The Intrapulse Polarization Agile Radar (IPAR) has been successfully utilized to collect data, some of which have been reduced and partially analyzed. These data will quantify IPAR's compression performance on simple scatterers, artificially-created complex targets, targets in clutter, and targets in multipath interference. Additional data were collected that will allow analysis of IPAR's compression performance on extended targets, targets in a sea multipath interference environment, sea clutter, tree clutter, and targets in sea and tree clutter. The data also provide amplitude and relative phase documentation for all of these experiments so that the reflected IPAR waveform may be studied in each of these scenarios. These data have been collected, reduced, and analyzed under contract with the Naval Sea Systems Command (NAVSEA) Contract N00024-82-K-5360 for which this document is the final report.

Section 1.2 of this report provides a historical overview of the research that led to this phase of the IPAR program. Section 1.3 provides a summary of the objectives that had been set forth for this phase of the research in the Statement of Work, and Section 1.4 provides a concise summary of the activities which were conducted under this contract to meet those objectives.

1.2 PROGRAM HISTORY

Under contract to the Air force in 1973, the Georgia Institute of Technology/Engineering Experiment Station (GIT/EES) was tasked to investigate stationary target discrimination techniques. During this project, the relative usefulness of employing frequency agility and polarization agility, independently and together, to stimulate clutter decorrelation was tested.(1,2,3) Two processors were constructed to discriminate between clutter and hard targets: Dynamic Threshold Gating (DTG) and Correlation Coefficient Discrimination (CCD) processors. Some success was achieved, but it was difficult to state the degree of improvement in quantitative terms. Of greater significance were two observations made by project personnel during the final phase of that program. The processors were designed to work primarily with amplitude discriminants, but discriminants (i.e., differences between clutter and targets) were also observed in the relative phase between the two orthogonal receive polarizations.

Subsequent to this program, several additional field experiments and demonstrations were conducted. One of the more notable demonstrations, in terms of technique exposure, was conducted at the Army Missile Command (MICOM) in Huntsville, Alabama. Conventional amplitude, phase, and threshold gated integrated digital video were simultaneously displayed on an A-scope. Targets included corner reflectors and tanks. A tank was moved around during the demonstration so that the observers could be certain of what they were viewing. Polarization, frequency agility, threshold setting, and integration time were all varied during the demonstration. Even when the relative clutter-to-target amplitude was approximately 20 dB, the target could still be detected and displayed without clutter false alarms.

In 1978, an analysis of Pseudo-Coherent Detection (PCD) techniques was conducted. This analysis is documented by report DELCS-TR-76-0961-F, entitled "Stationary Target Detection and Classification Studies," dated April, 1979.(4) The analysis did not predict the performance which had already been observed during the demonstrations mentioned above.

The first phase of the IPAR program itself was initiated by the Georgia Institute of Technology in March 1979 under the auspices of the Southeastern Center for Electrical Engineering Education (SCEE), subcontract SCEE-NAVSEA/79-2, Contract Number N00024-78-C-5338, with Mr. Charles Jedrey of NAVSEA acting as Technical Monitor. During this program, a demonstration state-of-the-art IPAR radar/processor was designed, built, and demonstrated. In addition, a detailed analysis of PCD was conducted to (1) better understand this technique which lies at the heart of the IPAR process, and (2) to attempt to clarify the apparent lack of consistency (as noted above) between the predicted and observed effects of PCD. The results of the program were documented in the project final report dated July 1981.(5) In summary, the IPAR system was designed, assembled, and successfully demonstrated. Essentially perfect intrapulse polarization coding and compression of the radar waveform was achieved at real-time rates of up to 50 MHz, and somewhat degraded compression was achieved at rates up to 100 MHz. Thus, the concept of compression on polarization modulation was successfully reduced to hardware implementation. The analyses of PCD were completed, and they successfully closed the gap between theory and experience, thus establishing a theoretical basis for future applications of IPAR.

The IPAR program continued during 1981 under the auspices of SCEEE, subcontract number SCEEE/81-2, Contract Number N00024-78-C-5338, again with Mr. Charles

Jedrey of NAVSEA acting as Technical Monitor. During this phase of the program, the IPAR radar/processor was outfitted as a data collection system incorporating a high speed digital tape recorder, an FM tape recorder, two strip chart recorders, and various oscilloscopes for system monitoring. The entire system was incorporated in the Georgia Tech GT-1 van, tested, tuned, and demonstrated as a viable data collection system. A detailed list of objectives and a data collection plan to meet these objectives were generated as well, and these formed the basis for the current program.(6)

In early 1982 Georgia Tech contracted with the Office of Naval Research (ONR), Contract No. N00014-82-K-0441 with Mr. Max Yoder acting as Technical Monitor, to build an advanced IPAR system (A-IPAR) as a test bed for state-of-the-art VHSIC technology. The system is currently being developed at Georgia Tech and will be completed by July, 1985. The A-IPAR system will feature computer-controlled operation, improved pulse compression ratios, extended range swath capability, and real-time 100 MHz operation with a design capable of supporting 500 MHz components, which are to be added as they become available.

1.3 PROGRAM OBJECTIVES

The purpose of this program was to collect and begin analyzing IPAR data that would (1) firmly establish proof of concept for achieving polarization-coded compression; (2) allow quantification of the target-to-clutter advantages of an IPAR process over an unmodulated, simple pulse radar; (3) allow quantification of IPAR performance in the presence of multipath interference; and (4) document IPAR's response to targets of various complexities and configurations.

1.4 PROGRAM SUMMARY

The IPAR data collection system was deployed in the Georgia Tech GT-1 van at Dobbins Air Force Base at program inception. The system was set up and tested at this site. Data were taken on a $36m^2$ trihedral to provide data for proof of concept. Portions of the digital data generated during this experiment were reduced in the laboratory, thus providing reassurance that the system was performing according to specification and recording the resulting data accurately.

The first data collection site was the Naval Coastal Systems Command (NCSC) at Panama City Beach, Florida. Extensive multipath-interference-over-bay experiments were conducted at this bayside site by erecting a prefabricated multipath pole in the bay shallows. Data were collected over many elevation lobes of the resulting multipath interference pattern. These data included exhaustive probing of the multipath interference fields that resulted from illumination with the 9.5 GHz narrowband (NB) carrier, the frequency modulated (FA) source, and the wideband noise (NO) source. In addition, a reflector was placed outside the multipath interference region and extensive data were collected on IPAR's performance under various signal-to-noise conditions. These latter data will provide a basis for computing the actual processing gain achieved by IPAR's 32-to-1 pulse compression coding.

The second test site was NCSC's Tower II on the beach overlooking the Gulf of Mexico. From this vantage point the system was utilized to generate and collect data on sea clutter, targets of opportunity, and multipath interference-over-sea. Again, a full set of data were collected utilizing each of the system's three carriers.

Upon completion of the Panama City Beach experiments, the Dobbins Air Force test site was reestablished. The range resolution of the IPAR system was documented through a series of experiments involving one and two range bin separation of various scatterers. The "capture" problem, due to the coarse quantization of received relative phase angle and hard limiting of received amplitude in the current system, was demonstrated by collecting data on two trihedrals (one 1 m^2 large and the other 10 m^2 large) which were separated by two range bins.

Various combinations of dihedrals and trihedrals were then deployed within a single range bin to form complex, but understandable, target combinations. Reflectors were moved a fraction of a wavelength at a time to demonstrate the effects of various in-cell phase relationships, and complex configurations involving as many as four trihedrals and three dihedrals were deployed in order to simulate various degrees of target complexity. Configurations where the total radar cross section (RCS) of the dihedrals was as much as 10 dB and as little as 0 dB greater than the total RCS of the trihedrals were probed. Analysis of these results will provide a method for quantification of IPAR's performance as a function of target complexity.

The final set of experiments was conducted at the Georgia Tech Research Facility at Cobb County. Full sets of data were collected on a tree line of evergreen trees, a target (trihedral) embedded in the tree line, and the same target set directly in front of the tree line. These experiments were repeated with a line of evergreen bushes that represented non-extended clutter in the sense that the line was less than one range bin deep. The data collection program was completed with the recording of IPAR backscatter from a pickup truck.

In all, the data collection encompassed recording of approximately 20 seconds of IPAR data from each of over 1,300 separate measurement runs. These data were recorded and stored on 22 digital and 17 FM magnetic tapes.

Initial reduction of the digital data collected was accomplished by loading the data onto a disk file system on Georgia Tech's Systems Engineering Lab (SEL) computer. The file contains approximately 2,000 digitized IPAR returns for each of 480 runs. These data were extracted from 14 of the digital tapes generated in the field. They were chosen to permit analysis of IPAR's response to (1) simple targets, (2) nonextended clutter, (3) simple targets in nonextended clutter, (4) multipath interference (over bay waters), and (5) artificially constructed complex targets.

An analysis program was developed and implemented on the disk-stored data. The program computes various statistical and signal processing measures from each of the subject data runs. These measures are available for human analysis and conclusion-forming.

Other activities engaged in during this program included development of a basic computer simulation of the IPAR process and consideration of potential applications of the IPAR system. The simulation aided in the design of the "complex target" experiments that led to theoretical results which proved to be of importance in understanding our experimental results. The investigations into potential applications aided the researchers in choosing and designing the experiments for the data collection program.

(THIS PAGE INTENTIONALLY LEFT BLANK)

SECTION 2
SITE DESCRIPTIONS

During the IPAR data collection effort, four different field sites were utilized. Because the IPAR system was completely self-sufficient, no external power was required and the sites could be chosen for specific physical conditions.

2.1 ST. ANDREWS BAY SITE

The first site was located along the shore of St. Andrews Bay in Panama City, Florida at the Naval Coastal Systems Center (NCSC). This site was chosen for multipath interference measurements over salt water because of the calm sea conditions present in the bay. The standard geometry for the bay measurements is demonstrated in Figure 1. The antenna was mounted on top of the van approximately 4.6 meters above mean waver level. The target, either a trihedral or a dihedral corner reflector, was mounted on a 4 meter pole at a distance of 256 meters from the radar. The target was raised and lowered to probe the elevation multipath field. The sea state conditions in the bay varied from a completely smooth sea to a rough sea with approximately 2 foot waves.

2.2 PANAMA CITY BEACH SITE

For more realistic sea state conditions a second site on the Gulf of Mexico was utilized, at NCSC Shore Tower #2 located in Panama City Beach, Florida. Tower #2 is a reinforced concrete shelter protected by a chain link fence with telephone communications to NCSC. The geometry of the Gulf measurements is presented in Figure 2. The pole used in the previous measurements was mounted on a 4 m fiberglass outboard motorboat. The antenna was positioned 6.4 meters above the mean sea level. By running the boat in toward the radar from 4 km to 0.5 km the multipath field was probed in the horizontal dimension. The approximate wave height for the Gulf measurements varied from .3 m to .9 m. Figure 3 is a map of the Panama City area showing the locations of the two Florida field sites.

2.3 DOBBINS AIR FORCE BASE SITE

Measurements of simple targets over clutter-free terrain were performed at Dobbins Air Force base in Marietta, Georgia. This location afforded an unimpeded range

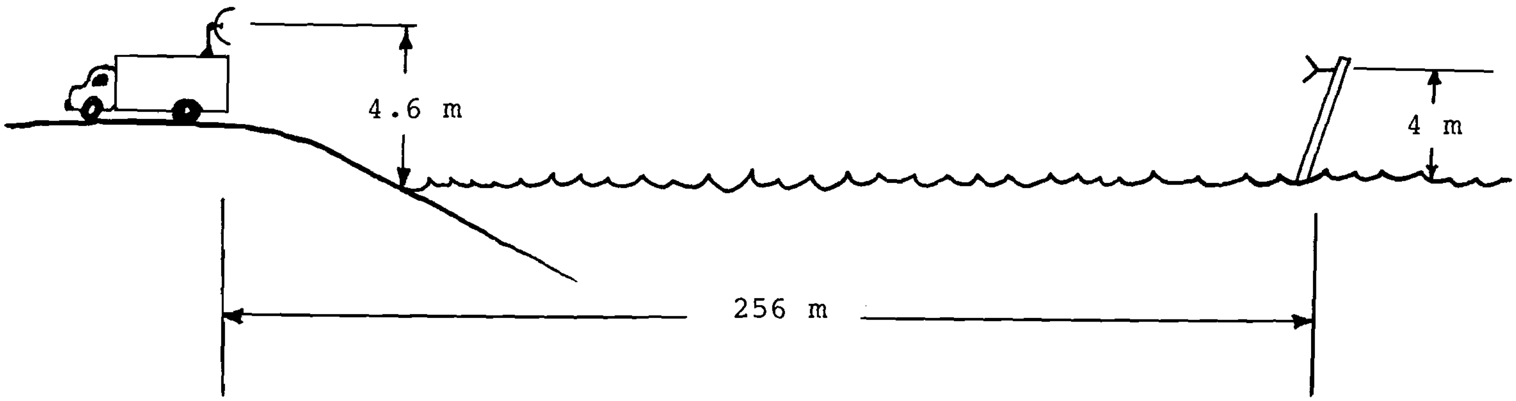


Figure 1. St. Andrews Bay field site geometry.

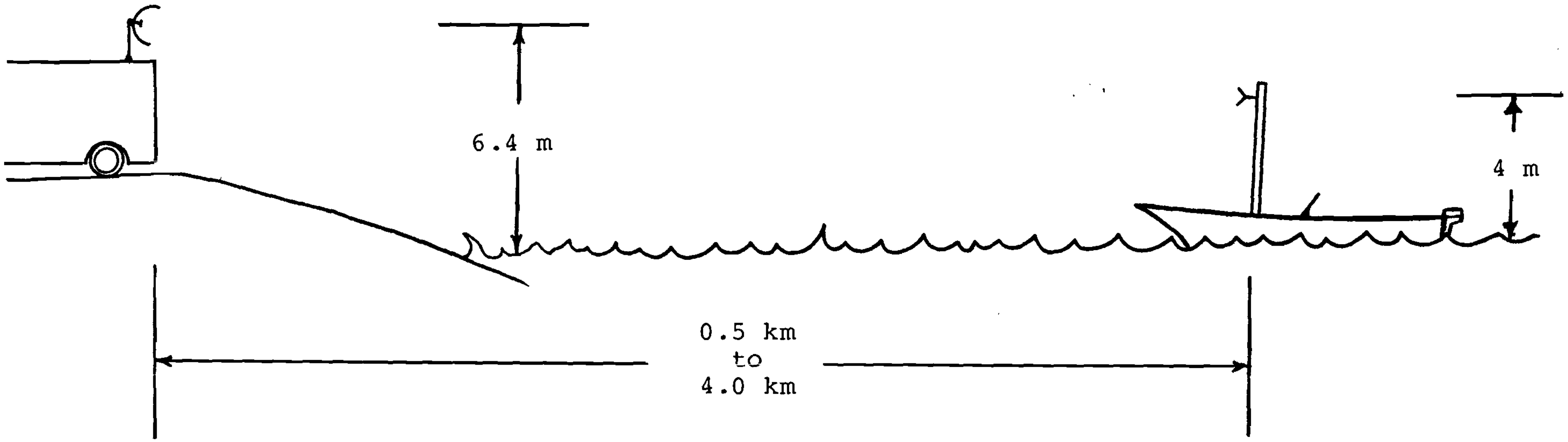


Figure 2. Panama City Beach field site geometry.

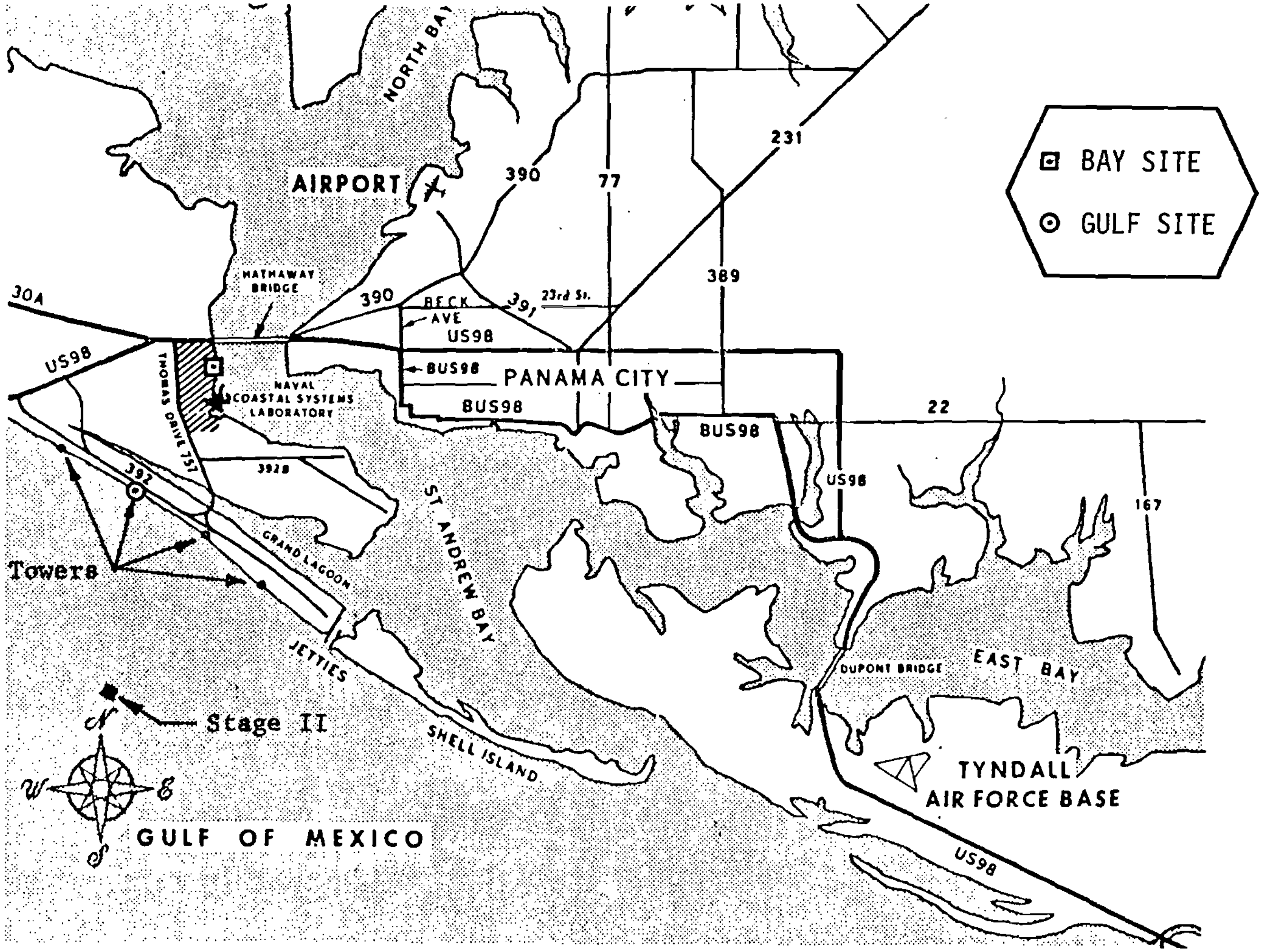


Figure 3. Panama City field site locations.

of 300 meters which was necessary for the long pulse length measurements. The geometry for these measurements is depicted in Figure 4. The targets were placed on the ground approximately 255 meters from the antenna which was elevated 6.7 meters above them.

2.4 GEORGIA TECH RESEARCH FACILITY/COBB COUNTY SITE

Finally, target-in-clutter data were collected at Georgia Tech Research Facility in Cobb County because of the convenience and the availability of extended and non-extended tree clutter. The geometry appears in Figure 5. The antenna was located 3.6 m above the ground at a range of 138 m from the target. The target was fixed to the multipath pole at a height of 3 meters. A summary of the IPAR field sites and the type of data collected at each site is presented in Table 1.

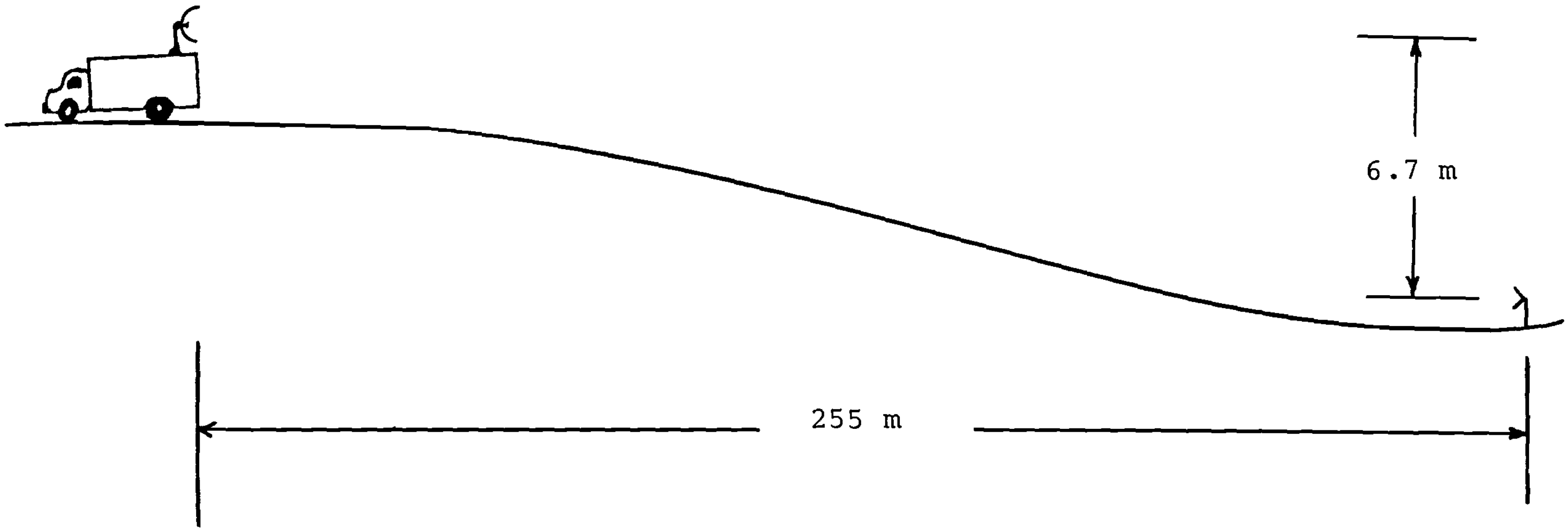


Figure 4. Dobbins Air Force Base field site geometry.

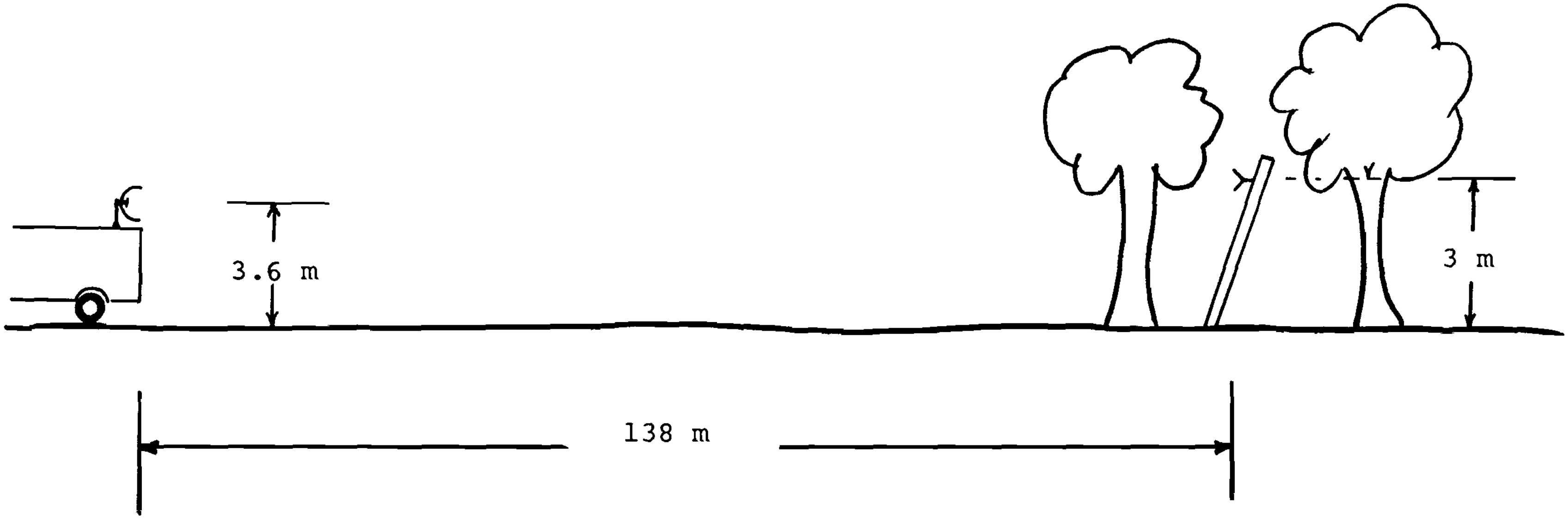


Figure 5. Georgia Tech Research Facility field site geometry.

TABLE 1. FIELD SITE AND DATA DESCRIPTIONS

1. Naval Coastal Systems Center - Bay Site
   - Vertical Multipath Probe

2. Panama City Beach - Gulf Site
   - Horizontal Multipath Probe
   - Sea Clutter
   - Target in Sea Clutter

3. Dobbins Air Force Base
   - Combinations of Simple Targets
     - Trihedrals and Dihedrals

4. Georgia Tech Research Facility - Cobb County
   - Simple Targets in Tree Clutter

SECTION 3
EQUIPMENT DESCRIPTION

The IPAR data collection system was a totally self-sufficient system housed in the mobile GT-1 radar van, Ford Model C-500. An Onan 15 kW generator was used to power the equipment located in the van. The steerable paraboloid antenna was mounted on top. The following equipment description is organized in terms of the radar system, the recording and monitoring equipment, and the ancillary support equipment.

3.1 RADAR SYSTEM DESCRIPTION

The IPAR radar system produces coded circularly polarized pulses by combining two linear polarizations in a dual mode coupler at the antenna feed. On receive, the two linear components are mixed in a phase detector, and the resultant phase angle is correlated with the original coded signal to determined the correlation properties of the target. This system implementation requires simultaneous transmission and reception of both horizontal and vertical polarization. Consequently, both the transmitter and the receiver have two complete channels. The radar system is described in more detail in the following sections. A list of the system parameters is presented in Table 2.

3.1.1 TRANSMITTER

The transmitter section of the IPAR radar consisted of a common oscillator source, power divider, two bi-phase modulators, a manual phase shifter, and two identical TWT amplifier chains.

A block diagram of the transmitter is shown in Figure 6. As an experimental variable, three different RF carriers were employed: narrow band, swept frequency, and wideband noise. For the first two modes the basic signal for the system was generated by a Hewlett Packard (HP) Model 624 X-band test set. The wideband noise was produced by a separate source. Table 3 lists the parameters for the three RF carriers including average power and bandwidth. The peak power of 1 kW listed in Table 2 was achieved by amplification of the 1 mW from the HP test set through the chain of TWT's. The peak power of the noise source was approximately 160 Watts.

The signal out of the HP test set was split in a 3 dB hybrid coupler and injected into the bi-phase modulators where the polarization code was created. The polarization

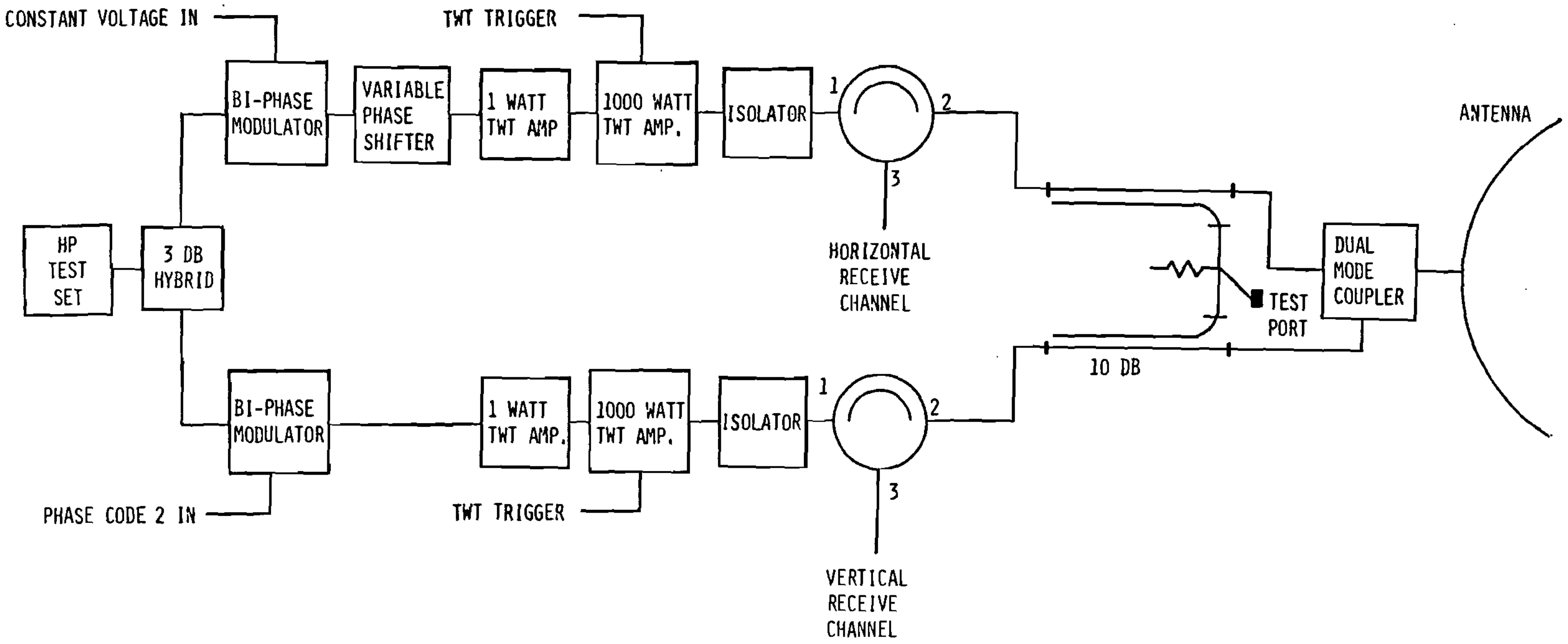


Figure 6. Transmitter block diagram.

TABLE 2. RADAR SYSTEM PARAMETERS

PARAMETERS	VALUE
Frequency	9.375 GHz
Peak Power	1 kW
Pulse Width	10 ns - 5.2 μs
Antenna Type	Steerable Parabolid
3 dB Beamwidths	1.8^o
Antenna Gain	38 dB
Maximum Sidelobe Level	-21 dB
Polarization	H & V Simultaneous Transmit and Receive
IF Center Frequency	300 MHz
IF Bandwidth	100 MHz
IF Response	Logarithmic
Noise Figure	10.3 dB
Dynamic Range	70 dB

TABLE 3. RADAR TRANSMITTER SOURCES

SOURCE	POWER	CENTER FREQUENCY	BANDWIDTH
Narrow Band	1 mW	9.45 GHz	-
Spread Spectrum	1 mW	9.45 GHz	< 40 MHz
Wideband Noise	.16 mW	9.39 GHz	140 MHz

modulation was achieved by shifting the phase of one split signal from a 90 degree leading phase angle to a 90 degree lagging phase angle and recombining these two split signals in a dual mode coupler located at the antenna feed horn. The required 180 degree phase shift was accomplished by supplying positive or negative currents, according to a predetermined code, to one of the bi-phase modulators. A constant current level was supplied to the other bi-phase modulator to generate a pulsed RF signal equal in length to the duration of the code. A phase shifter was included in the non-modulated channel to aid in the initial set-up of the radar and to provide the required 90 degree phase difference between the two components of the radiated signal at the dual mode coupler. Since the two components radiated from the dual mode coupler were spatially separated by 90 degrees, the resultant transmitted signal was right hand or left hand circularly polarized according to the code applied to the bi-phase modulator.

Each of the two amplifier chains was composed of a low power Traveling Wave Tube Amplifier (TWTA), which boosted the signal to a level of approximately 1W, and a high power pulsed TWTA which further increased the peak signal power to a level of 1 kW. The low power TWTAs are Model 495A amplifiers manufactured by Hewlett Packard, Inc., and were purchased as standard stock items. The high power TWTAs were built for Georgia Tech by the Electron Tube Division of Litton Industries. Their amplifiers consist of type L-5655-59 Traveling Wave Tubes modulated by a Model M-624 modulator assembly. Specifications for these amplifiers are given in Table 4.

A 1.2 meter diameter parabolic dish antenna was used in the IPAR radar. This antenna has a dual mode coupler with a square feed horn located at its focus. The antenna can be rotated about its polarization axis so that any linear polarization can be generated by feeding one port of the dual mode coupler. The antenna is normally oriented so that the polarization of the transmitted signal is either horizontal or vertical, depending on which port of the dual mode coupler is selected as the signal path. The beamwidth and gain of the antenna are 1.8 degrees and 38 dB, respectively.

If both ports of the dual mode coupler are fed simultaneously, the signal radiated by the antenna can be generally described as elliptically polarized, with either linear or circular polarizations being the limiting case. A rectangular waveguide port and video detector were provided at the center of the four foot dish to allow monitoring of the amplitude and phase of the transmitted waveform. This detector was also used to continuously monitor the transmit signal power for accurate calibration of the received signals.

TABLE 4. TRAVELING WAVE TUBE AMPLIFIER CHARACTERISTICS

Peak Power Output:	1,000 W
Frequency:	8 - 12.5 GHz
Gain:	35 dB minimum
Noise Figure:	40 dB
Duty Cycle:	2%
Pulse Width:	0.2 to 15 μs
Input Gate Amplitude:	+6 to +15 volts
Input Gate Width:	0.2 to 15 μs
Input Voltage:	120 VAC ±15%
Input Current:	8 amperes at 120 VAC

3.1.2 RECEIVER

The horizontal and vertical components of the reflected signal are received by the antenna. Two identical receivers were provided to amplify and detect these signals. Each receiver was equipped with a precision waveguide antennuator for calibration purposes, and one receiver channel included an adjustable phase shifter to compensate for slight differences in path length between the two receiver channels. The mixer-preamplifiers operated at an intermediate frequency (IF) of 300 MHz and had a 3 dB bandwidth of 100 MHz. The two model DM8-12/300 mixers were fed with a common local oscillator tuned to produce the 300 MHz IF frequency. The detected 300 MHz IF signals were amplified by two (model 1CLT-300-B) logarithmic amplifiers. These amplifiers had both a detected video output and a limited IF output. The receiver block diagram is shown in Figure 7.

The amplitude outputs from the logarithmic amplifiers were recorded on FM tape in order to document the radar cross section of the radar targets. The two limited IF outputs from the logarithmic amplifiers were mixed in a double balanced mixer to detect the relative phase difference between the horizontal and vertical components of the received signal. This phase difference was used in the IPAR correlation processor to determine the degree of correlation between the transmitted code and the received signal. The output of the correlator was recorded on digital tape.

The digital data were supplemented with analog data on FM tape to verify the correlator operation. For this mode of operation, phase information was not injected into the correlator, but was instead recorded on magnetic tape. A phase detection network replaced the double balanced mixer for this mode of operation.

The phase detection network, shown in Figure 8, employed two 3 dB hybrids and two DBM-200 double balanced mixers. The limited IF signals from the H and V channels were split in a pair of 3 dB hybrid couplers. One pair of H and V IF signals were combined in a mixer to produce the cosine of the relative phase angle. One channel of the other pair was delayed by 90°, then these two were also combined in a mixer to produce the sine of the relative phase angle. These sine and cosine signals were sampled and held at the PRF and recorded on FM tape along with the amplitude data.

3.1.3 IPAR PROCESSOR

The IPAR correlator and digital interface unit have been described in detail in two previous Final Technical Reports for Contract No. N00024-78-C-5338.(5,6) On transmit,

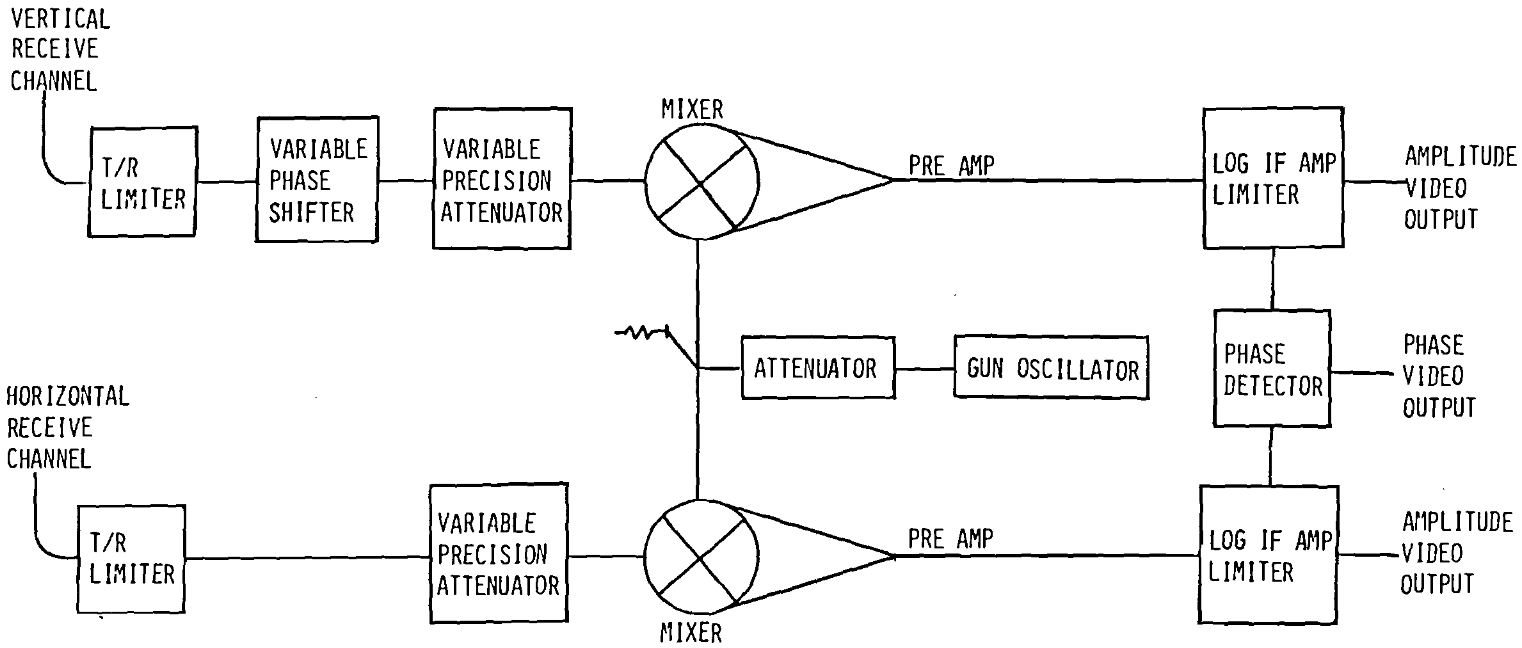


Figure 7. Receiver block diagram.

