

AD-A138 914

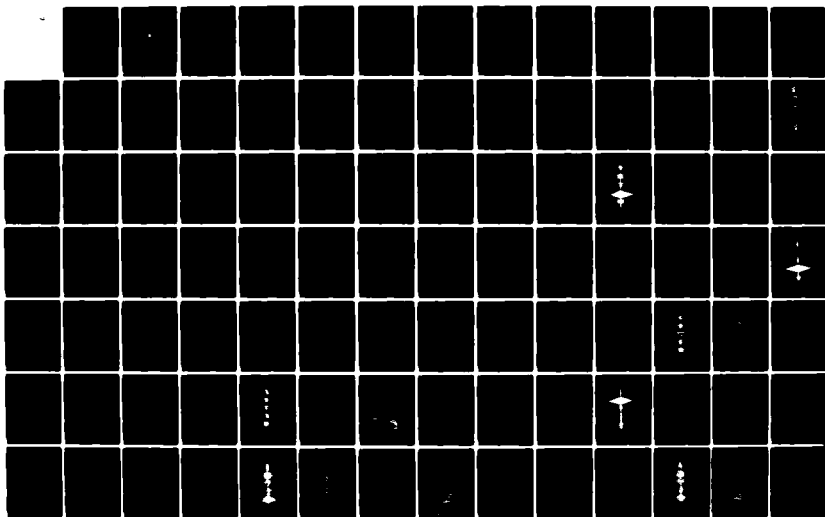
MULTIPLE TARGET IDENTIFICATION AND DIRECTION FINDING
USING MATCHED FILTERING TECHNIQUES(U) NAVAL
POSTGRADUATE SCHOOL MONTEREY CA J L JOHNSTON DEC 83

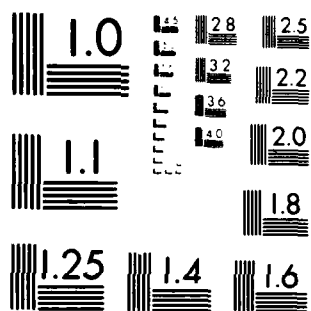
1/2

UNCLASSIFIED

F/G 17/10

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD A138914

NAVAL POSTGRADUATE SCHOOL
Monterey, California



THESIS

MULTIPLE TARGET IDENTIFICATION
AND DIRECTION FINDING
USING MATCHED FILTERING TECHNIQUES

by

James L. Johnston

December 1983

Thesis Advisor:

H. A. Titus

Approved for public release; distribution unlimited

DTIC FILE COPY

84 03 13 035

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. A138 914	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Multiple Target Identification and Direction Finding Using Matched Filtering Techniques		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis December 1983
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) James L. Johnston		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943		12. REPORT DATE December 1983
		13. NUMBER OF PAGES 169
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report)
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
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) seismic sensors, matched filtering, target identification, target direction.		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This research investigates seismic signal processing techniques for battlefield target classification and acquisition. Multiple target classification is performed by discrete time domain matched filtering. Multiple target directions are determined using the responses of the matched filters and least mean squares polynomial curve fitting. The least mean squares polynomial curve fitting procedure is also used for direction finding for recoil/blast sources, using the unfiltered seismic signals.		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102- LP-014-6001

1

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Approved for public release; distribution unlimited.

Multiple Target Identification and Direction Finding
Using Matched Filtering Techniques

by

James L. Johnston
Captain, United States Marine Corps
B.S.E.E., San Diego State University, 1975

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1983



Author: _____

Approved by: _____

Thesis Advisor

Second Reader

Chairman, Department of Electrical Engineering

Dean of Science and Engineering

ABSTRACT

This research investigates seismic signal processing techniques for battlefield target classification and acquisition. Multiple target classification is performed by discrete time domain matched filtering. Multiple target directions are determined using the responses of the matched filters and least mean squares polynomial curve fitting. The least mean squares polynomial curve fitting procedure is also used for direction finding for recoil/blast sources, using the unfiltered seismic signals.

TABLE OF CONTENTS

I.	INTRODUCTION	12
II.	THEORY OF SEISMIC SENSORS	15
III.	DEVELOPMENTAL REQUIREMENTS AND PROBLEM DEFINITION	18
	A. PASSIVE TARGET ACQUISITION AND SURVEILLANCE	18
	B. CURRENT CAPABILITIES AND DEFICIENCIES	20
IV.	MATCHED FILTER CONCEPT AND DISCRETE ALGORITHM	21
	A. MATCHED FILTER THEORY	21
	B. MATCHED FILTER ALGORITHM	22
	C. MULTIPLE TARGET MATCHED FILTERING	26
	D. COMPUTATIONAL REQUIREMENTS	26
	3. SUPPORT SOFTWARE	27
	1. Amplitude Analysis (Timeout)	27
	2. Frequency Analysis (Freqot)	28
	3. Simulation and Validation Software (SIMULT)	31
V.	MULTIPLE TARGET DIRECTION	34
	A. THEORY AND DESIGN CRITERION	34
	B. MULTIPLE TARGET FILTERING ALGORITHM (MULTI)	35
	1. Multiple Target Direction Phase Difference Algorithm	37
	2. Least Mean Square Polynomial Direction Finding	39
	3. Least Mean Squares Polynomial Algorithm Derivation	41

4. Adaptive Target Direction Finding	44
5. Software features of the Multiple Target Direction Routine	44
C. MULTIPLE TARGETS OF THE SAME TARGET CLASS . .	45
VI. ANALYSIS OF SEISMIC DATA	47
VII. CONCLUSIONS AND RECOMMENDATIONS	116
APPENDIX A: USERS MANUAL	118
APPENDIX B: SAMPLE INTERACTIVE PROGRAM SESSION	127
APPENDIX C: PROGRAM LISTING	132
LIST OF REFERENCES	168
INITIAL DISTRIBUTION LIST	169

LIST OF TABLES

I.	Target Detection Radii	19
II.	Test Plan for Simulated Data	48
III.	Matched Filter for Simulated Targets	48
IV.	Test Plan for Experimental Data	49
V.	Matched Filter for Experimental Data	49
VI.	Summary of Direction Finding Results	50
VII.	Missed or Incorrectly Identified Targets	51

LIST OF FIGURES

4.1	Working Array Configuration	24
4.2	Sample Matched Filter Output	25
4.3	Windowing of Experimental Data	26
4.4	Data Window Size vs Number of Operations	28
4.5	Sample Amplitude versus Time Output	29
4.6	Sample Frequency versus Power Output	30
4.7	Simulation of a Zero Degree Target	32
4.8	Sample Multiple Target and Simulated Target Output	33
5.1	Two Target Matched Filter Response	36
5.2	Circular Sensor Array Geometry	38
5.3	Relative Delay Times in a Nine Sensor Ring	40
5.4	Relative Time Delay versus Sensor Angle	42
6.1	Sample Least Mean Squares Initial Direction for Event 001	52
6.2	Sample Matched Filter Response for Event 001	53
6.3	Sample Amplitude Response for Event 001	54
6.4	Sample Frequency Response for Event 001	55
6.5	LMSP Matched Filter Direction for Event 001	56
6.6	LMSP Multiple Target Direction Summary for Event 001	57
6.7	LMSP Matched Filter Direction for Event 001	58
6.8	LMSP Multiple Target Direction Summary for Event 001	59
6.9	LMSP Matched Filter Direction for Event 001	60
6.10	LMSP Multiple Target Direction Summary for Event 001	61
6.11	LMSP Matched Filter Direction for Event 001	62

6.12	LMSP Multiple Target Direction Summary for Event 001	63
6.13	LMSP Initial Direction for Event 383	64
6.14	Matched Filter Response for Event 383	65
6.15	Amplitude Response for Event 383	66
6.16	Frequency Response for Event 383	67
6.17	Fourth Degree LMSP Matched Filter Direction for Event 383	68
6.18	Second Degree LMSP Matched Filter Direction for Event 383	69
6.19	LMSP Multiple Target Direction Summary for Event 383	70
6.20	LMSP Initial Direction for Event 382	71
6.21	Matched Filter Response for Event 382	72
6.22	Amplitude Response for Event 382	73
6.23	Frequency Response for Event 382	74
6.24	LMSP Matched Filter Direction for Event 382	75
6.25	LMSP Multiple Target Direction Summary for Event 382	76
6.26	LMSP Initial Direction for Event 372	77
6.27	Matched Filter Response for Event 372	78
6.28	Amplitude Response for Event 372	79
6.29	Amplitude Response of Malfunctioning Sensor for Event 372	80
6.30	Frequency Response for Event 372	81
6.31	LMSP Matched Filter Direction for Event 372	82
6.32	LMSP Multiple Target Direction Summary for Event 372	83
6.33	LMSP Initial Direction for Event 375	84
6.34	Matched Filter Response for Event 375	85
6.35	Amplitude Response for Event 375	86
6.36	Amplitude Response of Malfunctioning Sensor for Event 375	87

6.37	Frequency Response for Event 375	88
6.38	LMSP Matched Filter Direction for Event 375 . .	89
6.39	LMSP Multiple Target Direction Summary for Event 375	90
6.40	LMSP Initial Direction for Event 374	91
6.41	Matched Filter Response for Event 374	92
6.42	Amplitude Response for Event 374	93
6.43	Amplitude Response of Malfunctioning Sensor for Event 374	94
6.44	Frequency Response for Event 374	95
6.45	LMSP Matched Filter Direction for Event 374 . .	96
6.46	LMSP Multiple Target Direction Summary for Event 374	97
6.47	LMSP Initial Direction for Event 302	98
6.48	Matched Filter Response for Event 302	99
6.49	Amplitude Response for Event 302	100
6.50	Frequency Response for Event 302	101
6.51	LMSP Matched Filter Direction for Event 302 .	102
6.52	LMSP Multiple Target Direction Summary for Event 302	103
6.53	LMSP Initial Direction for Event 354 (2 - 6sec)	104
6.54	Matched Filter Response for Event 354 (2 - 6sec)	105
6.55	Amplitude Response for Event 354 (2 - 6sec) .	106
6.56	Frequency Response for Event 354 (2 - 6sec) .	107
6.57	LMSP Matched Filter Direction for Event 354 (2 - 6sec)	108
6.58	LMSP Multiple Target Direction Summary Event 354 (2 - 6sec)	109
6.59	LMSP Initial Direction for Event 354 (7 - 11sec)	110
6.60	Matched Filter Response for Event 354 (7 - 11sec)	111

6.61	Amplitude Response for Event 354 (7 - 11sec) .	112
6.62	Frequency Response for Event 354 (7 - 11sec) .	113
6.63	LMSP Matched Filter Direction for Event 354 (7 - 11sec)	114
6.64	LMSP Multiple Target Summary Event 354 (7 - 11sec)	115

ACKNOWLEDGEMENT

I wish to express my appreciation to Professor Harold A. Titus for his professional guidance and support in developing this thesis. I would also like to thank my wife, Pamela, for her constant support during the writing of this work.

I. INTRODUCTION

Timely and accurate combat intelligence is a integral part of the modern battlefield. A primary goal of combat intelligence is target acquisition. Hostile targets can be acquired by either passive or active means. An example of passive target acquisition is visual target identification. Radar, on the other hand, is an active target acquisition device. An effective combat intelligence system will include a mix of both active and passive target acquisition methods.

Rapidly advancing technology in the fields of electronic counter-measures and radiation-seeking weapons has enhanced interest in passive target acquisition methods. To be cost effective, as an additional target acquisition system, a passive system must be able to provide swift and accurate target identification, location and tracking information on hostile targets. A variety of seismic sensor systems have been used in this roll with varying degrees of sucess.

Naval Ocean System Command (NOSC) in San Diego has developed a system based on a circular ring of sensors with data collection managed by an array-processor/minicomputer system [Ref. 1]. The observation of enemy movements and activity beyond the Forward Edge of the Battle Area (FEBA) is the design objective of this system. This thesis uses data collected during a test of this system at the Marine Corps Air-Ground Combat Center at Twenty-nine Palms, California. Investigated are various methods of processing the seismic data collected. The objective of this research is to try to provide viable methods of satisfying system design objectives through signal processing techniques.

The following chapter addresses the design objectives and requirements for such a seismic sensor system. Additionally, capabilities and deficiencies of current systems and research are detailed. In order to intelligently address solutions to these requirements and deficiencies, an understanding of seismic theory and sensors is needed. General seismic theory is presented with emphasis on the constraining parameters for the use of the earth's surface as a medium for gathering seismic intelligence. Also highlighted are the similarities of the earth's surface to electro/optical phenomena and the resulting simplifying assumptions.

The sequence of design solutions investigated followed from the analogies and simplifying assumptions addressed in the study of seismic sensor theory. The problem of target identification or classification is approached using digital matched filtering of the time domain amplitude data. Frequency domain matched filtering was not considered, based upon the conclusion by NOSC that there appeared to be no consistent spectral lines for any of the possible target types, except for artillery [Ref. 1]. Matched filtering was used to identify single and multiple target classes occurring during a sample period. The chapter detailing the matched filter procedures and implementation also includes a description of support and validation software used in the analysis of the seismic data.

The validation software is primarily used to check the accuracy of the direction finding routines. These direction finding routines are the time domain phase difference procedure (TDPD) and a least mean squares polynomial (LMSP) curve fitting procedure. The combining of the matched filtering procedure and these direction routines allowed for multiple target direction finding. The theory and derivations of the two direction finding algorithms are presented in the multiple target direction chapter.

Application of these algorithms is performed first on simulated targets for validation of the procedures. The experimental data is then analyzed. A user's manual, which includes procedures for tape and mass storage operations, is provided as an appendix. This appendix describes how to set up and use the software system.

II. THEORY OF SEISMIC SENSORS

Elastic waves result from the stressing of an elastic media. The elastic media for seismic theory is the earth. Seismic theory is the study of the earth as a wave propagating media. Elastic waves propagate away from the source of seismic stress, e.g. an explosion [Ref. 2]. The energy, which propagates through the earth, travels via particle deformations. The elastic properties and densities of the earth media determine the velocities of these seismic waves. [Ref. 3]. Seismic wave sources of interest may be impulsive or continuous. Impulsive or short duration sources are artillery recoil or shell blasts. The time-limited nature of this type of seismic signal produces a broad range of frequencies. Continuous wave signals may be produced by tanks, trucks and low flying aircraft. These continuous wave signals may be described by narrow band frequency characteristics. The spectral power of a seismic source is a function of several parameters. A non-inclusive list of these parameters includes:

1. The vehicle's velocity and mass
2. The size of the explosive charge associated with the artillery or shell blast
3. The degree of coupling into the earth's surface
4. The geological structure over the wave's path

Information about the seismic source is contained in the waves which it generates. For example, in an array of seismic sensors, directional information is contained in the relative received signal phases. In otherwords, the relative phase differences between the signals received by the individual array elements can be used to compute the direction to the seismic source [Ref. 4]. These relative phase

differences represent the time delays of the waves as they pass the array's elements. The response of the array's elements is proportional to the amplitude and velocity of the earth's motion, relative to the geophone's sensing axis for the waves. [Ref. 3]

Assumptions about the propagation of seismic waves must be made to assist with their analysis. The earth is assumed to be made up of horizontal, homogeneous, and isotropic layers of material. These layers are assumed to be discontinuous in their elastic properties at their borders. This variance in the elasticity between the layers leads to an optical analogy for the wave propagation. Propagation paths may now be viewed as being direct, refracted, or reflected versions of the source's seismic waves. [Ref. 3]

There are four basic types of seismic waves. These types are compressional, shear, Love, and Rayleigh waves. Compressional waves are generated by impulsive sources such as shell blasts. Particle motion is along the direction of travel. Shear waves are characterized by particle motion orthogonal to the direction of motion. The Love wave is a surface wave which may occur as a result of the layering of the earth's surface. This layering effect acts as a wave guide for this type of wave. Particle motion is orthogonal to the direction of wave propagation. The Rayleigh wave is generally the strongest of the seismic waves. The Rayleigh wave travels along the free surface of the earth. Its particle motion direction is always in the vertical direction. It is the strongest wave generated by a compressional source. Its amplitude attenuates at a rate only inversely proportional to the square root of the distance. [Ref. 3]

The waves of primary interest for a seismic system are the Rayleigh and Love waves. This is due to the long-range propagation of these waves. Since these two wave types are orthogonal to each other, they must be sensed by different

geophones. Rayleigh waves may be sensed by vertical geophones and the Love waves by horizontal geophones. Since the Rayleigh wave is normally the strongest, and in order to reduce computational complexity, only verticle sensor data is used.

Rayleigh waves experience absorbtion losses, particularly at higher frequencies. This phenomenon occurs because of the lowpass filter effect of the earth. This filtering effect is further compounded due to the fact that the cutoff frequency of the earth diminishes with range. Further complications arise due to the dependence of wave velocity on frequency. The result being that the wave train may change with distance, reducing the correlation of the wave shape between its source and distant points. Other sources of error occur because of the weathering of the surface layer, irregularities in the sub-surface composition, variances in the earth's layers and surface geometry. [Ref. 3]

III. DEVELOPMENTAL REQUIREMENTS AND PROBLEM DEFINITION

A. PASSIVE TARGET ACQUISITION AND SURVEILLANCE

A battlefield commander possesses a definite requirement for real time combat intelligence. A significant tactical advantage is held by the commander who is able to integrate his available combat intelligence sources with his supporting arms, i.e., target acquisition and engagement. It follows that to be part of the target acquisition process, any real time, seismic sensing system must provide swift and accurate information on detected enemy targets. The specific requirements for such a system are the ability to detect, identify, and locate these targets [Ref. 5]. Additionally, the target's rate and direction of movement should be provided or made easily discernible.

Any seismic sensing system must be designed around the target acquisition cycle. The target acquisition cycle, as given by Dublin [Ref. 5], is as follows:

1. Search Time
2. Target Sensing
3. Information Processing
4. Display of Target Information
5. Analysis of Target Information
6. Time required to make a Decision
7. Time Required for Supporting Weapons to Respond

For a seismic system, a prioritized list of possible targets are as follows:

1. Artillery
2. Helicopters and Aircraft
3. Tracked and Wheeled Vehicles
4. Personnel

As may be expected, the relative amplitudes of these seismic targets vary widely. A seismic targeting system is therefore constrained as to the targets it can or can not be expected to effectively engage.

The variance in the relative amplitudes of seismic targets suggests a range of specifications for detection of these targets. As summarized by Dublin [Ref. 5], possible detection radii may be as shown in Table I. Radii are given for both short and long targeting systems.

TABLE I		
Target Detection Radii		
Target	Short Range System	Long Range System
Personnel	100M	None
Vehicles	1KM	10-20KM
Low Flying Aircraft	1KM	10-20KM
Hostile Weapons	1KM	15-20KM

The timeliness requirement is ancillary to the radii of detection specifications. Timeliness, as used here, refers to the total time commencing when the seismic sensor system first detects and processes the seismic target data and ending with the dissemination of the targeting information to command elements for disposition. This timeliness requirement ranges between five to fifteen minutes, depending upon the mobility of the target [Ref. 5].

The parameters having a direct effect on the timeliness of a system are the probabilities of false alarm and detection. These parameters directly relate to a system's value. Increased probability of detection with reasonable false alarm performance, combined with the ability to disregard

friendly targets, are practical design objectives for any targeting system. The ability to incorporate such design features into a seismic sensor system will reduce both the time wasted on invalid targets and the danger of undetected targets.

Once a valid target has been detected, target location information must be obtained. Stringent specifications for target locations allow for the system to support or enhance the effectiveness of: fire support systems, blind bombing, Harassing and Interdiction fire (H and I), and observerless artillery engagement.

B. CURRENT CAPABILITIES AND DEFICIENCIES

To date, numerous successful algorithms have been developed to determine direction to single targets. These algorithms include both time and frequency domain methods. Target identification of long range targets via seismic sensing has not yet met with equal success.

The modern battlefield is seldom a single target type environment. The complicated, real world problems of multiple target identification and multiple target engagement require solution before practical seismic sensor systems can be integrated into the target acquisition process.

IV. MATCHED FILTER CONCEPT AND DISCRETE ALGORITHM

A. MATCHED FILTER THEORY

As previously addressed, there exists a requirement for battlefield target identification/classification. The recovery and classification of target signals suggests a filtering requirement. Previous works and implementations have used frequency domain techniques [Ref. 4]. The Air Force's SKEET system and the U.S Army's Remote Battlefield Surveillance System (REMBASS) both have successfully implemented a spectral power approach for classification of seismic data. These systems, however, are for short range applications. Time domain approaches to target identification have been for the most part left unexplored.

The discrete matched filtering technique is an attempt to classify targets by their time domain amplitude pattern i.e., their seismic amplitude signature. The heuristic basis for this method evolved from the observation of visual differences between the amplitude versus time signals for the various classes of targets. The matched filter, being the optimum filter for detecting known signals, was selected [Ref. 6].

The discrete matched filter is described by equation 4.1, where $h(t)$ is the impulse response of a filter whose output signal to noise ratio at time t_0 is maximized. The unit step $u(t)$ has been added to assure causality for the system. The matched filter is [Ref. 7].

$$h(t) = s(t_0 - t)u(t) \quad (4.1)$$

The output signal is given by

$$s_{out} = \int_{-\infty}^{\infty} h(v) s(t - v) dv \quad (4.2)$$

The maximum signal to noise ratio is given by

$$(s_{out}^2/M^2)_{maxoutput} = E(t_0)/N_0 \quad (4.3)$$

Where M is the noise level at the filter output, N is the input noise level, and $E(t_0)$ is the energy in $s(t)$ up to time t_0 . In equation 4.2, the replica of the original known signal is reversed and translated in time to be convolved with the signal input to the filter, producing the optimum output signal to noise ratio.

For discrete realization of the matched filter as implemented, equation 4.2 becomes equation 4.4, where $h(k)$ is the reflected and translated known signal.

$$s_{out}(j) = \sum_{k=1}^N h(k) s(j - k) \quad (4.4)$$

Where N is the number of data points per sample period.

B. MATCHED FILTER ALGORITHM

Samples of known signals are stored in a data file and read into a 5120 array at the start of program execution. Five sample, or known filter signals, are recorded in this array. Each sample signal comprises 1024 of the 5120

elements of the matched filter array. To perform the matched filtering, the 1024 seismic data samples (unknown signal) are copied over five times on an array of 11264 size. These copies of the experimental data are separated by 1024 zeros on either side. Additionally, the leading and trailing elements of the experimental data are set to zero to eliminate switching spikes present in the data. Hence forth, this 11264 element array will be referred to as the working array for brevity. This working array will contain the results of the matched filtering. For M , given as the number of input signal and filter signal array elements and also the number of zeros, the requirement set forth in [Ref. 8] for nonoverlapping convolution of length L is satisfied. Equation 4.5 establishes the minimum length for nonoverlapping convolution of one matched filter segment.

$$L = 2M - 1$$

(4.5)

Figure 4.1 depicts the working array layout. Notice that the first filter signal, "Tracked Vehicle" is loaded into the $h(t)$ array and is convoluted with the working array elements one through 2048. Once the leading data point of $h(t)$ reaches the working array's element 2048, a new known signal is loaded into $h(t)$, "Wheeled Vehicle", and the convolution is continued for working array elements 2048 to 4096. This process continues until the last of the five $h(t)$ filter signals has been read in and the convolution has been performed on all copies of the data. Notice that working data array elements 10240 to 11264 are for array length over-run protection.

Prior to convolution, each filter sample signal and the input signal are equalized to the same power level through

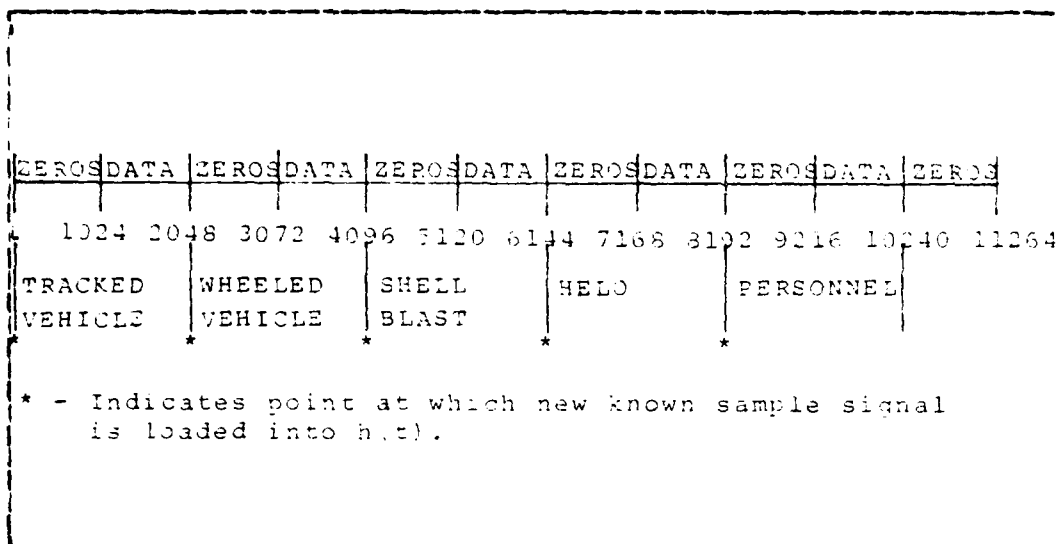


Figure 4.1 Working Array Configuration

division by their respective root mean square values. Additionally, after the convolution, the entire working array is normalized with respect to its maximum amplitude element. The maximum value in each target classification section is then compared with the interactively selected matched filter threshold. If the section's peak value is above the threshold value, that class of target is declared to be present. This method allows for the simultaneous detection and classification of multiple targets. It should be noted that this equalization forces both high and low amplitude signals to an equivalent average power level. This is felt to be justified in that signal amplification is not the goal, rather target detection and classification is. In the case of a single target, the known signal that most closely matches the unknown signal is anticipated to have the greatest amplitude matched filter spike. Figure 4.2 is a sample output.