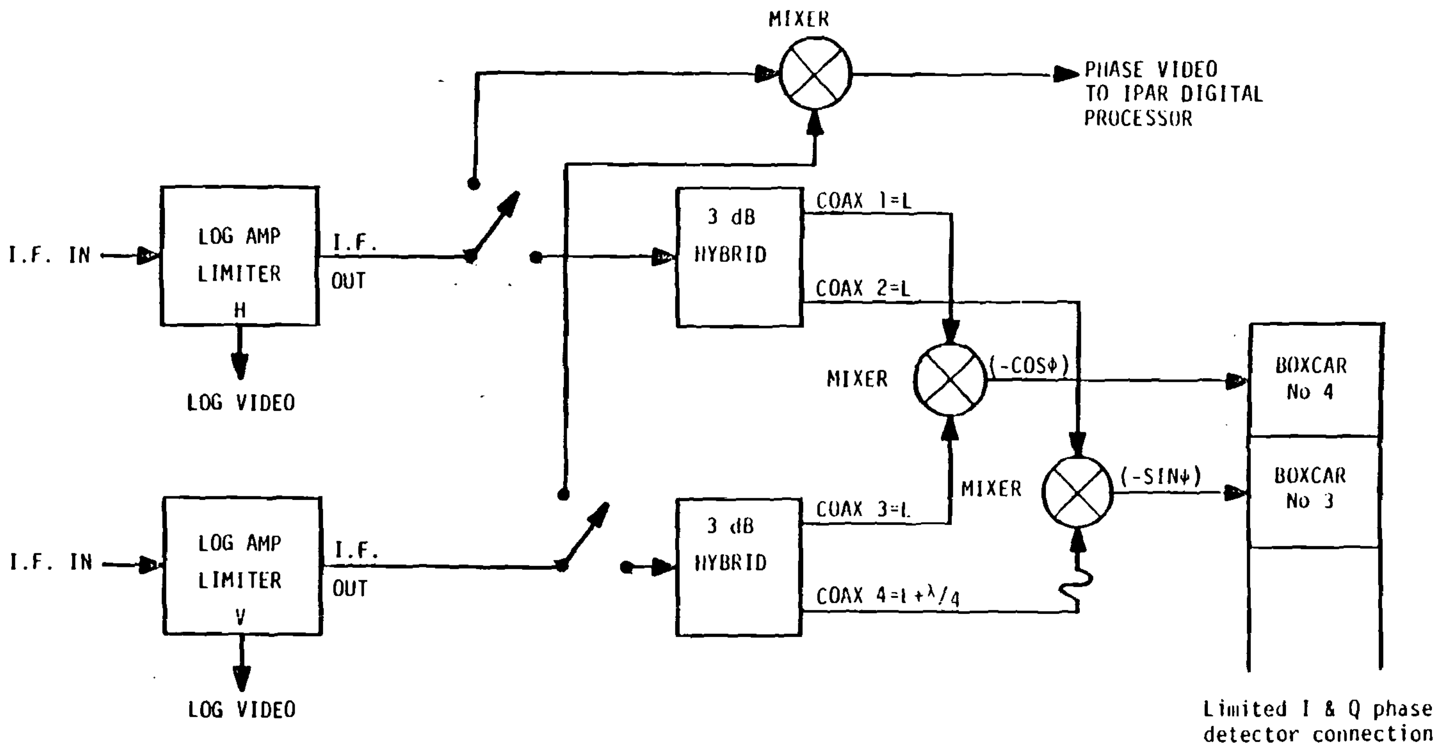


Figure 8. Phase detection network.

the IPAR processor controls the polarization code creation and triggers the transmitter modulators. On receive, the correlator quantizes the received phase information and produces a digitized received pulse code that is recorded on digital tape.

The input signal to the correlator comes from the double balanced mixer shown in the receiver block diagram, Figure 8; the signal consists of the relative phase angle between the received horizontal and vertical components. The response function of the phase detector (mixer) is shown in Figure 10. The thresholds that appear in the figure are set in the IPAR correlator and are used to digitize the phase information. If the H-V phase angle is close to -90°, the received signal is declared left circularly polarized and is assigned a value of -1. If the phase angle is near $+90^{\circ}$, the signal is right circularly polarized and assigned a +1. If the relative phase angle is between the two thresholds, the signal is assumed to be depolarized as is given a value of 0. An example of the resulting correlator output is shown in Figure 11. The transmitted code is a 13 bit Barker consisting of pure right and left circular subpulses (1 and -1). On receive, a portion of the pulse has been depolarized, resulting in the appearance of zeros. It is this received and detected pulse code that is written out to digital tape.

The IPAR processor controls the functioning of the radar system including the pulse code creation, the modulator triggering, and the clocking rate for the data samplers. The radar system parameters that were varied as part of the IPAR experiment were all controlled by the processor. A list of these capabilities is presented in Table 5. All of the parameters shown were employed as experimental variables and could be readily changed using switches on the front of the IPAR processor. This front panel is shown in Figure 12.

3.2 RECORDING AND MONITORING EQUIPMENT

Because the H-V phase angle information could not be recorded on FM tape and used in the IPAR correlator at the same time, two independent modes of data collection were employed. In the first, Mode A, the phase information was routed to the correlator, where the received pulse code was generated as described in the previous section. This code of 1's, 0's and -1's was recorded on digital tape on a pulse-by-pulse basis. Each recorded file of digital data was preceded by a header record containing the pertinent IPAR parameters such as code length, subpulse width, PRF, run number, and threshold setting.

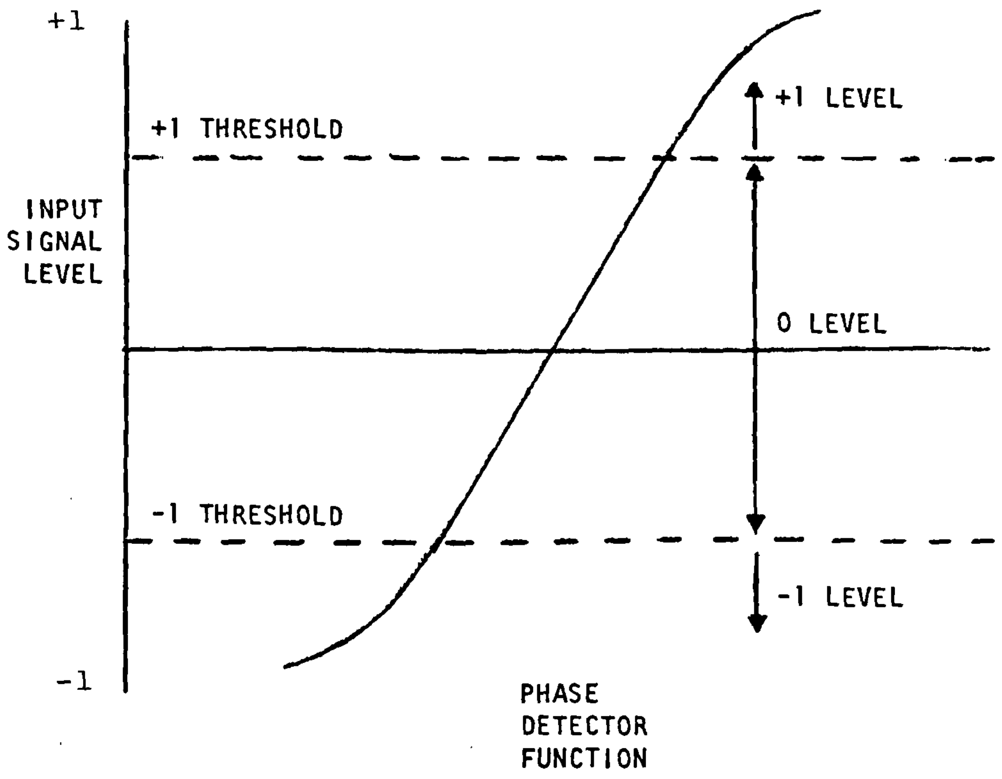


Figure 9. Phase detector response function.

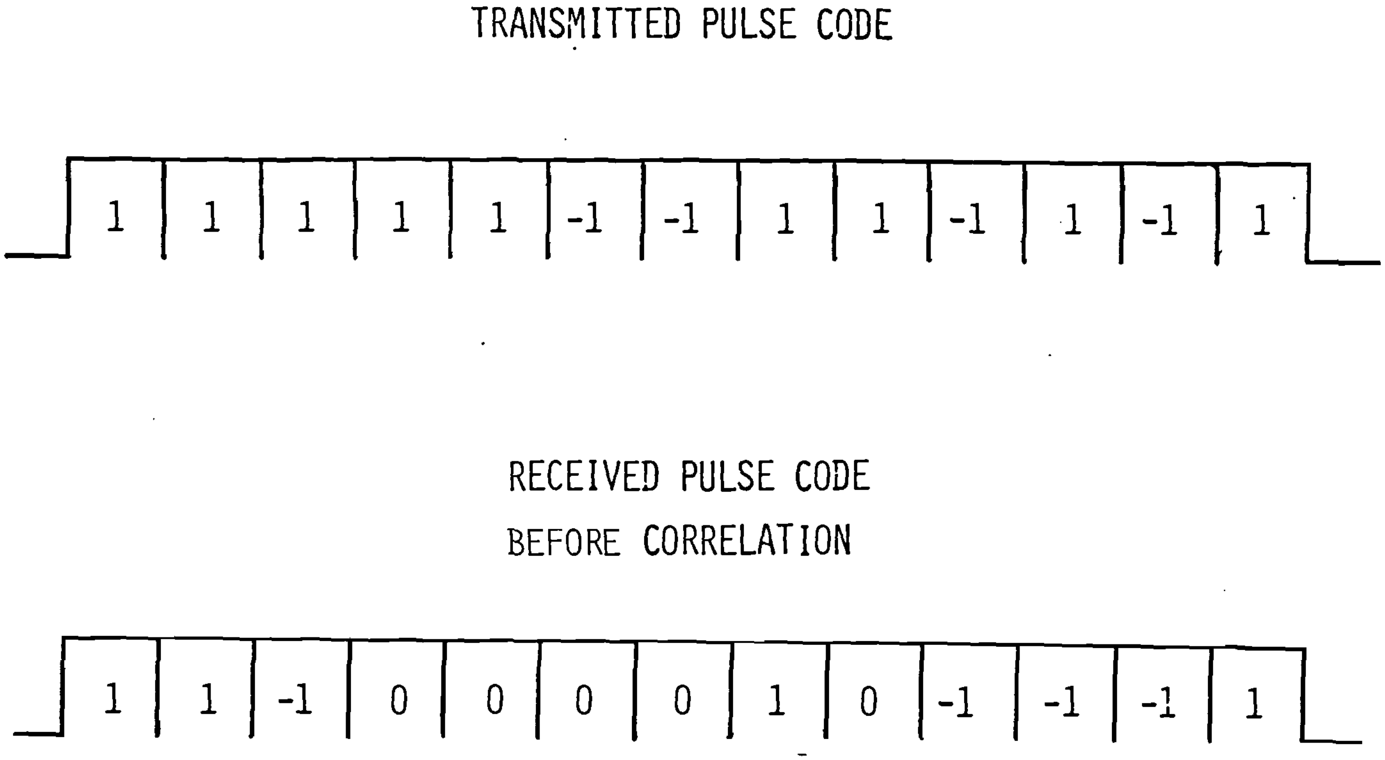


Figure 10. IPAR digital correlator output.

TABLE 5. IPAR PROCESSOR OPERATIONAL PARAMETERS

Subpulse Length:	10 - 160 ns (100 - 6.25 MHz)
PRF:	50 - 8,000 Hz plus manual
Range Gate:	0 - 15 km 0.75 m maximum accuracy
Correlation Processor:	True correlation for 32 bit code length Pseudo correlation for 1 - 31 bit code length
Adjustable Digital Threshold	
Built-In Self Test/Calibration	

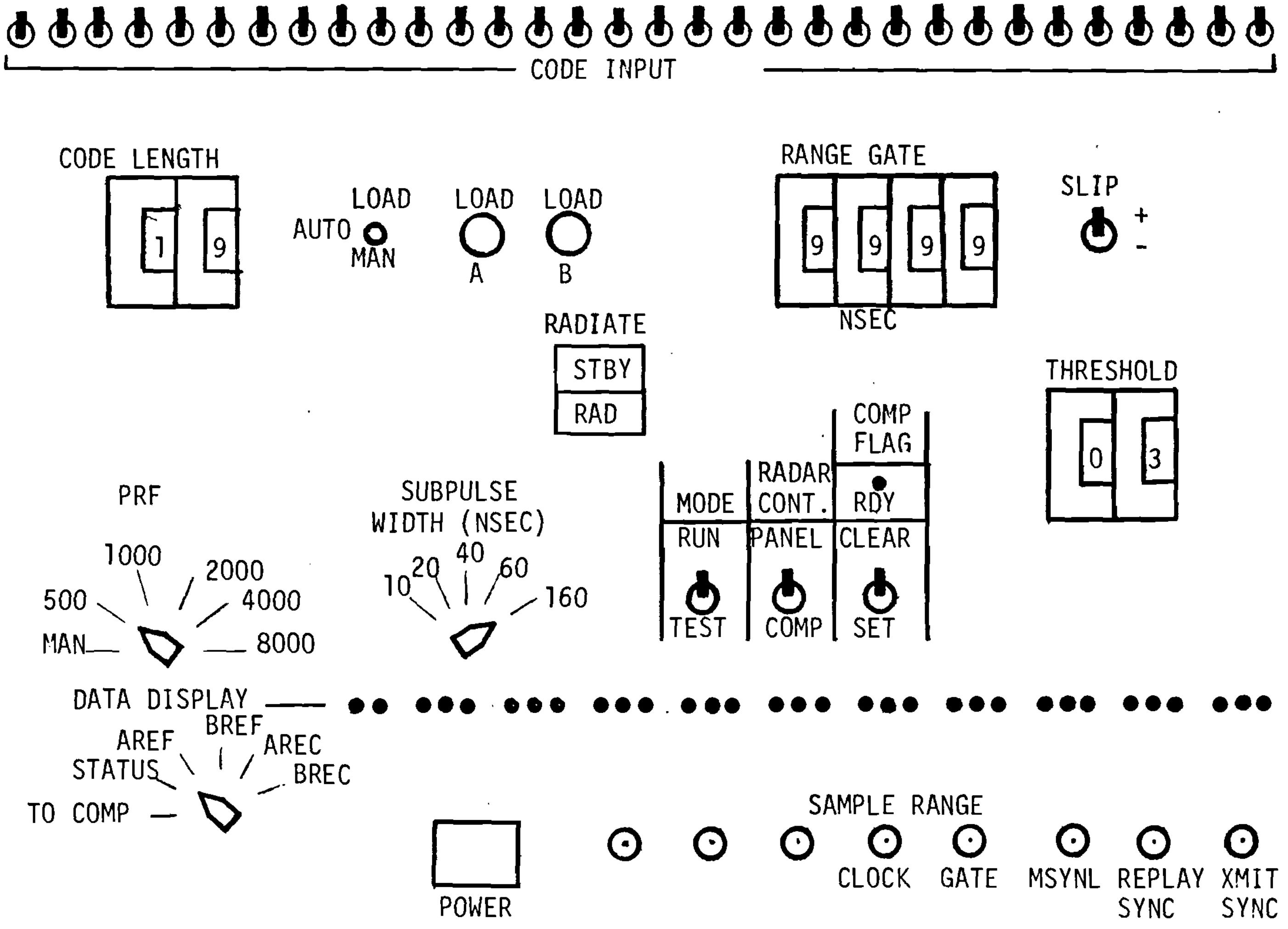


Figure 11. IPAR processor front panel.

In this data collection mode, the digital data were supplemented with analog amplitude data. The amplitude outputs of the logarithmic amplifiers were entered into a bank of range gated sample and hold circuits triggered at the PRF. The resulting time history of the received radar signal strength was recorded on a Honeywell 7-track FM tape recorder. In addition to the horizontal and vertical amplitudes, the FM tape contains a time code signal, a PRF triggered synchronization pulse, and voice annotation.

In the analog data collection mode (Mode B), the IPAR correlator was not employed. Instead, the phase information was translated in the phase detection network (Figure 8) into the sine and cosine of the relative H-V phase angle and recorded on FM tape along with the other five analog channels described. The four data channels were continuously monitored at the read heads of the FM tape recorder using a four channel oscilloscope. In addition, for both data collection modes, the two amplitude channels were routed to a strip chart recorder for observation and analysis. The block diagram of the recording and monitoring system is shown in Figure 12.

3.3 ANCILLARY EQUIPMENT

In addition to the radar system and the recording and monitoring equipment, several pieces of ancillary equipment were required for the data collection endeavor: in particular, the multipath pole and the supply of reflector targets. The multipath pole, shown in Figure 13, consisted of a 4 m wooden two-by-four fixed to a tilted base. The 15^{0} tilt of the base was designed to reduce the specular return from the pole itself so that the targets of interest would be dominant. A wooden traveler was afixed to the pole so the reflector targets could be raised and lowered. In addition, the pole was clearly marked in .15 m increments for accurate determination of the height of the target at all times.

The targets themselves consisted of standard trihedral and dihedral corner reflectors of various dimensions. The variety of sizes was included as an experimental variable so that the performance of the IPAR process could be related to target cross section. The trihedral radar targets were standard 90^{0} corner reflectors with triangular sides. The structure of a trihedral reflector and its backscatter pattern appear in Figure 14. The X-band cross sections of the trihedrals employed ranged from 1 m^2 to 36 m^2 and represented a decibel range of 15.6 dB. Table 6 lists the trihedral reflectors that were used along with their X-band cross sections.

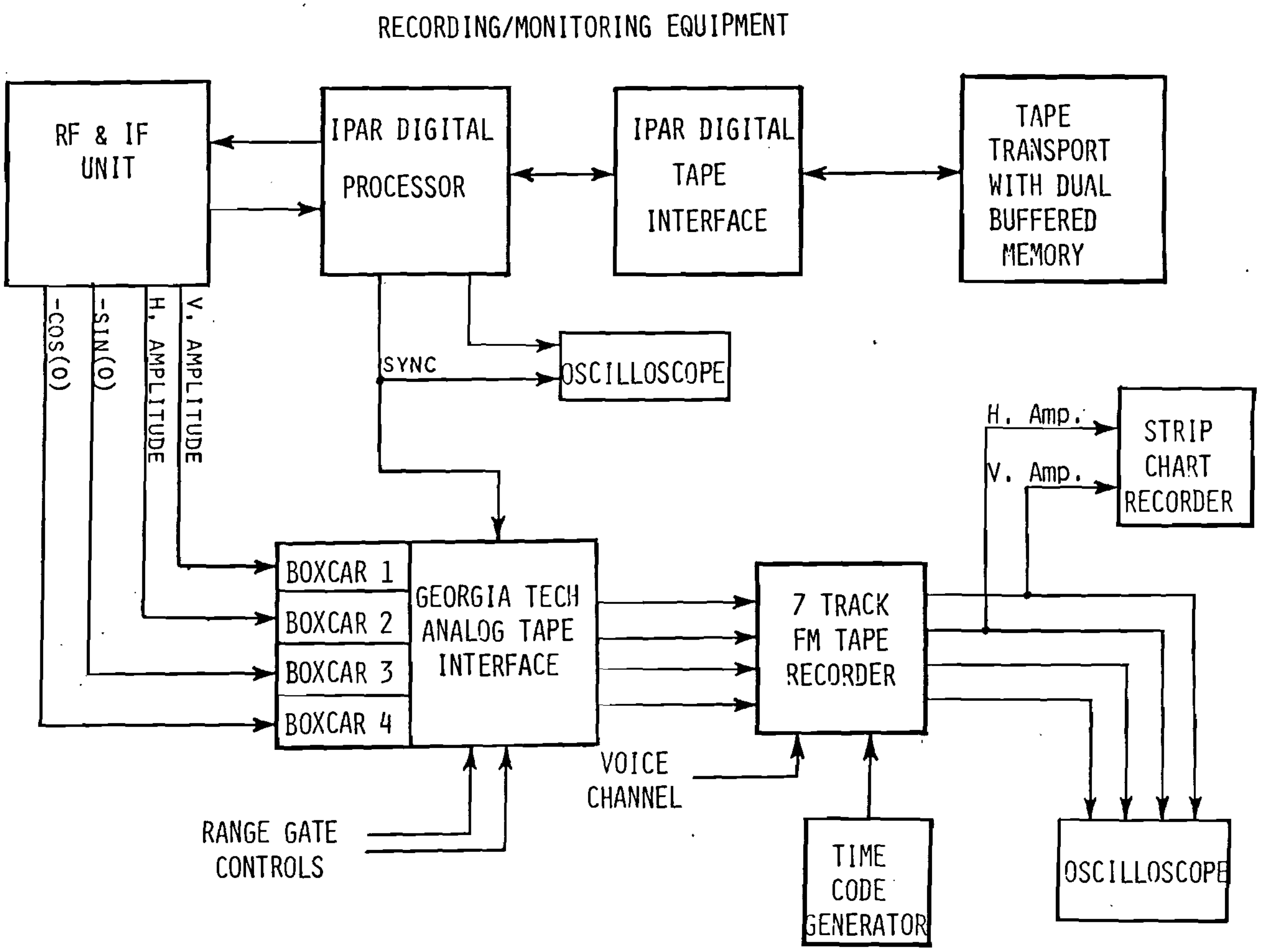


Figure 12. Recording and monitoring equipment block diagram.

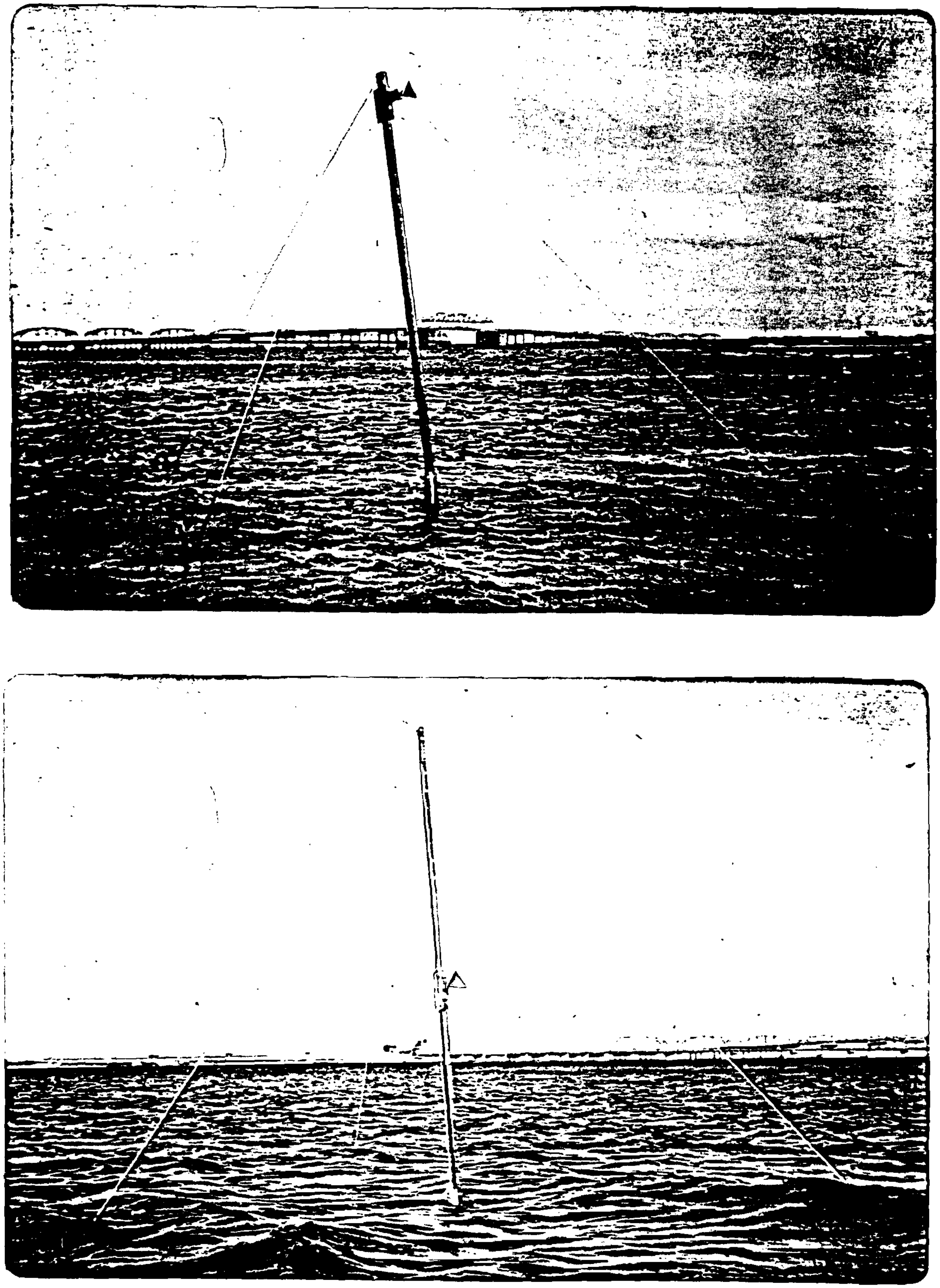


Figure 13. Multipath pole in St. Andrews Bay.

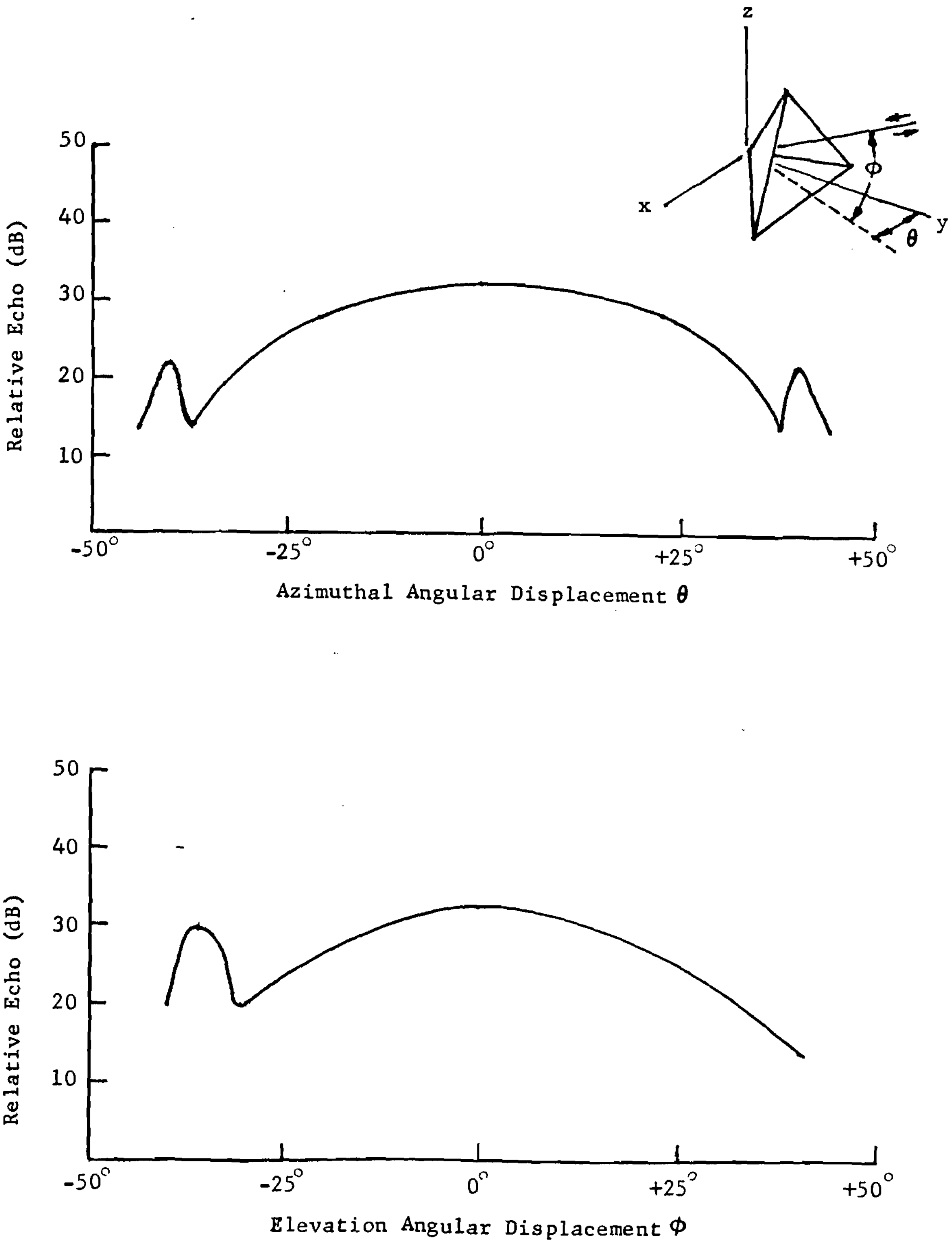


Figure 14. Trihedral reflector geometry and backscatter pattern.

TABLE 6. TRIHEDRAL AND DIHEDRAL CORNER REFLECTOR CHARACTERISTICS

TRIHEDRAL CORNER REFLECTORS

a (inches)	SIZE (m^2)	SIZE (dB)
4.9	1	0
8.7	10	10
10.3	20	13
11.3	28	14.5
12.0	36	15.6
37.0	3025	35

DIHEDRAL CORNER REFLECTORS

a, b (inches)	SIZE (m^2)	SIZE (dB)
2.2, 4.4	1	0
3.9, 7.9	10	10
4.7, 9.3	20	13

Dihedral targets were also employed to demonstrate the effect of even bounce reflectors on IPAR processing. These targets had rectangular sides of varying dimensions. A typical dihedral pattern as a function of plate dimensions is shown in Figure 15. The range of dihedral cross section was from 1 m^2 to 26.5 m^2 (a range of 14.2 dB). Table 6 lists the dihedral corner reflectors employed along with their X-band radar cross sections.

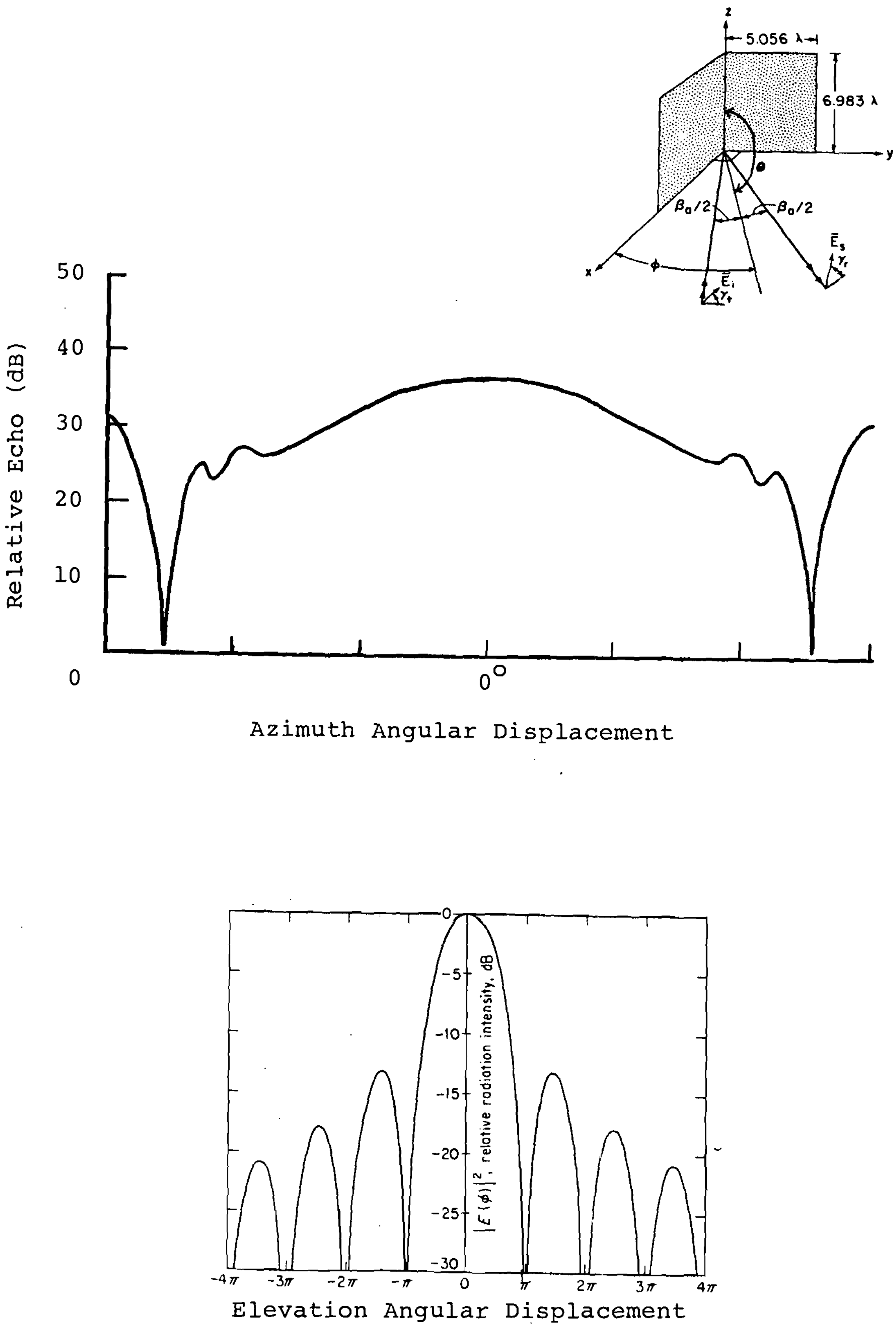


Figure 15. Dihedral reflector geometry and backscatter pattern.

(THIS PAGE INTENTIONALLY LEFT BLANK)

SECTION 4
EXPERIMENTAL PROCEDURE

During the IPAR field measurements, a systematic data collection procedure was employed. A complete calibration of the radar system was performed at the beginning and end of each day. In addition, the polarization circularity and power of the transmitted signal were monitored continuously, ensuring accurate received phase information and complete documentation of the amplitude of the returned signal. The data collection procedure itself was systematized into two data collection modes: mode A for digital and analog data and mode B for analog only data. These data collection procedures and the radar calibration methods are described in detail below.

4.1 RADAR CALIBRATION

The main purpose of these field experiments was to quantify the performance of the IPAR technique in a functional radar system. For this purpose, much of the performance evaluation was based on the phase only information processed by the IPAR correlator. However, it was essential for completeness that absolute radar cross section data be collected. Absolute RCS can be determined by comparing the amplitude of the radar return to either a signal of known power or to a signal reflected from a standard reflector target correcting for differences in range between the target and the reflector. Both methods require that the receiver transfer function be determined. In addition, the power comparison method requires that the transmitted power and the system losses be carefully determined.

Both of these methods were employed during the IPAR data collection program. If the two independent methods result in similar predicted radar cross sections for a particular target, then a high level of confidence in the RCS values is established.

4.1.1 POWER MEASUREMENTS

4.1.1.1 RF Source Power Measurements

There were three types of X-band RF carriers employed in the IPAR data collection exercise: narrowband, swept frequency, and wideband noise. The first step in the power measurement was to determine the power produced by each source to be injected into the TWT amplifier chain. The first two RF sources were produced by a

Hewlett Packard 624 X-band test set, the calibration and measurement of which were performed as follows:

1. Calibrate power dial of HP signal generator.
2. Read power out on power meter.
3. Switch mode to "FM" and read power out.
4. Dial in $\geq$ 20 dB of attenuation and disconnect cable.

For the wideband noise source no adjustments were necessary, and the output power was measured directly with the HP power meter as described in step 2 above.

4.1.1.2 Transmitter Power Measurements

In addition to measuring the basic power produced by the RF sources, the actual average transmitter power in each polarization channel was measured at the test port shown in Figure 6. The actual peak transmit power can be calculated from the measured average power using the equation

$$P_p = \frac{\text{Power meter reading}}{\text{Coupling efficiency}} \times \frac{1}{(\text{PRF})\ (N)\ (T_s)}$$

where the coupling efficiency is -13 dB (0.05) for single channel measurements (10 dB coupler + 3 dB Magic Tee), PRF is the pulse repetition frequency, N is the number of subpulses selected, and T_s is the selected subpulse width. Ratings on the thermistor mount dictate that the factors $(N \times T_s)$ must be kept below 100×10^{-9}.

The procedure for measuring the transmitted power in each channel was as follows:

1. Connect the power meter thermister to the test port.
2. Calibrate the HP test set.
3. Turn the 4 TWT amplifiers to "Standby" and allow them to warm up.
4. Set the correlator-timing and control.
5. After allowing sufficient warm up time,
   a. Engage "Radiate" switch on the correlator control panel.
   b. Select N = 5 on control panel.

c. Select T_s = 20 and PRF = 1000 on control panel.
d. Set all code switches to "1" position.
e. Load code with momentary load switch.
f. Reduce low power TWT gain to minimum.
g. Switch low power TWT No. 1 to "Radiate."
h. Engage "Radiate" switch on hi power TWT No. 1.
i. Increase low power TWT No. 1 gain pot until power meter reads desired level (2.5 mW).
j. Go to "Standby" on both low-power and high-power TWT amplifiers.
k. Repeat Steps 5a through 5j for TWT amplifier Chain No. 2.
l. Disconnect the thermistor mount from the waveguide.

4.1.2 TRANSMIT SIGNAL PHASE ADJUSTMENT

The ellipticity of the transmitted polarization is affected by the power balance and by the phase between the two X-band signals arriving at the dual mode coupler. A rectangular waveguide horn is located on the antenna dish to provide a means of establishing the required circular polarization. This horn is adjustable in 45 degree increments with respect to the primary antenna feed. The horn was positioned at the +45° position so that the final power and phase conditions could be established.

Several iterations of the following procedure were sometimes required to establish a linear 45° polarization. Once the 45° linear polarization has been obtained, the calibrated phase shifter in the transmitter was adjusted for a phase delay of 90° to achieve circular polarization.

1. Connect a 50 ohm coaxial cable between the crystal detector located on the antenna and a 100 MHz bandwidth oscilloscope. Be sure that the coaxial line is terminated at both ends with the proper impedance.
2. Position the test horn to 45° and lock the detent.
3. With the power balanced as indicated previously (must be disconnected at the roof test port), rotate the transmitter phase shifter to 0° and lock the calibration dial.
4. Set the code length to 32.
5. Set a 32-bit code of alternating 4-bits (11110000111 etc.) and load the code.
6. Turn on all 4 TWT amplifiers.

7. Observe the transmit pulse on the oscilloscope and adjust the transmit phase shifter for a null in the one's position of the detected code. This condition corresponds to transmission of linear 45^o polarization.
8. The initial adjustment should achieve a null greater than 20 dB below the peak observed for the code zeros.
9. Increase the vertical sensitivity on the oscilloscope and adjust the gain on one, not both, of the low power TWT amplifiers to achieve a better null if required. The TWT current should not be changed from its power-balance condition by more than 2 to 3 percent.
10. An iteration between power adjustment and phase adjustment may be required to achieve the best null. Repeat steps 7 through 9 as required.
11. Once the "best" null is achieved, unlock the phase shifter dial and set the phase to 90^o. This results in transmission of circular polarization.
12. Observe the transmit waveform to see that both the "1s" and "0s" of the code are the same amplitude; otherwise the polarization is elliptical.

Using this procedure, the system was adjusted so that the horizontally polarized and vertically polarized signals radiated from the primary feed were equal in power and 90^o out of phase. These conditions establish a circularly polarized radiated field.

4.1.3 RECEIVER AMPLITUDE CALIBRATION

The radar receiver was calibrated by injecting a signal of known frequency and power level into the same test port used to measure the transmitter power. The signal, generated by the HP 624 test set was adjusted to be 0 dBm as previously described. The receiver response function was documented using the attenuator on the HP test set as described in the following procedure. Note that the amplitude signals out of the logarithmic amplifiers were sampled and held at the PRF and recorded on FM tape as described in Section 3.2.

1. Turn on the HP 624 test set and allow 15 minutes warm up time.
2. Perform an internal calibration on the test set, as previously described.
3. Connect the test set to the power meter through the same coaxial cable used to connect to the waveguide test port.

4. Measure and record on the data sheet the power measured at the end of the cable.
5. Disconnect the cable from the power meter and re-connect to the waveguide test port.
6. Ensure that the receiver, transmitter, and test set are tuned to the same frequency.
7. Start at zero dB on the test set and on the receiver attenuators and record 10 second range gated samples for every 5 dB attenuation over the dynamic range of the receiver (usually around 70 dB).
8. Using the test set attenuator, record a series of maximum/minimum signal levels to aid in later setting of scaling amplifiers used in the data reduction process.

4.1.4 RECEIVER PHASE VERIFICATION

4.1.4.1 Receiver Phase Shifter Adjustment

The IPAR process uses the relative phase angle between the horizontal and vertical components to determine the polarization properties of the target. It is imperative that no spurious effect upon this phase angle is introduced in the receiver. For this reason the two receive channels were matched in path length as closely as possible. For fine tuning adjustments of this path length, a precision phase shifter was included in one receive channel, as shown in the receiver block diagram, Figure 7. Adjusting this phase shifter was a part of each calibration and proceeded as follows.

1. Perform the radar system calibration described above; set the HP test set to CW.
2. Boresight the antenna on a reference corner reflector.
3. Load an optimal 32-bit code into the correlator.
4. Switch the TWT's to "Radiate."
5. Observe the video correlation function out of the correlator on an oscilloscope.
6. Adjust the correlator range gate until the correlation function is centered on the target.
7. Add attenuation to each channel until the correlation function is drawn out of saturation.
8. Adjust the receive phase shifter until the correlation function is optimized.

9. Record the phase shifter setting; all CW data should be collected at this setting.
10. Repeat 5 through 9 for each RF carrier.

4.1.4.2 Phase Detector Verification

For the analog data collection mode (Mode B), the H-V phase information was not passed to the IPAR correlator but was processed in the phase detection network (Figure 8) to produce the sine and cosine of the relative H-V phase angle. The phase detectors consisted of two identical DBM-200 double balanced mixers. The required 90 degree phase shift was achieved with a pair of coaxial cables which were cut so that one cable was 1/4 wavelength longer than the other. It was important that the sine/cosine relationship be verified before data collection began. The following steps were performed to document the functioning of the phase detector network.

1. Attach a cable from boxcar circuit No. 3, containing the cosine of the H-V phase angle, to the Y axis of the oscilloscope and a cable from boxcar circuit No. 3, containing the sine of the phase angle, to the X axis input of the oscilloscope.
2. Apply a signal from the HP 624 test set to the waveguide test port.
3. Vary the phase shifter in the horizontal receiver and observe that the path of the dot on the oscilloscope traces out a circle.
4. If for the conditions set up in 3, the oscilloscope trace is not a circle, trim the cable length by using coaxial adapters until a circular Lissajous pattern is achieved.
5. Return the phase shifter to the setting determined in Section 4.1.4.1

4.1.5 CALIBRATION VERIFICATION

The receiver calibration will be verified against the radar return from a standard corner reflector during the next phase of the program. The theoretical power return from the standard corner reflector can be calculated from the radar range equation. The level of the video return from the calibration target will be compared with the receiver calibration which was described in Section 4.1.3. The calculated theoretical power should agree with the measured value within ± 2 dB. The peak received power can be calculated from:

$$P_r = \frac{P_t G^2 \lambda^2}{(4\pi)^3 R^4} \left(\frac{1}{L_1}\right) \left(\frac{1}{L_2}\right) \sigma$$

where:

P_t = measured value from paragraph 4.1.2
G_2 = 79.2 dB
λ^2 = -29.9 dB
$(4\pi)^3$ = 33.0 dB
L_1 = 1.2 dB (system losses)
L_2 = 3 dB (linear component of circular wave)
R = measured range to target
σ = 28 m^2

If the measured and calculated values agree within the acceptable limits, the calibration is closed and the RCS values are validated.

4.2 DATA COLLECTION PROCEDURE

The procedure for data collection followed a systematic format. A data log was maintained for each run in which the following parameters were included:

1. Date
2. FM Tape Number
3. Digital Tape Number
4. FM Channel Designations
5. Run Number
6. Time
7. Target Type and Size
8. Range to Target
9. Data Collection Mode
10. Transmission Carrier Type
11. Code Length
12. Code Type
13. Subpulse Width

14. Threshold Levels for Digitization
15. Receiver Attenuation
16. Comments.

The data collection parameters were varied in accordance with a test matrix. A list of the matrix variables is presented in Table 7. The test matrix was the same for the two data collection modes, but the procedures differed somewhat. An example of the steps followed for each mode is presented below.

4.2.1 MODE A: DIGITAL AND ANALOG

With the phase information routed through the IPAR correlator to the digital tape recorder and the amplitude data recorded on FM tape, the measurements procedure was as follows:

1. Set up trihedral target.
2. Transmit a fixed frequency (narrowband).
3. Set and load code per test matrix.
4. Set PRF to 1,000 Hz.
5. Select a 20 ns subpulse width.
6. Select N = 32.
7. Set threshold = 02.
8. Insert sufficient attenuation in receiver channels to avoid signal saturation.
9. Position analog range gate 1 (top) and the IPAR digital processor range gate to the range of the target.
10. Verify transmission of circular polarization.
11. Enter run number, date, and threshold setting on digital tape header file.
12. Start FM tape recorder and begin voice annotation.
13. Start digital tape recorder - record 20 seconds of data.
14. Select the next parameter in the test matrix.
15. Repeat steps 10 through 14.
16. Switch transmitter to FM (swept frequency).
17. Repeat steps 3 through 15.
18. Switch to wideband noise source.
19. Repeat steps 3 through 15.

TABLE 7. DATA COLLECTION VARIABLES

RF Carrier:	narrowband, swept frequency, or noise
Collection Mode:	digital or analog
Code Length:	13, 20, or 32
Subpulse Width:	20 ns or 40 ns
Threshold Setting:	0 to 29
Target Type:	trihedral or dihedral
Target Position:	as required for specific test

20. Move trihedral and repeat, above procedure.
21. Replace trihedral with dihedral and repeat, above procedure.

4.2.2 MODE B: ANALOG ONLY

For the analog data collection mode the phase information was routed through the phase detector network (Figure 8), which produced the sine and cosine of the relative H-V phase angle. These signals were recorded on FM tape along with the amplitude information. Because it was impossible to sample all of the received subpulses simultaneously with the single analog range gate, the mode B recording procedure was somewhate different from mode A. Instead of fixing the analog range gate on the target, it was placed in front of the target and slewed at a constant rate across it. This procedure is described below:

1. Set up trihedral target.
2. Connect the phase detector network as shown in Figure 8.
3. Transmit a fixed frequency.
4. Set and load code per test matrix.
5. Set PRF to 1,000 Hz.
6. Select a 20 ns subpulse width.
7. Select N = 32.
8. Insert sufficient attenuation in receiver to avoid signal saturation.
9. Position the analog range gate in front of the target.
10. Start FM tape recorder and begin voice annotation.
11. Slew range gate across target four times.
12. Select the next parameter in the test matrix.
13. Repeat steps 10 through 13.
14. Switch transmitter to FM (swept frequency)
15. Repeat steps 4 through 14.
16. Switch to wideband noise source.
17. Repeat steps 4 through 14.
18. Move trihedral and repeat, above procedure.
19. Replace trihedral with dihedral and repeat, above procedure.

SECTION 5
THE EXPERIMENTS

The major aim of this IPAR program was to generate, record, and make available IPAR data that would subsequently be used to document, understand, and quantify various characteristics of the IPAR waveform. Since IPAR technology and, indeed, the IPAR concept itself are still quite young, it was necessary to design and implement well-documented, tightly controlled experiments to ensure that each set of experimental data provided information on an isolated IPAR characteristic.

The following subsections describe each of the separate experiments conducted during the data collection program. In each case, the purpose, setting, and execution of the experiments are described.

5.1 COMPRESSION ON SIMPLE TARGETS

Data on simple targets in a multipath-interference-free, high target-to-clutter ratio environment were recorded at each of the test sites (Section 2), and an extensive set of data on simple targets was recorded at the Dobbins and Panama City Beach sites. The Dobbins data will serve as proof-of-concept data showing that the system actually achieves pulse compression on trihedrals and dihedrals. The St. Andrews Bay data for simple targets were recorded using a trihedral at various signal-to-noise (S/N) ratios. The signal-to-noise levels were varied between -5 dBm and +5 dBm in 5 dBm increments. The reflector was situated in a low clutter environment, out of any multipath interference, and the S/N ratio was varied by adding attenuation in the receiver precision attenuators. These data will aid in determining the compression gain (with respect to noise) of the pseudo coherent process that IPAR uses for phase detection.

5.2 MULTIPATH INTERFERENCE OVER BAY

These data were collected at the bay site, and will allow analysis of the behavior of the IPAR waveform and pulse compression processing in a multipath interference environment, where the reflecting surface is (fairly smooth) bay waters.

The one square meter trihedral was mounted on the multipath pole, and it's height above the mean water level (MWL) was varied in .15 meters (six inches) increments over a range from .45 meters (eighteen inches) to 1.82 meters (seventy-two inches) above the MWL. Since the geometry of the test site resulted in a lobing structure with a periodicity of one multipath interference peak every .91 meters (36 inches) above MWL, this experiment covered 1.5 full cycles of the lobing structure. At each height, a full set of runs (various threshold settings for each of the three carriers) was recorded.

The experiment continued with selective probes of the multipath interference lobing structure generated by each of the one square meter dihedral and one square meter trihedral targets. For each of the reflectors, full sets of data were collected at a multipath interference peak, a multipath interference null, and at two points in between.

5.3 MULTIPATH INTERFERENCE OVER SEA

A thirty-six square meter trihedral target was mounted on a Proline runabout to document the performance of the IPAR process in response to targets embedded in a multipath lobing structure, where the reflecting surface is the sea. The Proline operator was instructed to run a straight course toward the radar system from approximately four kilometers out to sea to one-third kilometer from the shore site (see Section 2.2.2). During each such run, the radar antenna was manually slewed so as to remain pointed toward the mounted corner reflector in azimuth and elevation, and the range gate on the IPAR box was slewed to track the target in range. In addition, a 20 bit code was employed so that our range estimate needed to be accurate to only ±6 range bins for the full return from the reflector to be recorded. Each physical run of the boat from four to one-third kilometers from the system was recorded as, approximately, fifteen separate data collection runs. The procedure employed was to allot approximately thirty seconds per collection run or to terminate a run and begin a new one if the nature of the return seemed to be changing substantially.

The resulting data represent fifteen "snapshots" of various regions of the multipath interference lobing structure for each physical run of the Proline. Separate "physical" runs were conducted for each of the system carriers and for various threshold settings.

5.4 BAY AND SEA CLUTTER

Each day of operation at the bay and beach sites, data were collected on the bay/sea state for purposes of documenting the environment for those days' experiments. In addition, full sets of runs (variable carrier, variable threshold, variable angles of depression) were recorded on the last day of operation at the bay site and the first day of operation at the sea site.

These data will be used to study and quantify two phenomena: (1) the loss in pulse compression gain achieved by IPAR when backscatter is from bay and sea clutter and (2) the time/frequency dependence of the angle between the received horizontal and vertical receive channels (ϕ_{HV}) of the IPAR system when the target is bay and sea clutter. The former phenomenon will form the basis for computation of target-to-clutter enhancement due to IPAR processing, and the latter will form the basis for computation of the target-to-clutter discrimination properties of IPAR.

5.5 SIMPLE TARGETS IN BAY AND SEA CLUTTER

Data consisting of backscatter from simple targets embedded in clutter were collected to study and quantify IPAR's ability to (1) distinguish between targets and clutter (target discrimination) and (2) generate greater pulse compression gain on target than clutter signatures (target-to-clutter enhancement).

In both cases (bay and sea) it was difficult to establish low signal-to-clutter ratio situations. This ratio was minimized for the bay measurements by using the smallest reflector available to us (1 m^2); the resulting signal-to-clutter ratio was still, however, high. Lower ratios were achieved during the sea experiments by sending the Proline out to four kilometers from the beach site; however, to keep the simple reflector dominant over its carrier vehicle (the Proline), a 36 m^2 trihedral was used. The signal-to-clutter ratio established was still not as low as would have been desired.

5.6 RESOLUTION EXPERIMENTS

After completion of the planned bay and sea experiments, the Dobbins Air Force Base site was reestablished in order to conduct experiments to document IPAR's resolving ability and probe IPAR's sensitivity to target complexity.

The resolution experiments consisted of collecting data on two targets separated by two range resolution cells (approximately 20 feet). Full sets of data were collected on each of three configurations: (1) two 1 m^2 trihedrals separated by two range bins, (2) a 1 m^2 trihedral and a 1 m^2 dihedral separated by two range bins, and (3) a 1 m^2 and a 10 m^2 trihedral separated by two range bins. The last of these experiments will exhibit the phenomenon of "capture," where a large reflector masks the return from a small reflector even though the two are located in different range bins. This phenomenon occurs with binary phase coded systems as well and is a function of coarseness of quantization in the receiver processing. For phase-coded systems, "capture" occurs when too few bits are used to quantize amplitude and phase. For IPAR "capture" is a result of hard limiting the amplitude and quantizing the relative phase of the two received polarization components to only two bits.

5.7 COMPLEX TARGETS

For the purposes of this document, define complex targets to be those which, on a range resolution cell basis, consist of more than one dominant scatterer. Thus, grass and leafy trees represent complex targets, and man-made objects may or may not represent complex targets in this sense depending on the particular target, target aspect, and system resolution.

Since IPAR waveforms respond exactly oppositely in relative polarization phase to even and odd bounce scatterers,(6) a target set composed of a dihedral and trihedral of equal radar cross sections and located at the same range is "invisible" to the IPAR pulse compression process. This property raises concern over IPAR's ability to pulse compress on man-made objects. The "complex target" experiments were designed to permit analysis and quantification of IPAR's response to such targets.

The experiments were carried out at the Dobbins site by collecting data on various configurations of dihedrals and trihedrals set in a single range bin. Each configuration was documented by photographs and by recording the backscatter from, first, all of the dihedrals in the configuration and, then, all of the trihedrals in the configuration. Thus, the total cross section of the even-bounce reflectors and the total cross section of the odd-bounce reflectors of the complex target were recorded for each configuration.

Data were collected from complex target configurations consisting of as few as two reflectors and as many as six reflectors in a single range bin, and the total difference in radar cross section between the ensembles of dihedrals and trihedrals was varied between 0 dBsm and 10 dBsm. In addition, the distance between reflectors (in the two-reflector experiments) was varied by a fraction (≈1/3) of a wavelength to document the effects of carrier constructive and destructive interference on the IPAR pulse compression scheme.

5.8 TREE AND GRASS CLUTTER

Each day of operation at the Dobbins and Georgia Tech sites data were recorded for documentation purposes from the grass and tree clutter that formed the background for the day's experiments. In addition, full sets of backscatter data were recorded from a tree line consisting of evergreen trees and a bush line consisting also of evergreens, both at the Georgia Tech site. The former represent extended (depth of more than one range bin) clutter signatures, and the latter represent non-extended clutter signatures.

As with the bay and sea clutter measurements, these data will be used to analyze and quantify IPAR's target-to-clutter enhancement and target discrimination properties.

5.9 TARGETS IN TREE CLUTTER

Trihedrals were embedded in and then placed in front of the aforementioned tree line. For each position, three different trihedrals were used to create various signal-to-clutter environments. Full sets of data were collected for each of these six configurations.

A 3 m^2 Trihedral was embedded in the bush-line mentioned above to create a 0 dB signal-to-clutter situation. Full sets of backscatter data were recorded from the clutter alone and from the clutter with the embedded target.

5.10 TARGETS OF OPPORTUNITY

Full sets of backscatter data from Stage II (an abandoned observation and test platform located offshore) and the Proline runabout were recorded at the beach site and from a pickup truck at the Georgia Tech site. These data will be utilized to investigate the interaction of the IPAR waveform and man-made objects as well as to investigate the nature of range profiles generated by these interactions.

(THIS PAGE INTENTIONALLY LEFT BLANK)

SECTION 6
DATA SUMMARY

6.1 SUMMARY

The data collection which formed a large portion of this program resulted in the generation of 22 digital tapes and 17 FM tapes of IPAR data, which in all contain the backscatter from more than 1300 system runs. These data are summarized in Table 8 according to run number, object studied, scenario, and the particular experiment involved. As described in Section 3.2, Mode A data are digitized IPAR returns that describe the relative phase of the received pulses on a subpulse basis. Each of these Mode A runs are, in addition, documented by the recording on FM tape of the amplitude in the H and V receive channels. As will be explained in Section 8, the Mode A data are useful only when the major portion of the radar backscatter can be assumed to emanate from a single range bin. Thus, the Mode B data will have to be utilized for analysis of the bay and sea clutter measurements (Runs 702-782), the signature of Stage II (Runs 935-946), the resolution experiments (Runs 950-1005), and almost all the background documentation. In all, the collection program generated 722 runs of usable Mode A (digital) data and 413 runs of usable Mode B data.

The Mode B data have the advantage of containing both the amplitude and non-quantized relative phase data of received signals on a subpulse basis; however, because of bandwidth limitations (as described in Section 3.2), these data are collected over many pulses so that any reconstituted single pulse actually represents sampled returns from a number of pulses. These data will prove invaluable for amplitude documentation of the various experimental scenarios, especially for those in which the background represents a distributed reflector. These data will also be used to characterize the actual relative H-V receive phase from various targets; however caution will have to be exercised in drawing conclusions since a time average and a large frequency excursion, when the frequency swept and noise carriers are employed, are by necessity intrinsic in all these data.

6.2 DATA SUMMARY ACCORDING TO ANALYSIS OBJECTIVES

The data collection was undertaken with various objectives in mind for subsequent data analysis. Table 9 summarizes the resulting IPAR data that will be available for use

TABLE 8. SUMMARY OF DATA COLLECTED.

DATE	RUN NO.	OBJECT	DIGITAL TAPE NO.	FM TAPE NO.	MODE	DESCRIPTION	EXPERIMENT
11/23	1-25	$36m^2$ Trihedral	1	1	A	Compression Tests	
11/26	28-50	$36m^2$ Trihedral	2	2	A	Compression Tests	Compression on Simple Target
	51-59	Grass	2	2	A	Background Documentation	
	60-78	$36m^2$ Trihedral	2	2	A	Compression Tests	
	79-96	$36m^2$ Trihedral	—	2	B	Compression Tests	
12/01	100-240	$1m^2$ Trihedral	3/4	3/4	A	Multipath Interference over bay	Performance in Multipath Interference Field
12/02	241-258	$1m^2$ Trihedral	5	4	A	Receiver Phase Adjustments	
	259-298	Pole	5	4	A	Background Documentation	
	299-360	$1m^2$ Trihedral	5/6	4/5	A	Multipath Interference (cont.)	
12/03	361-403	$1m^2$ Trihedral	7	5	A	Multipath Interference (cont.)	
12/07	411-545	$1m^2$ Trihedral	—	6/7	B	Performance as a Function of S/N Ratio	Documentation of Processing Gain
12/08	546-602	Pole	—	8	B	Background Documentation	

TABLE 8. SUMMARY OF DATA COLLECTED.

(continued)

DATE	RUN NO.	OBJECT	DIGITAL TAPE NO.	FM TAPE NO.	MODE	DESCRIPTION	EXPERIMENT
12/09	604-701	$1m^2$ Dihedral	8	8/9	A/B	Multipath Interference over bay	Performance in Multipath Interference over Bay (continued)
12/09	702-718	Bay Clutter	8	9	A/B	Bay Clutter	Clutter Measurements
12/15	719-782	Sea Clutter	9/10	9/10	A	Sea Clutter	
12/16	783-790	Proline	10	10	A	Background Documentation	Performance in Multipath Interference
	791-825	Sea Clutter	10	10	A	Background Documentation	Over Sea
12/17	826-887	$36m^2$ Trihedral	11/12	11	A/B	Multipath Interference over sea	
12/20	893-931	$36m^2$ Trihedral	14/15	12	A/B	Multipath Interference over sea	
	932-934	Proline with Pole	—	12	B	Background Documentation	
12/20	935-946	Stage II	15	13	A/B	Complex Target	Signature of Target of Opportunity
	947-948	Sea Clutter	15	13	A	Background Documentation	

TABLE 8. SUMMARY OF DATA COLLECTED.

(continued)

DATE	RUN NO.	OBJECT	DIGITAL TAPE NO.	FM TAPE NO.	MODE	DESCRIPTION	EXPERIMENT
1/17	950-963	Grass	16	13	A/B	Background Documentation	Resolution Experiments
	964-983	$1m^2$ Trihedral	16	13	A/B	$1M^2$ Trihedral Documentation	
1/25	984-994	Two Trihedrals	—	13	B	Trihedrals Separated by Two Range Bins	
1/31	995-1005	Trihedral and Dihedral	17	13	A/B	Two Range Bin Separation	
1/31	1010-1074	$1m^2$ Dihedral and Trihedral	18	14	A/B	Two Reflectors in One Range Bin	Simulated Complex Targets
2/1	1075-1093	$1m^2$ Dihedral and Trihedral $10m^2$ Trihedral	19	15	A/B	Two Reflectors	
	1094-1096	Grass, Reflectors	—	15	B	Background Documentation	

TABLE 8. SUMMARY OF DATA COLLECTED.

(continued)

DATE	RUN NO.	OBJECT	DIGITAL TAPE NO.	FM TAPE NO.	MODE	DESCRIPTION	EXPERIMENT
2/3	1097-1108	$1m^2$ Dihedral and Trihedral $10m^2$ Trihedral	20	15	A	Two Reflectors	Simulated Complex Targets
	1112-1148	$26.5m^2$ Dihedral $10m^2$ Trihedral.	20/21	15	A/B	Two Reflectors	(continued)
	1149-1162	$26.5m^2$ Dihedral $22m^2$ Trihedral	21	15	A/B	Two Reflectors	
	1163-1165	Individual Reflectors	—	15	B	Documentation	
2/4	1166-1173	Individual and Combinations of Reflectors	—	16	B	Documentation	
	1174-1189	1,$3m^2$ Trihedrals 7, 10, $26.5m^2$ Dihedrals	22	16	A/B	Five Reflectors	
	1190-1193	Individual and Combination of Reflectors	—	16	B	Documentation	

TABLE 8. SUMMARY OF DATA COLLECTED.

(continued)

DATE	RUN NO.	OBJECT	DIGITAL TAPE NO.	FM TAPE NO.	MODE	DESCRIPTION	EXPERIMENT
	1194-1212	3,7,10m^2 Trihedrals 7,10,26.5m^2 Dihedrals	22	16	A/B	Six Reflectors Targets	Simulated Complex
	1213-1214	Individuals and Combinations	--	16	B	Documentation	(continued)
	1215-1241	7,10,28m^2 Trihedrals 10,20,26.5m^2 Dihedrals	22	16	A/B	Six Reflectors	
	1242-1244	Individuals and Combinations	--	16	B	Documentation	
2/17	1248-1286	1m^2 Trihedral 3m^2 Trihedrals 28m^2 Trihedrals	--	17	B	Targets embedded in Clutter	Targets with Tree Clutter
	1287-1304	as above	--	17	B	Targets in front of Clutter	
2/21	1305-1334	Targets and Clutter	--	17	B	Documentation of Targets and Clutter	

TABLE 9. DATA SUMMARY ACCORDING TO ANALYSIS OBJECTIVES.

ANALYSES	NUMBER OF TARGETS/SCENARIOS	APPROXIMATE NUMBER OF RUNS
Compression on Simple Targets	11/5	190
Compression in Multipath and Interference	3/2	600
Characterization Clutter backscatter	5/3	160
Compression as a Function of Target Complexity	7/1	250
Detection of Targets in Clutter	4/5	100
System Range Resolution	2/1	50
Signatures of Targets of Opportunity	4/3	30

in each of the delineated analysis tasks. A brief description of each of the analyses tasks and any special properties of the available data with respect to these tasks are given below.

The analysis of compression on simple targets will provide measures of peak processing gain and variability of the peak as functions of threshold setting and carrier, and these analyses will provide irrefutable evidence of the ability to pulse compress on polarization modulation, the heart of the IPAR concept. In addition, analysis of peak gain as a function of S/N ratio will aid in the development of a theory to characterize the pulse compression gain realizable with a pseudocoherent detection process.

The multipath interference data will serve as a basis for a theoretical description of the behavior of the IPAR waveform in multipath interference and as a base for quantifying compression performance as a function of carrier, ground plane, and position on the multipath interference lobing structure. The bay data represent precisely documented, well controlled measurements which will be utilized for the phenomenological study, and the sea data represent a more realistic operational scenario, better suited for prediction of performance in an operational situation.

The clutter data include bay, sea, grass, tree, and bush clutter. (As noted in Section 6.1, since any Mode A date on extended targets may be misleading, only Mode B extended clutter data will be reduced for analysis. The number of runs shown in Table 9 include only these useful data). These data will be analyzed to determine features that (1) characterize each type of clutter individually and characterize the ensemble of types as "generic clutter" and (2) distinguish these (both individually and as an ensemble) IPAR returns from simple and complex target IPAR returns. Analysis will also determine the compression gain achievable by IPAR processing for each of these clutter types. All of these analyses will be based on the behavior of the relative angle between the received H and V components and the system's pulse compression characteristics as functions of carrier, integration time, and clutter type.

Based on the results of the separate target and clutter analyses, a target discrimination algorithm will be developed and tested on the collected "targets in clutter" data. The results of this experiment will form a basis for quantification of IPAR's ability to discriminate targets from various types of clutter. This discrimination ability will be quantified as a function of carrier, signal-to-clutter ratio, target type, and clutter type to the extent that the data permit.

The backscatter data collected from artificially-created complex targets are summarized in Table 10. In each case, all the reflectors were located in a single range resolution cell and the backscatter from the two different types of reflectors (trihedrals and dihedrals) were documented separately. The last column on the right in Table 10 represents the computed difference between the reflectivity of the ensemble of trihedrals and the ensemble of dihedrals utilized for each experiment. These cross sections were derived from the dimensions of the individual reflectors, not the measured backscatter from them, and were verified approximately by the relative A-scope levels observed. The actual relative cross sections will be computed from the recorded backscatter during the planned analyses.

The distance between the trihedral and dihedral in the two-reflector, 10 dB difference and 0 dB difference experiments was changed twice by approximately one-third of a wavelength (one-third of an inch) to sample the effects of various interference patterns on the IPAR process. Full sets of data (all three carriers, various thresholds, and both data collection modes) were taken for each of the complex target experiments listed in Table 10. The analysis that these data will be utilized for will be quantification of the IPAR pulse compression process as a function of target complexity, target make-up, and RF carrier.

TABLE 10. COMPLEX TARGETS EXPERIMENTS.

CONFIGURATION		RCS OF DIHEDRALS MINUS RCS OF TRIHEDRALS (in dB)
NO. OF TRIHEDRALS	NO. OF DIHEDRALS	
1	1	-10
		4
		1
		0
2	3	10
3	3	3.4
		1

SECTION 7

DATA REDUCTION AND ANALYSIS

As part of this program, approximately two-thirds (480 runs) of the recorded digital data were reformatted, read to disk, and analyzed. The software governing the writing to and reading from disk and the software that performs the analysis of the digital data were both developed expressly for this application, thus they are highly efficient and flexible for this purpose. Each of the digital runs analyzed has been documented by reduction and calibration of the accompanying FM amplitude data.

The following three subsections respectively describe the tape-to-disk procedure, the analysis program, and the method by which the FM data amplitude data were reduced and calibrated.

7.1 TAPE-TO-DISK TRANSFER

The digital data recorded during the field operation were written on 22 digital tapes. These tapes contain approximately 20 seconds of data for each of approximately 720 runs. It was decided to download the digital data from tape-to-disk and to compact the data in the process since these data were to be extensively analyzed over an extended period of time and 20 seconds of data per run were an order of magnitude more than was needed for observation and analysis of any physical phenomena (but the correct duration to conduct the necessary FM reduction).

The program created and utilized to effect this tape-to-disk transfer (developed and implemented by Mr. Michael Shannon) is included as Appendix A to this report. As a result of the application of this program, 480 runs of digital data have been written to and are accessible from disk. Hence, each of the runs is quickly accessible on a random access basis, which greatly reduces the time required for (and the cost of analysis of) the digital data. Furthermore, since an optimum packing procedure was employed, the processed 480 runs of data occupy only 12.5% of the disk storage space. It is thus expected that all of the digital and digitized FM data generated by our data collection program will fit onto the single, 80 megabyte, dedicated disk allocated to this effort.

The tape-to-disk program contains extensive error checking capability. Of the 481 digital runs processed, only one was found to contain errors, and thus 480 runs were packed to disk. Indeed, the data analysis program (which is described below) exhibited no problems in reading and analyzing the 480 "good" disk files.

7.2 ANALYSIS OF THE DIGITAL DATA

To perform automatic preliminary analysis of the digital data, an analysis computer program was developed and implemented by Mr. Michael Baden that computes the cross correlation between the received signals and the stored code matched filter and then compiles statistical measures-of-goodness for, and variability of, the output (compressed) waveforms. The computer program is reproduced as Appendix B to this report.

The analysis program may be applied interactively to select portions of the data or instructed to process all data present on disk. For each iteration of analysis the following may be specified: (1) run number(s) to be processed, (2) number of pulses to be processed per run, (3) measures to be computed, and (4) format of output.

The printed result of applying the analysis program to the data from run number 11 is reproduced here as Table 11. As a matter of background, run number 11 was the illumination of a 36 m^2 trihedral mounted on a tripod, the pulse repetition frequency was 500 Hz, the narrowband carrier was employed, the threshold level was set at 20, and the scenario was a multipath interference-free, high signal-to-background environment.

The first six lines on the output reproduce the header data that identify the run. The "Average" and "Standard Deviation" of correlator output represents these measures, on a subpulse basis, of the compressed (output) waveforms over the number of pulses specified. The next two lines report percentage of valid received pulses and total number of pulses processed. Since the PRF was 500 Hz and 576 pulses were processed for this run, just over a second's worth of data was processed in these analyses.

The "Average Peak Correlator Output" and attendant standard deviation are computed by considering the peak output resulting from each received pulse. The aforementioned averages and standard deviations were computed on a position, rather than value, basis. Thus, if the peak were to occur in varying positions from pulse-to-pulse, "Average Peak" and average of any single range bin would be different. However, for run number 11 that was not the case as can be garnered from "Average Peak Position" and "Standard Deviation of Peak Position" on the next lines in Table 11.

The mean and standard deviation of any particular bin may be computed and exhibited as in "Average Output of Bin Number ___ is:," thus obviating the necessity for the first two outputs (Average Correlator Output and Standard Deviation of Correlator Output) if the output from only one range bin is of interest.

TABLE 11. SAMPLE DIGITAL ANALYSIS OUTPUT

```
                                 PRF:   500.
          IPAR DISPLAY SWITCH POSITION:  DIGITAL TAPE
             IPAR COMPUTER FLAG SWITCH:   CLR
                         IPAR RUN/TEST:   RUN
                      SUB-PULSE LENGTH:    40
                  TRANSMIT CODE LENGTH:  32
****************************
ANALYSIS OF RUN NUMBER   11
****************************

******* STATISTICS REPORT *******

AVERAGE CORRELATOR OUTPUT (VOLTS):
  -.9 -1.9  -.7   .4   .7  1.7   .0 -2.2  1.6  2.7  1.3   .8  4.5   .5 -1.7 -2.3   .4  3.2   .1 -2.8  -.6
  2.7   .5 -1.6   .1  1.9 -1.0   .1   .4  1.8  2.9-29.3  -.2  1.8  2.5  -.6 -1.1  3.0  2.2 -2.6   .1  1.2
 -1.9 -1.1 -2.1  4.8  1.6 -2.6  -.1  -.6  5.8  1.4   .2  1.2  3.9 -1.0  -.3  1.0  1.7   .8  -.1 -1.1 -1.0

STANDARD DEVIATION OF CORRELATOR OUTPUT:
   .3   .3   .5   .7   .6  1.0  1.1   .8  1.5   .8  1.1  1.2  1.2  1.0   .9   .7   .9  1.1  1.2   .8   .7
  1.0   .8   .9  1.0  1.0   .9   .9  1.2  1.1  1.3  2.1  1.2   .9  1.0   .8   .9   .7   .7   .6   .6   .5
   .6   .6   .7   .6   .6   .6   .5   .6   .5   .6   .5   .5   .6   .5   .4   .5   .4   .4   .3   .3   .0

100.0 PERCENT OF THE PULSES WERE GOOD
  576 PULSES WERE PROCESSED

AVERAGE PEAK CORRELATOR OUTPUT:-29.3 VOLTS
STANDARD DEVIATION OF PEAK OUTPUT:  2.1

AVERAGE PEAK POSITION:    .0
STANDARD DEVIATION:     .0

AVERAGE OUTPUT OF BIN NUMBER 16 IS  -2.6 VOLTS
STANDARD DEVIATION:     .6


AVERAGE PEAK SIDELOBE LEVEL:-14.8 DB           PSL OF  AVERAGE OUTPUT:-14.0 DB
STANDARD DEVIATION:     .8

AVERAGE INTEGRATED SIDELOBE LEVEL: -5.7 DB      ISL OF  AVERAGE OUTPUT: -5.6 DB
STANDARD DEVIATION:     .5

AVERAGE LOSS IN PROCESSING GAIN:  -.8 DB       LPG OF  AVERAGE OUTPUT:  -.8 DB
STANDARD DEVIATION:     .6
```

TABLE 11. SAMPLE DIGITAL ANALYSIS OUTPUT. (Continued)

```
******* HISTOGRAM REPORT *******

HISTOGRAM OF PEAK (PEAK IS IN VOLTS):
PEAK:          -32 -31 -30 -29 -28 -27 -26 -25 -24 -23 -22 -21 -20 -19 -18 -17 -16 -15 -14 -13 -12 -11
OCCURENCES:     54 162 108  54 114  12  36  18   6  12   0   0   0   0   0   0   0   0   0   0   0   0

PEAK:          -10  -9  -8  -7  -6  -5  -4  -3  -2  -1   0   1   2   3   4   5   6   7   8   9  10  11
OCCURENCES:      0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0

PEAK:           12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32
OCCURENCES:      0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0


HISTOGRAM OF PEAK POSTTION:
POSITION  :    -31 -30 -29 -28 -27 -26 -25 -24 -23 -22 -21 -20 -19 -18 -17 -16 -15 -14 -13 -12 -11
OCCURENCES:      0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0

POSITION  :    -10  -9  -8  -7  -6  -5  -4  -3  -2  -1   0   1   2   3   4   5   6   7   8   9  10
OCCURENCES:      0   0   0   0   0   0   0   0   0   0 576   0   0   0   0   0   0   0   0   0   0

POSITION  :     11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31
OCCURENCES:      0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0


HISTOGRAM OF BIN NUMBER 16  (OUTPUT IS IN VOLTS):
OUTPUT:        -32 -31 -30 -29 -28 -27 -26 -25 -24 -23 -22 -21 -20 -19 -18 -17 -16 -15 -14 -13 -12 -11
OCCURENCES:      0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0

OUTPUT:        -10  -9  -8  -7  -6  -5  -4  -3  -2  -1   0   1   2   3   4   5   6   7   8   9  10  11
OCCURENCES:      0   0   0   0   0   0   0   0 372 180  24   0   0   0   0   0   0   0   0   0   0   0

OUTPUT:         12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32
OCCURENCES:      0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
```

The measures peak sidelobe level (PSL), integrated sidelobe level (ISL) and loss in processing gain (LPG) are standard measures-of-goodness for pulse compression codes.(8) These measures are computed here on an ensemble basis in two different ways. "Average ..." represents the process of computing each of these measures on a pulse by pulse basis, then computing the average of these measures over the ensemble of returns. "... of Average" represents the process of computing the average correlator output over the ensemble of all pulses, then computing the measures-of-goodness based on this "average" receive pulse. The "Average Measures" or the "Measures of Average" will be the more meaningful depending on the particular experiment being considered. For example, for moving targets such as were utilized in the multipath-interference-over-sea experiments, the range gate in which the target of interest appears changes during a run, and thus the "Average Measures" of PSL, ISL, and LPG are most meaningful for analysis of these data. On the other hand, the "Measures of Average" will be most meaningful for analysis of the non-extended clutter data, since changes in position of the peak return is due to decorrelation of the receive signal as opposed to movement in range of the reflecting surface.

The histogram data that appear in the continuation of Table 7.1 give the actual distributions of peak voltage, peak position, and voltage level from a particular bin over the full range of pulses processed from the subject run. These data provide more detailed information than the statistical summary parameters (average and standard deviation) described earlier. During the next phase of this research, plotting routines will be added to the analysis program so that these histograms may be graphed for more detailed analysis.