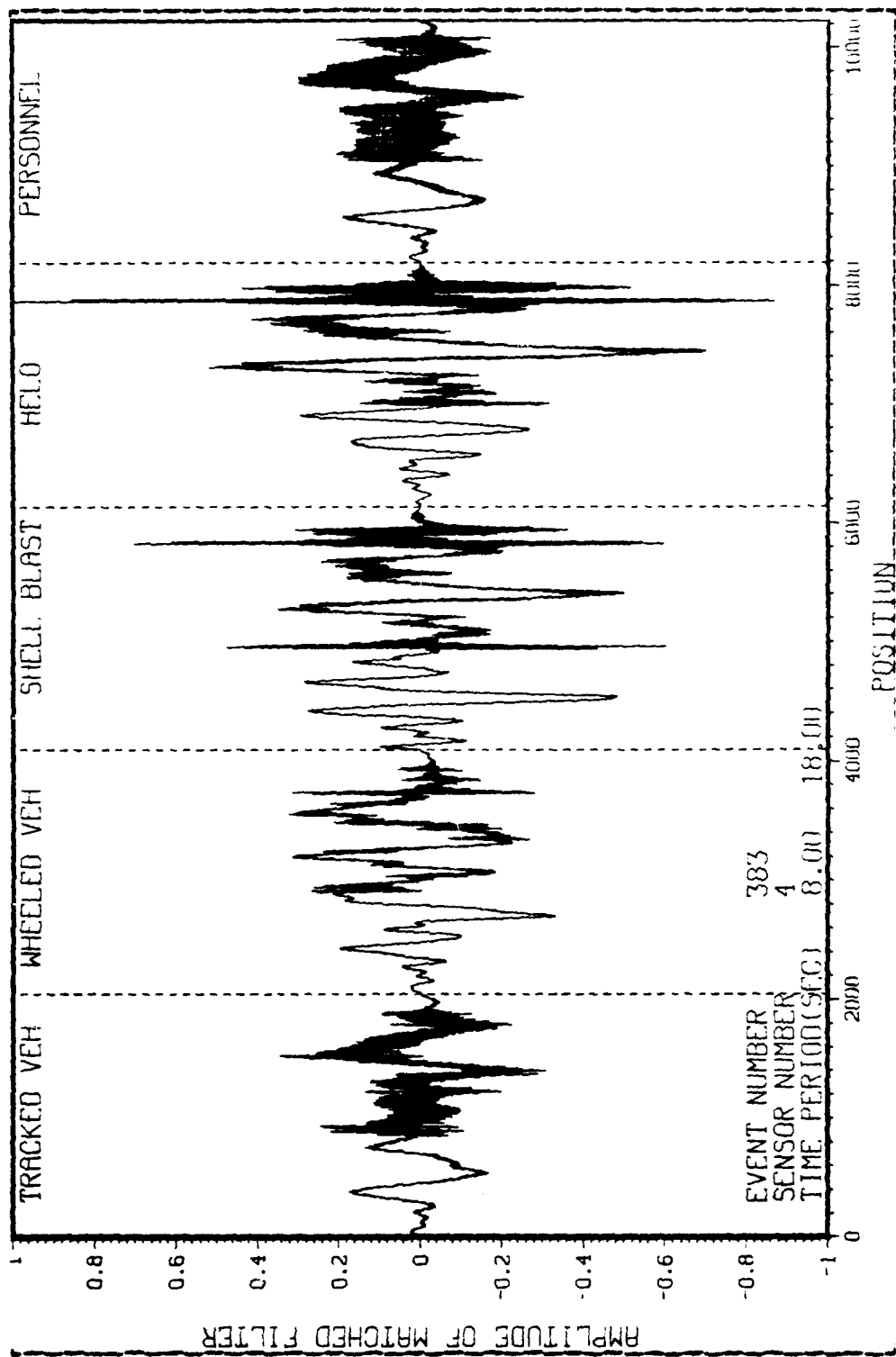


Figure 4.2 Sample Matched Filter Output

C. MULTIPLE TARGET MATCHED FILTERING

The upper limit of equation 4.4 may be selected to be from one to 1024 when called by the multiple target direction routine. This allows for the selection of reduced program execution times. Target identification, however, is always made by the full 1024 element buffer. When the matched filter target identification routine is used by the multiple target direction finding routine, data windows of less than 1024 are formed from a segment of the 1024 elements by extracting the segment size required around the maximum signal value. Figure 4.3 illustrates this procedure.

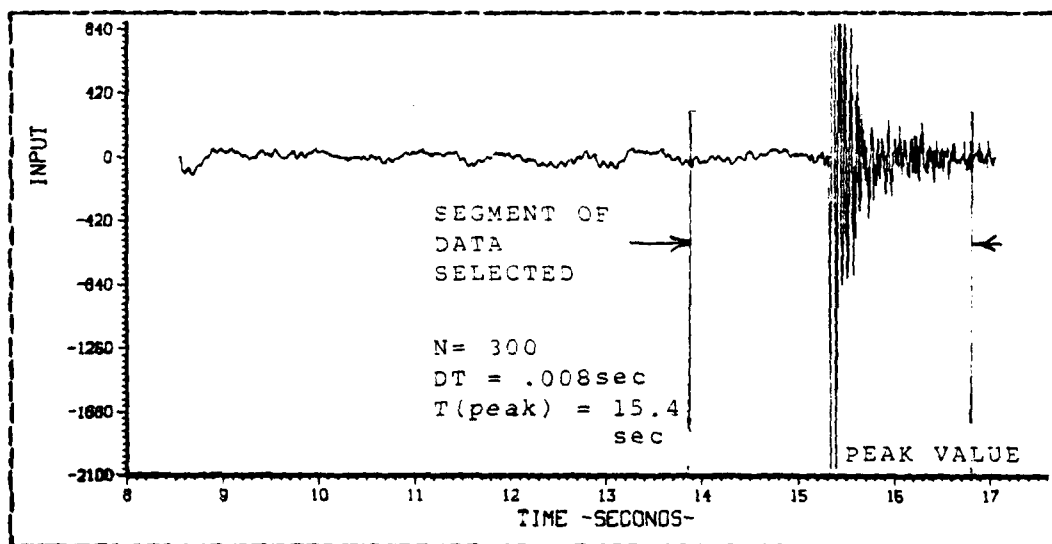


Figure 4.3 Windowing of Experimental Data

D. COMPUTATIONAL REQUIREMENTS

The computational requirements for this procedure are described by equation 4.6, where N_0 is the total number of operations required by the matched filter routine.

$$NO = (Ndf^2)(Nc) + (Ndf - 1)(Ndf)(Nc) \quad (4.6)$$

Ndf is the number of data elements and also the number of filter elements. Nc is the total number of target classes. The term Ndf^2Nc is the total number of multiplications required, while $(Ndf - 1)NdfNc$ is the total number of additions.

When used for target identification, Ndf equals 1024 and Nc equals five. Equation 4.6 gives a total of 10,480,640 operations for these array sizes. When the matched filter routine is initiated by the multiple target direction routine, the value of Ndf can be selected to range from one to 1024. Figure 4.4 is a plot of equation 4.6 and shows the computational consequence for selection of large values for Ndf. Note that equation 4.6 must be multiplied by the number of sensors in the ring when computing the number of operations for the multiple direction routine.

Window sizes of 100 and 200 were found to provide acceptable accuracy with greatly reduced computation times. As shown by figure 4.4, a window size of 200 requires 399,000 operations, while a window size of 100 requires only 99,500 operations. The number of operations required was found, as expected, to be proportional to the execution time of this routine.

B. SUPPORT SOFTWARE

1. Amplitude Analysis (Timeout)

A graphical output is provided by this amplitude routine which displays the relative amplitude versus time for a selected sensor. The initial target direction and target classifications found are also displayed. This routine allows for the interactive selection of the

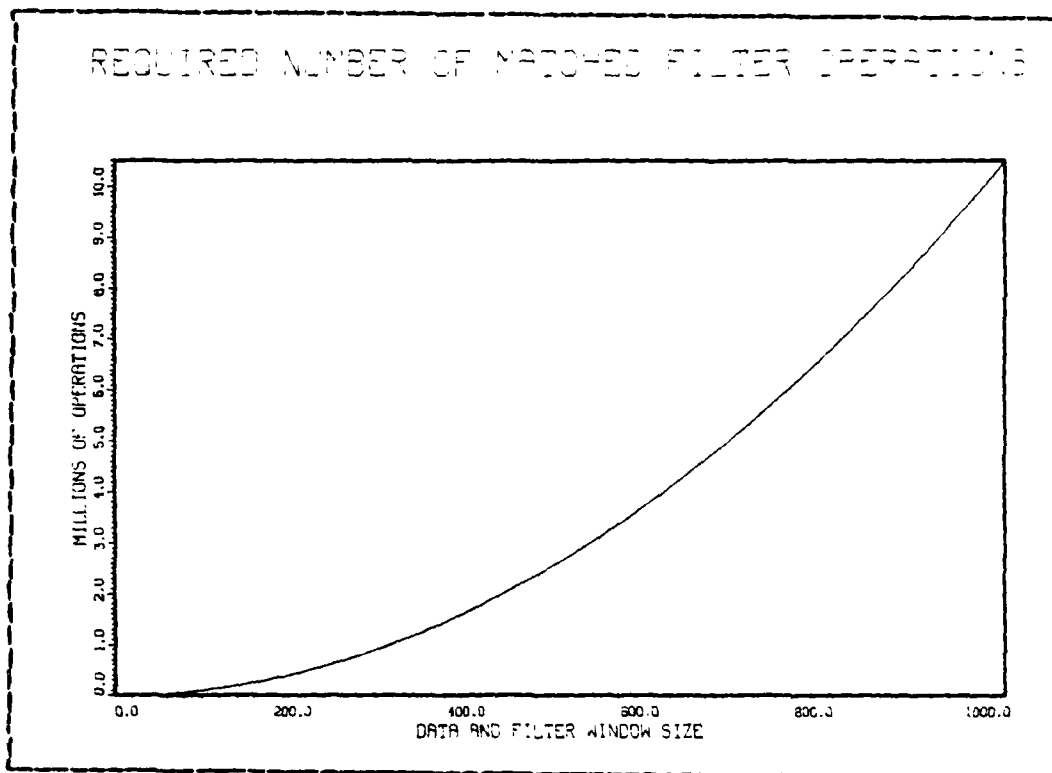


Figure 4.4 Data Window Size vs Number of Operations

amplitude response of any sensor as a sample target for later use in the matched filter analysis. In this way sample, signals can be catalogued and evaluated as filter signals. The axes of the graphical output adapts to the data's maximum amplitude and to the time period involved. Figure 4.5 is an example of amplitude analysis output.

2. Frequency Analysis (Freqot)

The frequency routine, as in the amplitude analysis routine, allows for initial primary target direction, target classifications, and for the time period of the experimental data to be displayed. Normalized spectral power versus frequency is graphically displayed as shown in figure 4.6.