7.3 SIGNAL-TO-NOISE DOCUMENTATION FROM FM TAPE

To support the digital data reduction, a portion of the FM data was analyzed using Honeywell SAI-48 correlator. The horizontal and vertical amplitude channels of the FM tape recorder were fed into the correlator, which was adjusted to produce probability amplitude density functions. A photograph of this reduction system block diagram appears in Figure 16.

The first step in producing comparable received amplitudes was to determine the receiver response function. This was done for each channel for each day of data to be analyzed. By passing the receiver amplitude calibration run (Section 4.1.3) through the correlator, a set of amplitudes as a function of injected power were produced. An example of a typical receiver response curve appears as Figure 17(a).

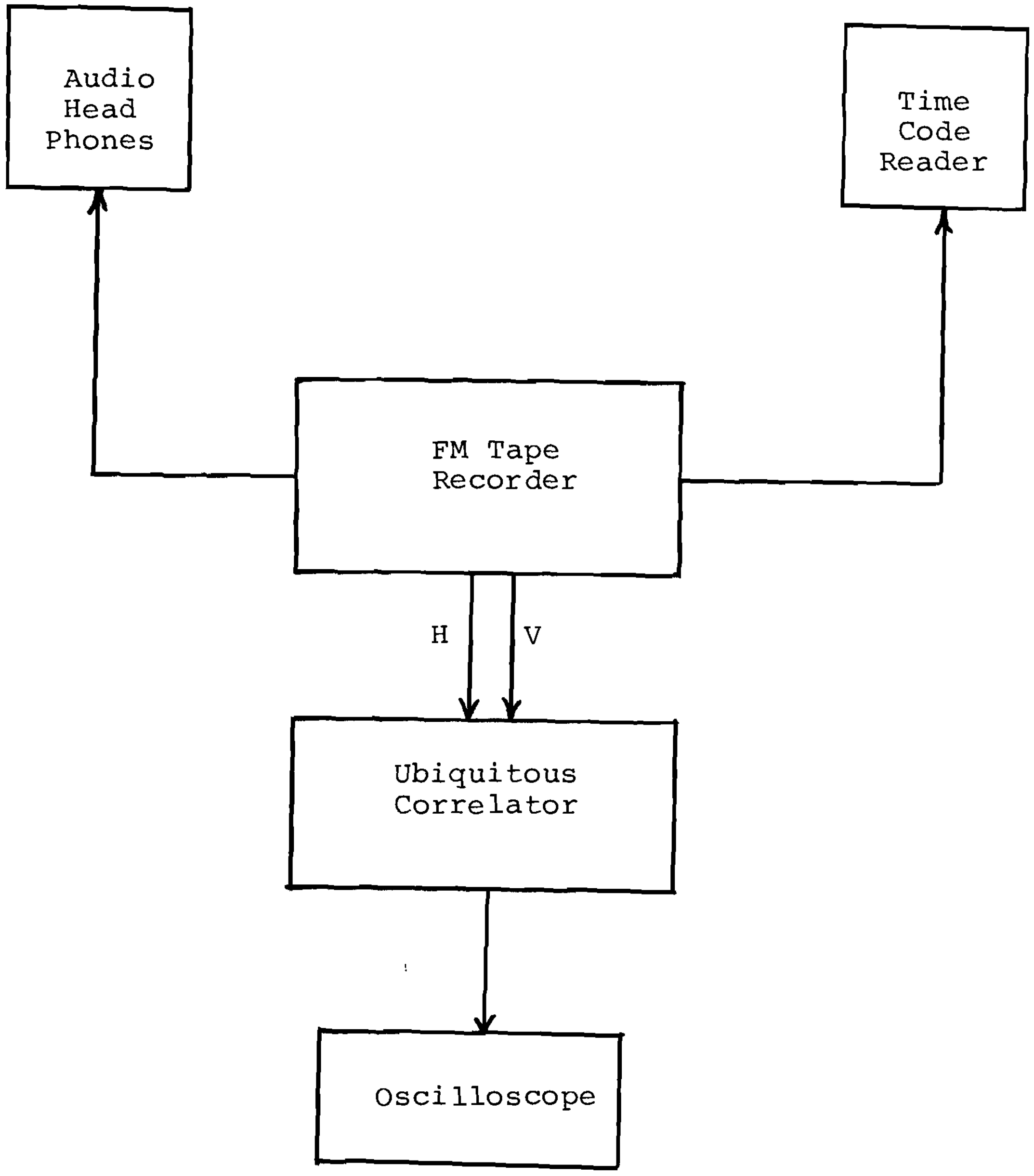
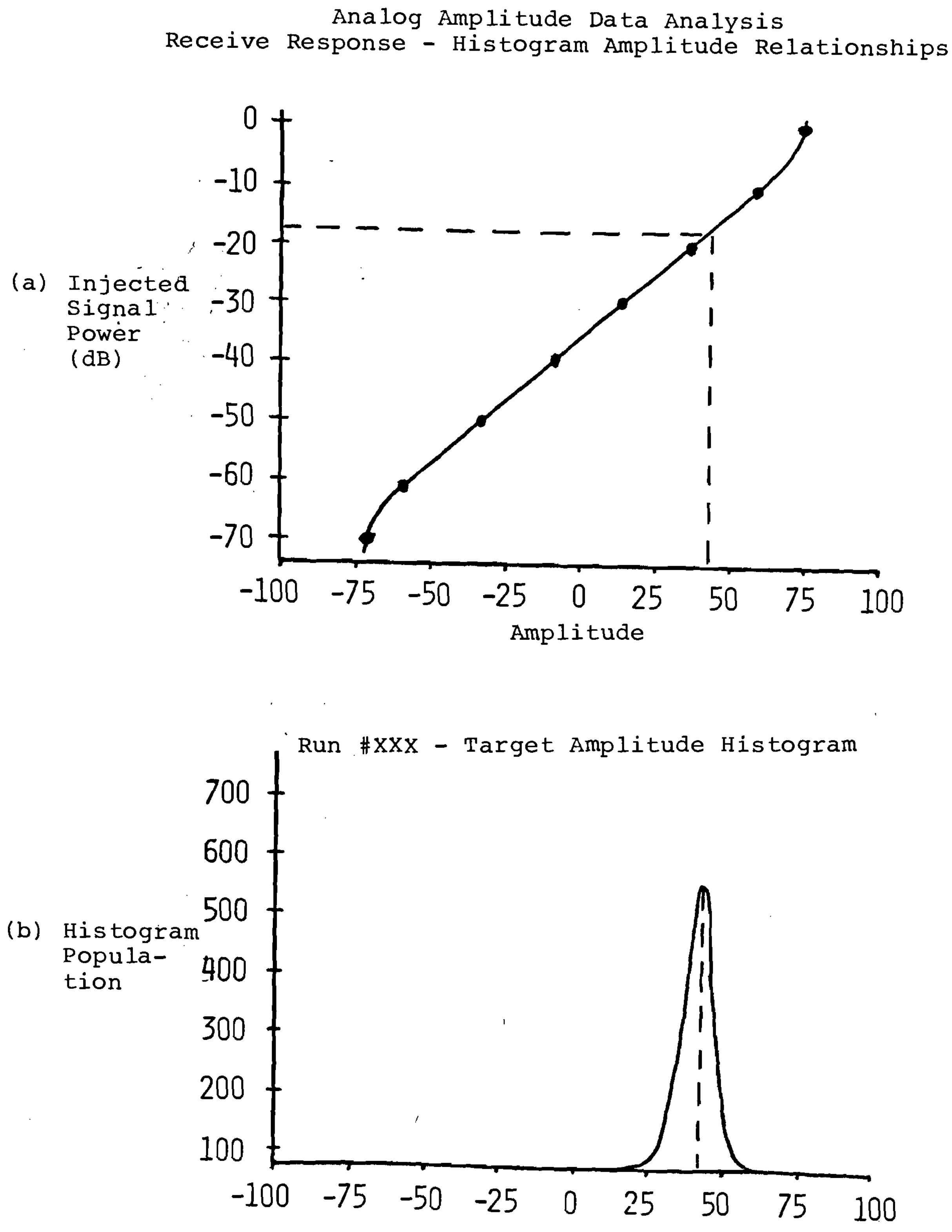


Figure 16. FM data reduction system block diagram.



Analog Amplitude Data Analysis
Receive Response - Histogram Amplitude Relationships

Run #XXX - Target Amplitude Histogram

Results in σ_t = -18 dB Relative To Injected Power

Figure 17. Examples of receiver response function and target amplitude distribution.

Once the correlator amplitude had been calibrated, the data runs were processed to produce probability amplitude density functions. An example of the data output appears as Figure 17(b). The amplitude position of the peak of this distribution can be related to the associated receiver response function to produce a target amplitude relative to known injected power. In this way, the target return from a variety of data runs can be compared directly.

SECTION 8
PROGRAM RESULTS

The major thrusts of this program were to collect, reduce, and analyze IPAR backscatter data from a variety of targets in various scenarios to quantify numerous theoretically predicted properties of the IPAR system. However, these activities were not the only program objectives. In particular, a simple but effective model of the IPAR process, developed to aid the researchers in developing the "complex target" experiments, led to other interesting results. In addition, the IPAR concept was publicized to the radar community in a sequence of professional presentations. These results are summarized in the subsections below.

8.1 DATA COLLECTION, REDUCTION, AND ANALYSIS

These efforts have been fully documented in Sections 2 through 7 of this report; thus a summary suffices here.

Over 1,300 runs of digital and analog IPAR backscatter data were recorded. These data were collected at four different test sites, each site being especially suited for the experiments conducted there.

These data will provide the basis for documentation, quantification, and analysis of IPAR's properties with respect to (1) compression on simple targets, (2) performance in a multipath interference environment, (3) performance in a clutter environment, (4) resolution on extended targets, and (5) compression on complex targets.

Two-thirds of the digital data have been compacted and written to computer disk, and the capability exists for doing the same with the remaining digital and to-be-digitized analog data. The data that have been transferred to disk have been documented in terms of the S/N ratios at which they were collected. These data have also been analyzed with respect to received-waveform pulse compression characteristics and variability. Again, the capability exists for doing the same with the remaining digital data.

8.2 MODELING AND THEORETICAL ANALYSES OF THE IPAR PROCESS

The electromagnetic propagation equations for right and left circularly polarized transmission pulses were derived and utilized as the basis for a computer analysis of

IPAR's response to backscatter from objects composed of two simple (dihedral or trihedral), major reflectors in a single range resolution cell. This modeling effort reassured the researchers that unless the two scatterers were almost identical in cross section or the utilized threshold setting was quite high, the target would compress quite well. These results seemed to be reproduced in the field.

While attempting to extend this model to include IPAR's response to extended targets (major reflectors spread over more than one resolution cell in range), it was noticed that, in fact, unless received amplitude was included as an input to the pulse compression correlator, the process of correlating the received signal with the filter matched to the transmit pulse is not a linear process in range. Thus, the current system configuration for IPAR represents a valid implementation of pulse compression coding on polarization for a communications system or for radar astronomy (where the scenario is a single range bin of backscatter surrounded by absence of backscatter from surrounding range bins). However, for most classical defense-oriented applications of radar (such as detection of extended targets in a clutter environment), the current configuration is inappropriate. These conclusions are based on the thesis that pulse compression processing must be linear in range (that is, superposition must hold) to be effective when returns are from extended targets. This thesis is explained below.

The process of pulse compression is an attempt to achieve a range resolution finer than the natural range resolution obtainable with a pulsed system.(7) If a radar transmit pulse is of duration T_0 seconds, then the backscatter from two objects separated in range by $1/2\ cT_0$ or more will be separated in time and thus will be resolvable in range (time). Pulse compression is achieved by modulating the transmit pulse at a bandwidth of $1/\tau$, $\tau \leq T_0$, according to some code, and then compressing the radar backscatter by passing it through a correlation filter (for many applications) matched to the transmitted code. In response to the backscatter from a point target, the output from the matched filter is a signal of duration $2T_0$ with a modulation, caused by the interaction of the coded pulse and its matched filter, whose bandwidth is $1/\tau$. If the code is chosen properly for pulse compression purposes, this output may look something like the output depicted in Figure 18. The relatively large return between $T = -\tau$ and $T = \tau$ emulates the return one would expect from a simple transmit pulse of duration τ, and it thus represents the output of the resolution cell of interest. The smaller returns found between $T = -T_0$ and $T = T_0$ represent responses that would not be present if the transmit pulse were indeed a short pulse of duration τ and thus represent undesirable

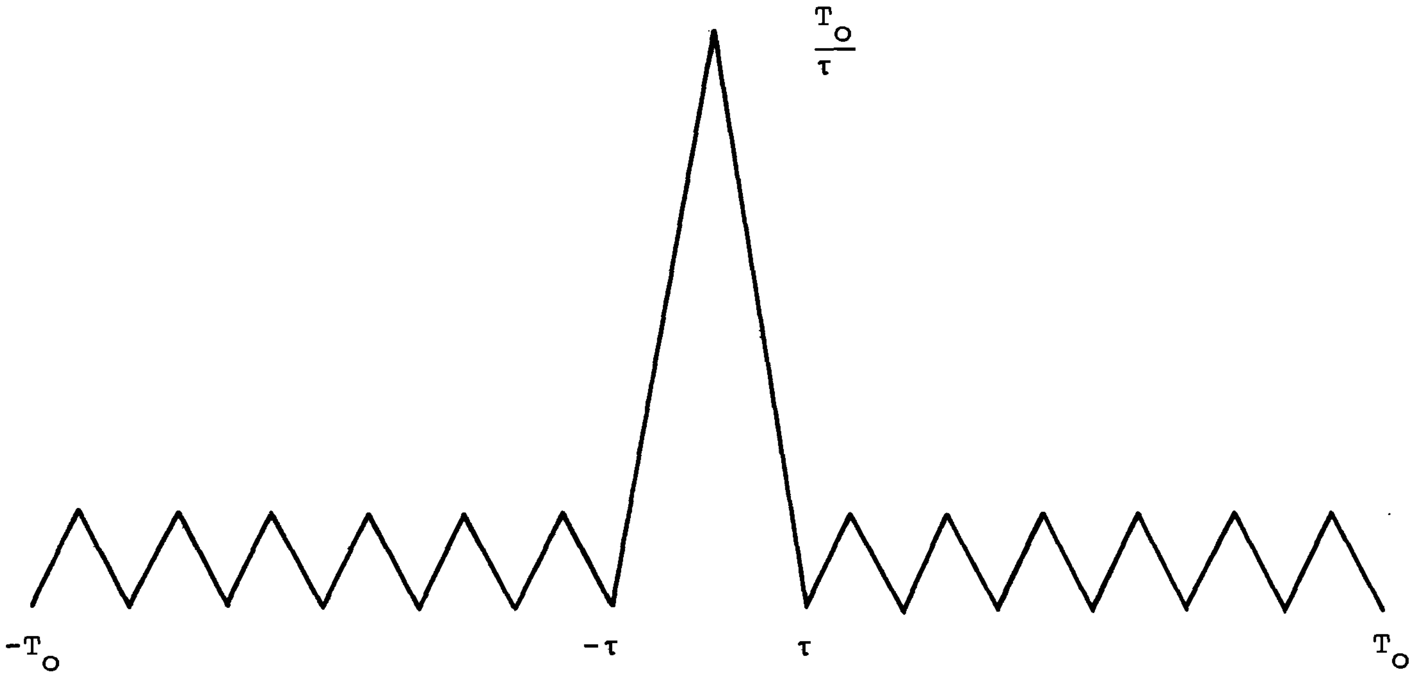


Figure 18. Pulse-compressed output.

transients in the attempt to emulate the behavior of a short pulse system by transmitting a long pulse and then pulse-compressing. These transients are known as range sidelobes.

The effects of range sidelobes, that is, the degree to which the transient responses from one range bin affect neighboring range bins, can be predicted and computed if the pulse compression process is linear in range. The argument for this can best be made by referring to Figure 19. The two five bit signals depicted entering two copies of a matched filter are identical, except that they are shown arriving delayed by three subpulse widths (3τ) with respect to each other and the latter is assumed to be twice as strong. Assuming that these signals are being generated by two separate experiments, the responses in each of the experiments can be depicted as the outputs from copies of the system's matched filter, as depicted in the figure. The result is a pair of outputs that are identical, except for the 2^{τ} delay between them. Now reconsider the two postulated return signals on the right side of the figure. This time assume that they actually were generated by two point scatterers separated by $\frac{1}{2}c\ (2^{\tau})$ units in range, so that they add in space as two complex signals. The result then is that a 7-bit return, as shown in the lower right-hand corner of the figure, consisting of the complex addition of the two signals is received and processed through the system's matched filter. The output shown on the lower-left of the figure is the actual result of passing this "combined" signal through the matched filter. Finally, by linearity, the response of the matched filter to the overlapped returns is precisely the sum of the outputs of the matched filter to the two component inputs taken separately! That is, the compressed return from an extended target may be computed as the sum (or superposition) of the compressed returns from each of the non-extended reflectors that make up the complex target.

Thus, the sidelobe structure generated by any one single return occurrence can be utilized to measure and predict the contribution of any one range cell to its neighbors, even when the simple return is part of an extended target, as long as the pulse compression system is linear. It follows that measures such as

$$\text{PSL} = \text{Peak Sidelobe Level} = \frac{\text{Maximum Sidelobe}}{\text{Peak Signal}} \text{ in dB}$$

and

$$\text{ISL} = \text{Integrated Sidelobe Level} = \frac{\text{Total Power in Sidelobe}}{\text{Power in Peak Signal}} \text{ in dB}$$

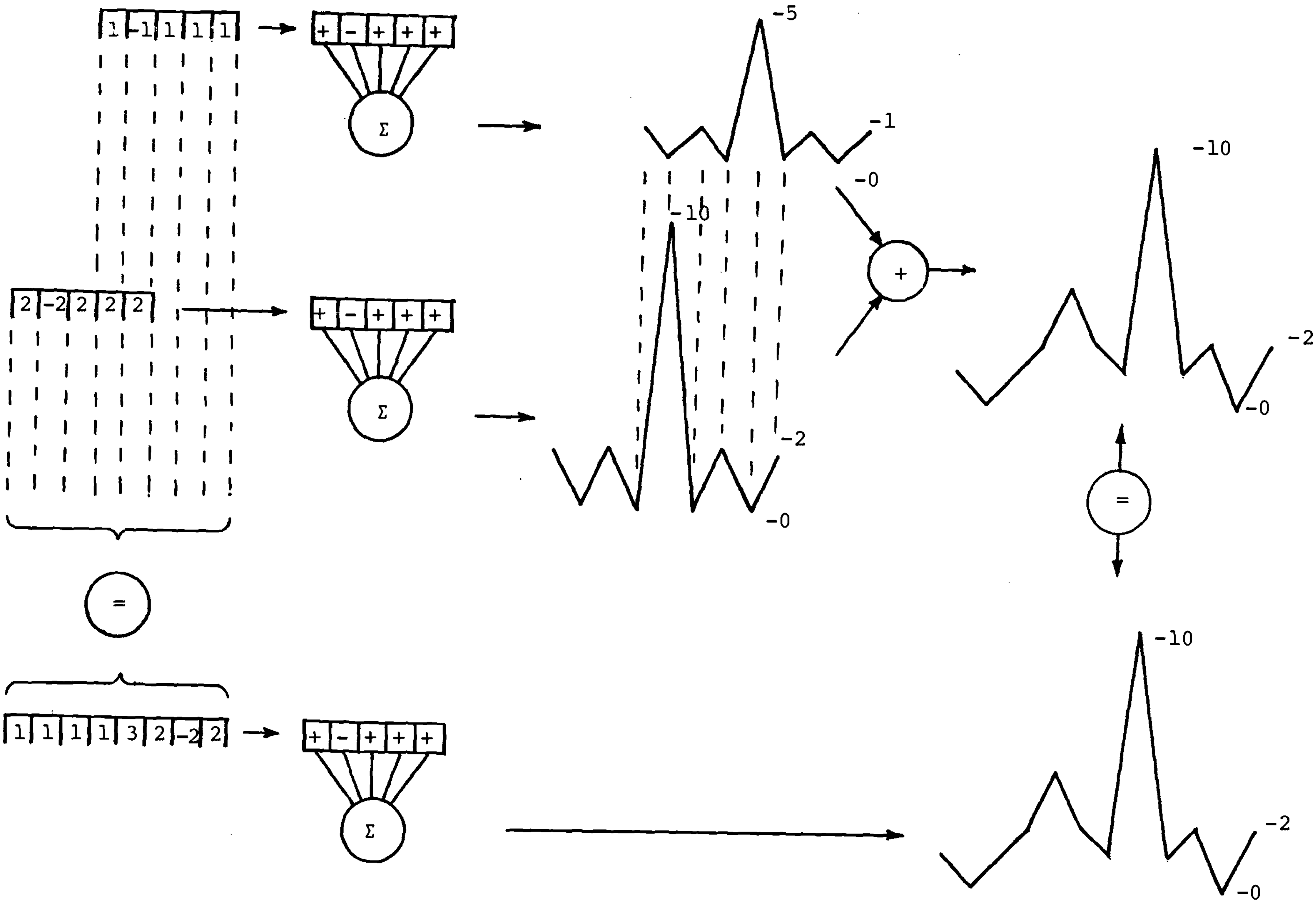


Figure 19. Response from an amplitude-dependent pulse compression system.

for a single occurrence of a pulse compression code are accurate measures of the contribution of each range bin to all of its neighboring range bins containing returns from an extended target as long as the compression processing is linear in range.

This is, in general, not true for systems whose pulse compression processing is non-linear in range. In particular, the current IPAR system, which does not utilize the amplitude of the received signals in its compression processing, is an example of such a non-linear pulse compression system. In fact, this implementation of IPAR leads to precisely the same effects as would be encountered in a binary phase-coded pulse compression system that did not utilize amplitude. Figure 20 represents the same scenario as that depicted in Figure 19, except that here it is assumed that amplitude is not taken into account in the compression process. Clearly, the principle of superposition does not hold here for the separate compressed returns and the compression of the extended return. Furthermore, the undesirable characteristic of "capture," where the compressed return from the larger reflector masks the compressed return from the smaller reflector is also present. Recall that this was not the case for the linear system configuration.

As a result of these findings, A-IPAR, the advanced design IPAR system, is being designed to pulse compress process on both relative phase and amplitude.

8.3 DELINEATION OF PRACTICAL APPLICATIONS

The real-world applications for which IPAR seemed to hold the most promise determined which experiments were to be performed during the data collection program. As a result of the consideration of applications, two research papers were presented and published.

The potential advantages of IPAR-like processing for use in a stationary-target identification radar system were described in a presentation at the Combat Identification Systems Conference, 1983, and the accompanying paper, "IPAR As A Target Identification Radar," was published in the conference proceedings.(8)

The characteristics of IPAR processing that give it unique potential in the area of lowered probability of intercept (LPI) operation were described at a technical session of the Microwave Systems and Technology Conference, 1983. This presentation appears in the proceedings of this conference as, "The Intrapulse Polarization Agile Radar."(9)

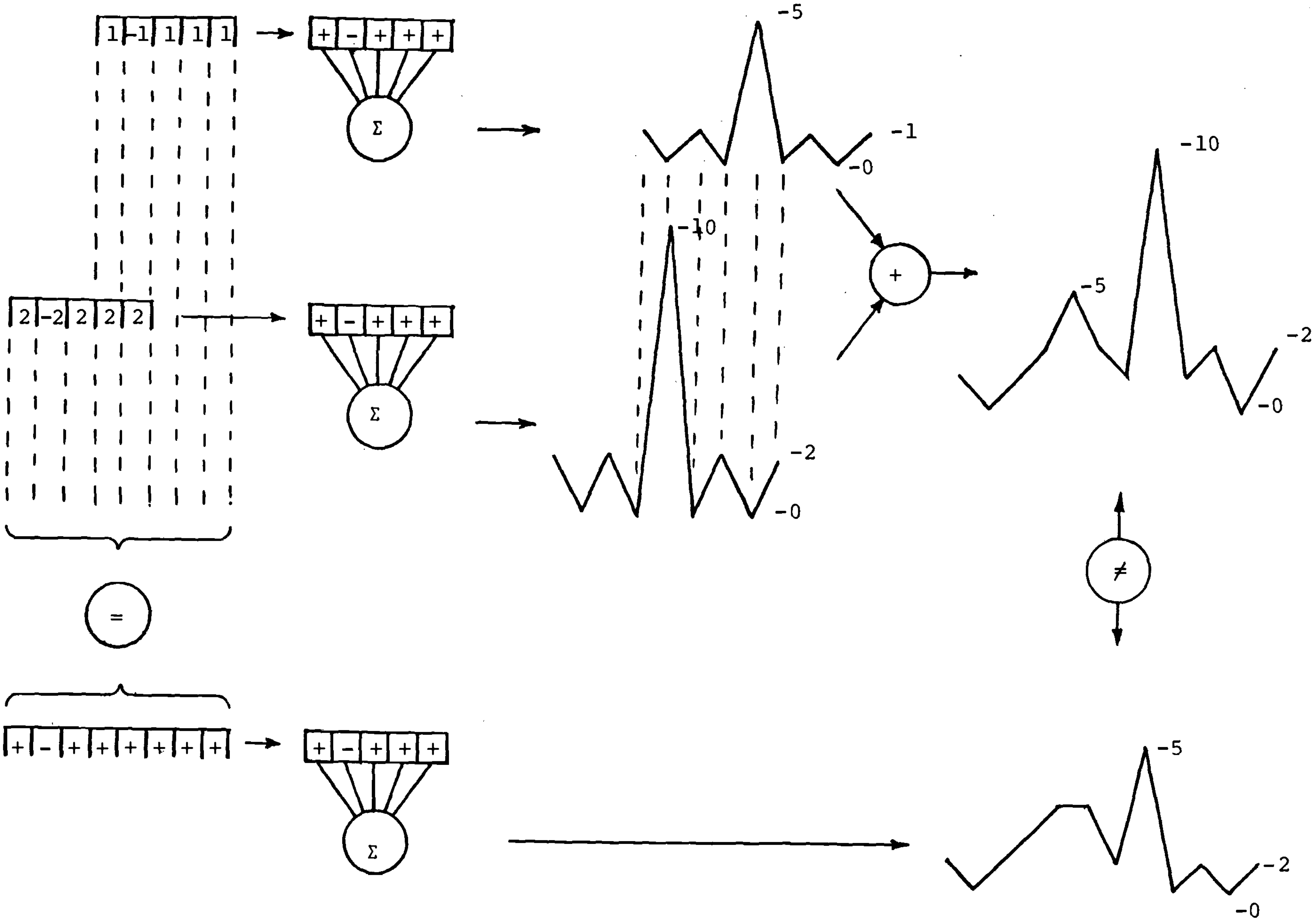


Figure 20. Response from an amplitude-independent pulse compression system.

The earliest presentation describing the IPAR system at a professional technical conference was the paper "Intrapulse Polarization Agile Radar," presented at RADAR '82 and published in that conference proceedings.(10)

SECTION 9

SUMMARY AND RECOMMENDATIONS

9.1 SUMMARY

IPAR data were generated, recorded, and partially analyzed. Twenty-two digital tapes and 17 FM tapes of data from over 1,300 system runs were generated. Analysis of these data will provide a basis for understanding and quantifying IPAR performance with respect to: (1) simple targets, (2) complex targets, (3) clutter, (4) targets in clutter, and (5) targets in multipath interference.

Approximately two-thirds (480 system runs) of the recorded digital data were compacted, stored on disk, and analyzed with respect to the statistical characteristics of the compressed returns resulting from each run. The results of this analysis have been documented in hard copy as well as stored on disk for future analyses. Each of these analyzed digital runs has been documented in amplitude through the reduction and compilation of the corresponding FM data. In addition, theoretical analysis established the need for incorporation of received amplitude information in the pulse compression process.

The IPAR process has been officially introduced to the radar community through a sequence of three technical presentations at professional conferences.(9,10,11) These presentations not only describe the IPAR concept, but also discuss IPAR's potentials in the areas of target identification by radar(9) and generation of lowered probability of intercept (LPI) waveforms.(10)

9.2 RECOMMENDATIONS FOR CONTINUED RESEARCH

The efficacy of IPAR processing under various conditions has been demonstrated, and documented IPAR performance data have been generated under these conditions. The next phase of the program should be to undertake a detailed analysis and evaluation of these data.

The analysis will begin with reduction and digitization of the recorded FM data. These digitized data will then be read to disk from which they can be conveniently integrated with the collected digital data. Each set of experimental data will then require special processing and consideration so that the particular features of interest for those data can be extracted. The aim of the analyses will be to quantify IPAR's interaction with, and performance in, the various experimental scenarios documented during the data collection program.

During the data collection, it was noticed that IPAR's response to targets was consistently distinguishable from its response to clutter. The next phase of the program should include development of an appropriate target/clutter discrimination algorithm and testing of the algorithm on the collected data. This will result in quantification of IPAR's discrimination potential.

Finally, it is believed that sufficiently much has been learned about IPAR that it is now possible to develop a sophisticated computer model of the IPAR process which could include (1) models of various carriers (narrowband, frequency agile, and noise) of variable spectral width and (2) the capability to emulate the return from complex scatterers composed of many simple objects (dihedrals, trihedrals, plates, wires, etc.) at arbitrary orientations. Such a model would prove invaluable for probing and, therefore, enhancing understanding of how IPAR might respond to real, highly complex objects.

REFERENCES

1. J. L. Eaves, et al. "An Airborne Investigation of Polarization Diversity for Target Discriminating," Proceedings of the 19th Annual Tri-Service Radar Symposium, July 1973.
2. J. L. Eaves, et al. "Investigation of Target Enhancement Techniques," Final Report, Air Force Contract No. F33615-71-C-1612, Georgia Institute of Technology, October 1974.
3. J. L. Eaves, "Flight Evaluation of Stationary Target Indication Techniques," Proceedings of the 22nd Annual Tri-Service Radar Symposium, July 1976.
4. L. C. Bomar, et al. "Evaluation of Selected Target Discrimination Techniques," 8th DARPA/Tri-Service Millimeter Wave Conference, April 1979.
5. B. C. Appling, E. S. Sjoberg, and E. E. Martin, "Intrapulse Polarization Agile Radar," Final Report, SCEEE Contract No. N00024-78-C-5338, SCEEE-NAVSEA/79-2, Georgia Institute of Technology, July 1981.
6. M. N. Cohen, E. E. Martin, and E. S. Sjoberg, "Intrapulse Polarization Agile Radar Development Program," Final Report, Contract No. N00024-78-C-5338, SCEEE/81-2, Georgia Institute of Technology, December 1981.
7. M. N. Cohen, "Pulse Compression," Principles of Modern Radar, Georgia Institute of Technology, February 1983.
8. M. N. Cohen and E. S. Sjoberg, "IPAR as a Target Identification Radar," Combat Identification System Conference (CISC) '83, Monterey, California, February 1983.
9. M. N. Cohen, E. E. Martin, and E. S. Sjoberg, "The Intrapulse Polarization Agile Radar," Microwave Systems and Technology (MSAT) Conference '83, Washington, D.C., March 1983.
10. M. N. Cohen and E. S. Sjoberg, "Intrapulse Polarization Agile Radar," RADAR '82, London, England, October 1982.

(THIS PAGE INTENTIONALLY LEFT BLANK)