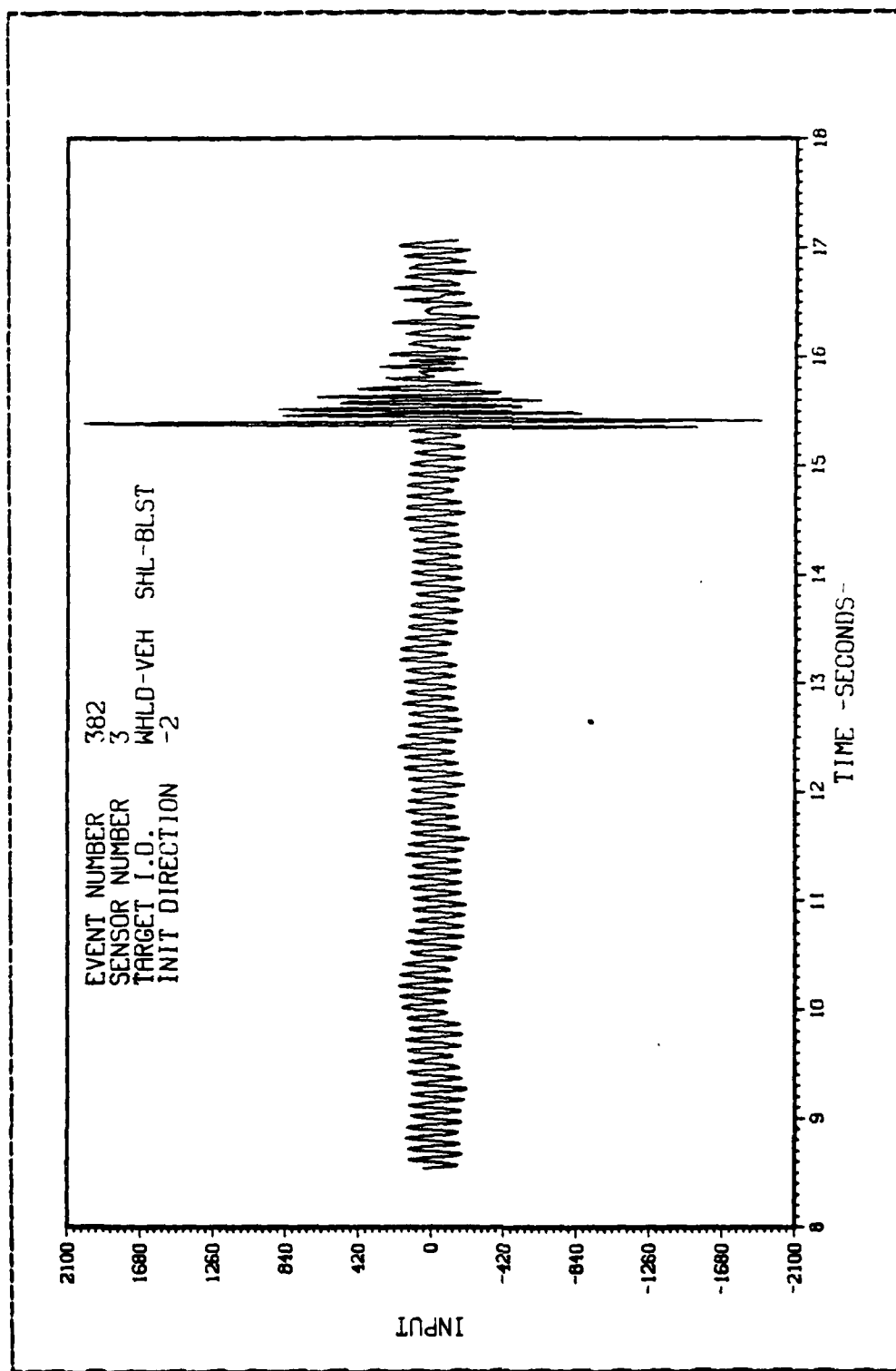


Figure 4.5 Sample Amplitude versus Time Output

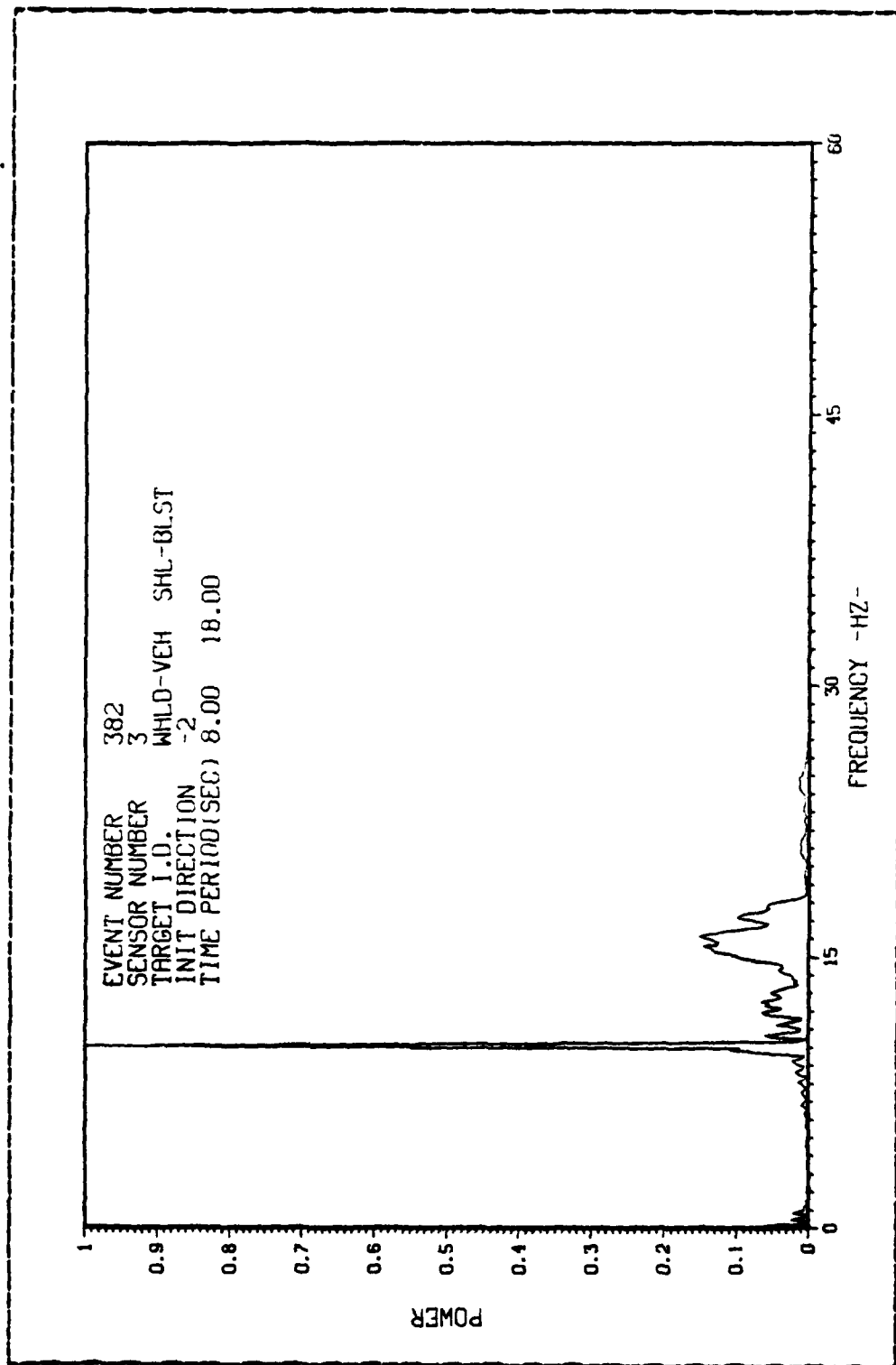


Figure 4.6 Sample Frequency versus Power Output

3. Simulation and Validation Software (SIMULT)

In order to validate the various algorithms and their implementation in software, a testing procedure was required. The specific algorithms which the simulation routine was designed to validate are the initial angle, phase difference and least mean square target direction routines. These routines, as previously described, use the relative time differences of the seismic signal's peak amplitude response or the peak matched filter response respectively. Validation of these routines is performed by allowing the creation of simulated targets with selected arrival angles.

Simulated targets can be created in the routine SIMULT. Up to four sine wave targets of selected frequency, amplitude, and direction can be input during each sample period. These simulated targets are added to the experimental seismic signal data for the sample period. Correspondingly, zero direction filter data must have been written into the matched filter data file at these selected sample target frequencies.

The directions for the simulated targets are created by introducing relative phase delays between the sine waves that are added to each sensor's seismic data. This is implemented by introducing a zero phase to the sensor in the desired direction of the simulated target. The phases of the other sensors are increased proportionally by their distance on the circular array away from the zero phase shift sensor. Figure 4.7 illustrates the case of a zero degree simulated target direction.

A summary of the target directions found by the multiple target direction routine and the simulated targets entered is provided by the multiple direction plotting routine. Figure 4.8 is a sample output.

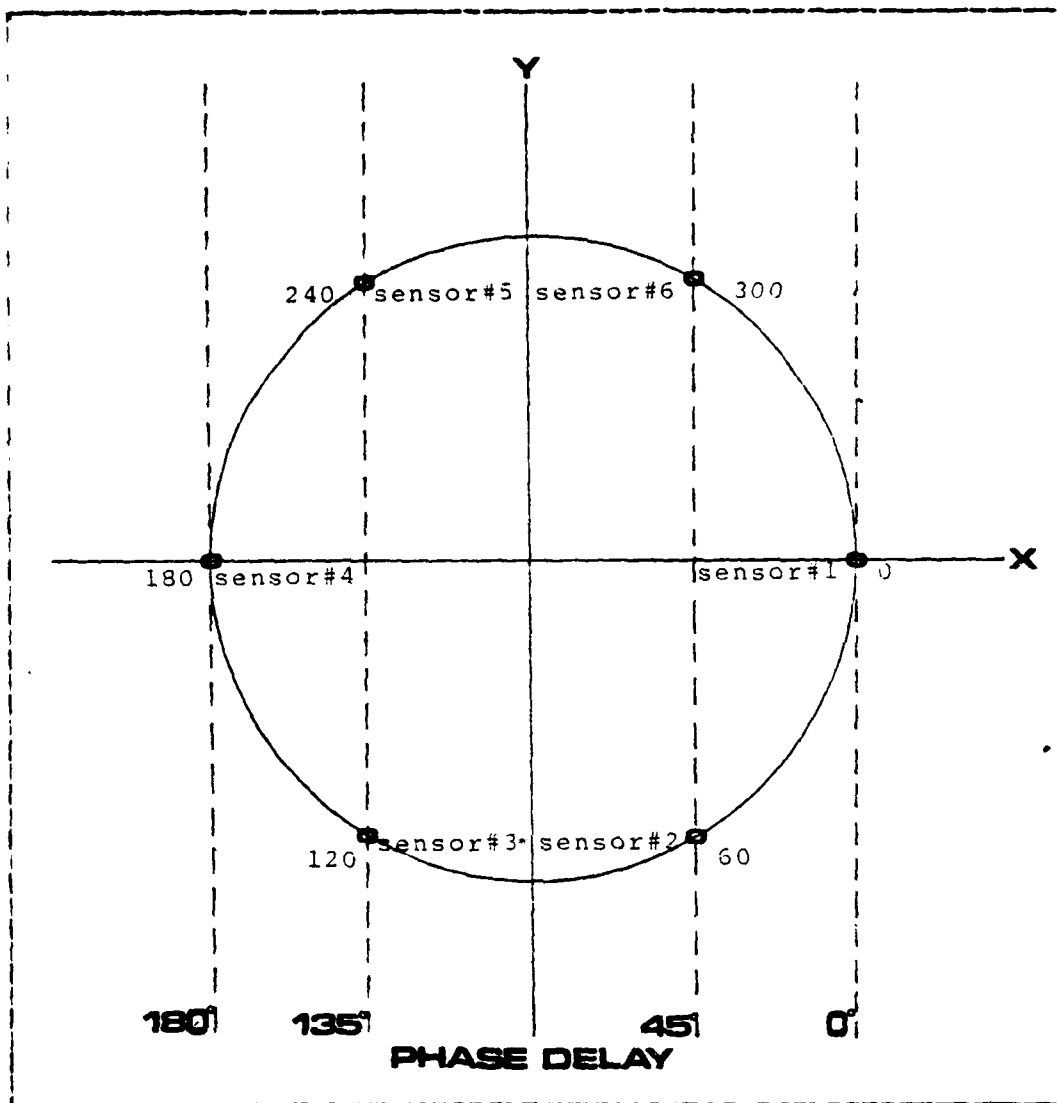


Figure 4.7 Simulation of a Zero Degree Target

MULTIPLE TARGET - MATCHED FILTER OUTPUT

EVENT NUMBER	375
TIME PERIOD(SEC)	25.00 35.00
WHEELED VEHICLE	DIRECTION - -59.00
SHELL BLAST	DIRECTION - 315.00
PERSONNEL	DIRECTION - -59.00
SIMULATED TRKD VEHICLE	TARGET FREQUENCY 0.00
AMPLITUDE	0.0000
DIRECTION	0.0000
SIMULATED WILD VEHICLE	TARGET FREQUENCY 0.00
AMPLITUDE	0.0000
DIRECTION	0.0000
SIMULATED HELICOPTER	TARGET FREQUENCY 0.00
AMPLITUDE	0.0000
DIRECTION	0.0000
SIMULATED PERSONNEL	TARGET FREQUENCY 0.00
AMPLITUDE	0.0000
DIRECTION	0.0000

Figure 4.8 Sample Multiple Target and Simulated Target Output

V. MULTIPLE TARGET DIRECTION

A. THEORY AND DESIGN CRITERION

The modern battlefield is comprised of many classes and quantities of seismic signals. Using these signals for target identification and acquisition is the purpose of battlefield seismic sensors. As presented in the last chapter, matched filters can be used for target identification. In this chapter, it will be shown that matched filter information may also be used to obtain target bearing.

Multiple target acquisition requires the concurrent separation of the target classes and the computation of direction for each of the target classes found. The output of the matched filter is a spike at t_0 , the time of peak signal detection. If matched filtering is performed for each of the sensors in the ring, the value of their respective t_0 , for each filter output, would be different. Since the seismic waves impinge upon each sensor at different times, dependent upon the target direction, the values for t_0 for each sensor may be expected to be directly related to the arrival angle of the seismic wave. The t_0 spike of the matched filter output may be thought of as a signal compression for both the continuous (tank, truck etc.) and the time limited (shell blast, artillery recoil etc.) seismic signals [Ref. 6].

Time domain methods [Ref. 1] use the positional differences of known wave points to geometrically estimate direction. The times associated with a sensor ring's peak amplitude responses may be explained by way of an illustrative example of a shell blast. A rough direction to the origin of this shell blast can be computed using the

relative time differences associated with the peak amplitudes of all of the ring's sensors [Ref. 9].

The enhanced time positional information, which is a by-product of the matched filtering, can be used to perform just such a time domain approach. The motivation for the method is that, unlike other time and frequency domain methods, which are very susceptible to noise corruption inaccuracies, matched filtering pulls the signal out of the noise and optimally detects the signal at time t_0 . Two time domain methods are evaluated for finding arrival angles of the seismic signal. The first being the time domain phase difference (TDPD) method. The second being a least mean squares polynomial (LMSP) curve fitting approach.

B. MULTIPLE TARGET FILTERING ALGORITHM (MULTI)

The routine MULTI calls the matched filter routine for each sensor's amplitude signal. Returned are the target classes found with their relative peak filter response positions. Figure 5.1 illustrates a two target case. The matched filter response and the relative time position for the two classes of targets can be seen. The figure shows that a shell blast target is present. The relative time for this target class is 5800. The relative time returned for the simulated wheeled vehicle target is 3000. These times or positions are relative since each sensor's filter response peaks are offset in time with respect to the peaks of the other sensors. This allows for simultaneous direction finding for each class of target. This algorithm allows selection of either a time domain phase difference or or a least means square polynomial algorithm for finding the direction to the targets. The time domain phase difference algorithm is derived first [Ref. 1]. The least mean squares polynomial direction algorithm then follows and is believed to be an original application to this field [Ref. 10].

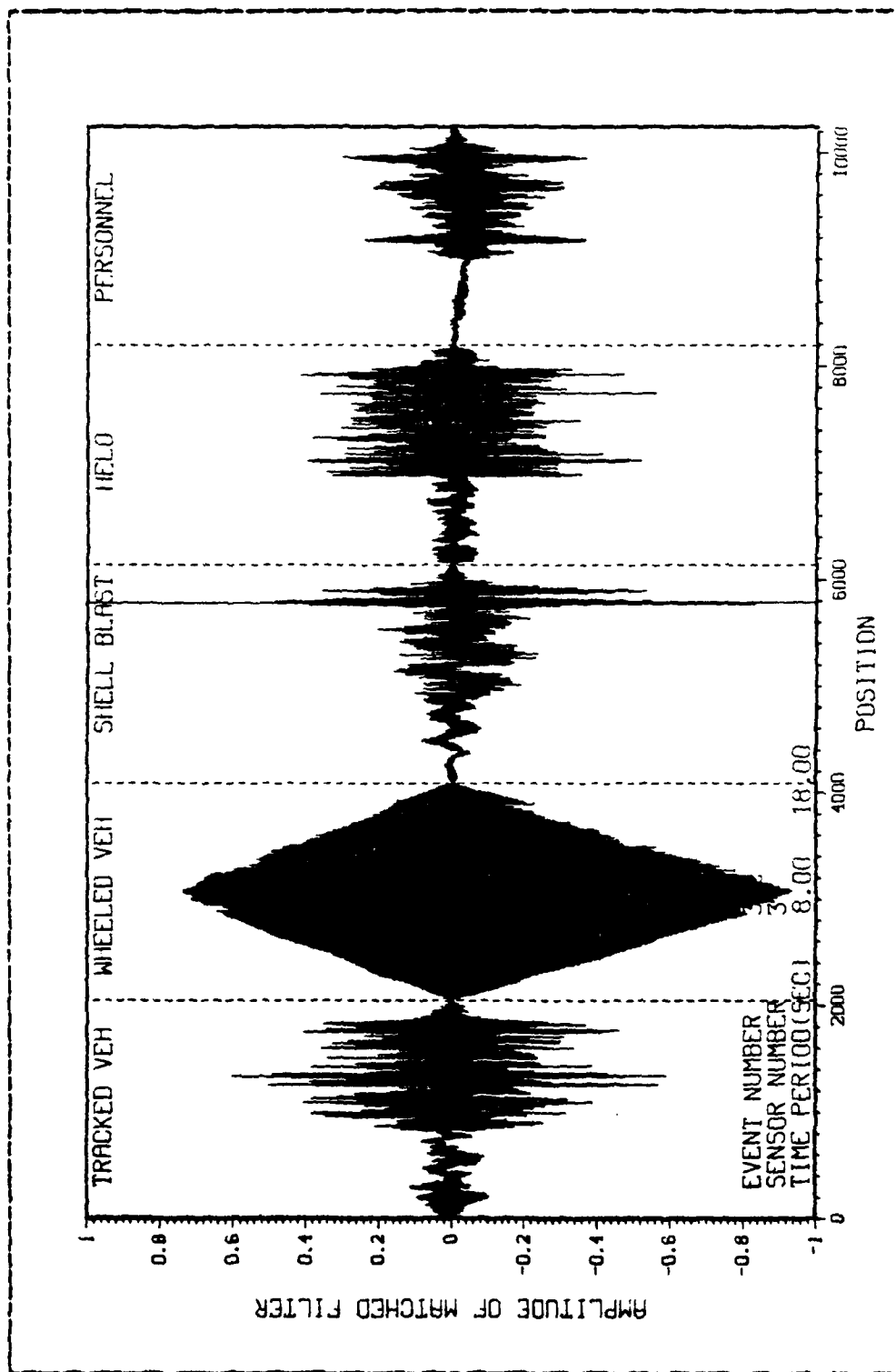


Figure 5.1 Two Target Matched Filter Response

1. Multiple Target Direction Phase Difference Algorithm

T_{ij} - the arrival time of the seismic signal at the I-th sensor for the J-th target class

θ_i - the angle of the I-th sensor to the x-axis

D_{ji} - the distance from the origin to where the wave front of the J-th target passes the I-th sensor

T_{0j} - the time when the J-th target class passes the origin

X_i, Y_i - the position of the I-th sensor

B_j - the arrival angle of the seismic wave for the J-th target class

V - the seismic wave velocity

R - the radius of the sensor ring

I - the sensor number where I has integer values from one to nine

J - the J-th target class

N - the number of sensors in the ring

Figure 5.2 illustrates these parameters and their interdependence. The derivation of the algorithm follows:

$$\theta_i = 2 (I - 1) / N$$

where zero degrees is set parallel to the x-axis

$$X_i = R \cos \theta_i$$

$$Y_i = R \sin \theta_i$$

$$D_{ji} = R \cos(\theta_i - B_j)$$

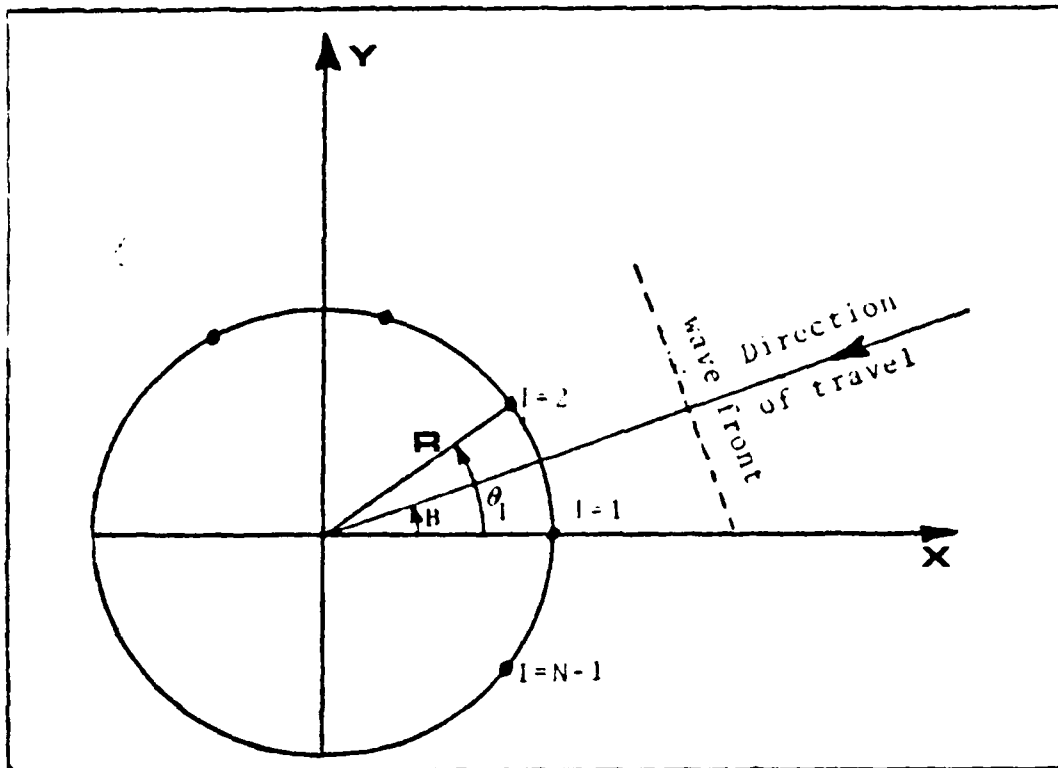


Figure 5.2 Circular Sensor Array Geometry

or equivalently $D_{ji} = (X_i) \cos B_j + (Y_i) \sin B_j$

$$Tc_j = (1/N) \sum_{i=1}^N T_{ji}$$

$$\tau_{ji} = Tc_j - T_{ji}$$

$$V = D_{ji} / \tau_{ji}$$

$$V = ((X_i) \cos B_j + (Y_i) \sin B_j) / (Tc_j - T_{ji}) \quad (5.1)$$

Since the wave velocity can be assumed to be constant when passing all sensors, then for any sensor I and K where $I \neq K$, equation 5.1 leads to

$$\frac{((X_i) \cos B_j + (Y_i) \sin B_j) / (T_{cj} - T_{ji})}{((X_i) \cos B_j + (Y_i) \sin B_j) / (T_{cj} - T_{jk})} =$$

Where $I \neq K$

Now solving for the arrival angle B_{jik} of the wave

$$B_{jik} = \arctan \left(\frac{((T_{cj} - T_{jk}) X_i - (T_{cj} - T_{ji}) X_k)}{((T_{cj} - T_{ji}) Y_k - (T_{cj} - T_{jk}) Y_i)} \right)$$

or equivalently;

$$B_{jik} = \arctan \left(\frac{((T_{cj} - T_{jk}) \cos \theta_i - (T_{cj} - T_{ji}) \cos \theta_k)}{((T_{cj} - T_{ji}) \sin \theta_k - (T_{cj} - T_{jk}) \sin \theta_i)} \right)$$

Where the values of T_{ji} and T_{jk} are returned values from the matched filter routine.

Now;

$$B_j = (1/(N)^2) \sum_{i=1}^N \sum_{k=1}^N B_{jik}$$

Where B_j is the direction in radians to the J-th class target. For the multiple direction routine as implemented, 'j' has values from one to five.

2. Least Mean Square Polynomial Direction Finding

The least mean square direction finding algorithm was developed in response to problems encountered with the phase difference direction finding algorithm. This new method is based on a least mean squares polynomial curve fit of the sensor data. This approach was selected since the least mean squares polynomial provides for best fit or a maximum likelihood curve fit for noisy data.

The least mean squares direction finding algorithm, as with the phase difference algorithm, assumes the seismic wave to be planar. Figure 5.2 illustrates the parameters for this model. Once the assumption of a planar seismic wave is made, the expected relation between the arrival

angle, relative delay times and sensor position in the circular array, can be made. Figure 5.3 illustrates these relations for a nine sensor circular array with a seismic wave arriving at zero degrees. Notice that the relative delay times have been scaled to be from zero to one.