APPENDIX A

TAPE-TO-DISK TRANSFER ROUTINE

```
$JOB A3366000 COHEN,MN
$OPTION 1 2 3 4 5 17
$EXECUTE FORT77
      PROGRAM DISCLD
C
C              ******************************
C              *                            *
C              *       PROGRAM DISCLD       *
C              *                            *
C              ******************************
C
      LOGICAL TWOEOF, NEWRANGE, ALL
      COMMON / IOUNIT / NTAP, NSLO, NHDR, NDTA
      COMMON / TRDATA / NTAPES   , NRUNS    , NRUNSER , NRUN     ,
     &                  NRUNERR , NTRECS   , NTRECSER, NTREC    ,
     &                  NTRECERR, NDRECS   , NDRECSER, NDREC    ,
     &                  NDRECERR, NTAPEN   , NRUNB    , NRUNE    ,
     &                  NBEGIN   , NEND     , NRUNTM   , NDRWE
      COMMON / DVSTAT / NSTATT, NSTATW, NSTATH, NSTATD, NSTATC, NSTATI,
     &                  NSTATHO, NSTATDO, NSTATHC, NSTATDC, NMOUNT,
     &                  NDSMNT
      COMMON / MISC   / CTIMER, CTIMEA, PLSPTREC, PLSPDREC, TWOEOF,
     &                  NEWRANGE, ALL, MTIME(3), MDATE(3), NROLL, BELL
      CALL OUTPUT (1)
      CALL FILES (1, IERR)
      IF (IERR .EQ. 2) THEN
100      CONTINUE
         CALL TAPEINIT (ITAPEN, IDREC, IERR)
         ITFILE = 0
         IF (ITAPEN .GT. 0 .AND. IERR .EQ. 2) THEN
200         CONTINUE
            CALL RUNSLCT (IRUNB, IRUNE, IDREC, IERR)
            IF (IERR .EQ. 2) THEN
300            CONTINUE
               CALL READER (1, ITREC, IERR)
               IF (IERR .EQ. 2) THEN
                  ITFILE = ITFILE + 1
                  CALL HDRDEC (IRUNB, IRUNE, IRUN, PRF, IDREC, IERR)
                  IF (IERR .EQ. 2) THEN
                     IBRANCH = NRUNTEST (IRUNB, IRUNE, IRUN, IERR)
                     IF (IBRANCH .EQ. 4 .AND. IERR .EQ. 2) THEN
                        CALL RUNTIME (RTIME, PRF, IRUN, ITIME, IERR)
                        IF (RTIME .GT. 0.0 .AND. IERR .EQ. 2) THEN
400                        CONTINUE
                           CALL READER (2, ITREC, IERR)
                           IF (IERR .EQ. 2) THEN
                              CALL DUMPDATA (ITREC, IDREC, ITIME, IRUN,
     &                                       PRF, IBRANCH, IERR)
                           ELSE
                              CALL DATACHEK (ITREC, IDREC, ITIME, IRUN,
     &                                       PRF, IBRANCH, IERR)
                           END IF
                           IF (IBRANCH.EQ.4 .AND. IERR.EQ.2) GO TO 400
                        ELSE IF (RTIME .EQ. 0.0 .AND. IERR .EQ. 2) THEN
                           IBRANCH = 3
                        ELSE
                           IBRANCH = NBRANCH (IERR)
                        END IF
                     ELSE IF (IERR .NE. 2) THEN
                        IBRANCH = NBRANCH (IERR)
                     END IF
                  ELSE
                     IBRANCH = NBRANCH (IERR)
                  END IF
               ELSE IF (IERR .EQ. 3) THEN
                  IBRANCH = NTAPECHK (INDEOF, ITFILE, IERR)
               ELSE
                  ITFILE  = ITFILE + 1
                  IBRANCH = NBRANCH (IERR)
               END IF
               IF (IBRANCH .EQ. 3 .AND. IERR .EQ. 2) GO TO 300
            ELSE
               IBRANCH = NBRANCH (IERR)
            END IF
            IF (IBRANCH .EQ. 2 .AND. IERR .EQ. 2) GO TO 200
         ELSE
            IBRANCH = 0
         END IF
         REWIND (NTAP, IOSTAT = NSTATRW)
         IF (NSTATRW .NE. 0) CALL OUTPUT (15)
         NRUNS     = NRUNS     + NRUN
         NRUNSER   = NRUNSER   + NRUNERR
         NTRECS    = NTRECS    + NTREC
         NTRECSER  = NTRECSER  + NTRECERR
         NDRECS    = NDRECS    + NDREC
         NDRECSER  = NDRECSER  + NDRECERR
         IF (NTAPEN   .GT. 0) CALL OUTPUT (4)
         IF (IBRANCH .EQ. 1) GO TO 100
         CALL OUTPUT (30)
      END IF
      CALL FILES (2, IERR)
      CALL OUTPUT (28)
      STOP
      END
      SUBROUTINE RUNSLCT (IRUNB, IRUNE, IDREC, IERR)
C
C             *******************************
C             *                             *
C             *    SUBROUTINE RUNSLCT       *
C             *                             *
C             *******************************
C
      LOGICAL TWOEOF, NEWRANGE, ALL, ERROR
      COMMON / DVSTAT / NSTATT, NSTATW, NSTATH, NSTATD, NSTATC, NSTATI,
     &                  NSTATHD, NSTATDD, NSTATHC, NSTATDC, NMOUNT,
     &                  NDSMNT
      COMMON / TRDATA / NTAPES  , NRUNS   , NRUNSER , NRUN    ,
     &                  NRUNERR , NTRECS  , NTRECSER, NTREC   ,
     &                  NTRECERR, NDRECS  , NDRECSER, NDREC   ,
     &                  NDRECERR, NTAPEN  , NRUNB   , NRUNE   ,
     &                  NBEGIN  , NEND    , NRUNTM  , NDRWE
      COMMON / MISC   / CTIMER, CTIMEA, PLSPTREC, PLSPDREC, INDEOF,
     &                  NEWRANGE, ALL, MTIME(3), MDATE(3), NROLL, BELL
100   CONTINUE
```

```
      ERROR = .FALSE.
      WRITE ('UT', 200, IOSTAT = NSTATC) BELL
      IF (NSTATC .NE. 0) THEN
         IERR = 9
         IRUNB = IRUNE = 0
         CALL OUTPUT (3)
      ELSE
         READ ('UT', *, IOSTAT = NSTATC) IRUNB, IRUNE
         IF (NSTATC .EQ. 0) THEN
            IF ((IRUNB .LE. 0) .OR. (IRUNE .LE. 0)) THEN
               IERR = -2
               CALL OUTPUT (33)
            ELSE IF (IRUNB .GT. IRUNE) THEN
               ERROR = .TRUE.
            ELSE
               NRUNB = IRUNB
               NRUNE = IRUNE
               IF (IRUNB .EQ. 9999 .AND. IRUNE .EQ. 9999) THEN
                  ALL = .TRUE.
                  CALL OUTPUT (6)
               ELSE
                  ALL = .FALSE.
                  CALL OUTPUT (6)
                  CALL DISKCHEK (IRUNB, IDREC, 1, IERT)
                  IF (IERT .EQ. 22) THEN
                     ERROR = .TRUE.
                     IERT  = 0
                  END IF
               END IF
            END IF
         ELSE
            IF (NSTATC .NE. 15) THEN
               IERR = 10
               IRUNB = IRUNE = 0
               CALL OUTPUT (2)
            ELSE
               ERROR = .TRUE.
            END IF
         END IF
      END IF
      IF (ERROR) GO TO 100
      NEWRANGE = .FALSE.
      RETURN
200   FORMAT(/A1,'ENTER>  START AND STOP RUN NUMBERS IN INTEGER ',
     &       'FORMAT'/9X,'(ENTER 9999 FOR BOTH TO PROCESS ALL RUNS ON ',
     &       'THE TAPE)'/6X,'>  NON-POSITIVE VALUE FOR EITHER TO SKIP ',
     &       'THE REST OF THE TAPE')
      END
C****
C****
      SUBROUTINE DUMPDATA (ITREC, IDREC, ITIME, IRUN, PRF, IBRANCH,IERR)
C
C               ******************************
C               *                            *
C               *    SUBROUTINE DUMPDATA     *
C               *                            *
C               ******************************
```

```
C
      COMMON / TRDATA / NTAPES   , NRUNS    , NRUNSER , NRUN     ,
     &                  NRUNERR , NTRECS   , NTRECSER, NTREC    ,
     &                  NTRECERR, NDRECS   , NDRECSER, NDREC    ,
     &                  NDRECERR, NTAPEN   , NRUNB   , NRUNE    ,
     &                  NBEGIN  , NEND     , NRUNTM  , NDRWE
      IF (ITREC .GT. 0) THEN
         CALL SWITCH
         CALL SHUFFLE (IERR)
         IF (IERR .EQ. 12) THEN
            NTRECERR = NTRECERR + 1
         ELSE IF (IERR .EQ. 2) THEN
            NTREC = NTREC + 1
            CALL TRANSFER (ITREC, IDREC, IRUN, IERR)
         END IF
      ELSE
         IERR = 11
         CALL OUTPUT (19)
      END IF
      CALL DATACHEK (ITREC, IDREC, ITIME, IRUN, PRF, IBRANCH, IERR)
      RETURN
      END
C****
C****
      SUBROUTINE DUMPHEDR (IDREC, IRUN, PRF, IERR )
C
C               ******************************
C               *                            *
C               *    SUBROUTINE DUMPHEDR     *
C               *                            *
C               ******************************
C
      LOGICAL TWOEOF, NEWRANGE, ALL
      COMMON / DATA    / IHDR4(3), IBUFD(256)
      COMMON / IOUNIT / NTAP, NSLO, NHDR, NDTA
      COMMON / DVSTAT / NSTATT, NSTATW, NSTATH, NSTATD, NSTATC, NSTATI,
     &                  NSTATHO, NSTATDO, NSTATHC, NSTATDC, NMOUNT,
     &                  NDSMNT
      COMMON / TRDATA / NTAPES   , NRUNS    , NRUNSER , NRUN     ,
     &                  NRUNERR , NTRECS   , NTRECSER, NTREC    ,
     &                  NTRECERR, NDRECS   , NDRECSER, NDREC    ,
     &                  NDRECERR, NTAPEN   , NRUNB   , NRUNE    ,
     &                  NBEGIN  , NEND     , NRUNTM  , NDRWE
      COMMON / MISC    / CTIMER, CTIMEA, PLSPTREC, PLSPDREC, TWOEOF,
     &                  NEWRANGE, ALL, MTIME(3), MDATE(3), NRULL, BELL
      COMMON / DIRECT / NDIR (1400,2)
      WRITE (NHDR, REC = IRUN, IOSTAT = NSTATH) IHDR4, NBEGIN, IDREC
      IF (NSTATH .EQ. 0) THEN
         NEND = IDREC
         NDIR(IRUN, 1) = NBEGIN
         NDIR(IRUN, 2) = NEND
         NRUN   = NRUN + 1
         NDREC  = NDREC + (IDREC - NBEGIN + 1)
         CTIMEA = FLOAT(NEND - NBEGIN + 1) * PLSPDREC / PRF
         CALL OUTPUT (20)
         NBEGIN = NEND + 1
      ELSE
```

```
         IERR = 7
         IDREC = NEND
         NRUNERR = NRUNERR + 1
         NDREC  = 0
         CTIMEA = 0.
         CALL OUTPUT (21)
      END IF
      RETURN
      END
C****
C****
C****
C****
      SUBROUTINE TAPEINIT (ITAPEN, IDREC, IERR)
C
C             ******************************
C             *                            *
C             *    SUBROUTINE TAPEINIT     *
C             *                            *
C             ******************************
C
      LOGICAL TWOEOF, NEWRANGE, ALL, ERROR
      COMMON / DVSTAT / NSTATT, NSTATW, NSTATH, NSTATD, NSTATC, NSTATI,
     &                  NSTATHD, NSTATDD, NSTATHC, NSTATDC, NMOUNT,
     &                  NDSMNT
      COMMON / TRDATA / NTAPES  , NRUNS   , NRUNSER , NRUN    ,
     &                  NRUNERR , NTRECS  , NTRECSER, NTREC   ,
     &                  NTRECERR, NDRECS  , NDRECSER, NDREC   ,
     &                  NDRECERR, NTAPEN  , NRUNB   , NRUNE   ,
     &                  NBEGIN  , NEND    , NRUNTM  , NDRWE
      COMMON / MISC   / CTIMER, CTIMEA, PLSPTREC, PLSPDREC, TWOEOF,
     &                  NEWRANGE, ALL, MTIME(3), MDATE(3), NROLL, BELL
      COMMON / DISK   / NSZHEDR, NSZDATA, NBPSCTR, NBPHREC, NBPDREC,
     &                  NBPPLSE, RHSPACE, NRMHEDR, RDSPACE, NRMDATA,
     &                  NRMPLSE
      IF (NTAPES .EQ. 0) THEN
100      CONTINUE
         ERROR = .FALSE.
         WRITE ('UT', 300, IOSTAT = NSTATC) BELL
         IF (NSTATC .EQ. 0) THEN
            READ ('UT', *, IOSTAT = NSTATC) IDREC
            IF (NSTATC .EQ. 0) THEN
               IF (IDREC .GT. 0) THEN
                  CALL DISKCHEK (0, IDREC, 2, IERR)
                  NBEGIN = IDREC
                  NEND   = IDREC - 1
                  PLSPDREC = FLOAT(NBPDREC) / FLOAT(NBPPLSE)
               ELSE
                  ERROR = .TRUE.
               END IF
            ELSE
               IF (NSTATC .NE. 15) THEN
                  IERR = 10
                  IDREC  = 0
                  ITAPEN = 0
                  CALL OUTPUT (2)
               ELSE
```

```
                  ERROR = .TRUE.
               END IF
            END IF
         ELSE
            IERR = 9
            IDREC  = 0
            ITAPEN = 0
            CALL OUTPUT (3)
         END IF
         IF (ERROR) GO TO 100
      END IF
      IF (NSTATC .EQ. 0 .AND. IERR .NE. 23) THEN
200      CONTINUE
         ERROR = .FALSE.
         WRITE ('UT', 400, IOSTAT = NSTATC) BELL, NTAPEN
         IF (NSTATC .EQ. 0) THEN
            READ ('UT', *, IOSTAT = NSTATC) ITAPEN
            IF (NSTATC .EQ. 0) THEN
               IERR   = 2
               NRUN   = NRUNERR  = 0
               NTREC  = NTRECERR = 0
               NDREC  = NDRECERR = 0
               NTAPEN = ITAPEN = MAX0 (ITAPEN, 0)
               IF (ITAPEN .GT. 0) THEN
                  IDREC  = NBEGIN
                  NRUNTM = -1
                  NTAPES = NTAPES + 1
                  NROLL  = NSTATT = 0
                  TWOEOF = ALL = NEWRANGE = .FALSE.
                  CALL OUTPUT (5)
               END IF
            ELSE
               IF (NSTATC .NE. 15) THEN
                  IERR = 10
                  ITAPEN = 0
                  CALL OUTPUT (2)
               ELSE
                  ERROR = .TRUE.
               END IF
            END IF
         ELSE
            IERR = 9
            ITAPEN = 0
            CALL OUTPUT (3)
         END IF
         IF (ERROR) GO TO 200
      END IF
      RETURN
300   FORMAT(/A1,'ENTER>  RECORD NUMBER OF DISK FILE <DATA> AT WHICH'/
     &         1X,'           THE DATA TRANSFER IS TO BEGIN  ( >0 )')
400   FORMAT(/A1,'ENTER>  MOUNT A NEW DATA TAPE AND ENTER THE',
     &             ' TAPE ID NUMBER WHEN READY'/
     &         1X,'           (THE PREVIOUS TAPE ID NUMBER IS',I5,')'/
     &         1X,'       >  NON-POSITIVE VALUE TO STOP PROCESSING')
      END
      SUBROUTINE RUNTIME (RTIME, PRF, IRUN, ITIME, IERR)
C
```

```
C            ******************************
C            *                            *
C            *     SUBROUTINE RUNTIME     *
C            *                            *
C            ******************************
C
      LOGICAL INOEOF, NEWRANGE, ALL, ERROR
      COMMON / DVSTAT / NSTATT, NSTATW, NSTATH, NSTATO, NSTATC, NSTATI,
     &                  NSTATHO, NSTATOO, NSTATHC, NSTATOC, NMOUNT,
     &                  NDSMNT
      COMMON / MISC   / CTIMER, CTIMEA, PLSPTREC, PLSPDREC, INOEOF,
     &                  NEWRANGE, ALL, MTIME(3), MDATE(3), NNULL, BELL
100   CONTINUE
      ERROR = .FALSE.
      WRITE ('UT', 200, IOSTAT = NSTATC) BELL, IRUN
      IF (NSTATC .NE. 0) THEN
         IERR = 9
         CTIMER = RTIME = 0.
         ITIME  = 0
         CALL OUTPUT (3)
      ELSE
         READ ('UT', *, IOSTAT = NSTATC) RTIME
         IF (NSTATC .EQ. 0) THEN
            RTIME  = AMAX1 (RTIME, 0.)
            CTIMER = RTIME
            ITIME  = INT (RTIME * PRF / PLSPTREC)
            IF (RTIME .GT. 0.) THEN
               CALL OUTPUT (25)
            ELSE
               CALL OUTPUT (34)
               CALL SKPFRSET (IERR)
            END IF
         ELSE
            IF (NSTATC .NE. 32) THEN
               IERR = 10
               CTIMER = RTIME = 0.
               ITIME  = 0
               CALL OUTPUT (2)
            ELSE
               ERROR = .TRUE.
            END IF
         END IF
      END IF
      IF (ERROR) GO TO 100
      RETURN
200   FORMAT(/A1,'ENTER>  NUMBER OF SECONDS OF DATA TO PROCESS FOR',
     &       ' RUN NUMBER',I5/6X,'>  NON-POSITIVE VALUE TO SKIP THE '
     &       ,'RUN')
      END
      FUNCTION NBRANCH (IERR)
C
C            ******************************
C            *                            *
C            *     FUNCTION NBRANCH       *
C            *                            *
C            ******************************
C
```

```
      LOGICAL IWDEOF, NEWRANGE, ALL
      COMMON / TRDATA / NTAPES   , NRUNS    , NRUNSER , NRUN     ,
     &                  NRUNERR  , NTRECS   , NTRECSER, NTREC    ,
     &                  NTRECERR, NDRECS   , NDRECSER, NDREC    ,
     &                  NDRECERR, NTAPEN   , NRUNB    , NRUNE    ,
     &                  NBEGIN   , NEND     , NRUNTH   , NDRNE
      COMMON / MISC   / CTIMER, CTIMEA, PLSPTREC, PLSPDREC, IWDEOF,
     &                  NEWRANGE, ALL, MTIME(3), MDATE(3), NROLL, BELL
      IF ((IERR .GE.  1 .AND. IERR .LE.  4) .OR.
     &    (IERR .GE. 11 .AND. IERR .LE. 16)) THEN
         IF (NEWRANGE) THEN
            NBRANCH = 2
         ELSE
            NBRANCH = 3
         END IF
         IF (IERR .NE. 3) THEN
            CALL SKPFRSET (IERR)
            IF (IERR .EQ. 6) NBRANCH = 1
         END IF
         IF (IERR .NE. 6) IERR = 2
      ELSE IF (IERR .EQ. -2 .OR. IERR .EQ. 0) THEN
         NBRANCH = 1
      ELSE
         NBRANCH = 0
      END IF
      RETURN
      END
C**
      SUBROUTINE TRANSFER (ITREC, IDREC, IRUN, IERR)
C
C              ********************************
C              *                              *
C              *     SUBROUTINE TRANSFER      *
C              *                              *
C              ********************************
C
C**** SUBROUTINE TRANSFER
C**** THIS SUBROUTINE CONVERTS 3 TAPE RECORDS TO 4 DISC RECORDS
C**** AND DUMPS THE NEW RECORDS TO THE DISC PACK.
C****   THE RECORD SIZES ARE CHANGED TO PERMIT EASIER STORAGE.
C**** 48 TAPE RECORDS = 64 DISC RECORDS = 64 SECTORS ON THE DISC.
C
      COMMON / DATA   / IHDR4(3), IBUFD(256)
      COMMON / IOUNIT / NTAP, NSLO, NHDR, NDTA
      COMMON / DVSTAT / NSTATT, NSTATN, NSTATH, NSTATD, NSTATC, NSTATI,
     &                  NSTATHO, NSTATDO, NSTATHC, NSTATDC, NMOUNT,
     &                  NDSMNT
      DIMENSION IBTEMP(128), IBOUT(192)
      N = MOD (ITREC - 1, 3) + 1
C**** TRANSFER FIRST FILE AND STORE EXCESS.
      IF (N .EQ. 1) THEN
         DO I = 1, 192
            IBOUT(I) = IBUFD(I)
            IF (I .LE. 64) IBTEMP(I) = IBUFD(I + 192)
         END DO
         WRITE (NDTA, REC = IDREC, IOSTAT = NSTATD) IBOUT
C**** TRANSFER EXCESS AND FILL SECOND FILE, STORE EXCESS.
```

```
      ELSE IF (N .EQ. 2) THEN
         DO I = 1, 64
            IBOUT(I) = IBTEMP(I)
         END DO
         DO J = 1, 128
            IBOUT(J + 64) = IBUFD(J)
            IBTEMP(J) = IBUFD(J + 128)
         END DO
         WRITE (NDTA, REC = IDREC, IOSTAT = NSTATD) IBOUT
C**** TRANSFER EXCESS AND FILL THIRD FILE, STORE EXCESS.
      ELSE IF (N .EQ. 3) THEN
         DO I = 1, 128
            IBOUT(I) = IBTEMP(I)
            IF (I .LE. 64) IBOUT(I + 128) = IBUFD(I)
         END DO
         WRITE (NDTA, REC = IDREC, IOSTAT = NSTATD) IBOUT
C**** TRANSFER EXCESS TO FOURTH FILE.
         IDREC = IDREC + 1
         CALL DISKCHEK (IRUN, IDREC, 2, IERR)
         IF (IERR .NE. 23) THEN
            DO J = 1, 192
               IBOUT(J) =    IBUFD(J + 64)
            END DO
            WRITE (NDTA, REC = IDREC, IOSTAT = NSTATD) IBOUT
         END IF
      END IF
C**** ERROR CHECK
      IF (NSTATD .NE. 0) THEN
         IERR = 5
         NDRWE = IDREC
         NDRECERR = NDRECERR + 1
         CALL OUTPUT (18)
      END IF
      RETURN
      END
C****
C****
      SUBROUTINE SWITCH
C
C             *******************************
C             *                             *
C             *    SUBROUTINE SWITCH        *
C             *                             *
C             *******************************
C**
C**** THIS SUBROUTINE SWITCHES THE BYTES IN A WORD.
C**** THE BYTES IN A WORD ARE IN THE ORDER 4,3,2,1.
C**** THIS SUBROUTINE REFARANGES THEM TO 1,2,3,4.
C**** THIS MAKES THE WORDS EASIER TO WORK WITH.
C**
      COMMON / DATA   / IHDR4(3), IBUFD(256)
      INTEGER*1 IBYTE(1024), IBTMP
      EQUIVALENCE (IBUFD,IBYTE)
      IBTMP = 0
      DO I = 1, 1024, 4
         IBTMP = IBYTE(I)
         IBYTE(I) = IBYTE(I + 3)
```

```
        IBYTE(I+3) = IBTMP
        IBTMP = IBYTE(I + 1)
        IBYTE(I + 1) = IBYTE(I + 2)
        IBYTE(I + 2) = IBTMP
      END DO
      RETURN
      END
C****
C****
C**
      SUBROUTINE SHUFFLE (IERR)
C
C              ****************************
C              *                          *
C              *     SUBROUTINE SHUFFLE   *
C              *                          *
C              ****************************
C
C****
C**** THIS SUBROUTINE "SHUFFLES" TWO WORDS BY TAKING THE FIRST
C**** BIT OF TWO WORDS AND MAKING THEM THE FIRST AND SECOND OF
C**** A NEW WORD AND SO ON, UNTIL THERE ARE TWO NEW SHUFFLED WORDS.
C****  THE RAW DATA IS FORMATTED AS FOLLOWS:
C**** A-REC,B-REC,A-REC,B-REC,ETC...
C**** WHERE EACH PAIR (A-REC,B-REC) IS COMPOSED OF 8 BYTES OR
C**** ONE PULSE.  THIS SUBROUTINE RECOMBINES EACH PAIR INTO
C**** THE SHUFFLED FORM DESCRIBED ABOVE.  THE SUBPLUSE CODES
C**** CAN THEN BE COMPAIRED AND A POLARIZATION VALUE DETERMINED,
C**** BUT THIS OCCURS IN ANOTHER PROGRAM.
C**
      COMMON / DATA    / IHDR4(3), IBUFD(256)
      DO J = 1,256,2
         ISTOA = 0
         ISTOB =0
         ITEMP1 = 0
         ITEMP2 = 0
         IDUMA = IBUFD(J)
         IDUMB = IBUFD(J + 1)
         IF (IERR .NE. 12) THEN
C**** BEGIN SHUFFLE OF FIRST WORD.
            DO I = 1,16
               ISTOA = IBITF(IDUMA, I-1, 1, IERR)
               ISTOB = IBITF(IDUMB, I-1, 1, IERR)
               IF (IERR .NE. 12)
     &         ITEMP1 = ISHFT(ISTOA,2*I-1)+ISHFT(ISTOB,2*(I-1))+ITEMP1
            END DO
         END IF
         IF (IERR .NE. 12) THEN
C**** BEGIN SHUFFLE OF SECOND WORD.
            DO I = 17,32
               ISTOA = IBITF(IDUMA, I-1, 1, IERR)
               ISTOB = IBITF(IDUMB, I-1, 1, IERR)
               IF (IERR .NE. 12)
     &         ITEMP2 = ISHFT(ISTOA,2*(I-16)-1)+ISHFT(ISTOB,2*(I-17))
     &                  + ITEMP2
            END DO
         END IF
         IF (IERR .NE. 12) THEN
            IBUFD(J  ) = NOT(ITEMP1)
            IBUFD(J+1) = NOT(ITEMP2)
         END IF
      END DO
      IF (IERR .EQ. 12) CALL OUTPUT (17)
      RETURN
      END
C****
      FUNCTION NTAPECHK (EOF, IFILE, IERR)
C
C              ******************************
C              *                            *
C              *     FUNCTION NTAPECHK      *
C              *                            *
C              ******************************
C
      LOGICAL EOF
      COMMON / IOUNIT / NTAP, NSLO, NHDR, NDTA
      COMMON / DVSTAT / NSTATT, NSTATW, NSTATH, NSTATD, NSTATC, NSTATI,
     &                  NSTATHD, NSTATDD, NSTATHC, NSTATDC, NMOUNT,
     &                  NDSMNT
      IF (IFILE .EQ. 0) THEN
         IF (EOF) THEN
            NTAPECHK = 1
            CALL OUTPUT (26)
         ELSE
            NTAPECHK = 3
            EOF = .TRUE.
            CALL OUTPUT (27)
         END IF
      ELSE
         NTAPECHK = 1
      END IF
      IF (IERR .EQ. 3) IERR = 2
      RETURN
      END
      SUBROUTINE SKPFRSET (IERR)
C
C              ******************************
C              *                            *
C              *    SUBROUTINE SKPFRSET     *
C              *                            *
C              ******************************
C
      COMMON / IOUNIT / NTAP, NSLO, NHDR, NDTA
      COMMON / DVSTAT / NSTATT, NSTATW, NSTATH, NSTATD, NSTATC, NSTATI,
     &                  NSTATHD, NSTATDD, NSTATHC, NSTATDC, NMOUNT,
     &                  NDSMNT
      IERS = IERR
      SKIPFILE (NTAP, IOSTAT = NSTATT)
      CALL STATUS (NTAP, IERR, IBYTE)
      IF (IERR .NE. 3) THEN
         IERR = 6
         CALL OUTPUT (14)
      ELSE
         IERR = IERS
      END IF
      RETURN
      END
      SUBROUTINE HDRDEC (IRUNB, IRUNE, IRUN, PRF, IDREC, IER)
C
C             *******************************
C             *                             *
C             *     SUBROUTINE HDRDEC       *
C             *                             *
C             *******************************
C
C*
C*    THIS SUBROUTINE DECODES THE FIRST RECORD ON THE IPAR DIGITAL TAPE,
C*    A 12-BYTE IDENTIFIER RECORD, INTO THE FOLLOWING PARAMETERS:
C*              BYTES 1- 3   BCD-FORMAT FILE IDENTIFIER
C*              BYTE      4   BCD-FORMAT RUN NUMBER
C*              BYTES 5- 8   IPAR STATUS DATA
C*              BYTES 9-12   A-REFERENCE DATA
C*    THE IPAR STATUS DATA IS DECODED BY A SEPARATE ROUTINE, STADEC.
C*    THE FILE HEADER INFORMATION IS OUTPUT TO THE USER.  AN ERROR CODE,
C*    IER, IS RETURNED AS FOLLOWS:
C*              1 = NO ERROR
C*              2 = BAD DATA OR INCORRECT FORMAT FOR SPECIFIED RECORD
C*              3 = END-OF-FILE ENCOUNTERED
C*
      INTEGER*1 IHDR1(12), IASCII(6), ITEMPL, ITEMPH, IBCDAS, MASK
      CHARACTER*2 RNCHAR
      LOGICAL TWOEOF, NEWRANGE, ALL, SKIP
      COMMON / DATA   / IHDR4(3), IBUFD(256)
      COMMON / IOUNIT / NTAP, NSLO, NHDR, NDTA
      COMMON / TRDATA / NTAPES  , NRUNS   , NRUNSER , NRUN    ,
     &                  NRUNERR , NTRECS  , NTRECSER, NTREC   ,
     &                  NTRECERR, NDRECS  , NDRECSER, NDREC   ,
     &                  NDRECERR, NTAPEN  , NRUNB   , NRUNE   ,
     &                  NBEGIN  , NEND    , NRUNTM  , NDRWE
      COMMON / DVSTAT / NSTATT, NSTATW, NSTATH, NSTATD, NSTATC, NSTATI,
     &                  NSTATHD, NSTATDD, NSTATHC, NSTATDC, NMOUNT,
     &                  NDSMNT
      COMMON / MISC   / CTIMER, CTIMEA, PLSPTREC, PLSPDREC, TWOEOF,
     &                  NEWRANGE, ALL, MTIME(3), MDATE(3), NRULL, BELL
      COMMON / RNBIAS / IBIAS(100)
      EQUIVALENCE (IHDR1, IHDR4)
      DATA MASK, IBCDAS / X'0F', X'30' /
C*****RUN NUMBER
      ITEMPL = IAND (IHDR1(4), MASK) + IBCDAS
      ITEMPH = ISHFT(IHDR1(4), -4)   + IBCDAS
      WRITE (RNCHAR, '(2A1)',IOSTAT = NSTATI) ITEMPH, ITEMPL
      READ  (RNCHAR, '(I2)', IOSTAT = NSTATI) IRUN
      IF (NSTATI .NE. 0) THEN
         IER  = 13
         IRUN = 0
         CALL OUTPUT (10)
      ELSE
         IF (IRUN .LT. 0) THEN
            IER = 13
            CALL OUTPUT (10)
         ELSE
```

```
            IRUN = IRUN + IBIAS(NTAPEN) + NROLL*100
            IF (IRUN .LT. NRUNTM) THEN
               IRUN = IRUN + 100
               NROLL = NROLL + 1
            END IF
            IF (IRUN .LE. 0) THEN
               IER = 13
               CALL OUTPUT (10)
            ELSE
               CALL DISKCHEK (IRUN, IDREC, 1, IER)
               IF (IER .NE. 22) THEN
                  NRUNTM = IRUN
                  IF ((IRUN.GE.IRUNB .AND. IRUN.LE.IRUNE) .OR. ALL) THEN
                     SKIP = .FALSE.
                     CALL HDRFIX (IRUN, IHDR4(1))
C*****FILE HEADER
                     CALL UNPACK (IHDR1, IASCII)
                     WRITE (NSLO, 30, IOSTAT = NSTATW) (IASCII(I),I=1,6)
                     WRITE (NSLO, 35, IOSTAT = NSTATW) IRUN
                  ELSE
                     SKIP = .TRUE.
                  END IF
               END IF
            END IF
         END IF
      END IF
      IF (IER .EQ. 13) NRUNERR = NRUNERR + 1
      IF (IER .NE. 13 .AND. IER .NE. 22 .AND. .NOT.(SKIP)) THEN
C*****OTHER PARAMETERS
C*****RE-ORDER BYTES
         IOFF = 4
         DO JJ  =  1, 2
         DO II  =  1, 2
            ITEMPL = IHDR1(IOFF+II)
            IHDR1(IOFF+II) = IHDR1(IOFF+5-II)
            IHDR1(IOFF+5-II) = ITEMPL
         END DO
            IOFF = IOFF+4
         END DO
         CALL STADEC (IHDR4(2), PRF, IER)
C*****A-REFERENCE DATA
         WRITE (NSLO, 40, IOSTAT = NSTATW) (IHDR1(I), I = 9, 12)
      END IF
      RETURN
C*****FORMAT STATEMENTS:
30    FORMAT(//11X,76(1H+)//11X,'DECODED RUN HEADER RECORD'/11X,45(1H-)/
     &       32X,'FILE ID:  ',6A1)
35    FORMAT(33X,'RUN ID:  ',I4)
40    FORMAT(28X,'A-REFERENCE:',4(2X,Z2)/11X,45(1H-))
      END
      SUBROUTINE STADEC (IPARSW, PRF, IER)
C
C             ******************************
C             *                            *
C             *     SUBROUTINE STADEC      *
C             *                            *
C             ******************************
```

```
C
C*
C*    THIS SUBROUTINE PERFORMS THE DECODING OF THE FILE HEADER STATUS
C*    WORD INTO THE FOLLOWING PARAMETERS:
C*                   IPAR DISPLAY SWITCH
C*                   IPAR COMPUTER FLAG SWITCH
C*                   IPAR RUN/TEST
C*                   SUB-PULSE LENGTH
C*                   PRF
C*                   TRANSMIT CODE LENGTH
C*    THESE PARAMETERS ARE RETURNED TO THE USER WITH THE HELP OF
C*    SUBROUTINE TABLE.
C*
      COMMON / IOUNIT / NTAP, NSLO, NHUR, NUTA
      ISTORE = IPARSW
      IERD = IERF = IERR = IERS = 0
      IERC = IERP = IERT = IERU = 0
C*****PRF
      IPRF = IBITF (ISTORE, 22, 4, IERP)
      IF (IERP .NE. 12) THEN
         SELECT CASE IPRF
         CASE 0
            PRF = 4000.
         CASE 4
            PRF = 2000.
         CASE 6
            PRF = 1000.
         CASE 7
            PRF = 500.
         CASE 8
            PRF = 8000.
         CASE DEFAULT
            IERU = 14
            PRF  = 0.
            NRUNERR = NRUNERR + 1
            CALL OUTPUT (13)
         END SELECT
         IF (IERU .NE. 14) THEN
            WRITE (NSLO, 50, IOSTAT = NSTATW) PRF
C*****IPAR DISPLAY SWITCH
            IDC = IBITF (ISTORE, 16, 1, IERU)
            NWD = 3
            INDX = 17+3*IDC
            IF (IERD .NE. 12) CALL TABLE (INDX, NWD, 1, IERT)
C*****IPAR COMPUTER FLAG SWITCH
            IDC = IBITF (ISTORE, 17, 1, IERF)
            NWD = 1
            INDX = 2+2*IDC
            IF (IERF .NE. 12) CALL TABLE (INDX, NWD, 2, IERT)
C*****IPAR RUN/TEST
            IDC = IBITF (ISTORE, 18, 1, IERR)
            NWD = 1
            INDX = 9+IDC
            IF (IERR .NE. 12) CALL TABLE (INDX, NWD, 3, IERT)
C*****SUB-PULSE LENGTH
            IDC = IBITF (ISTORE, 19, 3, IERS)
            NWD = 1
            IF (IFRS .NE. 12) CALL TABLE (IDC , NWD, 4, IERT)
C*****TRANSMIT CODE LENGTH
            IDC = IBITF (ISTORE, 26, 6, IERC)
            IF (IERC .NE. 12) THEN
               IP2 = 32
               ITCL = 0
               DO I = 1, 6
                  KBIT = IBITF (IDC, I-1, 1, IERC)
                  ITCL = ITCL+IP2*(1-KBIT)
                  IP2 = IP2/2
               END DO
               IF (IERC .NE. 12) WRITE (NSLO, 35, IOSTAT = NSTATW) ITCL
            END IF
         END IF
      ELSE
         PRF = 0.
         NRUNERR = NRUNERR + 1
      END IF
      IF (IERD + IERF + IERR + IERS + IERC + IERP .NE. 0) THEN
         IF (IERT + IERU .NE. 0) THEN
            IER = 15
         ELSE
            IER = 12
         END IF
      ELSE
         IF (IERT + IERU .NE. 0) IER = 14
      END IF
      RETURN
C*****FORMAT STATEMENTS:
30    FORMAT(36X,'PRF:  ',F5.0)
35    FORMAT(19X,'TRANSMIT CODE LENGTH:  ',I2)
      END
      SUBROUTINE TABLE (INDEX, NWORD, IPARAM, IERR)
C
C             ******************************
C             *                            *
C             *      SURROUTINE TABLE      *
C             *                            *
C             ******************************
C
C*
C*    SUBROUTINE TABLE PERFORMS A TABLE LOOK-UP IN ORDER TO DEFINE THE
C*    VARIOUS PARAMETERS OF THE IPAR FILE HEADER STATUS WORD.
C*
      LOGICAL ERROR
      COMMON / IOUNIT / NTAP, NSLO, NHUR, NDTA
      COMMON / DVSTAT / NSTATT, NSTATW, NSTATH, NSTATU, NSTATC, NSTATI,
     &                  NSTATHO, NSTATUO, NSTATHC, NSTATUC, NMOUNT,
     &                  NDSMNT
      DIMENSION ITAB(22),LIT(3)
      DATA ITAB/4H  80,4H SET,4H  20,4H CLR,4H  40,4H 160,4H  10,4H4000,
     1          4H RUN,4HTEST,4H    ,4H2000,4H MAN,4H1000,4H 500,4H8000,
     2          4HPANE,4HL   ,4H    ,4HDIGI,4HTAL ,4HTAPE/
      IF (NWORD .LT. 1 .OR. NWORD .GT. 3) THEN
         ERROR = .TRUE.
      ELSE
         ERROR = .FALSE.
```

```
C*****INITIALIZE LITERAL ARRAY
         DO I = 1, 3
            LIT(I) = ITAB(11)
         END DO
C*****DEFINE THE SELECTED PARAMETER FROM THE TABLE
         DO J = 1, NWORD
            IOFF = INDEX+J-1
            IF (IOFF .LT. 1 .OR. IOFF .GT. 22) THEN
               ERROR = .TRUE.
            ELSE
               LIT(J) = ITAB(IOFF)
            END IF
         END DO
      END IF
      IF (.NOT.(ERROR)) THEN
         SELECT CASE IPARAM
         CASE 1
            WRITE (NSLU, 10, IOSTAT = NSTATW) (LIT(I), I = 1, NWORD)
         CASE 2
            WRITE (NSLU, 15, IOSTAT = NSTATW) (LIT(I), I = 1, NWORD)
         CASE 3
            WRITE (NSLU, 20, IOSTAT = NSTATW) (LIT(I), I = 1, NWORD)
         CASE 4
            WRITE (NSLU, 25, IOSTAT = NSTATW) (LIT(I), I = 1, NWORD)
         CASE DEFAULT
            ERROR = .TRUE.
         END SELECT
      END IF
      IF (ERROR) THEN
         IERR = 14
         CALL OUTPUT (11)
      END IF
      RETURN
C*****FORMAT STATEMENTS:
10    FORMAT(11X,'IPAR DISPLAY SWITCH POSITION:  ',3A4)
15    FORMAT(14X,'IPAR COMPUTER FLAG SWITCH:  ',A4)
20    FORMAT(26X,'IPAR RUN/TEST:  ',A4)
25    FORMAT(23X,'SUB-PULSE LENGTH:  ',A4)
      END
      SUBROUTINE UNPACK (IHDR1, IASCII)
C
C              ******************************
C              *                            *
C              *     SUBROUTINE UNPACK      *
C              *                            *
C              ******************************
C
C*
C*    SUBROUTINE UNPACK SPLITS A SINGLE BYTE INTO TWO SEPARATE 4-BIT
C*    BCD QUANTITIES.  THESE QUANTITIES ARE THEN CONVERTED TO ASCII
C*    FORMAT AND STORED IN THE BYTE ARRAY, IASCII.
C*
      INTEGER*1 IHDR1(12), IASCII(6), MASK, ITMP, JTMP
      DATA MASK/X'0F'/
C*****UNPACK EACH OF THE NBT BYTES
      DO I = 1, 3
         IREF = (4 - I) * 2
```

```
C*****LOWER BYTE
          ITMP=IAND(IHDR1(I),MASK)
          ENCODE(1, '(I1)', JTMP) ITMP
          DECODE(1, '(A1)', JTMP) IASCII(IREF)
C*****UPPER BYTE
          ITMP=ISHFT(IHDR1(I),-4)
          ENCODE(1, '(I1)', JTMP) ITMP
          DECODE(1, '(A1)', JTMP) IASCII(IREF-1)
      END DO
      RETURN
      END
      SUBROUTINE READER (IRF, ITREC, IER)
C
C              ******************************
C              *                            *
C              *     SUBROUTINE READER      *
C              *                            *
C              ******************************
C
C*
C*    THIS SUBROUTINE IS DESIGNED TO READ A HEADER OR DATA RECORD FROM
C*    THE IPAR DATA TAPE.  AN ERROR CODE, IERC, IS RETURNED AS FOLLOWS:
C*              1 = INCORRECT NUMBER OF BYTES FOR SPECIFIED RECORD TYPE
C*              2 = NO ERROR
C*              3 = END-OF-FILE ENCOUNTERED
C*              4 = PARITY ERROR OR LOST DATA ON READ
C*    THE RECORD FORMAT CODE IS AS FOLLOWS:
C*              1 = FILE HEADER RECORD
C*              2 = DATA RECORD
C*
      LOGICAL TWOEOF, READY
      COMMON / DATA   / IHDR4(3), IBUFD(256)
      COMMON / IOUNIT / NTAP, NSLO, NHDR, NDTA
      COMMON / TRDATA / NTAPES  , NRUNS   , NRUNSER , NRUN    ,
     &                  NRUNERR , NTRECS  , NTRECSER, NTREC   ,
     &                  NTRECERR, NDRECS  , NDRECSER, NDREC   ,
     &                  NDRECERR, NTAPEN  , NRUNB   , NRUNE   ,
     &                  NBEGIN  , NEND    , NRUNTM  , NDRWE
      COMMON / DVSTAT / NSTATT, NSTATW, NSTATH, NSTATD, NSTATC, NSTATI,
     &                  NSTATHO, NSTATDD, NSTATHC, NSTATDC, NMOUNT,
     &                  NDSMNT
      COMMON / TAPEFT / NBH, NWH, NBD, NWD, NBYT
      COMMON / MISC   / CTIMER, CTIMEA, PLSPTREC, PLSPDREC, TWOEOF,
     &                  NEWRANGE, ALL, MTIME(3), MDATE(3), NROLL, BELL
C*****FILE HEADER RECORD
      IF (IRF .EQ. 1 ) THEN
100      CONTINUE
         READY = .TRUE.
C*****READ THE RECORD
         READ (NTAP, '(20A4)', IOSTAT=NSTATI) (IHDR4(II), II = 1, NWH)
         CALL M:WAIT (NTAP)
         CALL STATUS (NTAP, IERC, NBYT)
C*****ERROR PROCESSING
         SELECT CASE IERC
C*****UNIT NOT READY
         CASE 1
            READY = .FALSE.
```

```
C*****GOOD READ.  CHECK FOR PROPER BYTE COUNT.
         CASE 2
            TWOEOF = .FALSE.
            IF (NBYT .NE. NBH) THEN
               IERC = 1
               NRUNERR = NRUNERR + 1
               CALL OUTPUT (7)
            ELSE
               ITREC = 0
            END IF
C*****BAD READ.  FLAG THE ERROR AND RETURN.
         CASE 4
            TWOEOF = .FALSE.
            NRUNERR = NRUNERR + 1
            CALL OUTPUT (8)
         END SELECT
         IF (.NOT.(READY)) GO TO 100
C*****DATA RECORD
      ELSE IF (IRF .EQ. 2) THEN
200      CONTINUE
         READY = .TRUE.
C*****READ THE RECORD
         CALL BUFFERIN (NTAP, 1, IBUFD, NBD)
         CALL M:WAIT (NTAP)
         CALL STATUS (NTAP, IERC, NBYT)
C*****ERROR PROCESSING
         SELECT CASE IERC
C*****UNIT NOT READY
         CASE 1
            READY = .FALSE.
C*****GOOD READ.  CHECK FOR PROPER BYTE COUNT.
         CASE 2
            IF (NBYT .NE. NBD) THEN
               IERC = 1
               NTRECERR = NTRECERR + 1
               CALL OUTPUT (7)
            ELSE
               ITREC = ITREC + 1
            END IF
C*****BAD READ.  FLAG THE ERROR AND RETURN.
         CASE 4
            NTRECERR = NTRECERR + 1
            CALL OUTPUT (8)
         END SELECT
         IF (.NOT.(READY)) GO TO 200
      ELSE
         IERC = 16
         CALL OUTPUT (9)
      END IF
      IF (IERC .NE. 2) IER = IERC
      RETURN
      END
      FUNCTION IBITF (IWORD, ISBIT, LEN, IERR)
C
C              *******************************
C              *                             *
C              *       FUNCTION IBITF        *
```

```
C              *                              *
C              ********************************
C
C*
C*    FUNCTION IBITF EXTRACTS A FIELD OF BITS FROM IWORD, STARTING
C*    FROM BIT POSITION ISBIT AND EXTENDING TO THE LEFT FOR LEN BITS.
C*    THE RIGHTMOST BIT IS BIT 0 AND THE LEFTMOST BIT IS BIT 31.
C*
      LOGICAL ERROR
C*****ERROR CONDITION CHECK
      IF (ISBIT+LEN.LE.32 .AND. LEN.GT.0 .AND. ISBIT.GE.0) THEN
C*****UNPACK THE SPECIFIED BIT FIELD
         ERROR = .FALSE.
         IBITF = ISHFT(IWORD,32-(ISBIT+LEN))
         IBITF = ISHFT(IBITF,-32+LEN)
      ELSE
         ERROR = .TRUE.
      END IF
      IF (ERROR) THEN
         IERR = 12
         CALL OUTPUT (12)
      END IF
      RETURN
      END
      FUNCTION NRUNTEST (IRUNB, IRUNE, IRUN, IERR)
C
C              ********************************
C              *                              *
C              *      FUNCTION NRUNTEST       *
C              *                              *
C              ********************************
C
      LOGICAL TWOEOF, NEWRANGE, ALL
      COMMON / IOUNIT / NTAP, NSLO, NHDR, NDTA
      COMMON / DVSTAT / NSTATT, NSTATW, NSTATH, NSTATD, NSTATC, NSTATI,
     &                  NSTATHO, NSTATDO, NSTATHC, NSTATDC, NMOUNT,
     &                  NDSMNT
      COMMON / MISC   / CTIMER, CTIMEA, PLSPIREC, PLSPOREC, TWOEOF,
     &                  NEWRANGE, ALL, MTIME(3), MDATE(3), NROLL, BELL
      IF (.NOT.(ALL)) THEN
         IF (IRUN .LT. IRUNB) THEN
            CALL SKPFRSET (IERR)
            IF (IERR .NE. 0) THEN
               NRUNTEST = 3
            ELSE
               NRUNTEST = 1
            END IF
         ELSE IF (IRUN .GT. IRUNE) THEN
            CALL OUTPUT (32)
            BACKSPACE (NTAP, IOSTAT = NSTATT)
            IF (NSTATT .NE. 0) THEN
               CALL OUTPUT (16)
               CALL SKPFRSET (IERR)
               IF (IERR .NE. 0) THEN
                  NRUNTEST = 2
               ELSE
                  NRUNTEST = 1
                  END IF
               ELSE
                  NEWRANGE = .TRUE.
                  NRUNTEST = 2
               END IF
            ELSE
               IF (IRUN .EQ. IRUNE) NEWRANGE = .TRUE.
               NRUNTEST = 4
            END IF
         ELSE
            NRUNTEST = 4
         END IF
         RETURN
         END
         SUBROUTINE DATACHEK (ITREC, IDREC, ITIME, IRUN, PRF,
        &                     IBRANCH, IERR)
C
C             ******************************
C             *                            *
C             *     SUBROUTINE DATACHEK    *
C             *                            *
C             ******************************
C
         LOGICAL TWOEOF, NEWRANGE, ALL
         COMMON / TRDATA / NTAPES   , NRUNS    , NRUNSER , NRUN     ,
        &                  NRUNERR , NTRECS   , NTRECSER, NTREC    ,
        &                  NTRECERR, NDRECS   , NDRECSER, NDREC    ,
        &                  NDRECERR, NTAPEN   , NRUNB   , NRUNE    ,
        &                  NBEGIN  , NEND     , NRUNTM  , NDRWE
         COMMON / MISC   / CTIMER, CTIMEA, PLSPTREC, PLSPDREC, TWOEOF,
        &                  NEWRANGE, ALL, MTIME(3), MDATE(3), NROLL, BELL
         IF (IERR .EQ. 2) THEN
            IF (ITREC .GE. ITIME) THEN
               CALL DUMPHEDR (IDREC, IRUN, PRF, IERO)
               IBRANCH = NBRANCH (IERR)
            ELSE
               IBRANCH = 4
            END IF
         ELSE IF (IERR .EQ. 3) THEN
            IDREC = IDREC - 1
            IF (ITREC .EQ. 0) THEN
               CALL OUTPUT (22)
            ELSE
               CALL DUMPHEDR (IDREC, IRUN, PRF, IERO)
            END IF
            IBRANCH = NBRANCH (IERR)
         ELSE
            IDREC = IDREC - 1
            IF ((IDREC - NBEGIN + 1) .EQ. 0) THEN
               CALL OUTPUT (23)
            ELSE
               CALL DUMPHEDR (IDREC, IRUN, PRF, IERO)
            END IF
            IBRANCH = NBRANCH (IERR)
         END IF
         CALL DISKCHEK (IRUN, IDREC + 1, 2, IERR)
         IF (IERR .EQ. 23) THEN
         IF (IBRANCH .EQ. 4) CALL DUMPHEDR (IDREC, IRUN, PRF, IERD)
         IBRANCH = 0
      ELSE
         IDREC = IDREC + 1
         IF (IERD .EQ. 7) THEN
            IF (IERR .EQ. 6) THEN
               IERR = 8
            ELSE
               IERR = IERD
            END IF
         END IF
      END IF
      RETURN
      END
      BLOCK DATA INITIAL
C
C              ********************************
C              *                              *
C              *     BLOCK DATA INITIAL       *
C              *                              *
C              ********************************
C
      LOGICAL IWDEOF, NEWRANGE, ALL
      COMMON / DATA   / IHDR4(3), IBUFD(256)
      COMMON / IOUNIT / NTAP, NSLO, NHDR, NDTA
      COMMON / TRDATA / NTAPES  , NRUNS   , NRUNSER , NRUN    ,
     &                  NRUNERR , NTRECS  , NTRECSER, NTREC   ,
     &                  NTRECERR, NDRECS  , NDRECSER, NDREC   ,
     &                  NDRECERR, NTAPEN  , NRUNB   , NRUNE   ,
     &                  NBEGIN  , NEND    , NRUNTM  , NDRWE
      COMMON / DVSTAT / NSTATT, NSTATW, NSTATH, NSTATD, NSTATC, NSTATI,
     &                  NSTATHD, NSTATDD, NSTATHC, NSTATDC, NMOUNT,
     &                  NDSMNT
      COMMON / TAPEFT / NBH, NWH, NBD, NWD, NBYT
      COMMON / MISC   / CTIMER, CTIMEA, PLSPTREC, PLSPDREC, IWDEOF,
     &                  NEWRANGE, ALL, MTIME(3), MDATE(3), NROLL, BELL
      COMMON / DIRECT / NDIR(1400, 2)
      COMMON / RNBIAS / IBIAS(100)
      COMMON / DISK   / NSZHEDR, NSZDATA, NBPSCTR, NBPHREC, NBPDREC,
     &                  NBPPLSE, RHSPACE, NRMHEDR, RDSPACE, NRMDATA,
     &                  NRMPLSE
C
      DATA IHDR4, IBUFD / 3*0, 256*0 /
      DATA NTAP, NSLO, NHDR, NDTA / 1, 6, 2, 3 /
      DATA NTAPES  , NRUNS   , NRUNSER , NRUN    , NRUNERR , NTRECS  ,
     &     NTRECSER, NTREC   , NTRECERR, NDRECS  , NDRECSER, NDREC   ,
     &     NDRECERR, NTAPEN  , NRUNB   , NRUNE   , NBEGIN  , NEND    ,
     &     NRUNTM  , NDRWE   / 16*0, 1, 0, -1, 0 /
      DATA NSTATT, NSTATW, NSTATH, NSTATD, NSTATC, NSTATI, NSTATHD,
     &     NSTATDD, NSTATHC, NSTATDC, NMOUNT, NDSMNT / 12*0 /
      DATA NBH, NWH, NBD, NWD, NBYT / 12, 3, 1024, 256, 0 /
      DATA CTIMER, CTIMEA, PLSPTREC, PLSPDREC, IWDEOF, NEWRANGE,
     &     ALL, MTIME, MDATE, NROLL, BELL / 2*0.0, 128.0, 0.0,
     &                                     3*.FALSE., 7*0, X'07000000'/
      DATA ((NDIR(I, J), I = 1, 1400), J = 1, 2) / 1400*0, 1400*0 /
      DATA IBIAS / 0, 0, 100, 100, 200, 300, 300, 600, -1000, 700,
     &            800, 800, -1000, 800, 900, 900, 900, 1000, 1000,
&              1000, 1100, 1100, 1300, 17*-1000 /
 DATA NSZHEDR, NSZDATA, NBPSCTR, NBPHREC, NBPDREC, NBPPLSE,
&     RHSPACE, NRMHEDR, RDSPACE, NRMDATA, NRMPLSE
&     / 50, 50000, 768, 20, 768, 8, 50., 1920, 50000., 50000,
&        4800000 /
 END
 SUBROUTINE HDRFIX (IRUN, IFHDR)
C
C           ******************************
C           *                            *
C           *      SUBROUTINE HDRFIX     *
C           *                            *
C           ******************************
C
 DIMENSION JRUN(5), JFIX(5)
 DATA JRUN / 4, 65, 70, 677, 1112 /
 DATA JFIX / X'20000000', X'01000000', X'00000000',
&              X'12000000', X'20000000' /
 DATA MASK / X'00FFFFFF' /
 DO I = 1, 5
    IF (IRUN .EQ. JRUN(I)) IFHDR = IOR (IAND (MASK,IFHDR), JFIX(I))
 END DO
 RETURN
 END
 SUBROUTINE OUTPUT (ICODE)
C
C           ******************************
C           *                            *
C           *      SUBROUTINE OUTPUT     *
C           *                            *
C           ******************************
C
 LOGICAL TWOEOF, NEWRANGE, ALL
 COMMON / DATA   / IHDR4(3), IBUFD(256)
 COMMON / IOUNIT / NTAP, NSLD, NHDR, NDTA
 COMMON / TRDATA / NTAPES  , NRUNS   , NRUNSER , NRUN    ,
&                  NRUNERR , NTRECS  , NTRECSER, NTREC   ,
&                  NTRECERR, NDRECS  , NDRECSER, NDREC   ,
&                  NDRECERR, NTAPEN  , NRUNB   , NRUNE   ,
&                  NBEGIN  , NEND    , NRUNTM  , NDRWE
 COMMON / DVSTAT / NSTATT, NSTATN, NSTATH, NSTATD, NSTATC, NSTATI,
&                  NSTATHD, NSTATDD, NSTATHC, NSTATDC, NMOUNT,
&                  NDSMNT
 COMMON / TAPEFI / NBH, NWH, NBD, NWD, NBYT
 COMMON / MISC   / CTIMER, CTIMEA, PLSPTREC, PLSPDREC, TADEOF,
&                  NEWRANGE, ALL, MTIME(3), MDATE(3), NROLL, BELL
 COMMON / DIRECT / NDIR (1400,2)
 COMMON / RNBIAS / IBIAS(100)
 COMMON / DISK   / NSZHEDR, NSZDATA, NBPSCTR, NBPHREC, NBPDREC,
&                  NBPPLSE, RHSPACE, NRMHEDR, RDSPACE, NRMDATA,
&                  NRMPLSE
 DIMENSION MONTH(12)
 DATA MONTH / ' JAN', ' FEB', ' MAR', ' APR', ' MAY', ' JUN',
&             ' JUL', ' AUG', ' SEP', ' OCT', ' NOV', ' DEC' /
 DATA BLK / X'20000000' /
 SELECT CASE ICODE
 CASE  1
```

```
CALL TIME (MTIME)
CALL DATE (MDATE)
   WRITE (NSLO,  10, IOSTAT = NSTATW) MDATE(3), MONTH(MDATE(2)),
&                                     MDATE(1), MTIME
 CASE  2
   WRITE (NSLO,  20, IOSTAT = NSTATW) BLK
 CASE  3
   WRITE (NSLO,  30, IOSTAT = NSTATW) BLK
 CASE  4
   WRITE (NSLO,  40, IOSTAT = NSTATW) BLK, NTAPEN, NRUN, NRUNERR,
&          NTREC, NTRECERR, NUREC, NURECERR
 CASE  5
   WRITE (NSLO,  50, IOSTAT = NSTATW) NTAPEN
 CASE  6
   IF (ALL) THEN
      WRITE (NSLO,  65, IOSTAT = NSTATW) BLK, NTAPEN
   ELSE
      WRITE (NSLO,  60, IOSTAT = NSTATW) BLK, NRUNB, NRUNE
   END IF
 CASE  7
   WRITE ('UT',  70, IOSTAT = NSTATC) BELL, NBYT
   WRITE (NSLO,  70, IOSTAT = NSTATW) BLK , NBYT

 CASE  8
   WRITE ('UT',  80, IOSTAT = NSTATC) BELL
   WRITE (NSLO,  80, IOSTAT = NSTATW) BLK
 CASE  9
   WRITE ('UT',  90, IOSTAT = NSTATC) BELL
   WRITE (NSLO,  90, IOSTAT = NSTATW) BLK
 CASE 10
   WRITE ('UT', 100, IOSTAT = NSTATC) BELL, NTAPEN, IHDR4,
&                                     IBIAS(NTAPEN)
   WRITE (NSLO, 100, IOSTAT = NSTATW) BLK , NTAPEN, IHDR4,
&                                     IBIAS(NTAPEN)
 CASE 11
   WRITE ('UT', 110, IOSTAT = NSTATC) BELL
   WRITE (NSLO, 110, IOSTAT = NSTATW) BLK
 CASE 12
   WRITE ('UT', 120, IOSTAT = NSTATC) BELL
   WRITE (NSLO, 120, IOSTAT = NSTATW) BLK
 CASE 13
   WRITE ('UT', 130, IOSTAT = NSTATC) BELL, IHDR4
   WRITE (NSLO, 130, IOSTAT = NSTATW) BLK , IHDR4
 CASE 14
   WRITE ('UT', 140, IOSTAT = NSTATC) BELL, NTAPEN
   WRITE (NSLO, 140, IOSTAT = NSTATW) BLK , NTAPEN
 CASE 15
   WRITE ('UT', 150, IOSTAT = NSTATC) BELL, NTAPEN
   WRITE (NSLO, 150, IOSTAT = NSTATW) BLK , NTAPEN
 CASE 16
   WRITE ('UT', 160, IOSTAT = NSTATC) BELL, NTAPEN, NRUNTM
   WRITE (NSLO, 160, IOSTAT = NSTATW) BLK , NTAPEN, NRUNTM
 CASE 17
   WRITE ('UT', 170, IOSTAT = NSTATC) BELL
   WRITE (NSLO, 170, IOSTAT = NSTATW) BLK
 CASE 18
   WRITE ('UT', 180, IOSTAT = NSTATC) BELL, NDTA, NRUNTM, NDRWE
   WRITE (NSLO, 180, IOSTAT = NSTATW) BLK , NDTA, NRUNTM, NDRWE
```

[redacted] [redacted]

```
 CASE 19
    WRITE ('UT', 190, IOSTAT = NSTATC) BELL, NRUNTM
    WRITE (NSLO, 190, IOSTAT = NSTATW) BLK , NRUNTM
 CASE 20
    WRITE ('UT', 200, IOSTAT = NSTATC) BELL,NRUNTM, CTIMER, CTIMEA,
&                                 NBEGIN, NEND  , NEND - NBEGIN + 1
    WRITE (NSLO, 200, IOSTAT = NSTATW) BLK, NRUNTM, CTIMER, CTIMEA,
&                                 NBEGIN, NEND  , NEND - NBEGIN + 1
 CASE 21
    WRITE ('UT', 210, IOSTAT = NSTATC) BELL, NRUNTM
    WRITE (NSLO, 210, IOSTAT = NSTATW) BLK , NRUNTM
 CASE 22
    WRITE ('UT', 220, IOSTAT = NSTATC) BELL, NRUNTM
    WRITE (NSLO, 220, IOSTAT = NSTATW) BLK , NRUNTM
 CASE 23
    WRITE ('UT', 230, IOSTAT = NSTATC) BELL, NRUNTM
    WRITE (NSLO, 230, IOSTAT = NSTATW) BLK , NRUNTM
 CASE 24
    WRITE ('UT', 240, IOSTAT = NSTATC) BELL, NTAPEN
    WRITE (NSLO, 240, IOSTAT = NSTATW) BLK , NTAPEN
 CASE 25
    WRITE (NSLO, 250, IOSTAT = NSTATW) BLK , CTIMER, NRUNTM
 CASE 26
    WRITE ('UT', 260, IOSTAT = NSTATC) BELL, NTAPEN
    WRITE (NSLO, 260, IOSTAT = NSTATW) BLK , NTAPEN
 CASE 27
    WRITE ('UT', 270, IOSTAT = NSTATC) BELL, NTAPEN
    WRITE (NSLO, 270, IOSTAT = NSTATW) BLK , NTAPEN
 CASE 28
    WRITE (NSLO, 280, IOSTAT = NSTATW) BLK, NSTATT, NSTATC, NSTATW,
&           NSTATT, NSTATHO, NSTATH, NSTATHC, NSTATDO, NSTATD,
&           NSTATDC
    CALL TIME (MTIME)
    WRITE (NSLO, 285, IOSTAT = NSTATW) BLK, MTIME
 CASE 29
    WRITE ('UT', 290, IOSTAT = NSTATC) BELL
    WRITE (NSLO, 290, IOSTAT = NSTATW) BLK
 CASE 30
    WRITE (NSLO, 300, IOSTAT = NSTATW) NTAPES, NRUNS, NRUNSER,
&           NTRECS, NTRECSER, NDRECS, NDRECSER
    DO I = 1, 1400
       IF (NDIR(I, 1) .NE. 0) WRITE (NSLO, 304, IOSTAT = NSTATW) I,
&                                      NDIR(I, 1), NDIR(I, 2)
    END DO
    WRITE (NSLO, 308, IOSTAT = NSTATW) BLK    , NSZHEDR, NSZDATA,
&                                      RHSPACE, RDSPACE, NRMHEDR,
&                                      NRMDATA
 CASE 31
    WRITE ('UT', 310, IOSTAT = NSTATC) BELL
    WRITE (NSLO, 310, IOSTAT = NSTATW) BLK
 CASE 32
    WRITE ('UT', 320, IOSTAT = NSTATC) BELL, NRUNTM
 CASE 33
    WRITE (NSLO, 330, IOSTAT = NSTATW) BLK , NTAPEN
 CASE 34
    WRITE (NSLO, 340, IOSTAT = NSTATW) BLK , NRUNTM
 CASE 35
```

```
         WRITE ('UT', 350, IOSTAT = NSTATC) BELL, NRUNTM
         WRITE (NSLO, 350, IOSTAT = NSTATW) BLK , NRUNTM
      CASE 36
         WRITE ('UT', 360, IOSTAT = NSTATC) BELL, NRUNTM
         WRITE (NSLO, 360, IOSTAT = NSTATW) BLK , NRUNTM
      CASE 37
         WRITE ('UT', 370, IOSTAT = NSTATC) BELL, NMOUNT
         WRITE (NSLO, 370, IOSTAT = NSTATW) BLK , NMOUNT
      CASE 38
         WRITE ('UT', 380, IOSTAT = NSTATC) BELL, NDSMNT
         WRITE (NSLO, 380, IOSTAT = NSTATW) BLK , NDSMNT
      CASE DEFAULT
         WRITE (NSLO, 999, IOSTAT = NSTATW) BLK
      END SELECT
      IF (NSTATW .NE. 0 .AND. NSTATC .EQ. 0)
     &    WRITE ('UT', 998, IOSTAT = NSTATC) BELL
C*****FORMAT STATEMENTS:
10    FORMAT(1H1,'IPAR MAGNETIC TAPE TO DISK DATA TRANSFER PROGRAM'///
     &   1X,'RUN DATE',6X,':  ',I2,A4,I5//1X,'RUN START TIME:  ',
     &   I2,':',I2,':',I2)
20    FORMAT(/A1,'***** ERROR:  FATAL READ ERROR ENCOUNTERED ON ',
     &   'CONSOLE')
30    FORMAT(/A1,'***** ERROR:  FATAL WRITE ERROR ENCOUNTERED ON ',
     &   'CONSOLE')
40    FORMAT(/A1,86(1H*)///1X,'SUMMARY INFORMATION FOR IPAR DATA ',
     &   'TAPE NUMBER:',I3//50X,'ACCEPTED',5X,'REJECTED'/1X,70(1H-)/1X,
     &   'NUMBER OF USER SELECTED RUNS ON DISK',10X,':  ',I8,5X,I8/1X,
     &   'NUMBER OF TAPE DATA RECORDS READ AND PROCESSED:  ',I8,5X,
     &   I8/1X,'NUMBER OF WRITE ATTEMPTS TO DISK FILE <DATA>  :  ',
     &   I8,5X,I8////1X,86(1H*))
50    FORMAT(1H1,'IPAR DATA TAPE NUMBER:',I3)
60    FORMAT(/A1,86(1H*)///11X,'USER REQUESTED PROCESSING OF RUNS',I5,
     &   ' THROUGH',I5,'.')
65    FORMAT(/A1,86(1H*)///11X,'USER REQUESTED PROCESING OF ALL RUNS ',
     &   'ON IPAR TAPE NUMBER',I3,'.')
70    FORMAT(/A1,'***** ERROR:  UNEXPECTED BYTE COUNT ON INPUT FOR',
     &   ' THE SELECTED RECORD TYPE'/1X,'*****',9X,'ENCOUNTERED IN ',
     &   'SUBROUTINE READER.',5X,'BYTES TRANSFERRED:',I5/1X,'*****',9X,
     &   'REMAINDER OF RUN FILE SKIPPED.')
80    FORMAT(/A1,'***** ERROR:  BAD DATA OR PARITY ERROR ENCOUNTERED ',
     &   'ON INPUT IN SUBROUTINE'/1X,'*****',9X,'READER.  REMAINDER OF',
     &   ' RUN FILE SKIPPED.')
90    FORMAT(/A1,'***** ERROR:  INVALID RECORD TYPE SPECIFIER IN ',
     &   'SUBROUTINE READER.'/1X,'*****',9X,'REQUEST FOR DATA INPUT ',
     &   'DENIED.  REMATNDER OF RUN FILE SKIPPED.')
100   FORMAT(/A1,'***** ERROR:  BAD RUN HEADER RECORD DATA, INCORRECT ',
     &   'BIAS FOR TAPE',I3,','/1X,'*****',9X,'OR I/O ERROR IN ',
     &   'INTERNAL FILE <RNCHAR>.'/1X,'*****',9X,'HEADER RECORD:',
     &   3(2X,Z8),' (HEX)'/1X,'*****',9X,'RUN BIAS:',I6,'.  ',
     &   'REMAINDER OF RUN FILE SKIPPED.')
110   FORMAT(/A1,'***** ERROR:  INVALID ARGUMENT(S) PASSED TO ',
     &   'SUBROUTINE TABLE.'/1X,'*****',9X,'REQUEST FOR RUN ',
     &   'PARAMETER INFORMATION DENIED.'/1X,'*****',9X,'REMAINDER OF ',
     &   'RUN FILE SKIPPED.')
120   FORMAT(/A1,'***** ERROR:  INVALID ARGUMENT(S) PASSED TO FUNCTION',
     &   ' IBITF.'/1X,'*****',9X,'REQUEST FOR BIT FIELD EXTRACTION ',
     &   'DENIED.'/1X,'*****',9X,'REMAINDER OF RUN FILE SKIPPED.')
```

```
130   FORMAT(/A1,'***** ERROR:  UNDEFINED PRF CODE OBTAINED FROM RUN ',
     &    /1X,'*****',9X,'HEADER RECORD (',3(2X,Z8),' (HEX) ).'/1X,
     &    '*****',9X,'REMAINDER OF RUN FILE SKIPPED.')
140   FORMAT(/A1,'***** ERROR:  SKIPFILE ERROR ENCOUNTERED ON TAPE ',
     &    'DRIVE WHILE PROCESSING'/1X,'*****',9X,'TAPE NUMBER',I3,'.  ',
     &    'INDETERMINATE TAPE POSITION.'/1X,'*****',9X,'REMAINDER OF ',
     &    'TAPE FILE SKIPPED.  DISMOUNT TAPE.')
150   FORMAT(/A1,'***** ERROR:  REWIND ERROR ENCOUNTERED ON TAPE DRIVE',
     &    ' WHILE PROCESSING '/1X,'*****',9X,'TAPE NUMBER',I3,'.  ',
     &    'REWIND THE TAPE USING THE TAPE'/1X,'*****',9X,'DRIVE FRONT',
     &    ' PANEL CONTROLS.')
160   FORMAT(/A1,'***** ERROR:  BACKSPACE ERROR ENCOUNTERED ON TAPE ',
     &    'DRIVE WHILE PROCESSING'/1X,'*****',9X,'TAPE NUMBER',I3,
     &    ', RUN NUMBER',I5,'.  INDETERMINATE TAPE POSITION.'/1X,
     &    '*****',9X,'REMAINDER OF TAPE SKIPPED.  MOUNT A NEW TAPE IF ',
     &    'DESIRED.')
170   FORMAT(/A1,'***** ERROR:  PROCESSING TERMINATED IN SUBROUTINE ',
     &    'SHUFFLE DUE TO ERROR IN'/1X,'*****',9X,'FUNCTION IBITF.  ',
     &    'PROBABLE OVER-WRITE CONDITION.'/1X,'*****',9X,'REMAINDER OF',
     &    ' RUN FILE SKIPPED.')
180   FORMAT(/A1,'***** ERROR:  ERROR ENCOUNTERED IN SUBROUTINE ',
     &    'TRANSFER DURING WRITE ATTEMPT'/1X,'*****',9X,'TO DISK FILE ',
     &    '<DATA>, UNIT NUMBER',I3,'.  THE ERROR OCCURRED WHILE'/1X,
     &    '*****',9X,'PROCESSING RUN NUMBER',I5,', DISK RECORD NUMBER',
     &    I6,'.')
190   FORMAT(/A1,'***** ERROR:  ILLEGAL DATA RECORD NUMBER PASSED TO ',
     &    'SUBROUTINE DUMPDATA WHILE'/1X,'*****',9X,'PROCESSING RUN ',
     &    'NUMBER',I5,'.  PROBABLE OVER-WRITE CONDITION.'/1X,'*****',
     &    9X,'REMAINDER OF RUN FILE SKIPPED.')
200   FORMAT(//A1,10X,'DISK DATA LOAD FOR RUN NUMBER',I5,' IS COMPLETE.'
     &    //11X,'AMOUNT OF DATA REQUESTED:',F8.4,' SECONDS'/11X,'AMOUNT',
     &    ' OF DATA PROCESSED:',F8.4,' SECONDS'/11X,'FIRST DISK ',
     &    'RECORD NUMBER:',I8/11X,'LAST  DISK RECORD NUMBER:',I8/11X,
     &    'TOTAL NUMBER OF DISK RECORDS WRITTEN FOR THIS RUN:',I5)
210   FORMAT(/A1,'***** ERROR:  UNABLE TO WRITE HEADER RECORD AND ',
     &    'DIRECTORY INFORMATION TO DISK'/1X,'*****',9X,'FILE <HEADER>',
     &    ' FOR RUN NUMBER',I5,'.  THE DATA FOR THIS RUN'/1X,'*****',9X,
     &    'WILL BE DELETED SINCE NO IDENTIFIER CAN BE STORED.')
220   FORMAT(/A1,'***** WARNING:  NO DATA RECORDS WERE FOUND IN THE ',
     &    'TAPE FILE FOR RUN'/1X,'*****',11X,'NUMBER',I5,'.  NO DATA ',
     &    'WAS WRITTEN TO DISK.')
230   FORMAT(/A1,'***** ERROR:  NO VALID DATA FROM RUN NUMBER',I5,
     &    ' IS AVAILABLE TO STORE'/1X,'*****',9X,'ON DISK.  NO HEADER',
     &    ' RECORD OR DIRECTORY INFORMATION FOR THIS'/1X'*****',9X,
     &    'RUN WILL BE STORED ON DISK.')
240   FORMAT(/A1,'***** ERROR:  A FATAL TAPE ERROR HAS OCCURRED ON ',
     &    'TAPE NUMBER',I3,'.'/1X,'*****',9X,'CONTINUE PROCESSING ON ',
     &    'A NEW TAPE IF DESIRED.')
250   FORMAT(/10X,A1,'THE USER HAS REQUESTED THAT',F8.4,' SECONDS OF ',
     &    'RUN NUMBER',I5,' BE PROCESSED.')
260   FORMAT(/A1,'***** WARNING:  TWO CONSECUTIVE END-OF-FILE MARKERS ',
     &    'WERE FOUND AT THE START'/1X,'*****',11X,'OF TAPE NUMBER',I3,
     &    '.  THE TAPE IS ASSUMED TO BE BLANK.'/1X,'*****',11X,
     &    'DISMOUNT TAPE.')
270   FORMAT(/A1,'***** WARNING:  AN END-OF-FILE MARKER WAS FOUND AT ',
     &    'THE START OF TAPE NUMBER',I3,'.'/1X,'*****',11X,'PROCESSING',
     &    ' CONTINUES.')
```

```
280   FORMAT(///A1,'DEVICE STATUS CODES AT PROGRAM TERMINATION (IN HEX)'
     &   /1X,59(1H-)/1X,'TAPE DRIVE',12X,':  ',Z8/1X,'CONSOLE',15X,
     &   ':  ',Z8/1X,'LINE PRINTER',10X,':  ',Z8/1X,'INTERNAL ',
     &   'FILE <RNCHAR>:  ',Z8//1X,'DISK FILES',12X,':    OPEN',
     &   9X,'WRITE',8X,'CLOSE'/15X,'<HEADER>:  ',Z8,2(5X,Z8)/15X,
     &   '<DATA  >:  ',Z8,2(5X,Z8))
285   FORMAT(///A1,'RUN STOP TIME:  ',I2,':',I2,':',I2)
290   FORMAT(/A1,'***** ERROR:  ERROR IN CLOSING <HEADER> AND/OR ',
     &   '<DATA> FILES.  CHECK THE IOSTAT'/1X,'*****',9X,'ERROR CODE ',
     &   'AND THE PARAMETERS IN THE CLOSE STATEMENTS.'/1X,'*****',9X,
     &   'THE OPERATING SYSTEM WILL CLOSE ALL OPEN FILES AT THE ',
     &   'TERMINATION'/1X,'*****',9X,'OF THE PROGRAM.')
300   FORMAT(1H1,'1PAR MAGNETIC TAPE TO DISK DATA TRANSFER PROGRAM HAS '
     &   ,'COMPLETED PROCESSING.  THE TOTALS ARE AS FOLLOWS:'////1X,
     &   'NUMBER OF TAPES PROCESSED:',I3////1X,'PROGRAM STATISTICS',
     &                       31X,'ACCEPTED',5X,'REJECTED'/1X,70(1H-)/1X,
     &   'NUMBER OF USER SELECTED RUNS ON DISK',10X,':  ',I8,5X,I8/1X,
     &   'NUMBER OF TAPE DATA RECORDS READ AND PROCESSED:  ',I8,5X,
     &   I8/1X,'NUMBER OF WRITE ATTEMPTS TO DISK FILE <DATA>  :  ',
     &   I8,5X,I8////1X,'RUN NUMBER',5X,
     &   'FIRST DISK RECORD     LAST DISK RECORD'/1X,55(1H-))
304   FORMAT(4X,I4,14X,I5,16X,I5)
308   FORMAT(///A1,'FILE NAME',42X,':  <HEADER>',5X,'<DATA  >'/
     &   1X,75(1H-)/1X,'FILE SIZE (IN SECTORS)',29X,':  ',I8,5X,I8
     &   /1X,'NUMBER OF SECTORS AVAILABLE BEYOND CURRENT POSITION:',
     &   F10.2,3X,F10.2/1X,'NUMBER OF RECORDS AVAILABLE BEYOND ',
     &   'CURRENT POSITION:  ',I8,5X,I8)
310   FORMAT(/A1,'***** ERROR:  UNABLE TO OPEN <HEADER> AND/OR <DATA> ',
     &   'DISK FILES.'/1X,'*****',9X,'CHECK THE PARAMETERS IN THE ',
     &   'OPEN STATEMENTS.')
320   FORMAT(/A1,10X,'THE NEXT RUN NUMBER ON THE CURRENT DATA TAPE IS',
     &   I5,'.'/11X,'SELECT A NEW RANGE OF RUNS OR A NEW DATA TAPE.')
330   FORMAT(/10X,A1,'THE USER HAS REQUESTED THAT THE REMAINDER OF ',
     &   'DATA TAPE NUMBER',I3,' BE SKIPPED.')
340   FORMAT(/10X,A1,'THE USER HAS REQUESTED THAT RUN NUMBER',I5,
     &   ' BE SKIPPED.')
350   FORMAT(/A1,'***** WARNING:  THE MAXIMUM AVAILABLE RECORD NUMBER ',
     &   'FOR DISK FILE <DATA> HAS'/1X,'*****',11X,'BEEN EXCEEDED.  ',
     &   'NO ADDITIONAL DATA STORAGE FOR RUN NUMBER',I5/1X,'*****',11X,
     &   'WILL BE EXECUTED.')
360   FORMAT(/A1,'***** WARNING:  THE MAXIMUM AVAILABLE RECORD NUMBER ',
     &   'FOR DISK FILE <HEADER> HAS'/1X,'*****',11X,'BEEN EXCEEDED.  ',
     &   'RUN NUMBER',I5,' WILL NOT BE PROCESSED.')
370   FORMAT(/A1,'***** ERROR:  UNABLE TO MOUNT THE DISK.'/1X,'*****',
     &   9X,'ERROR CODE:  ',Z8,' (HEX).')
380   FORMAT(/A1,'***** ERROR:  UNABLE TO DISMOUNT THE DISK.  ',
     &   'REMOVE THE DISK MANUALLY.'/1X,'*****',9X,'ERROR CODE:  ',
     &   Z8,' (HEX).')
998   FORMAT(/A1,'***** WARNING:  AN ERROR HAS OCCURRED ON OUTPUT',
     &   ' TO THE SLU FILE.'/1X,'*****',11X,'TABULATE DISK RECORD ',
     &   'INFORMATION FOR EACH RUN BY HAND'/1X,'*****',11X,'IN CASE ',
     &   'PRINTED OUTPUT IS LOST.')
999   FORMAT(/A1,'***** WARNING:  AN UNDEFINED OUTPUT CODE HAS BEEN ',
     &   'PASSED TO SUBROUTINE OUTPUT.'/1X,'*****',11X,'CHECK ALL ',
     &   'CALLS TO SUBROUTINE OUTPUT FOR CORRECT OUTPUT CODES.')
      RETURN
      END
      SUBROUTINE FILES (IOP, IERR)
C
C             ******************************
C             *                            *
C             *     SUBROUTINE FILES       *
C             *                            *
C             ******************************
C
      LOGICAL PUBLIC, SHARE, MESSAGE
      CHARACTER*4 HSTAT, DSTAT
      COMMON / TOUNIT / NTAP, NSLO, NHDR, NDTA
      COMMON / DVSTAT / NSTATT, NSTATW, NSTATH, NSTATD, NSTATC, NSTATI,
     &                  NSTATHO, NSTATDO, NSTATHC, NSTATDC, NMOUNT,
     &                  NDSMNT
      COMMON / DISK   / NSZHEDR, NSZDATA, NBPSCTR, NBPHREC, NBPDREC,
     &                  NBPPLSE, RHSPACE, NRMHEDR, RDSPACE, NRMDATA,
     &                  NRMPLSE
      COMMON / MISC   / CTIMER, CTIMEA, PLSPTREC, PLSPDREC, TAOEOF,
     &                  NEWRANGE, ALL, MTIME(3), MDATE(3), NROLL, BELL
      DIMENSION MTBUF(192)
      DATA PUBLIC, SHARE, MESSAGE / 2 * .FALSE., .TRUE. /
      IF (IOP .EQ. 1) THEN
         CALL X←MOUNT ('DISC3', 'DM', PUBLIC, MESSAGE, SHARE, -10,
     &               MTBUF, NMOUNT)
         IF (NMOUNT .EQ. 0) THEN
            WRITE ('UT', 100, IOSTAT = NSTATC) BELL
            IF (NSTATC .EQ. 0) THEN
               READ  ('UT', '(2A4)', IOSTAT = NSTATC) HSTAT, DSTAT
               IF (NSTATC .EQ. 0) THEN
                  OPEN (NHDR, FILE = '@DISC3T(SHANNON)HEADER',
     &                  FILESIZE=NSZHEDR,  ACCESS = 'DIRECT',
     &                  FORM = 'UNFORMATTED', RECL = NBPHREC,
     &                  STATUS = HSTAT ,IOSTAT = NSTATHO)
                  OPEN (NDTA, FILE = '@DISC3T(SHANNON)DATA',
     &                  FILESIZE=NSZDATA,  ACCESS = 'DIRECT',
     &                  FORM = 'UNFORMATTED', RECL = NBPDREC,
     &                  STATUS = DSTAT, IOSTAT = NSTATDO)
                  IF (NSTATHO .NE. 0 .OR. NSTATDO .NE. 0) THEN
                     IERR = 20
                     CALL OUTPUT (31)
                  ELSE
                     IERR = 2
                  END IF
               ELSE
                  IERR = 10
                  CALL OUTPUT (2)
               END IF
            ELSE
               IERR = 9
               CALL OUTPUT (3)
            END IF
         ELSE
            IERR = 17
            CALL OUTPUT (37)
         END IF
      ELSE
         IF (NSTATHO .EQ. 0)
     &   CLOSE (NHDR , STATUS = 'KEEP', IOSTAT = NSTATHC)
         IF (NSTATDO .EQ. 0)
     &   CLOSE (NDTA, STATUS = 'KEEP', IOSTAT = NSTATDC)
         IF (NSTATHC .NE. 0 .OR. NSTATDC .NE. 0) THEN
            IERR = 21
            CALL OUTPUT (29)
         END IF
         CALL X←DISMNT ('DISC3', NDSMNT)
         IF (NDSMNT .NE. 0) CALL OUTPUT (38)
      END IF
      RETURN
C*****FORMAT STATEMENTS:
100   FORMAT(/A1,'ENTER>  FILE STATUS FOR FILE <HEADER> AND FILE ',
     &        '<DATA>'/9X,'(TYPE <NEW > OR <OLD > LEFT JUSTIFIED AND ',
     &        'IN (2A4) FORMAT)')
      END
      SUBROUTINE DISKCHEK (IRUN, IDREC, ICHEK, IERR)
C
C               *******************************
C               *                             *
C               *     SUBROUTINE DISKCHEK     *
C               *                             *
C               *******************************
C
      COMMON / DVSTAT / NSTATT, NSTATW, NSTATH, NSTATD, NSTATC, NSTATI,
     &                  NSTATHO, NSTATDO, NSTATHC, NSTATDC, NMOUNT,
     &                  NDSMNT
      COMMON / DISK   / NSZHEDR, NSZDATA, NBPSCTR, NBPHREC, NBPDREC,
     &                  NBPPLSE, RHSPACE, NRMHEDR, RDSPACE, NRMDATA,
     &                  NRMPLSE
      COMMON / MISC   / CTIMER, CTIMEA, PLSPTREC, PLSPDREC, TXDEUF,
     &                  NEWRANGE, ALL, MTIME(3), MDATE(3), NROLL, BELL
      DATA IHM, IDM / 0, 0/
      SPHR = FLOAT(NBPHREC) / FLOAT(NBPSCTR)
      SPDR = FLOAT(NBPDREC) / FLOAT(NBPSCTR)
      IF (ICHEK .EQ. 1) THEN
         RHSPACE = FLOAT(NSZHEDR) - FLOAT(IRUN - 1) * SPHR
         NRMHEDR = INT((FLOAT(NSZHEDR) / SPHR) + 1.E-3) - (IRUN - 1)
         RDSPACE = FLOAT(NSZDATA) - FLOAT(IDREC - 1) * SPDR
         NRMDATA = INT((RDSPACE / SPDR) + 1.E-3)
         NRMPLSE = INT((FLOAT(NRMDATA * NBPDREC) / FLOAT(NBPPLSE))
     &                 + 1.E-3)
         IF (NRMHEDR .LT. 1) THEN
            RHSPACE = 0.
            NRMHEDR = 0
            IERR = 22
            CALL OUTPUT (36)
            WRITE ('UT', 100, IOSTAT = NSTATC) BELL, IRUN
         ELSE IF (NRMHEDR .GE. 1 .AND. NRMHEDR .LE. 5) THEN
            IF (NRMHEDR .LE. 5 - IHM) THEN
               WRITE ('UT', 200, IOSTAT = NSTATC) BELL, RHSPACE,
     &                                           NRMHEDR+IRUN-1
               IHM = IHM + 1
            END IF
         END IF
      ELSE
         RHSPACE = FLOAT(NSZHEDR) - FLOAT(IRUN) * SPHR
      NRMHEDR = INT((RHSPACE / SPHR) + 1.E-3)
      RDSPACE = FLOAT(NSZDATA) - FLOAT(IDREC - 1) * SPDR
      NRMDATA = INT((FLOAT(NSZDATA) / SPDR) + 1.E-3) - (IDREC - 1)
      NRMPLSE = INT((FLOAT(NRMDATA * NBPDREC) / FLOAT(NBPPLSE))
     &                 + 1.E-3)
      IF (NRMDATA .LT. 1) THEN
         RDSPACE = 0.
         NRMDATA = NRMPLSE = 0
         IERR = 23
         CALL OUTPUT (35)
         WRITE ('UT', 300, IOSTAT = NSTATC) BELL, IDREC
      ELSE IF (NRMDATA .GE. 1 .AND. NRMDATA .LE. 500) THEN
         IF (NRMDATA .LE. 500 - 100*IDM) THEN
            WRITE ('UT', 400, IOSTAT = NSTATC) BELL,RDSPACE, NRMDATA,
     &                                           NRMPLSE
            IDM = IDM + 1
         END IF
      END IF
    END IF
    RETURN
C
C   FORMAT STATEMENTS:
100 FORMAT(/1X,A1,'***** WARNING:  DISK FILE <HEADER> IS NOT LARGE ',
   &   'ENOUGH TO STORE HEADER'/1X,'*****',11X,'INFORMATION FOR ',
   &   'RUN NUMBER',I5,'.')
200 FORMAT(/1X,A1,'***** WARNING:  THERE ARE ONLY',F7.2,' SECTORS ',
   &   'AVAILABLE IN DISK FILE <HEADER>.'/1X,'*****',11X,'THE LAST',
   &   ' RUN THAT CAN BE PROCESSED IS NUMBER',I5,'.')
300 FORMAT(/1X,A1,'***** WARNING:  DISK FILE <DATA> IS NOT LARGE ',
   &   'ENOUGH TO STORE DATA'/1X,'*****',11X,'RECORD NUMBER',I6,'.')
400 FORMAT(/1X,A1,'***** WARNING:  THERE ARE ONLY',F7.2,' SECTORS ',
   &   'AVAILABLE IN DISK FILE <DATA>.'/1X,'*****',11X,'ONLY',I5,
   &   ' MORE DATA RECORDS (',I8,' PULSES) CAN BE STORED.')
    END
$A1 L01=SRILIB,,U
$A1 D01=SRTDIR,,U
$EXECUTE CATALOG
OPTION PROMPT
A2 6=SLU,2000
A3 1=M91000,INPT,,U
BUILD LOAD U NOM
$EOJ
$$
```