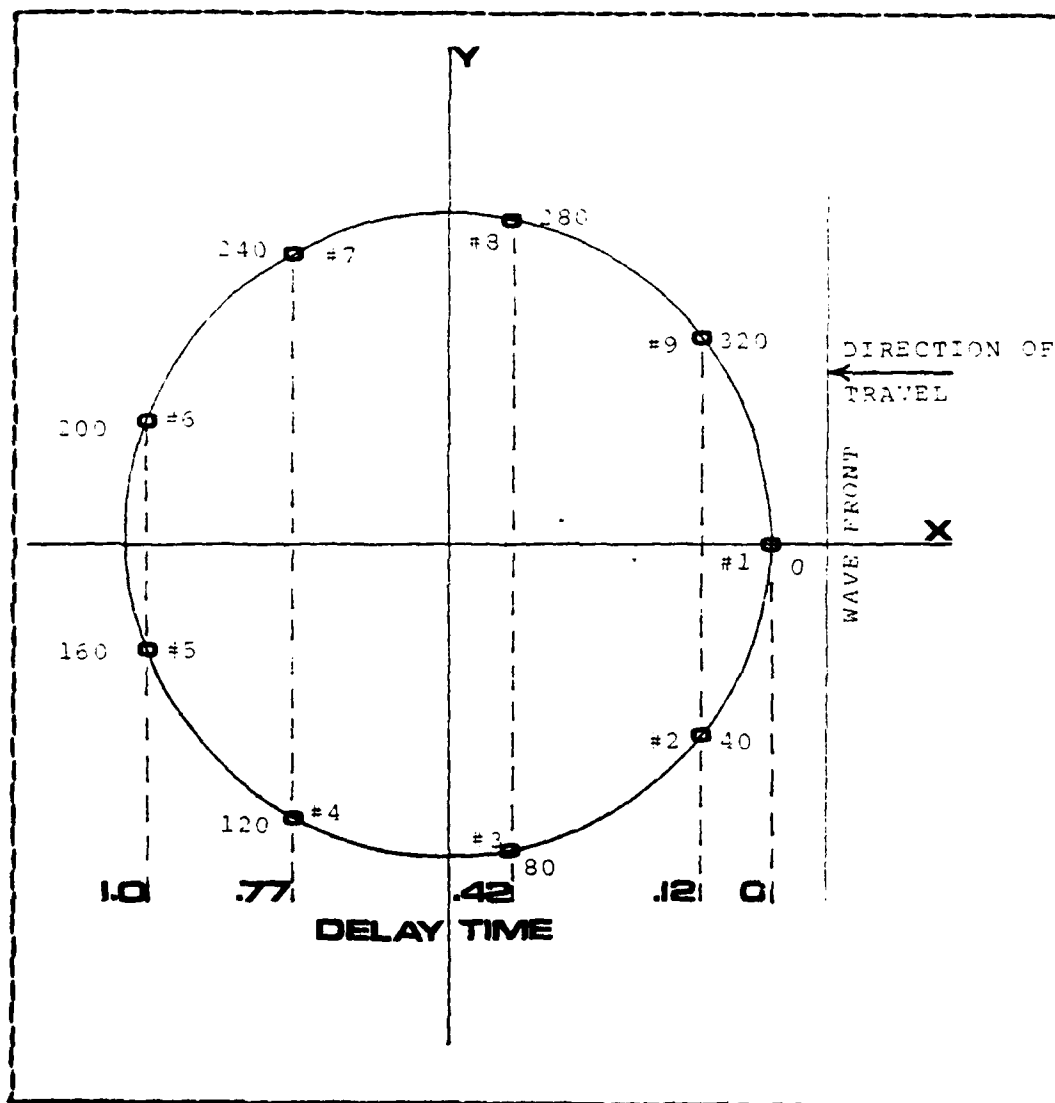


Figure 5.3 Relative Delay Times in a Nine Sensor Ring

Since the sensors in a nine sensor ring are at increments of forty degrees relative to the x-axis, a correlation can be seen between the relative time delays at each sensor and the arrival angle of the seismic wave. A plot can now be made to illustrate the relationship of sensor angles versus relative time delays. Figure 5.3 is a plot for a nine sensor ring with a planar seismic wave arriving at zero degrees. Figure 5.4 shows that for a wave at an arrival angle of 160 degrees, the delay time at sensor number five will be zero. It can be seen that the fitting of these ideal data points with a least mean squares polynomial will produce an equation for a curve whose minimum value is also at the arrival angle of the seismic wave. The minimum degree of the polynomial to fit this ideal data is four. This results from noting that the curve in figure 5.4 has three curve inflections. For experimental data, this minimum curve point corresponds to the predicted arrival angle.

Polynomials of degrees higher than four may be expected to enhance arrival angle errors since the polynomial would distort to fit noisy data. Least means squares polynomials of degree two and three however, may be useful in reducing curve sensitivity to one or two malfunctioning sensors or excessively noisy data.

3. Least Mean Squares Polynomial Algorithm Derivation

Let:

N - number of data point pairs

Y_i - the observed or experimental data position values

X_i - independent degree values with a range of 0 to 360 degrees

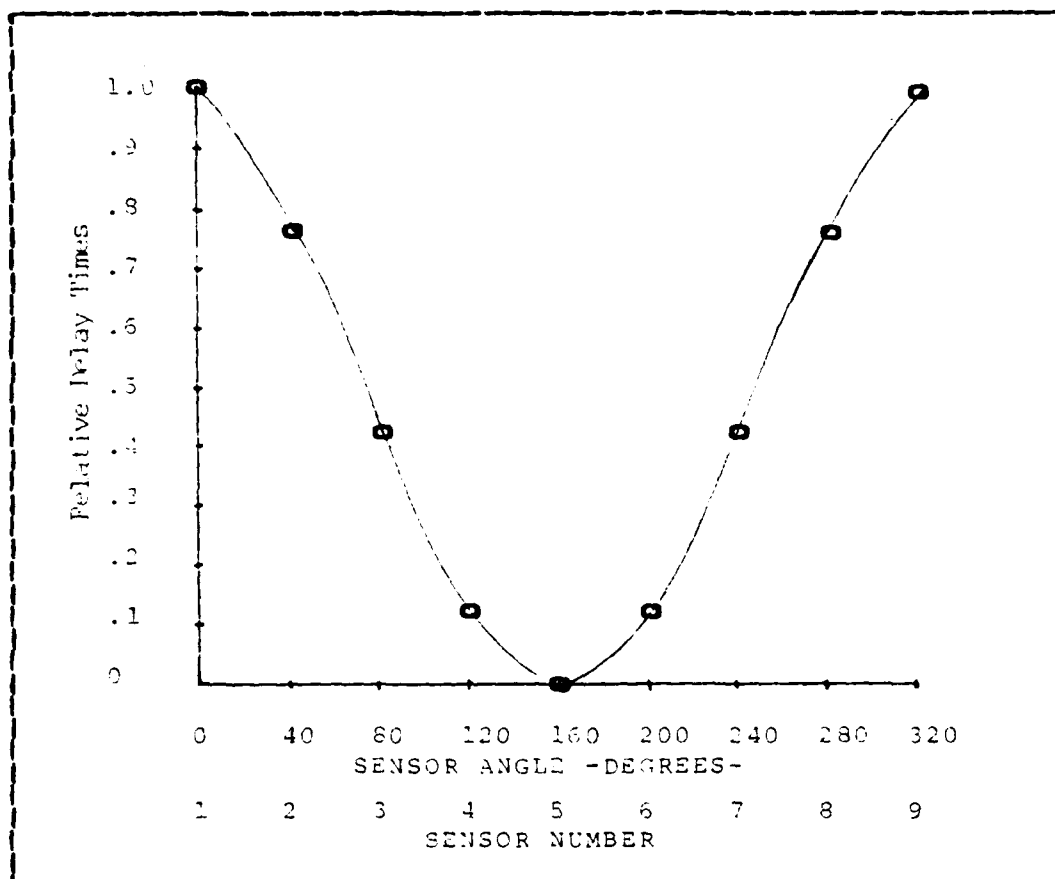


Figure 5.4 Relative Time Delay versus Sensor Angle

P_i - dependent predicted delay time values found by the least mean squares polynomial

A_i, B_i - coefficient values for the system of simultaneous equations

E_i - error between the experimental data values and predicted values of time delays

S - sum of the square errors between each data and predicted points

The derivation of the least mean squares polynomial follows:

$$S = E_1^2 + E_2^2 + E_3^2 + \dots + E_N^2$$

$$P_i = A_0 + A_1 X_i + A_2 X_i^2 + A_3 X_i^3 + \dots + A_n X_i^n \quad (5.2)$$

$$E_i = Y_i - P_i \quad (5.3)$$

$$S = \sum_{i=1}^N E_i^2 \quad (5.4)$$

Combining equations 5.2 and 5.3 yields

$$E_i = Y_i - A_0 - A_1 X_i - A_2 X_i^2 - \dots - A_n X_i^n$$

where n is the degree of the polynomial such that $N > n + 1$ and $1 < i < N$

Equation 5.4 now becomes, after substituting for E , equation 5.5

$$S = \sum_{i=1}^N (Y_i - A_0 - A_1 X_i - A_2 X_i^2 - \dots - A_n X_i^n)^2 \quad (5.5)$$

To find the minimum of the sum of the squares expressed by equation 5.5, the partial derivatives of S with respect to all of the coefficients are taken. At the minimum, these partial derivatives all vanish.

$$\begin{aligned} \partial S / \partial A_0 &= 0 = \sum_{i=1}^N 2(Y_i - A_0 - A_1 X_i - \dots - A_n X_i^n) (-1) \\ \partial S / \partial A_1 &= 0 = \sum_{i=1}^N 2(Y_i - A_0 - A_1 X_i - \dots - A_n X_i^n) (-X_i) \\ &\quad \cdot \\ &\quad \cdot \\ \partial S / \partial A_n &= 0 = \sum_{i=1}^N 2(Y_i - A_0 - A_1 X_i - \dots - A_n X_i^n) (-X_i^n) \end{aligned}$$

Dividing by two and rearranging gives $n + 1$ normal simultaneous equations. Expressed in matrix notation these equations become

$$\begin{bmatrix} N & \sum X_i & \sum X_i^2 & \dots & \sum X_i \\ \sum X_i & \sum X_i^2 & \sum X_i^3 & \dots & \sum X_i^4 \\ \sum X_i^2 & \sum X_i^3 & \sum X_i^4 & \dots & \sum X_i^5 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum X_i^n & \sum X_i^{n+1} & \sum X_i^{n+2} & \dots & \sum X_i^{2n} \end{bmatrix} \begin{bmatrix} A_0 \\ A_1 \\ A_2 \\ \vdots \\ A_n \end{bmatrix} = \begin{bmatrix} \sum Y_i \\ \sum X_i Y_i \\ \sum X_i^2 Y_i \\ \vdots \\ \sum X_i^n Y_i \end{bmatrix}$$

The coefficient matrix in the above system of equations can be solved for. The minimum value for P_i can then be found. The X corresponding to this minimum value is declared to be the predicted arrival angle of the seismic wave. [Ref. 10]

4. Adaptive Target Direction Finding

An adaptive method is used to improve the directions found for all single target cases. To enhance the resolution accuracy of the peak filter output position, the single target seismic data is copied into the filter data section for its class of target. Subsequently, when the matched filter routine is called for each sensor, filtering is performed with sensor data from that time period.

5. Software features of the Multiple Target Direction Routine

As first addressed in the previous chapter, the multiple direction routine allows for the selection of the number of data points to be used from the 1024 size buffer of sensor data. This was included to investigate the algorithm's performance for various data/filter window sizes and to allow an option for reduced CPU time utilization for interactive program runs. Tabular output from this routine includes the event number, the time period for the data, identification and directions for up to five target classes, and up to four simulated target specifications. Also, notice that the routine adapts to the number of sensors

specified for the ring. This was necessary since the experimental data included sensor rings of six and nine vertical sensors.

C. MULTIPLE TARGETS OF THE SAME TARGET CLASS

A limitation on the system, as presented up to this point, is its inability to engage multiple targets of the same class. Discrete targets may appear as separate entities since the seismic signals generated by the two same class targets are not likely to be incident at the same time. This situation is greatly complicated for continuous time signals, such as tracked or wheeled vehicles. For such continuous time targets, mutual distortion would be the likely result.

An algorithm is needed to identify these multiple peaks without erroneously declaring elements in the same peak as targets. By using only the positive half of the matched filter output and performing data smoothing on the remaining points, a curve with its number of peaks equalling the number of targets present could be generated. Numerical methods for curve fitting or interpolation are available [Ref. 11.]. Polynomial curve fitting, Sterling's method and variations of Newton's method are only a few of the possible approaches applicable for equally spaced data. By differentiating the smoothed data, the peaks and valleys of the filter output can be found. The height of the curve corresponding to the points where the derivative is zero can now be compared to the selected matched filter threshold. This excludes the valley points and leaves the points remaining which correspond to the relative times of the same class targets.

This method is constrained by resolution of close proximity, time limited targets and near phase synchronization

of continuous targets. The variables to be optimized through experimentation may be the degree of the smoothing of the curve data and the exclusion of erroneous valleys associated with the same target's data.

VI. ANALYSIS OF SEISMIC DATA

Analysis of the simulated and experimental seismic data will be conducted as detailed in Table II. Table III lists the matched filter contents for the simulated data. Table IV is the test plan for the experimental data. Table V lists the matched filter signals used for the experimental data analysis.

Table VI summarizes the results of the simulated and experimental data runs for direction finding. The window size, used for all multiple target direction finding results, was 300. Table VII lists the events in which targets were missed or incorrectly identified.

The time domain phase difference directions found, are not presented for the reasons noted earlier. Errors of up to eighty degrees were not uncommon with this method.