(THIS PAGE INTENTIONALLY LEFT BLANK)

APPENDIX B

DATA ANALYSIS ROUTINE

```
$JOB A3366000 COHEN,MN
$OPTION 1 3 4 5
$EXECUTE FORT77
      PROGRAM DATAR
C
C
C
C    THIS PROGRAM IS WRITTEN TO REDUCE THE IPAR DATA RECORDED IN PANAMA
C    CITY IN DECEMBER OF 1982.  THE DATA THAT WAS COLLECTED ON THAT FIELD
C    TRIP WAS RECORDED ON MAGNETIC TAPES.  THIS DATA WAS THEN TRANSFERED
C    TO A HARD DISK UNIT ON THE GT SEL.  THIS PROGRAM READS THAT DISK AND
C    ANALYZES THE DATA AS SPECIFIED BY THE USER.
C
C    THE DATA MENTIONED ABOVE CONSISTS OF THE RETURN SIGNAL OF A
C    BINARY PHASE ENCODED TRANSMIT PULSE.  THESE PULSES ARE ARRANGED
C    IN 'RUNS' OF 2000 OR SO RETURN PULSES.  EACH PULSE CONSISTS OF
C    32 2-BIT VALUES INDICATING POLARIZATION.  THIS PROGRAM READS THOSE
C    RETURNS, CORRELATES THE RETURN SIGNAL WITH THE TRANSMITTED SIGNAL,
C    AND ANALYZES THE SPECIFIED SUBSET OR SUBSETS OF THE CORRELATIONS.
C    THE USER SPECIFIES THE SUBSETS, CHOOSING FROM THE FOLLOWING:
C
C       (1)  THE CORRELATION OF ALL RETURN SIGNALS.
C       (2)  THE OUTPUT OF ONE PARTICULAR BIN.
C       (3)  THE PEAK OF THE RETURN SIGNAL.
C       (4)  THE POSTIION AT WHICH THE PEAK OCCURED.
C
C    THE USER THEN SPECIFIES THE ANALYSIS TO BE PERFORMED ON THE CHOSEN
C    DATA SUBSET.  THE CHOICES ARE THE FOLLOWING:
C
C       (1)  STATISITCAL ANALYSIS (AVERAGE AND STANDARD DEVIATION).
C       (2)  HISTOGRAM ANALYSIS (OCCURENCES OF A GIVEN LEVEL OUTPUT).
C       (3)  AUTOCORRELATION OF OUTPUT (GIVES INDICATION OF DECORRELATION
C            TIME OF TARGET.  THIS IS MEANINGFUL ONLY FOR DATA SETS (2) O
C            (3) ABOVE).
C
C    THEN THE OUPUT FORM MUST BE SPECIFIED.  THE CHOICES ARE:
C
C       (1)  PRINT TO PAPER.
C       (2)  PRINT TO THE TERMINAL.
C       (3)  PLOT THE DATA.
C
C    NOT ALL COMBINATIONS ARE POSSIBLE.  FOR EXAMPLE, IF DATA TYPE (2)
C    IS SPECIFIED ALONG WITH THE STATISTICAL ANALYSIS AND PLOT OUTPUT
C    OPTION, THE DATA WILL NOT BE PLOTTED SINCE THE STATISTICS WOULD
C    CONSIST OF ONLY ONE AVERAGE AND STANDARD DEVIATION, WHICH WOULD NOT
C    PRODUCE A REASONABLE PLOT.  SO IN CASES WHERE SEVERAL DIFFERENT
C    DATA TYPES, ANALYSES, OR OUTPUT OPTIONS ARE SPECIFIED ALL REASON-
C    ABLE COMBINATIONS ARE PERFORMED, AND THE UNREASONABLE COMBINATIONS
C    ARE IGNORED.
C
C
C    ** PROGRAM ORGANIZATION **
C    DUE TO THE AMOUNT OF DATA TO BE ANALYZED, THE DATA IS READ IN
C    TO AN ARRAY IN BLOCKS OF 96 RETURNS.  ALL IMPORTANT INFORMATION
C    IS EXTRACTED FROM THIS BLOCK OF RETURNS, AND THEN THE NEXT BLOCK
C    IS READ IN.  IN THIS MANNER AT NO TIME IS IT NECESSARY TO HAVE
C    ALL 2000 OR SO RETURNS IN THE COMPUTER MEMORY SIMULTANEOUSLY.
```

```
C     THE MAIN PROGRAM CONSISTS OF TWO LOOPS, ONE NESTED INSIDE THE
C     OTHER.  THE OUTSIDE LOOP ALLOWS THE SAME ANALYSES TO BE PERFORMED
C     ON MANY SUCCESSIVE RUNS (WHERE A 'RUN' IS DEFINED ABOVE).
C     THE INNER LOOP PERFORMS THE ANALYSES FOR THE SPECIFIED PULSE
C     NUMBERS WITHIN A GIVEN RUN.  WITHIN THIS LOOP THE SUBROUTINES
C     STAT, HIST, AND AUTOC ARE CALLED.  THESE ROUTINES SAVE THE
C     INFORMATION EXTRACTED FROM A GIVEN RUN AND PULSE NECESSARY FOR
C     THE CALCULATIONS OF THEIR RESPECTIVE ANALYSES.  AFTER ALL SPEC-
C     IFIED PULSES FOR A GIVEN RUN ARE ANALYZED, THE SUBROUTINE
C     WRAPUP IS CALLED TO TAKE CARE OF THE DETAILS OF THE ANALYSES
C     WHICH COULD NOT HAVE BEEN DONE UNTIL ALL PULSES WERE
C     CONSIDERED.
C
C     THE LOGIC OF MOST OF THIS CODE SHOULD BE EASILY UNDERSTOOD
C     BY ANOTHER PROGRAMMER, THOUGH ONE SECTION OF THE MAIN
C     PROGRAM IS CONFUSING.  THE CONFUSION ARISES FROM THE FACT
C     THAT NUMPREC(96 FOR IPAR) RETURNS ARE READ AT ONCE FROM
C     THE DISK, AND FROM THE UNCERTAINITY CONCERNING THE VALIDITY
C     OF THOSE RETURNS.  DUE TO THESE FACTORS THREE VARIABLES OF
C     DISTINCT BUT SIMILAR (AND SO CONFUSING) USES WERE REQUIRED.
C     THOUGH THESE VARIABLES ARE DESCRIBED BELOW, IT MIGHT HELP
C     TO EXPLAIN THEIR DIFFERENCES HERE.  INDP IS A LOOP VARIABLE
C     THAT EXECUTES THE ANALYSIS SUBROUTINES ONCE FOR EACH GOOD
C     PULSE RETURNED FROM GTPULS.  THE ACTUAL NUMBER OF THE PULSE
C     IS NOT REPRESENTED BY INDP, ONLY THE NUMBER OF THE PULSE
C     WITHIN THE NUMPREC PULSE BLOCK JUST READ IN.  INDPULS, ON
C     THE OTHER HAND, REPRESENTS THE ACTUAL PULSE NUMBER OF THE
C     FIRST PULSE IN THE PRESENT BLOCK OF RETURNS.  THIS IS NEEDED
C     TO READ THE APPROPRIATE BLOCK OF RETURNS FROM THE DISK.
C     FINALLY, ISOFAR REPRESENTS THE TOTAL NUMBER OF GOOD PULSES
C     PROCESSED OR BEING PROCESSED SO FAR.  FOR EXAMPLE, IF THE
C     PROGRAM IS PRESENTLY PROCESSING THE 200TH PULSE IN A GIVEN
C     RUN, INDP SHOULD BE 8 SINCE THE 200TH PULSE IS THE 8TH
C     PULSE IN THE THIRD BLOCK OF RETURNS (THE BLOCK NUMBER IS
C     IRRELEVANT TO INDP).  INDPULS WOULD BE 193, THE NUBMER OF
C     THE FIRST PULSE IN THE THIRD BLOCK OF RETURNS, AND ISOFAR
C     WOULD BE THE NUMBER OF GOOD PULSES PROCESSED OR BEING
C     PROCESSED SO FAR, SAY 195 IF 5 OF THE PULSES FROM PULSE
C     ONE TO PULSE 200 HAD BEEN BAD.
C
C     ** VARIABLES **
C
C     NOTE: VARIABLES ENDING IN 'YN' ARE INTENDED TO BE ASSIGNED THE
C     VALUE Y OR N.  THESE VARIABLES INDICATE WHETHER OR NOT THAT
C     SPECIFIC ITEM IS REQUESTED BY THE USER.
C
C     ALBINYN    ALL BINS TO BE ANALYZED?
C     ALBNSC     THE SUM OF THE CORRELATOR OUTPUT FOR ALL BINS.
C     ALBNSC2    THE SUM OF THE CORRELATOR OUTPUT SQUARED FOR ALL BINS.
C     AUTOBIN    THE SEQUENCE CONSISTING OF THE OUTPUT OF ONE
C                PARTICULAR BIN.
C     AUTOPK     THE AUTOCORRELATION OF THE SEQUENCE CONSITING OF THE
C                PEAK OUTPUT FOR EACH PULSE UNDER CONSIDERATION.
C     AUTOYN     PERFORM AUTOCORRELATION ANALYSIS?
C     AVGCOR     AVERAGE CORRELATOR OUTPUT FOR EACH BIN.
C     BAUTO      AUTOCORRELION OF THE SEQUENCE CONSISTING OF THE
C                OUTPUT OF ONE PARTICULAR (USER SPECIFIED) BIN.
```

```
C               (THE AUTOCORRELATION OF AUTOBIN).
C     BAVG      THE AVERAGE OUTPUT OF ONE PARTICULAR BIN.
C     BSIG      THE STANDARD DEVIATION OF THE OUTPUT OF ONE PARTICULAR
C               BIN.
C     CODE      USED ONLY IN SUBROUTINE CORREL, CODE IS THE CODE TO
C               BE CORRELATED.  USUSALLY THIS IS ONE OF THE SEQUENCES
C               IN 'RETURNS'.  WHEN CORREL1 IS CALED FROM SUBROUTINE
C               WRAPUP, HOWEVER, CODE IS THE SEQUENCE SPECIFIED FOR
C               THE AUTOCORRELATION ANALYSIS.
C     CORR      CALCULATED IN SUBROUTINE CORREL.  THIS VARIABLE REPRESENTS
C               THE CORRELATION OF A CODE THROUGH A FILTER, USUALLY 'CODE'
C               THROUGH 'FILTER', BUT ALSO REPRESENTS THE AUTOCORRELATION
C               OF THE DATA SPECIFIED FOR USE IN THE AUTOCORRELATION
C               ANALYSIS IF CALLED FROM ROUTINE WRAPUP.
C     FILTER    THE FILTER, OR TRANSMIT SIGNAL, THIS IS USED AS THE REFER-
C               ENCE OF CORRELATION OF THE VARIABLE 'CODE'.  IT IS READ FR
C               THE HEADER INFORMATION BY ROUTINE GETHEAD.
C     HISTYN    HISTOGRAM ANALYSIS OPTION CHOSEN?
C     I         LOOP VARIABLE
C     IBINNO    SPECIFIED BIN NUMBER.
C     IERR      ERROR FLAG GENERATED IN READ TO DISK.
C     IHSTBIN   THIS IS THE HISTOGRAM CALCULATED FOR ONE PARTICULAR BIN.
C     IHSTPK    THIS IS THE HISTOGRAM FOR THE PEAK OUTPUT OF THE CORRELATO
C     IHSTPOS   THE HISTOGRAM OF THE POSITION OF THE PEAK OUTPUT OF THE
C     INDP      INDEX USED TO ANALYZE EACH PULSE.
C     INDPULS   INDEX FOR THE PULSE LOOP (INNER LOOP AS DESCRIBED ABOVE).
C     INDRUN    INDEX FOR THE RUN LOOP (OUTER LOOP AS DESCRIBED ABOVE).
C     IPULFN    LAST PULSE NUMBER SPECIFIED.
C     IPULST    FIRST PULSE NUMBER SPECIFIED.
C     IRUNFN    LAST RUN NUMBER SPECIFIED.
C     IRUNST    FIRST RUN NUMBER SPECIFIED.
C     ISOFAR    THE VARIABLE KEEPING COUNT OF HOW MANY PULSES LOOKED AT
C               SO FAR HAVE BEEN VALID.
C     ISTART    THIS IS A POINTER TO THE STARTING RECORD NUMBER
C               FOR THIS RUN.
C     ISTOP     POINTER TO THE STOPING RECORD NUMBER FOR THIS RUN.
C     LENFILT   LENGTH OF THE FILTER (FROM HEADER).
C     LENGTH    THE LENGTH OF THE INPUT CODE, SHOULD BE 32.
C     NOMORE    LOGICAL VARIABLE INDICATING THAT THE LAST PULSE IN A
C               GIVEN RUN HAS BEEN READ.  THIS IS USED IN THE CASE THAT
C               MORE PULSES HAVE BEEN REQUESTED THAN EXIST FOR THAT RUN,
C               IN WHICH CASE 'NOMORE' RETURNS FROM SUBROUTINE GTPULS SET
C               TRUE.
C     NONE      BOOLEAN VARIABLE INDICATING THAT THERE WAS NO SUCH RUN.
C     NUMGOOD   THE NUMBER OF GOOD PULSES IN THE MOST RECENTLY READ BLOCK,
C               CALCULATED IN GTPULS.
C     NUMPREC   THE NUMBER OF PULSES PER DISK RECORD.  SHOULD BE 96.
C     ONEBNS    THE SUM OF THE CORRELATOR OUTPUT FOR ONE PARTICULAR BIN.
C     ONEBNS2   THE SUM OF THE SQUARE OF THE CORRELATOR OUTPUT FOR ONE
C               PARTICULAR BIN.
C     ONEBNYN   ANALYZE ONE PARTICULAR BIN?
C     PAUTO     THE SEQUENCE CONSISTING OF THE PEAK CORRELATOR OUTPUT FOR
C               EACH PULSE UNDER CONSIDERATION, TO BE USED IN SUBROUTINE
C               AUTOC.
C     PCTGOOD   INDICATES WHAT PERCENTAGE OF THE REQUESTED PULSES WERE
C               GOOD.  PULSES THAT ARE NON-EXISTENT WERE NOT CONSIDERED.
C     PEAK      CALCULATED IN SUBROUTINE CORREL, PEAK IS THE PEAK OUTPUT
```

```
C                 OF THE CORRELATOR FOR THE GIVEN PULSE.
C     PEAKYN      ANALYZE THE PEAK CORRELATOR OUTPUT?
C     PKAVG       THE AVERAGE PEAK CORRELATOR OUTPUT.
C     PKPOS       THE POSITION OF THE PEAK CORRELATOR OUTPUT.
C     PKPOSAV     THE AVERAGE POSITION OF THE PEAK CORRELATOR OUTPUT.
C     PKPOSSG     THE STANDARD DEVIATION OF THE PEAK CORRELATOR OUTPUT POSIT
C     PKPOSYN     ANALYZE THE PEAK POSITION?
C     PKS         SUM OF THE PEAK CORRELATOR OUTPUTS, TO BE USED IN CALCULAT
C                 AVERAGE PEAK.
C     PKSIG       STANDARD DEVIATION OF PEAK CORRELATOR OUTPUT.
C     PKS2        SUM OF THE PEAK CORRELATOR OUTPUTS, TO BE USED IN CALCULAT
C                 STANDARD DEVIATION OF PEAK (PKSIG).
C     PLDAYN      PLOT DATA?
C     POSS        SUM OF PEAK POSITIONS.
C     POSS2       SUM OF PEAK POSITIONS SQUARED.
C     PRPAYN      PRINT DATA TO PAPER?
C     PRSCYN      PRINT DATA TO SCREEN?
C     PSAVG       AVERAGE PEAK SIDELOBE LEVEL.
C     PSSIG       STANDARD DEVIATION OF PSL.
C     PSL         PEAK SIDELOBE LEVEL, IN DB WHEN EXITING CORREL.
C     PSLCOM      PEAK SIDELOBE LEVEL OF AVERAGE CORRELATOR OUTPUT.
C     RIAVG       AVERAGE INTEGRATED SIDELOBE LEVEL.
C     RISIG       STANDARD DEVIATION OF ISL.
C     RISLCOM     INTEGRATED SIDELOBE LEVEL OF AVERAGE CORRELATOR OUTPUT.
C     RLPGCOM     LOSS IN PROCESSING GAIN OF AVERAGE CORRELATOR OUTPUT.
C     RNEW        INDICATES THAT THE USER WANTS AN ENTIRELY NEW DATA SET.
C     RETURNS     THE SET OF 96 RETURN SIGNALS (CODES) TO BE ANALYZED.
C     RISL        INTEGRATED SIDELOBE LEVEL, IN DB WHEN EXITING CORREL.
C     RLPG        THE LOSS IN PROCESSING GAIN, IN DB.
C     SIGCOR      STANDARD DEVIATION OF THE CORRELATOR OUTPUT FOR ALL BINS.
C     STATYN      STATISITICS ANALYSIS REQUESTED?
C     STOP        INDICATES THAT THE USER WANTS TO STOP.
C     SUMISL      THE SUM OF THE INTEGRATED SIDELOBE LEVELS FOR USE IN
C                 CALCULATING AVERAGE ISL.
C     SUMISL2     SUM OF INTEGRATED SIDELOBE LEVELS SQUARED.
C     SUMLPG      THE SUM OF THE LOSSES IN PROCESSING GAIN.
C     SUMLPG2     THE SUM OF THE SQUARE OF THE LPG VALUES.
C     SUMPSL      THE SUM OF THE PEAK SIDELOBE LEVELS FOR USE IN CALCULATING
C                 THE AVERAGE PSL.
C     SUMPLS2     SUM OF PEAK SIDELOBE LEVELS SQUARED.
C     TCODE       TEMPORARY STORAGE FOR THE CODE TO BE USED IN SUBROUTINE
C                 CORREL, WHICH EXPECTS A ONE DIMENSIONAL ARRAY.
C
C
C
C     ** SUBROUTINES **
C
C     INITIAL     CLEARS ALL ARRAY VALUES TO ZERO.  THIS IS DONE BECAUSE
C                 MANY ARRAYS ARE CALCULATED INCREMENTALLY THROUGHOUT THE
C                 EXECUTION OF THE PROGRAM.
C     SLECTIO     SELECT INPUT/OUTPUT.  THIS ROUTINE ALLOWS THE USER TO
C                 SPECIFY THE INPUT DATA SETS, THE ANALYSES TO BE PERFORMED,
C                 AND THE OUTPUT FORM.
C     ALTERIO     THIS ALLOWS THE USER TO ALTER PART OR ALL OF THE INPUTS
C                 WITHOUT TYPING ALL OF THEM IN AGAIN.
C     GETHEAD     THIS ROUTINE GETS THE HEADER INFORMATION OFF THE DISK.
C     GTPULS      THIS ROUTINE GETS THE SPECIFIED BLOCK OF 96 PULSES FROM
```

```
C                THE DISK, AND RETURNS WITH THOSE PULSES AND A COUNT OF HOW
C                MANY OF THE PULSES WERE GOOD (CONTAINED NO ILLEGAL CHAR-
C                ACTERS).
C    CORREL      SUBROUTINE CORREL CALCULATES THE CORRELATION OF A CODE
C                THROUGH A FILTER.  IT ALSO CALCULATES THE PEAK SIDELOBE
C                AND INTEGRATED SIDELOBE LEVELS.
C    CORREL1     SIMILAR TO CORREL EXCEPT FOR THE ARRAY SIZE.  THIS
C                IS TO CORRELATE THE LARGER ARRAYS FOR THE AUTO-
C                CORRELATION ANALYSIS.
C    STAT        SUBROUTINE STAT UPDATES THE STATITSICS ARRAYS.
C    HIST        SUBROUTINE HIST UPDATES THE HISTOGRAM ARRAYS.
C    AUTOC       SUBROUTINE AUTOC UPDATES THE AUTOCORRELATION ARRAYS.
C    WRAPUP      THIS ROUTINE TAKES CARE OF THE LOOSE ENDS NOT TAKEN
C                CARE OF IN STAT, HIST, OR AUTOC.
C    DATAOUT     THIS TS THE GENERAL OUTPUT ROUTINE.
C    STADEC      EXTRACTED FROM MIKE SHANNON'S DISCLD ROUTINE
C    TABLE       EXTRACTED FROM MIKE SHANNON'S DISCLD ROUTINE
C    UNPACK      EXTRACTED FROM MIKE SHANNON'S DISCLD ROUTINE
C    IBITF       EXTRACTED FROM MIKE SHANNON'S DISCLD ROUTINE
C
C
C
      LOGICAL NOMORE,NONE,PUBLIC,MESSAGE,SHARE
      DIMENSION CORR(63),FILTER(32),ICODE(32),MTBUF(192)
      COMMON/ARRAYS/ ALBNSC(2000),ALBNSC2(2000),RETURNS(96,32),
     &   AVGCOR(63),SIGCOR(63),IHSTBIN(65),IHSTPK(65),IHSTPUS(63),
     &   BAUTO(2000),PAUTO(2000),AUTOBIN(4000),AUTOPK(4000)
      DATA PUBLIC,MESSAGE,SHARE/.FALSE.,.TRUE.,.TRUE./
         CALL X←MOUNT ('DISC3', 'DM', PUBLIC, MESSAGE, SHARE, -10,
     &                 MTBUF, NMOUNT)
                  OPEN (2, FILE = '@DISC3↑(SHANNON)HEADER',
     &                  FILESIZE=50,  ACCESS = 'DIRECT',
     &                  FORM = 'UNFORMATTED', RECL = 20,
     &                  STATUS = 'OLD' )
                  OPEN (3, FILE = '@DISC3↑(SHANNON)DATA',
     &                  FILESIZE=50000,  ACCESS = 'DIRECT',
     &                  FORM = 'UNFORMATTED', RECL = 768,
     &                  STATUS = 'OLD')
   40 CALL SLECTIO(IRUNST,IRUNFN,IPULST,IPULFN,ALBINYN,PEAKYN,
     &   PKPUSYN,ONEBNYN,IBINNU,PRPAYN,PLDAYN,STATYN,HISTYN,AUTOYN,
     &   PRSCYN)
   50 CALL ALTERIO(IRUNST,IRUNFN,IPULST,IPULFN,ALBINYN,PEAKYN,
     &   PKPUSYN,ONEBNYN,IBINNU,PRPAYN,PLDAYN,STATYN,HISTYN,AUTOYN,
     &   PRSCYN,RNEW,STOP)
      IF(RNEW.EQ.'Y')GOTO40
      IF(STOP.EQ.'Y')GOTO2000
      IF (IRUNFN .GT. 1500) IRUNFN = 1500
      DO 1000 INORUN = IRUNST,IRUNFN
      CALL INITIAL(SUMPSL,SUMISL,PKS,PKS2,POSS,POSS2,ONEBNS,ONEBNS2,
     &   SUMPSL2,SUMISL2,SUMLPG,SUMLPG2,FILTER,CORR)
      NONE = .FALSE.
      CALL GETHEAD(FILTER,LENFILT,INORUN,ISTART,ISTOP,NONE)
      IF(NONE) GOTO 1000
      ISOFAR = 0
      NUMPREC = 96
      LENGTH = 32
      INOPULS = IPULST
```

```
  100 CALL GTPULS(INDRUN,INDPULS,NUMGOOD,NUMORE,ISTART,ISTOP,LENFILT)
      IF(NUMGOOD.LT.1) GOTO 300
      DO 250 INDP = 1,NUMGOOD
      ISOFAR = ISOFAR + 1
      DO 125 I = 1,LENGTH
      TCODE(I) = RETURNS(INDP,I)
  125 CONTINUE
      CALL CORREL(CORR,TCODE,FILTER,LENGTH,PEAK,PKPOS,PSL,RISL,LENFILT,
     &   RLPG)
      CALL STAT(LENGTH,ALBINYN,PEAKYN,CORR,
     &   PEAK,PKS,PKS2,PKPOSYN,POSS,POSS2,PKPOS,ONEBNYN,ONEBNS,
     &   ONEBNS2,IBINNO,PSL,RISL,SUMPSL,SUMISL,SUMPSL2,SUMISL2,RLPG,
     &   SUMLPG,SUMLPG2)
      CALL HIST(LENGTH,ONEBNYN,IBINNO,PEAKYN,PEAK,CORR,
     &   PKPOSYN,PKPOS)
      CALL AUTOC(CORR,PEAK,PEAKYN,ONEBNYN,IBINNO,ISOFAR)
  250 CONTINUE
  300 IF(NUMORE) GOTO 500
      INDPULS = INDPULS + NUMPREC
      IF(INDPULS.LT.IPULFN) GOTO 100
  500 CONTINUE
      PCTGOOD = 100.0 * FLOAT(ISOFAR)/(INDPULS-1)
      CALL WRAPUP(LENGTH,ISOFAR,STATYN,ONEBNYN,BAVG,BSIG,ONEBNS,
     &   ONEBNS2,PEAKYN,PKAVG,PKSIG,PKS,PKS2,PKPOSYN,PKPOSAV,PKPOSSG,
     &   POSS,POSS2,AUTOYN,SUMPSL,SUMISL,SUMPSL2,SUMISL2,PSAVG,PSSIG,
     &   RIAVG,RISIG,SUMLPG,SUMLPG2,RLPGAVG,RLPGSIG,LENFILT,PSLCOM,
     &   RISLCOM,RLPGCOM)
      CALL DATAOUT(STATYN,HISTYN,AUTOYN,ALBINYN,ONEBNYN,IBINNO,
     &   PEAKYN,PKPOSYN,PKAVG,PKSIG,PKPOSAV,PKPOSSG,BAVG,BSIG,ISOFAR,
     &   LENGTH,PSAVG,PSSIG,RIAVG,RISIG,RLPGAVG,RLPGSIG,PSLCOM,RISLCOM,
     &   RLPGCOM,INDRUN,PRPAYN,PCTGOOD)
 1000 CONTINUE
      GOTO 50
 2000    CLOSE (2 , STATUS = 'KEEP')
         CLOSE (3, STATUS = 'KEEP')
         CALL X$DISMNT ('DISC3', NDSMNT)
      END
C
C
C
C**************************** SUBROUTINE INITIAL **********************
      SUBROUTINE INITIAL(SUMPSL,SUMISL,PKS,PKS2,POSS,POSS2,ONEBNS,
     &   ONEBNS2,SUMPSL2,SUMISL2,SUMLPG,SUMLPG2,FILTER,CORR)
C
C  THIS ROUTINE CLEARS ALL THE ARRAYS IN THE COMMON BLOCK, AND A FEW OF
C  THE SIMPLE VARIABLES USED IN SUBROUTINE STAT.
C
      DIMENSION FILTER(32),CORR(63)
      COMMON/ARRAYS/ ALBNSC(2000),ALBNSC2(2000),RETURNS(96,32),
     &   AVGCOR(63),SIGCOR(65),IHSTBIN(55),IHSTPK(65),IHSTPOS(63),
     &   RAUTO(2000),PAUTO(2000),AUTOBIN(4000),AUTOPK(4000)
      SUMPSL = 0.
      SUMPSL2 = 0.
      SUMISL = 0.
      SUMISL2 = 0.
      SUMLPG = 0.
      SUMLPG2 = 0.
      PKS = 0.
      PKS2 = 0.
      POSS = 0.
      POSS2 = 0.
      ONEBNS = 0.
      ONEBNS2 = 0.
      DO 10 I=1,32
      FILTER(I) = 0
   10 CONTINUE
      DO 20 I=1,63
      CORR(I) = 0
      AVGCOR(I) = 0
      SIGCOR(I) = 0
      IHSTBIN(I) = 0
      IHSTPK(I) = 0
      IHSTPOS(I) = 0
      IHSTBIN(64) = 0
      IHSTBIN(65) = 0
      IHSTPK (64) = 0
      IHSTPK (65) = 0
   20 CONTINUE
      DO 30 I=1,2000
      ALBNSC(I) = 0
      ALBNSC2(I) = 0
      BAUTO(I) = 0
      PAUTO(I) = 0
   30 CONTINUE
      DO 40 I=1,4000
      AUTOBIN(I) = 0
      AUTOPK(I) = 0
   40 CONTINUE
      RETURN
      END
C
C
C
C********************* SUBROUTINE SLECTIO ************************
      SUBROUTINE SLECTIO(IRUNST,IRUNFN,IPULST,IPULFN,ALBINYN,PEAKYN,
     &   PKPOSYN,ONEBNYN,IBINNO,PRPAYN,PLOAYN,STATYN,HISTYN,AUTOYN,
     &   PRSCYN)
C
C  THIS ROUTINE READS FILE INPUT TO DETERMINE WHAT THE DESIRED INPUT/
C  OUTPUT IS.
C
      LENGTH = 32
      WRITE('OT',10)
   10 FORMAT(' *** SELECT INPUT QUANTITIES ***')
      WRITE('OT',*)
      WRITE('OT',20)
   20 FORMAT(' INPUT RUN NUMBER RANGE (FOR EXAMPLE 35,40)')
      READ('OT',*)IRUNST,IRUNFN
      WRITE('OT',30)
   30 FORMAT(' INPUT PULSE NUMBER RANGE')
      READ('OT',*)IPULST,IPULFN
      WRITE('OT',40)
   40 FORMAT(' CHOOSE ONE OR MORE OF THE FOLLOWING:')
      WRITE('OT',50)
   50 FORMAT(' ALL BINS? (Y OR N)')
      READ('UT',51)ALBINYN
   51 FORMAT(A1)
      WRITE('UT',60)
   60 FORMAT(' PEAK? (Y OR N)')
      READ('UT',51)PEAKYN
      WRITE('UT',70)
   70 FORMAT(' PEAK POSITION? (Y OR NO)')
      READ('UT',51)PKPOSYN
      WRITE('UT',80)
   80 FORMAT(' ONE PARTICULAR BIN? (Y OR NO)')
      READ('UT',51)ONEFBNYN
      IF(ONEFBNYN .EQ. 'N') GOTO 100
      WRITE('UT',90)
   90 FORMAT(' WHICH BIN?')
      READ('UT',*)IBINNO
      IBINNO = IBINNO + LENGTH
  100 WRITE('UT',*)
      WRITE('UT',110)
  110 FORMAT(' *** SELECT OUTPUT DEVICES ***')
      WRITE('UT',*)
      WRITE('UT',120)
  120 FORMAT(' PRINT TO SCREEN? (Y OR N)')
      READ('UT',51)PRSCYN
      WRITE('UT',130)
  130 FORMAT(' PRINT TO PAPER? (Y OR N)')
      READ('UT',51)PRPAYN
      WRITE('UT',140)
  140 FORMAT(' PLOT DATA? (Y OR N)')
      READ('UT',51)PLDAYN
      WRITE('UT',*)
      WRITE('UT',150)
  150 FORMAT(' *** SELECT FUNCTIONS ***')
      WRITE('UT',*)
      WRITE('UT',160)
  160 FORMAT(' STATISTICS? (Y OR N)')
      READ('UT',51)STATYN
      WRITE('UT',170)
  170 FORMAT(' HISTOGRAM? (Y OR N)')
      READ('UT',51)HISTYN
      WRITE('UT',180)
  180 FORMAT(' AUTOCORRELATION? (Y OR N)')
      READ('UT',51)AUTOYN
      WRITE('UT',190)
  190 FORMAT(' *** DONE WITH SELECTION ROUTINE ***')
      RETURN
      END
C
C
C
C************************* SUBROUTINE ALTERIO ***************************
      SUBROUTINE ALTERIO(IRUNST,IRUNFN,IPULST,IPULFN,ALBINYN,PEAKYN,
     &   PKPOSYN,ONEFBNYN,IBINNO,PRPAYN,PLDAYN,STATYN,HISTYN,AUTOYN,
     &   PRSCYN,RNEW,STOP)
C
C     THIS ROUTINE ALLOWS THE USER TO ALTER THE INPUTS PREVIOUSLY ENTERED
C
```

```
      LENGTH = 32
      IBINNO = IBINNO - LENGTH
      WRITE('UT',*)
      WRITE('UT',*)
      WRITE('UT',*)
    5 WRITE('UT',10)
   10 FORMAT('1***** THE PRESENT INPUT/OUTPUT SELECTIONS *****')
      WRITE('UT',*)
      WRITE('UT',20)
   20 FORMAT(' * INPUT QUANTITIES *')
      WRITE('UT',30)IRUNST,IRUNFN
   30 FORMAT('  (1)   RUN NUMBER RANGE:   ',I4,' TO ',I4)
      WRITE('UT',40)IPULST,IPULFN
   40 FORMAT('  (2)   PULSE NUMBER RANGE: ',I4,' TO ',I4)
      WRITE('UT',50)ALBINYN
   50 FORMAT('  (3)   ALL BINS:           ',A1)
      WRITE('UT',60)PEAKYN
   60 FORMAT('  (4)   PEAK:               ',A1)
      WRITE('UT',70)PKPOSYN
   70 FORMAT('  (5)   PEAK POSITION:      ',A1)
      WRITE('UT',80)ONEBNYN
   80 FORMAT('  (6)   ONE PARTICULAR BIN: ',A1)
      IF (ONEBNYN.EQ.'N') GOTO 100
      WRITE('UT',90)IBINNO
   90 FORMAT('  (7)   BIN NUMBER:        ',I3)
  100 WRITE('UT',*)
      WRITE('UT',110)
  110 FORMAT(' * OUTPUT DEVICES SELECTED *')
      WRITE('UT',120)PRSCYN
  120 FORMAT('  (8)   PRINT TO SCREEN:    ',A1)
      WRITE('UT',130)PRPAYN
  130 FORMAT('  (9)   PRINT TO PAPER:     ',A1)
      WRITE('UT',140)PLDAYN
  140 FORMAT(' (10)   PLOT DATA:          ',A1)
      WRITE('UT',*)
      WRITE('UT',150)
  150 FORMAT(' * FUCTIONS SELECTED *')
      WRITE('UT',160)STATYN
  160 FORMAT(' (11)   STATISTICS:         ',A1)
      WRITE('UT',170)HISTYN
  170 FORMAT(' (12)   HISTOGRAM:          ',A1)
      WRITE('UT',180)AUTOYN
  180 FORMAT(' (13)   AUTOCORRELATION:    ',A1)
      WRITE('UT',*)
      WRITE('UT',182)
  182 FORMAT(' (14)   DATA IS CORRECT')
      WRITE('UT',184)
  184 FORMAT(' (15)   ALL NEW DATA')
      WRITE('UT',186)
  186 FORMAT(' (16)   STOP')
      WRITE('UT',190)
  190 FORMAT(' ENTER NUMBER OF ITEM YOU WANT TO CHANGE:')
      READ('UT',*)ICHNGNO
      RNEW = 'N'
      STOP = 'N'
      IF(ICHNGNO.EQ.14)GOTO1000
      IF(ICHNGNO.NE.15)GOTO 187
    RNEW = 'Y'
    GOTO 1000
187 IF(ICHNGNO.NE.16)GOTO 195
    STOP = 'Y'
    GOTO 1000
195 GOTO(200,210,220,230,240,250,260,270,280,290,300,310,320)ICHNGNO
200 WRITE('UT',205)
205 FORMAT(' INPUT RUN NUMBER RANGE')
    READ('UT',*)IRUNST,IRUNFN
    GOTO5
210 WRITE('UT',215)
215 FORMAT(' INPUT PULSE NUMBER RANGE')
    READ('UT',*)IPULST,IPULFN
    GOTO5
220 WRITE('UT',225)
225 FORMAT(' ALL BINS? (Y OR N)')
    READ('UT',226)ALBINYN
226 FORMAT(A1)
    GOTO5
230 WRITE('UT',235)
235 FORMAT(' PEAK? (Y OR N)')
    READ('UT',226)PEAKYN
    GOTO5
240 WRITE('UT',245)
245 FORMAT(' PEAK POSITION? (Y OR N)')
    READ('UT',226)PKPOSYN
    GOTO5
250 WRITE('UT',255)
255 FORMAT(' ONE PARTICULAR BIN? (Y OR N)')
    READ('UT',226)ONEBNYN
    GOTO5
260 WRITE('UT',265)
265 FORMAT(' BIN NUMBER?')
    READ('UT',*)IBINNO
    GOTO5
270 WRITE('UT',275)
275 FORMAT(' PRINT TO SCREEN? (Y OR N)')
    READ('UT',226)PRSCYN
    GOTO5
280 WRITE('UT',285)
285 FORMAT(' PRINT TO PAPER? (Y OR N)')
    READ('UT',226)PRPAYN
    GOTO5
290 WRITE('UT',295)
295 FORMAT(' PLOT DATA? (Y OR N)')
    READ('UT',226)PLDAYN
    GOTO5
300 WRITE('UT',305)
305 FORMAT(' STATISTICS? (Y OR N)')
    READ('UT',226)STATYN
    GOTO5
310 WRITE('UT',315)
315 FORMAT(' HISTOGRAM? (Y OR N)')
    READ('UT',226)HISTYN
    GOTO5
320 WRITE('UT',325)
325 FORMAT(' AUTOCORRELATION? (Y OR N)')
```

```
      READ('UT',226)AUTOYN
      GOTO5
 1000 WRITE('UT',1010)
 1010 FORMAT(' **** END DATA ALTERING ROUTINE ****')
      IF(PRPAYN.EQ.'Y')GOTO 9999
      IF(PRPAYN.EQ.'N')GOTO 9999
      WRITE(1,10)
      WRITE(1,*)
      WRITE(1,20)
      WRITE(1,30)IRUNST,IRUNFN
      WRITE(1,40)IPULST,IPULFN
      WRITE(1,50)ALBINYN
      WRITE(1,60)PEAKYN
      WRITE(1,70)PKPUSYN
      WRITE(1,80)ONEBNYN
      IF (ONEBNYN.EQ.'N')GOTO 2000
      WRITE(1,90)IBINNO
 2000 WRITE(1,*)
      WRITE(1,110)
      WRITE(1,120)PRSCYN
      WRITE(1,130)PRPAYN
      WRITE(1,140)PLOAYN
      WRITE(1,*)
      WRITE(1,150)
      WRITE(1,160)STATYN
      WRITE(1,170)HISTYN
      WRITE(1,180)AUTOYN
      WRITE(1,*)
 9999 IBINNO = IBINNO + LENGTH
      RETURN
      END
C
C
C
C************************* SUBROUTINE GETHEAD *************************
      SUBROUTINE GETHEAD(FILTER,LENFILT,INDRUN,ISTART,ISTOP,NONE)
C
C   THIS ROUTINE EXTRACTS THE HEADER INFORMATION.
C
      COMMON /DATA/ IHDR4(3),IBUF(192)
      DIMENSION FILTER(32),IFL(32),INDR4(3),IRUNDAT(58)
      LOGICAL NONE
      NONE = .TRUE.
      DO 100 I=1,57,2
      IF(IRUNDAT(I).LE.INDRUN.AND.INDRUN.LE.IRUNDAT(I+1))NONE=.FALSE.
  100 CONTINUE
      IF(NONE) GOTO 20
      READ(2,REC=INDRUN,IOSTAT=IERR)IHDR4,ISTART,ISTOP
      IF(IERR.NE.0) WRITE('UT',*)'ERROR IN GETHEAD'
      CALL STADEC (IHDR4(2),PRF,LENFILT,IERR)
      DO I=1,LENFILT
      FILTER(I)=FLOAT((IBITF(IHDR4(3),I-1,1,IERR))*2-1)
      END DO
   20 CONTINUE
      RETURN
      DATA IRUNDAT/
     6    1,  25,  26,  50,  60,  78, 106, 110, 115, 155,
     &  158, 240, 299, 312, 314, 328, 329, 344, 347, 360,
     &  364, 375, 381, 391, 393, 403, 663, 682,1010,1023,
     & 1031,1046,1054,1065,1075,1087,1097,1108,1112,1119,
     & 1126,1136,1143,1148,1149,1155,1174,1180,1194,1206,
     & 1215,1234,1247,1247,1360,1374,1383,1397/
      END
C
C
C
C***************************** SUBROUTINE GTPULS *************************
      SUBROUTINE GTPULS(INDRUN,INDPULS,NUMGOOD,NOMORE,ISTART,ISTOP,
     &   LENFILT)
C
C    THIS ROUTINE GETS 96 PULSES FROM THE DISK, AND RETURNS WITH THE
C    NUMBER OF GOOD PULSES (NUMGOOD) AND AN ARRAY (RETURNS) FILLED WITH
C    THE GOOD RETURNS.
C
      LOGICAL GOOD
      COMMON/DATA/IHDR4(3),IBUF(192)
      COMMON/ARRAYS/ ALBNSC(2000),ALBNSC2(2000),RETURNS(96,32),
     &   AVGCOR(63),SIGCOR(63),IHSTBIN(65),IHSTPK(65),IHSTPOS(63),
     &   BAUTO(2000),PAUTO(2000),AUTOBIN(4000),AUTOPK(4000)
      LOGICAL NOMORE
      NUM = ISTART + MOD(INDPULS,96) - 1
      READ(3,REC=NUM)IBUF
      N=1
      DO K=1,96
      GOOD = .TRUE.
      DO L=1,32,2
      RETURNS(N,(L+1)/2)=RLOOK(IBITF(IBUF(2*K-1),L-1,2,IERR))
      IF(ABS(RETURNS(N,(L+1)/2)).GT.1.1)GOOD=.FALSE.
      END DO
      DO M=33,64,2
      RETURNS(N,(M+1)/2)=RLOOK(IBITF(IBUF(2*K),M-33,2,IERR))
      IF(ABS(RETURNS(N,(M+1)/2)).GT.1.1)GOOD=.FALSE.
      END DO
      IF(GOOD) N=N+1
      END DO
      NUMGOOD = N-1
      NOMORE = .FALSE.
      IF(NUM.EQ.ISTOP)NOMORE = .TRUE.
      RETURN
      END
C
C
C
      FUNCTION RLOOK(ICODE)
      IPLSE = ICODE + 1
      SELECT CASE IPLSE
      CASE 1
         RLOOK = -1.0
      CASE 2
         RLOOK = 0.0
      CASE 3
         RLOOK = 999.0
      CASE 4
         RLOOK = 1.0
      END SELECT
      RETURN
      END
C
C
C
C**************************** SUBROUTNE CORREL *************************
      SUBROUTINE CORREL(CORR,CODE,FILTER,LENGTH,PEAK,PKPOS,
     &   PSL,RISL,LENFILT,RLPG)
C
C THIS ROUTINE CALCULATES THE CORRELATION OF A CODE THROUGH A FILTER.
C IT ALSO CALCULATES THE PEAK,PEAK POSITION, PEAK SIDELOBE, AND
C THE INTEGRATED SIDELOBE LEVEL.
C
      DIMENSION CORR(63),CODE(32),FILTER(32)
      COMMON/ARRAYS/ ALBNSC(2000),ALBNSC2(2000),RETURNS(96,32),
     &   AVGCOR(63),SIGCOR(63),IHSTBIN(65),IHSTPK(65),IHSTPOS(63),
     &   BAUTO(2000),PAUTO(2000),AUTOBIN(4000),AUTOPK(4000)
      LENCOR = 2 * LENGTH - 1
      DO 10 I = 1,LENCOR
      CORR(I) = 0.0
      IND = I - LENGTH
      IF (IND.LT.0)GOTO5
      ILOW = 1
      IHIGH = LENGTH - IND
      GOTO 7
    5 ILOW = - IND + 1
      IHIGH = LENGTH
    7 DO 20 J = ILOW,IHIGH
      CORR(I) = CORR(I) + CODE(J) * FILTER(I + J - LENGTH)
   20 CONTINUE
   10 CONTINUE
      PEAK = 0
      DO 30 I = 1,LENCOR
      IF(ABS(CORR(I)).LE.ABS(PEAK))GOTO30
      PEAK = CORR(I)
      PKPOS = I - LENGTH
   30 CONTINUE
      PSL = 0
      RISL = 0
      DO 40 I=1,LENCOR
      IF(CORR(I).EQ.PEAK)GOTO 35
      RISL = RISL + (CORR(I))**2
   35 IF(CORR(I).EQ.PEAK.OR.ABS(CORR(I)).LE.PSL)GOTO40
      PSL = ABS(CORR(I))
   40 CONTINUE
      IF(RISL .EQ. 0.) RISL = .00000001
      IF(PSL .EQ. 0.) PSL = .00000001
      RISL = RISL**.5/(ABS(PEAK)+.00000001)
      PSL = PSL/(ABS(PEAK)+.00000001)
      RLPG = (ABS(PEAK)+.00000001)/LENFILT
      RETURN
      END
C
C
C
C**************************** SUBROUTNE CORREL1 *************************
```

```
      SUBROUTINE CORRFLT(CORR,CODE,FILTER,LENGTH)
C
C THIS ROUTINE CALCULATES THE CORRELATION OF A CODE THROUGH A FILTER.
C
      DIMENSION CORR(4000),CODE(2000),FILTER(2000)
      COMMON/ARRAYS/ ALBNSC(2000),ALBNSC2(2000),RETURNS(96,32),
     &   AVGCOR(63),STGCOR(63),IHSTBIN(63),IHSTPK(63),IHSTPOS(63),
     &   BAUTO(2000),PAUTO(2000),AUTOBIN(4000),AUTOPK(4000)
      DO 10 I=1,LENGTH
      CORR(I) = 0.
      K = LENGTH - I + 1
      DO 20 J=1,K
      CORR(I) = CORR(I) + CODE(J) * FILTER(J + I - 1)
   20 CONTINUE
   10 CONTINUE
      RETURN
      END
C
C
C
C****************************** SUBROUTINE STAT **************************
      SUBROUTINE STAT(LENGTH,ALBINYN,PEAKYN,CORR,
     &   PEAK,PKS,PKS2,PKPOSYN,POSS,POSS2,PKPOS,ONEBNYN,ONEBNS,
     &   ONEBNS2,IBINNO,PSL,RISL,SUMPSL,SUMISL,SUMPSL2,SUMISL2,
     &   RLPG,SUMLPG,SUMLPG2)
C
C THIS SUBROUTINE SAVES THE DATA NECESSARY FOR CALCULATING THE
C REQUESTED STATISTICS INFORMATION.
C
      DIMENSION CORR(63)
      COMMON/ARRAYS/ ALBNSC(2000),ALBNSC2(2000),RETURNS(96,32),
     &   AVGCOR(63),STGCOR(63),IHSTBIN(63),IHSTPK(63),IHSTPOS(63),
     &   BAUTO(2000),PAUTO(2000),AUTOBIN(4000),AUTOPK(4000)
      LENCOR = LENGTH * 2 - 1
      SUMPSL = SUMPSL + PSL
      SUMPSL2 = SUMPSL2 + PSL**2
      SUMISL = SUMISL + RISL
      SUMISL2 = SUMISL2 + RISL**2
      SUMLPG = SUMLPG + RLPG
      SUMLPG2 = SUMLPG2 + RLPG**2
      IF (ALBINYN.EQ.'N') GOTO 15
      DO 10 I = 1,LENCOR
      ALBNSC(I) = ALBNSC(I) + CORR(I)
      ALBNSC2(I) = ALBNSC2(I) + (CORR(I))**2
   10 CONTINUE
   15 IF(PEAKYN.EQ.'N') GOTO 20
      PKS = PKS + PEAK
      PKS2 = PKS2 + PEAK**2
   20 IF (PKPOSYN.EQ.'N')GOTO60
      POSS = POSS + PKPOS
      POSS2 = POSS2 + PKPOS**2
   60 IF(ONEBNYN.EQ.'N')GOTO80
      ONEBNS = ONEBNS + CORR(IBINNO)
      ONEBNS2 = ONEBNS2 + (CORR(IBINNO)**2)
   80 RETURN
      END
C
```

```
C
C
C****************************** SUBROUTINE HIST ******************************
      SUBROUTINE HIST(LENGTH,ONEBNYN,IBINNO,PEAKYN,PEAK,CORR,
     &    PKPOSYN,PKPOS)
C
C  THIS SUBROUTINE UPDATES THE FILES NECESSARY FOR PRODUCING THE
C  HISTOGRAMS.
C
      DIMENSION ICORR(63),CORR(63)
      COMMON/ARRAYS/ ALBNSC(2000),ALBNSC2(2000),RETURNS(96,32),
     &    AVGCOR(63),STGCOR(63),IHSTBIN(65),IHSTPK(65),IHSTPOS(63),
     &    BAUTO(2000),PAUTO(2000),AUTOBIN(4000),AUTOPK(4000)
      LENCUR = 2 * LENGTH - 1
      DO 10 I = 1,LENCUR
      ICORR(I) = (CORR(I) + .5)
   10 CONTINUE
      IF (ONEBNYN.EQ.'N')GOTO30
      IHSTBIN(ICORR(IBINNO)+LENGTH+1)=IHSTBIN(ICORR(IBINNO)+LENGTH+1)+1
   30 IF(PEAKYN.EQ.'N')GOTO50
      IPEAK = (PEAK + .5) + LENGTH + 1
      IHSTPK(IPEAK) = IHSTPK(IPEAK) + 1
   50 IF (PKPOSYN.EQ.'N')GOTO70
      IHSTPOS(PKPOS + LENGTH) = IHSTPOS(PKPOS + LENGTH) + 1
   70 RETURN
      END
C
C
C
C****************************** SUBROUTINE AUTOC ******************************
      SUBROUTINE AUTOC(CORR,PEAK,PEAKYN,ONEBNYN,IBINNO,ISOFAR)
C
C  THIS ROUTINE UPDATES THE FILES NECESSARY FOR THE AUTOCORRELATION
C  OF THE RETURN SIGNAL'S CORRELATIONS.
C
      DIMENSION CORR(63)
      COMMON/ARRAYS/ ALBNSC(2000),ALBNSC2(2000),RETURNS(96,32),
     &    AVGCOR(63),STGCOR(63),IHSTBIN(65),IHSTPK(65),IHSTPOS(63),
     &    BAUTO(2000),PAUTO(2000),AUTOBIN(4000),AUTOPK(4000)
      IF(ONEBNYN.EQ.'N')GOTO10
      BAUTO(ISOFAR) = CORR(IBINNO)
   10 IF (PEAKYN.EQ.'N')GOTO20
      PAUTO(ISOFAR) = PEAK
   20 RETURN
      END
C
C
C
C****************************** SUBROUTINE WRAPUP ******************************
      SUBROUTINE WRAPUP(LENGTH,ISOFAR,STATYN,
     &    ONEBNYN,BAVG,BSIG,ONEBNS,ONEBNS2,PEAKYN,PKAVG,PKSIG,
     &    PKS,PKS2,PKPOSYN,PKPOSAV,PKPOSSG,POSS,POSS2,AUTOYN,
     &    SUMPSL,SUMISL,SUMPSL2,SUMISL2,PSAVG,PSSIG,RIAVG,RISIG,
     &    SUMLPG,SUMLPG2,RLPGAVG,RLPGSIG,LENFILT,PSLCOM,RISLCOM,RLPGCOM)
C
C  THIS ROUTINE TAKES CARE OF THE DETAILS OF THE HISTOGRAM, STATISTICS
C  AND AUTOCORRELATION FUNCTIONS THAT WERE NOT EAISILY PERFORMED IN
```

```
C  THE ITERATIVE LOOPING STRUCTURE OF THE OTHER SUBROUTINES.
C
      COMMON/ARRAYS/ ALBNSC(2000),ALBNSC2(2000),RETURNS(96,32),
     &   AVGCOR(63),SIGCOR(63),IHSTBIN(65),IHSTPK(65),IHSTPOS(63),
     &   BAUTO(2000),PAUTO(2000),AUTOBIN(4000),AUTOPK(4000)
      LENCOR = 2 * LENGTH - 1
C * WRAPUP FOR STATISTICS *
      IF(STATYN.EQ.'N')GOTO50
      DO 10 I = 1,LENCOR
      AVGCOR(I) = ALBNSC(I)/ISOFAR
      SIGCOR(I) = (ALBNSC2(I)/ISOFAR - (AVGCOR(I))**2)**.5
   10 CONTINUE
      PSAVG = SUMPSL/ISOFAR
      PSSIG = 20*ALOG10((SUMPSL2/ISOFAR - PSAVG**2)**.5)
      RIAVG = SUMISL/ISOFAR
      RISIG = 20*ALOG10((SUMISL2/ISOFAR - RIAVG**2)**.5)
      RLPGAVG = SUMLPG/ISOFAR
      RLPGSIG = 20*ALOG10((SUMLPG2/ISOFAR - RLPGAVG**2)**.5)
      PSAVG = 20*ALOG10(PSAVG)
      RIAVG = 20*ALOG10(RIAVG)
      RLPGAVG = 20*ALOG10(RLPGAVG+.000011)
C   CALCULATE PSL,ISL, AND LPG FOR AVERAGE CORRELATOR OUTPUT.
      PKCOM = 0.
      PSLCOM = 0.
      RISLCOM = 0.
      DO 25 I=1,LENCOR
      RISLCOM = RISLCOM + AVGCOR(I)**2
      IF(PKCOM.GE.ABS(AVGCOR(I)))GOTO23
      PSLCOM = PKCOM
      PKCOM = ABS(AVGCOR(I))
      GOTO 25
   23 IF(PSLCOM.GE.ABS(AVGCOR(I)))GOTO 25
      PSLCOM = ABS(AVGCOR(I))
   25 CONTINUE
C  THE NEXT LINE SUBTRACTS THE PEAK FROM ISL CONSIDERATION.
      RISLCOM = RISLCOM - PKCOM**2
      PSLCOM = 20 * ALOG10(PSLCOM/PKCOM)
      RISLCOM = 20 * ALOG10(RISLCOM**.5/PKCOM)
      RLPGCOM = 20 * ALOG10(PKCOM/FLOAT(LENFILT))
   20 IF(ONEBNYN.EQ.'N')GOTO30
      BAVG = ONEBNS/ISOFAR
      BSIG = (ONEBNS2/ISOFAR - BAVG**2)**.5
   30 IF(PEAKYN.EQ.'N')GOTO40
      PKAVG = PKS/ISOFAR
      PKSIG = (PKS2/ISOFAR - PKAVG**2)**.5
   40 IF(PKPOSYN.EQ.'N')GOTO50
      PKPOSAV = POSS/ISOFAR
      PKPOSSG = (POSS2/ISOFAR - PKPOSAV**2)**.5
   50 CONTINUE
C
C * WRAPUP FOR HISTOGRAM - NONE NEEDED *
C
C * WRAPUP FOR AUTOCORRELATION *
      IF (AUTOYN.EQ.'N')GOTO100
C  THIS SECTION ALTERS BAUTO AND PAUTO TO FIT THE REQUIREMENTS OF THE
C  AUTOCORRELATION ANALYSIS.  IF THE STANDARD DEVIATION OF EITHER OF
C  THE VARIABLES IS VERY SMALL, THEN THAT VARIABLE IS SET EQUAL TO
```

```
C  THE RECIPROCAL OF THE NUMBER OF PULSES PROCESSED SO FAR.  OTHERWISE
C  THE MEAN IS REMOVED FROM THE VARIABLE, AND THE RESULT IS DIVIDED
C  BY THE STANDARD DEVIATION AND THE SQUARE ROOT OF THE NUMBER OF
C  PULSES PROCESSED SO FAR.  WHEN THE RESULTING ARRAY IS PROCESSED
C  THROUGH THE STANDARD CORRELATION FUNCTION, THE RESULT IS IN THE
C  DESIRED FORM:
C      OUTPUT = SUM((X(I)-MEAN)*(X(I-J)-MEAN))/SUM((X(I)-MEAN)**2)
      IF(BSIG.GT..000001)GOTO 60
      DO 61 I=1,ISOFAR
      BAUTO(I) = 1.0 / FLOAT(ISOFAR)**.5
   61 CONTINUE
      GOTO62
   60 DO 63 I=1,ISOFAR
      BAUTO(I) = (BAUTO(I) - BAVG)/(FLOAT(ISOFAR)**.5*BSIG)
   63 CONTINUE
   62 IF(PKSIG.GT..000001)GOTO 64
      DO 65 I=1,ISOFAR
      PAUTO(I) = 1.0 / FLOAT(ISOFAR)**.5
   65 CONTINUE
      GOTO 69
   64 DO 66 I=1,ISOFAR
      PAUTO(I) = (PAUTO(I) - PKAVG)/(FLOAT(ISOFAR)**.5*PKSIG)
   66 CONTINUE
   69 IF(ONEBNYN.EQ.'N')GOTO70
      CALL CORREL1(AUTOBIN,BAUTO,BAUTO,ISOFAR)
   70 IF(PEAKYN.EQ.'N')GOTO100
      CALL CORREL1(AUTOPK,PAUTO,PAUTO,ISOFAR)
  100 RETURN
      END
C
C
C
C****************************** SUBROUTINE DATAOUT ***********************
      SUBROUTINE DATAOUT(STATYN,HISTYN,AUTOYN,ALBINYN,ONEBNYN,I3INNO,
     &   PEAKYN,PKPOSYN,PKAVG,PKSIG,PKPOSAV,PKPOSSG,BAVG,BSIG,ISOFAR,
     &   LENGTH,PSAVG,PSSIG,RIAVG,RISIG,RLPGAVG,RLPGSIG,PSLCUM,
     &   RISLCUM,RLPGCUM,INDRUN,PRPAYN,PCTGOOD)
C
C   THIS ROUTINE PRINTS OR PLOTS THE DATA AS SPECIFIED BY THE USER.
C
      COMMON/ARRAYS/ ALBNSC(2000),ALBNSC2(2000),RETURNS(96,32),
     &   AVGCOR(63),STGCOR(63),INSTBIN(65),INSTPK(65),INSTPOS(63),
     6   BAUTO(2000),PAUTO(2000),AUTOBIN(4000),AUTOPK(4000)
      IBINNO = IBINNO - LENGTH
      IF(PRPAYN.EQ.'N')GOTO 90
      WRITE(1,997)
      WRITE(1,998)INDRUN
      WRITE(1,997)
  997 FORMAT(' ****************************')
  998 FORMAT(' ANALYSIS OF RUN NUMBER',I5)
      WRITE(1,*)
C*** STATISTICS INFORMATION ***
      IF (STATYN.EQ.'N')GOTO50
      WRITE(1,999)
  999 FORMAT(' ******* STATISTICS REPORT *******')
      WRITE(1,*)
      IF(ALBINYN.EQ.'N')GOTO 20
      WRITE(1,1000)
 1000 FORMAT(' AVERAGE CORRELATOR OUTPUT (VOLTS):')
      WRITE(1,1001)(AVGCOR(I),I=1,63)
      WRITE(1,*)
 1001 FORMAT(' ',21F5.1)
      WRITE(1,1002)
 1002 FORMAT(' STANDARD DEVIATION OF CORRELATOR OUTPUT:')
      WRITE(1,1001)(SIGCOR(I),I=1,63)
      WRITE(1,*)
      WRITE(1,996)PCTGOOD
  996 FORMAT(' ',F5.1,' PERCENT OF THE PULSES WERE GOOD')
      WRITE(1,995)ISOFAR
  995 FORMAT(' ',I5,' PULSES WERE PROCESSED')
      WRITE(1,*)
   20 IF(PEAKYN.EQ.'N')GOTO 30
      WRITE(1,1003)PKAVG
 1003 FORMAT(' AVERAGE PEAK CORRELATOR OUTPUT:',F5.1,' VOLTS')
      WRITE(1,1004)PKSIG
 1004 FORMAT(' STANDARD DEVIATION OF PEAK OUTPUT:',F5.1)
      WRITE(1,*)
   30 IF(PKPOSYN.EQ.'N')GOTO40
      WRITE(1,1005)PKPOSAV
 1005 FORMAT(' AVERAGE PEAK POSITION:',F5.1)
      WRITE(1,1008)PKPOSSG
      WRITE(1,*)
   40 IF(ONEBNYN.EQ.'N')GOTO45
      WRITE(1,1007)IBINNO,BAVG
 1007 FORMAT(' AVERAGE OUTPUT OF BIN NUMBER',I3,' IS ',F5.1,' VOLTS')
      WRITE(1,1008)BSIG
 1008 FORMAT(' STANDARD DEVIATION: ',F5.1)
      WRITE(1,*)
      WRITE(1,*)
      WRITE(1,*)
   45 WRITE(1,1100)PSAVG,PSLCOM
 1100 FORMAT(' AVERAGE PEAK SIDELOBE LEVEL:',F5.1,' DB          PSL OF
     & AVERAGE OUTPUT:',F5.1,' DB')
      WRITE(1,1008)PSSIG
      WRITE(1,*)
      WRITE(1,1102)RIAVG,RISLCOM
 1102 FORMAT(' AVERAGE INTEGRATED SIDELOBE LEVEL:',F5.1,' DB    ISL OF
     & AVERAGE OUTPUT:',F5.1,' DB')
      WRITE(1,1008)RISIG
      WRITE(1,*)
      WRITE(1,1103)RLPGAVG,RLPGCOM
 1103 FORMAT(' AVERAGE LOSS IN PROCESSING GAIN:',F5.1,' DB      LPG OF
     & AVERAGE OUTPUT:',F5.1,' DB')
      WRITE(1,1008)RLPGSIG
      WRITE(1,*)
C*** HISTOGRAM INFORMATION ***
   50 IF(HISTYN.EQ.'N') GOTO 60
      WRITE(1,*)
      WRITE(1,1009)
 1009 FORMAT('1******* HISTOGRAM REPORT *******')
      WRITE(1,*)
      IF(PEAKYN.EQ.'N')GOTO70
      WRITE(1,1010)
 1010 FORMAT(' HISTOGRAM OF PEAK (PEAK IS IN VOLTS):')
```

```
      M1=-32
      M2=-11
      WRITE(1,1011)((I),I=M1,M2)
 1011 FORMAT(' PEAK:       ',22I4)
      WRITE(1,1012)(IHSTPK(I),I=1,22)
 1012 FORMAT(' OCCURENCES:',22I4)
      WRITE(1,*)
      M1=-10
      WRITE(1,1011)((I),I=M1,11)
      WRITE(1,1012)(IHSTPK(I),I=23,44)
      WRITE(1,*)
      WRITE(1,1011)((I),I=12,32)
      WRITE(1,1012)(IHSTPK(I),I=45,65)
      WRITE(1,*)
      WRITE(1,*)
   70 IF(PKPOSYN.EQ.'N') GOTO 80
      WRITE(1,1013)
 1013 FORMAT(' HISTOGRAM OF PEAK POSITION:')
      M1=-31
      M2=-11
      WRITE(1,1014)((I),I=M1,M2)
 1014 FORMAT(' POSITION  :',21I4)
      WRITE(1,1012)(IHSTPOS(I),I=1,21)
      WRITE(1,*)
      M1=-10
      WRITE(1,1014)((I),I=M1,10)
      WRITE(1,1012)(IHSTPOS(I),I=22,42)
      WRITE(1,*)
      WRITE(1,1014)((I),I=11,31)
      WRITE(1,1012)(IHSTPOS(I),I=43,63)
      WRITE(1,*)
      WRITE(1,*)
   80 IF(ONEBNYN.EQ.'N')GOTO 60
      WRITE(1,1015)IBINNO
 1015 FORMAT(' HISTOGRAM OF BIN NUMBER',I3,'  (OUTPUT IS IN VOLTS):')
      M1=-32
      M2=-11
      WRITE(1,1016)((I),I=M1,M2)
 1016 FORMAT(' OUTPUT:     ',22I4)
      WRITE(1,1012)(IHSTBIN(I),I=1,22)
      WRITE(1,*)
      M1=-10
      WRITE(1,1016)((I),I=M1,11)
      WRITE(1,1012)(IHSTBIN(I),I=23,44)
      WRITE(1,*)
      WRITE(1,1016)((I),I=12,32)
      WRITE(1,1012)(IHSTBIN(I),I=45,65)
      WRITE(1,*)
      WRITE(1,*)
C*** AUTOCORRELATION INFORMATION ***
   60 IF(AUTOYN.EQ.'N') GOTO 90
      WRITE(1,1017)
 1017 FORMAT(' ****** AUTOCORRELATION REPORT ******')
      WRITE(1,*)
      IF(PEAKYN.EQ.'N') GOTO 100
      WRITE(1,1018)
 1018 FORMAT(' AUTOCORRELATION OF PEAK SEQUENCE:')
```

```
      K = ISUFAR - 1
      WRITE(1,1020)(AUTOPK(I),I=1,K)
 1020 FORMAT(' ',15F6.4)
      WRITE(1,*)
      WRITE(1,*)
      WRITE(1,*)
  100 IF(ONEBNYN.EQ.'N') GOTO 90
      WRITE(1,1019)IBINNO
 1019 FORMAT(' AUTOCORRELATION OF OUTPUT FROM BIN NUMBER',I3)
      WRITE(1,1020)(AUTOBIN(I),I=1,K)
   90 CONTINUE
      IBINNO = IBINNO + LENGTH
      RETURN
      END
      SUBROUTINE STADEC (IPARSW, PRF, ITCL, IER)
C
C                ******************************
C                *                            *
C                *      SUBROUTINE STADEC     *
C                *                            *
C                ******************************
C
C*
C*    THIS SUBROUTINE PERFORMS THE DECODING OF THE FILE HEADER STATUS
C*    WORD INTO THE FOLLOWING PARAMETERS:
C*                IPAR DISPLAY SWITCH
C*                IPAR COMPUTER FLAG SWITCH
C*                IPAR RUN/TEST
C*                SUB-PULSE LENGTH
C*                PRF
C*                TRANSMIT CODE LENGTH
C*    THESE PARAMETERS ARE RETURNED TO THE USER WITH THE HELP OF
C*    SUBROUTINE TABLE.
C*
      INTEGER*1 IHDR1(12), IASCII(6)
      COMMON /DATA/ IHDR4(3),IBUF(192)
C     CALL UNPACK(IHDR1,IASCII)
      ISTORE = IPARSW
C*****PRF
      IPRF = IBITF (ISTORE, 22, 4, IERR)
         SELECT CASE IPRF
         CASE 0
            PRF = 4000.
         CASE 4
            PRF = 2000.
         CASE 6
            PRF = 1000.
         CASE 7
            PRF = 500.
         CASE 8
            PRF = 6000.
         END SELECT
            WRITE (1, 30, IOSTAT = IOSTATW) PRF
C*****IPAR DISPLAY SWITCH
            IDC = IBITF (ISTORE, 16, 1, IERR)
            NWD = 3
            INDX = 17+3*IDC
            CALL TABLE(INDX,NWD,1,IERT)
C*****IPAR COMPUTER FLAG SWITCH
            IDC = IBITF (ISTORE, 17, 1, IERR)
            NWD = 1
            INDX = 2+2*IDC
            CALL TABLE (INDX, NWD, 2, IERT)
C*****IPAR RUN/TEST
            IDC = IBITF (ISTORE, 18, 1, IERR)
            NWD = 1
            INDX = 9+IDC
            CALL TABLE (INDX, NWD, 3, IERT)
C*****SUB-PULSE LENGTH
            IDC = IBITF (ISTORE, 19, 3, IERR)
            NWD = 1
            CALL TABLE (IDC , NWD, 4, IERT)
C*****TRANSMIT CODE LENGTH
            IDC = IBITF (ISTORE, 26, 6, IERC)
               IP2 = 32
               ITCL = 0
               DO I = 1, 6
                  KBIT = IBITF (IDC, I-1, 1, IERC)
                  ITCL = ITCL+IP2*(1-KBIT)
                  IP2 = IP2/2
               END DO
               WRITE (1, 35, IOSTAT = NSTATW) ITCL
      RETURN
C*****FORMAT STATEMENTS:
30    FORMAT('1',35X,'PRF:  ',F5.0)
35    FORMAT(19X,'TRANSMIT CODE LENGTH:  ',I2)
      END
      SUBROUTINE TABLE (INDEX, NWORD, IPARAM, IERR)
C
C              ******************************
C              *                            *
C              *      SUBROUTINE TABLE      *
C              *                            *
C              ******************************
C
C*
C*    SUBROUTINE TABLE PERFORMS A TABLE LOOK-UP IN ORDER TO DEFINE THE
C*    VARIOUS PARAMETERS OF THE IPAR FILE HEADER STATUS WORD.
C*
      COMMON / IOUNIT / NTAP, NSLU, NHOR, NDTA
      COMMON / DVSTAT / NSTATT, NSTATW, NSTATH, NSTATD, NSTATC, NSTATI,
     6                  NSTATHD, NSTATDD, NSTATHC, NSTATDC, NMOUNT,
     6                  NDSMNT
      DIMENSION ITAB(22),LIT(3)
      DATA ITAB/4H  80,4H SET,4H  20,4H CLR,4H  40,4H 160,4H  10,4H4000,
     1          4H RUN,4HTEST,4H    ,4H2000,4H MAN,4H1000,4H 500,4H8000,
     2          4HPAGE,4HL   ,4H    ,4HDIGI,4HTAL ,4HTAPE/
C*****INITIALIZE LITERAL ARRAY
         DO I = 1, 3
            LIT(I) = ITAB(11)
         END DO
C*****DEFINE THE SELECTED PARAMETER FROM THE TABLE
         DO J = 1, NWORD
            IOFF = INDEX+J-1
            LIT(J) = ITAB(IOFF)
         END DO
         SELECT CASE IPARAM
         CASE 1
            WRITE (1, 10, IOSTAT = NSTATW) (LIT(I), I = 1, NWORD)
         CASE 2
            WRITE (1, 15, IOSTAT = NSTATW) (LIT(I), I = 1, NWORD)
         CASE 3
            WRITE (1, 20, IOSTAT = NSTATW) (LIT(I), I = 1, NWORD)
         CASE 4
            WRITE (1, 25, IOSTAT = NSTATW) (LIT(I), I = 1, NWORD)
         END SELECT
      RETURN
C*****FORMAT STATEMENTS:
10    FORMAT(11X,'IPAR DISPLAY SWITCH POSITION:  ',3A4)
15    FORMAT(14X,'IPAR COMPUTER FLAG SWITCH:  ',A4)
20    FORMAT(20X,'IPAR RUN/TEST:  ',A4)
25    FORMAT(23X,'SUB-PULSE LENGTH:  ',A4)
      END
      SUBROUTINE UNPACK (IHDR1, IASCII)
C
C              *******************************
C              *                             *
C              *     SUBROUTINE UNPACK       *
C              *                             *
C              *******************************
C
C*
C*    SUBROUTINE UNPACK SPLITS A SINGLE BYTE INTO TWO SEPARATE 4-BIT
C*    BCD QUANITITIES.  THESE QUANTITIES ARE THEN CONVERTED TO ASCII
C*    FORMAT AND STORED IN THE BYTE ARRAY, IASCII.
C*
      INTEGER*1 IHDR1(12), IASCII(6), MASK, ITMP, JTMP
      DATA MASK/X'0F'/
C*****UNPACK EACH OF THE NOT BYTES
      DO I = 1, 3
         IREF = (4 - I) * 2
C*****LOWER BYTE
         ITMP=IAND(IHDR1(I),MASK)
         ENCODE(1, '(I1)', JTMP) ITMP
         DECODE(1, '(A1)', JTMP) IASCII(IREF)
C*****UPPER BYTE
         ITMP=ISHFT(IHDR1(I),-4)
         ENCODE(1, '(I1)', JTMP) ITMP
         DECODE(1, '(A1)', JTMP) IASCII(IREF-1)
      END DO
      RETURN
      END
      FUNCTION IBITF (IWORD, ISBIT, LEN, IERR)
C
C              *******************************
C              *                             *
C              *       FUNCTION IBITF        *
C              *                             *
C              *******************************
C
C*
```

```
C*     FUNCTION IBITF EXTRACTS A FIELD OF BITS FROM IWORD, STARTING
C*     FROM BIT POSITION ISBIT AND EXTENDING TO THE LEFT FOR LEN BITS.
C*     THE RIGHTMOST BIT IS BIT 0 AND THE LEFTMOST BIT IS BIT 31.
C*
      LOGICAL ERROR
C*****ERROR CONDITION CHECK
      IF (ISBIT+LEN.LE.32 .AND. LEN.GT.0 .AND. ISBIT.GE.0) THEN
C*****UNPACK THE SPECIFIED BIT FIELD
         ERROR = .FALSE.
         IBITF = ISHFT(IWORD,32-(ISBIT+LEN))
         IBITF = ISHFT(IBITF,-32+LEN)
      ELSE
         ERROR = .TRUE.
      END IF
      IF (ERROR) THEN
         IERR = 12
      END IF
      RETURN
      END
$A1 L01=SRTLIB,,U
$A1 D01=SRTDIR,,U
$EXECUTE CATALOG
A2 1=SLU,9000
OPTION PROMPT
BUILD CORC U NOM
$EOJ
$$
```