Each event run will be accompanied by the following graphic output:

1. Least Mean Square Initial Direction
2. Matched Filter Response
3. Amplitude Response
4. Amplitude Response of any Malfunctioning Sensor
5. Frequency Response
6. Least Mean Squares Polynomial Curve Fitting (LMSP)
Using Matched Filter Outputs
7. Multiple Target Direction Summary Resulting from
Least Mean Squares Curve Fitting

TABLE II
Test Plan for Simulated Data

<u>Event</u>	<u>#Sen</u>	<u>#Tots</u>	<u>Frequency</u>	<u>Amplitude</u>	<u>Direction</u>
001	9	1	10	3000	0
001	9	1	10	3000	40
001	9	1	10	3000	120
001	9	1	10	3000	240

TABLE III
Matched Filter for Simulated Targets

<u>Filter</u>	<u>Frequency</u>	<u>Amplitude</u>	<u>Direction</u>
Tracked Veh	30	2000	0
Wheeled Veh	10	2000	0
Shell blast data from event #383			0
Helicopter	15	2000	0
Personnel	20	2000	0

TABLE IV
Test Plan for Experimental Data

<u>Event</u>	<u>#Sen</u>	<u>#Tgts</u>	<u>Dir</u>	<u>Target</u>	<u>Distance</u>
383	9	1	0	Shot	5KM
382	9	1	0	Shot	5Km
372	6	1	315	Helicopter	5 - 15KM
375	6	1	0	Tank	5 - 0Km
374	6	1	315	Helicopter	15 - 5KM
302	6	1	0	Mortar	1KM
314	6	1	315	LVT	4 - 5KM
354	6	5	0	105mm How	5Km
			225	175mm Gun	4KM
			315	LVT	4 - 5KM
			0	M-60 Tank	4 - 5KM

TABLE V
Matched Filter for Experimental Data

<u>Target</u>	<u>Event Used as Filter</u>
Tracked Vehicle	375
Wheeled Vehicle	none (background noise)
Blast/Recoil	383
Helicopter	372
Personnel	none (background noise)

TABLE VI
Summary of Direction Finding Results

<u>Event</u>	<u>#Sen</u>	<u>#Tqts</u>	<u>Distance</u>	<u>Dir</u>	<u>Initial</u> <u>Dir</u>	<u>Error</u>	<u>LMSP</u> <u>Dir</u>	<u>Error</u>
001	9	1	N/A	0	N/A		0	0
001	9	1	N/A	40	N/A		40	0
001	9	1	N/A	120	N/A		120	0
001	9	1	N/A	240	N/A		240	0
383	9	1	5KM	0	4 (3)	1.1	28 (4) -5	7.8 1.4
382	9	1	5KM	0	-14 (4)	3.9	-6	1.67
372	6	1	5 - 15KM	315	-32	3.6	-59	3.9
375	6	1	5 - 0KM	0	312	13.3	FAILED	
374	6	1	15 - 5KM	315	-59	4.0	FAILED	
302	6	1	1KM	0	4	1.1	6	1.7
354	6	5	5KM	0	-3	.8	3	.8
			4 KM	225	291	18.0	FAILED	
			4 - 5KM	315	FAILED			
			4 - 5KM	0	FAILED			

*Note: The matched filter threshold was set at .9 for all single targets and .6 for all multiple target events. Brackets indicate the use of other than a second degree polynomial.

TABLE VII
Missed or Incorrectly Identified Targets

<u>Event</u>	<u>#Targets</u>	<u>Target</u>	<u>Nature of Errors</u>
375	1	Tank	1. Matched filter was not based on a high S/N sample signal
374	1	Helo	2. Small seismic signal amplitudes
354	1	LVT	3. Malfunctioning sensor(s)
	1	Tank	
	1	175mm Gun	Sever clipping distortion of input signal

Note: The numbered error sources apply to all events listed.

LEAST MEAN SQUARES POLYNOMIAL

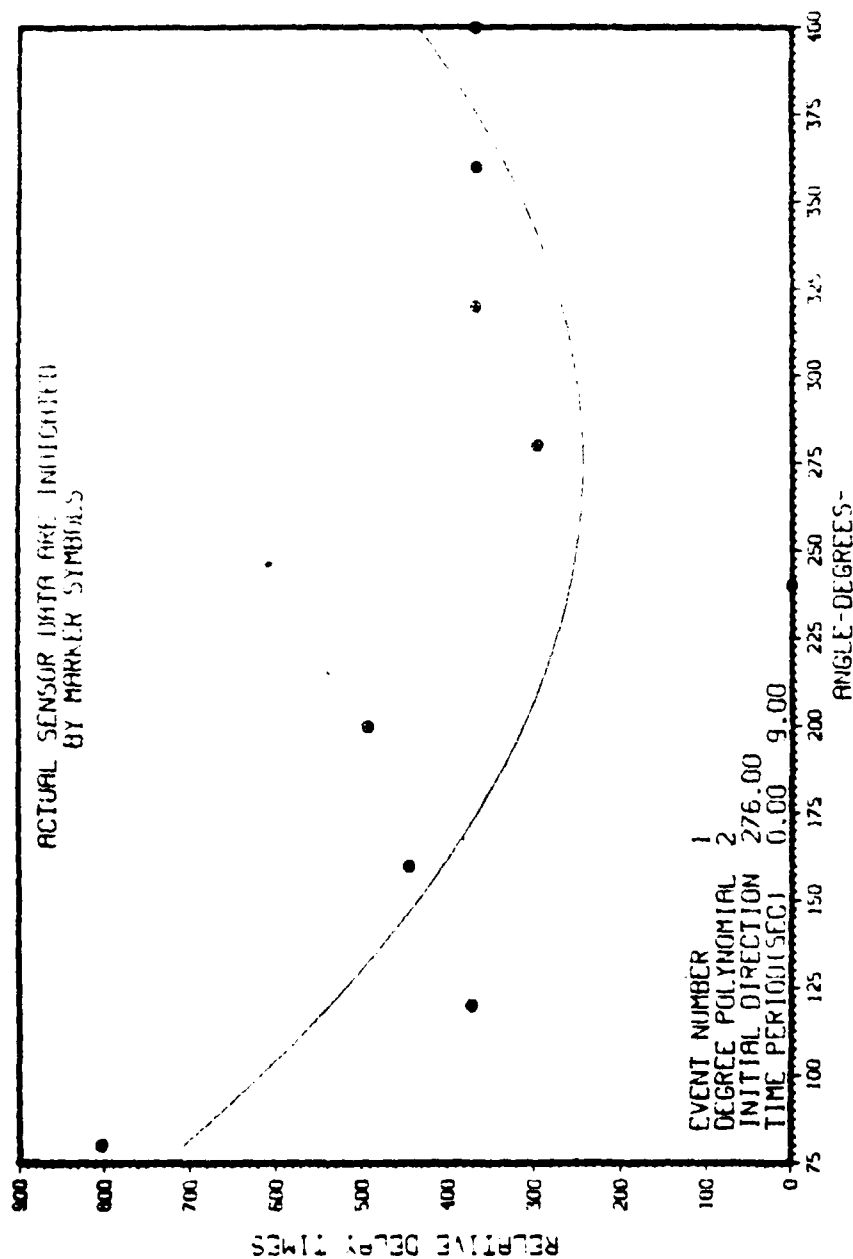


Figure 6.1 Sample Least Mean Squares Initial Direction for Event 001

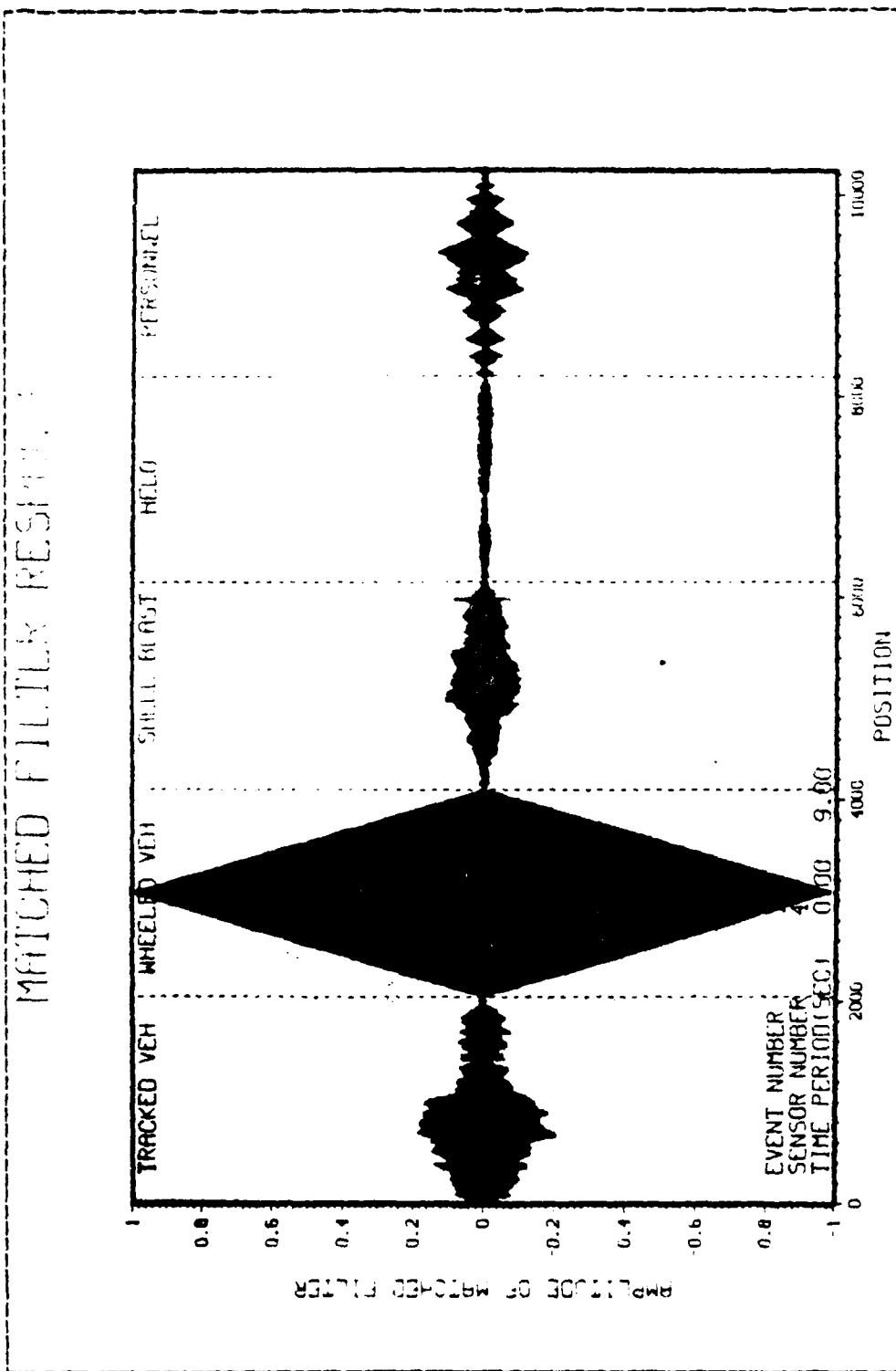


Figure 6.2 Sample Matched Filter Response for Event 001

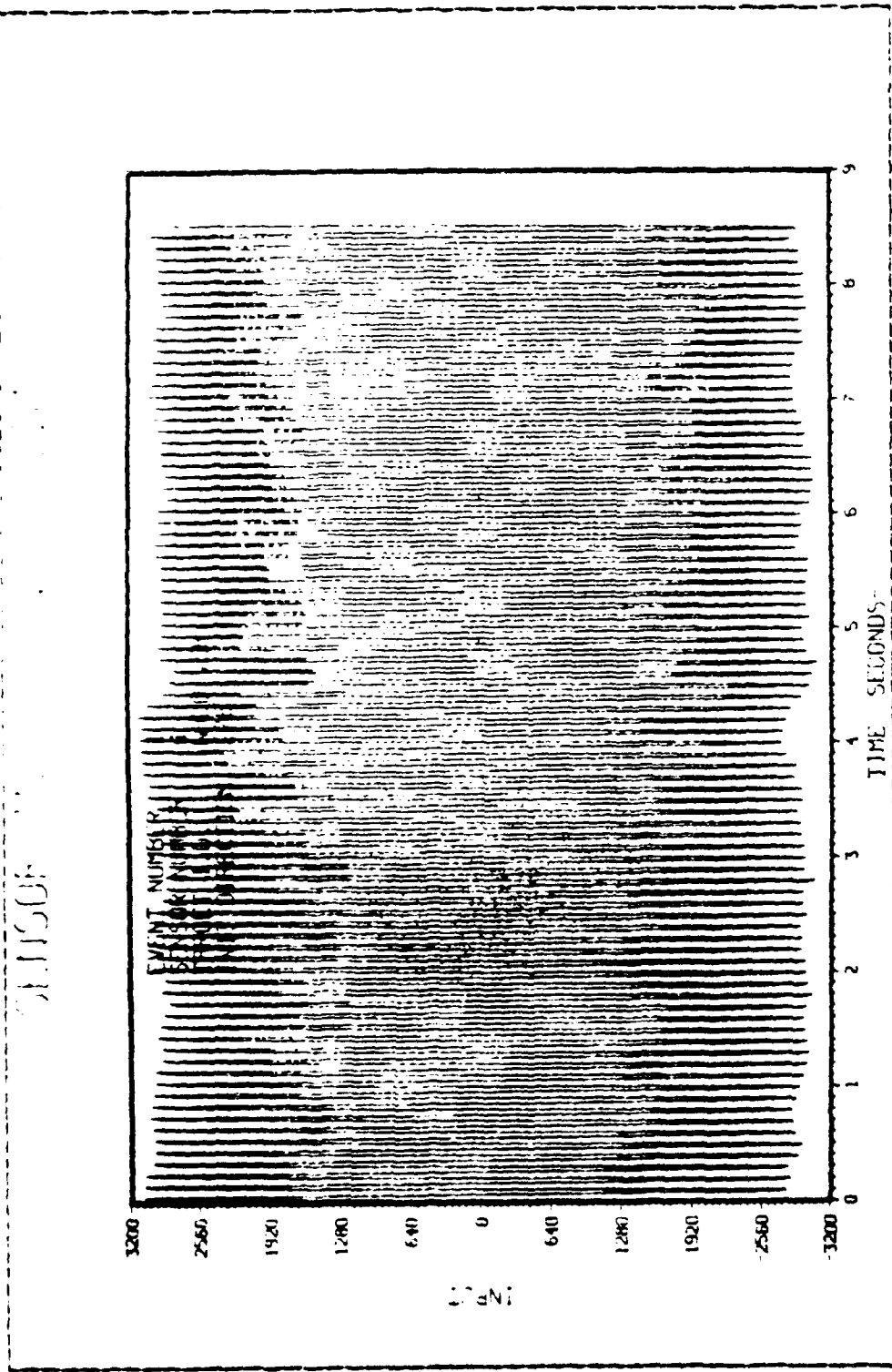


Figure 6.3 Sample Amplitude Response for Event 001

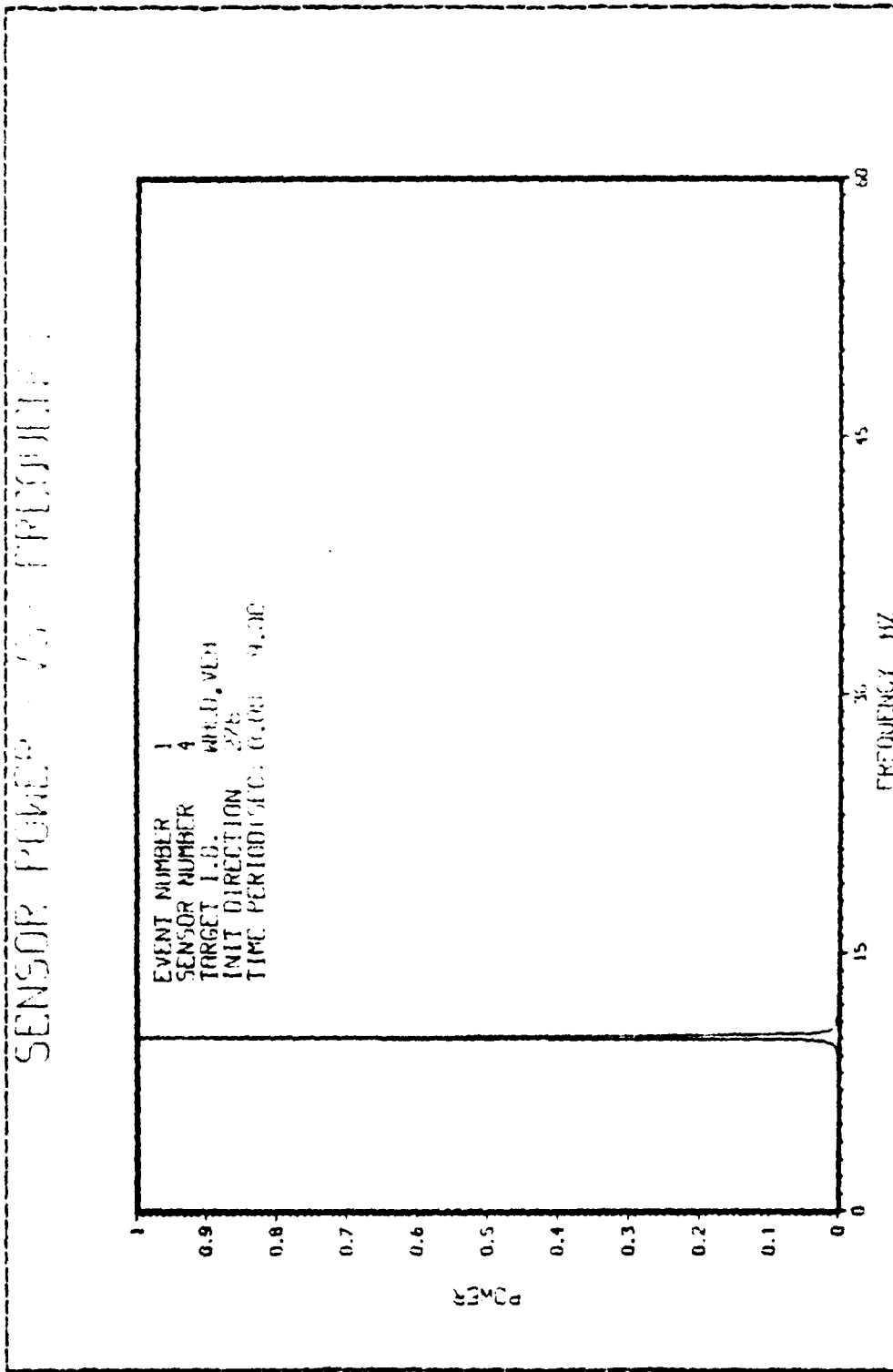


Figure 6.4 Sample Frequency Response for Event 001

LEAST MEAN SQUARES POLYNOMIAL

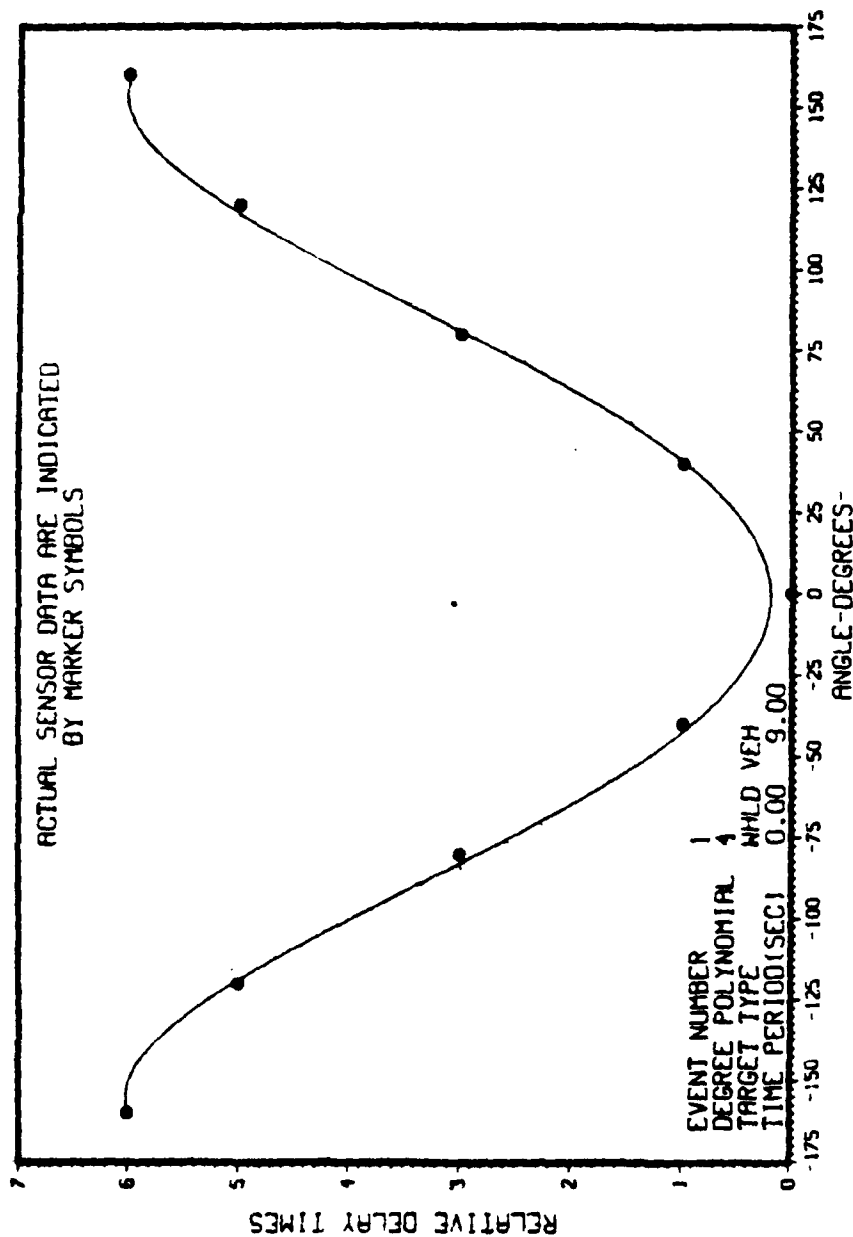


Figure 6.5 LMSP Matched Filter Direction for Event 001

MULTIPLE TARGET - MATCHED FILTER OUTPUT

EVENT NUMBER 1

TIME PERIOD(SEC) 0.00 9.00

WHEELED VEHICLE DIRECTION - 0.00

SIMULATED TRKD VEHICLE	TARGET FREQUENCY	0.00
AMPLITUDE	0.0000	
DIRECTION	0.0000	
SIMULATED WILD VEHICLE	TARGET FREQUENCY	10.00
AMPLITUDE	3000.0000	
DIRECTION	0.0000	
SIMULATED HELICOPTER	TARGET FREQUENCY	0.00
AMPLITUDE	0.0000	
DIRECTION	0.0000	
SIMULATED PERSONNEL	TARGET FREQUENCY	0.00
AMPLITUDE	0.0000	
DIRECTION	0.0000	

Figure 6.6 LMSP Multiple Target Direction Summary for Event 001

LEAST MEAN SQUARES POLYNOMIAL

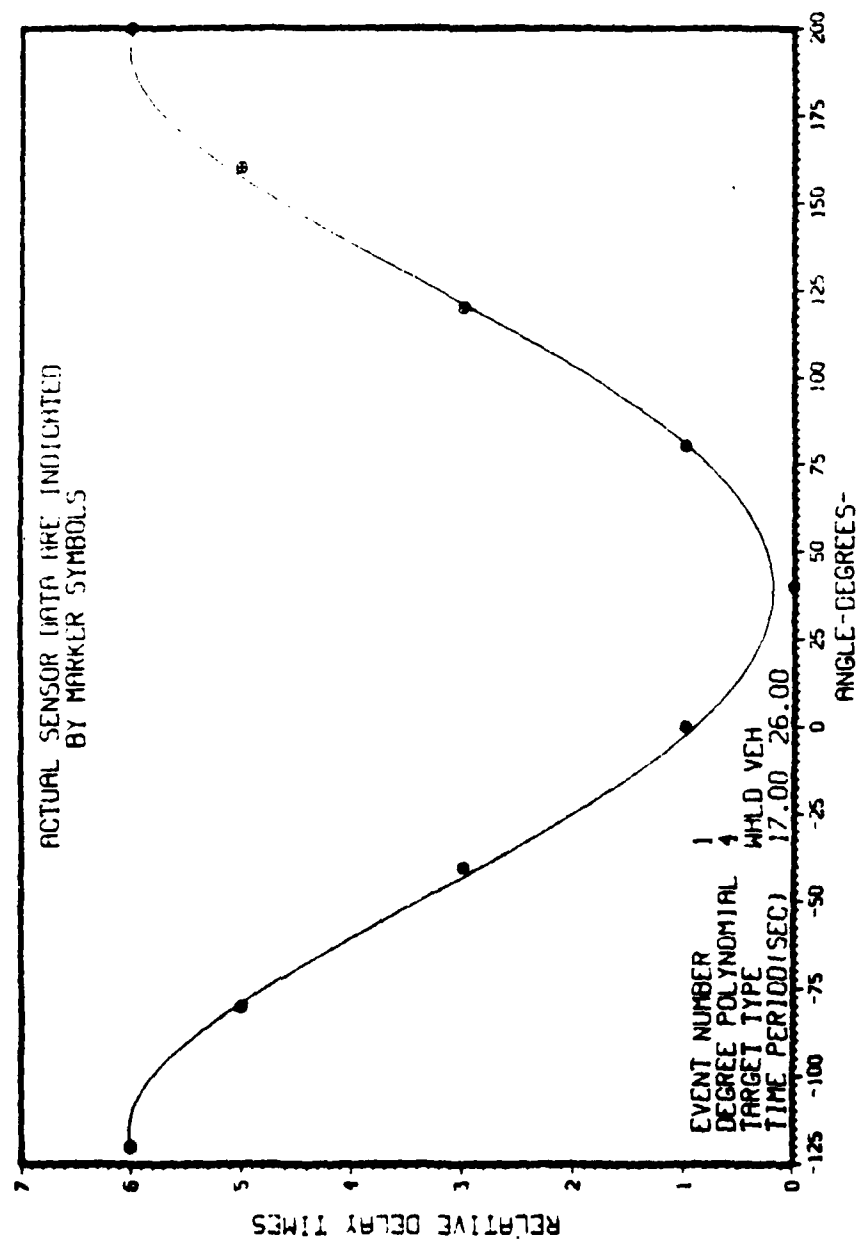


Figure 6.7 LMSP Matched Filter Direction for Event 001

MULTIPLE TARGET - MATCHED - LMSPP OUTPUT

```

EVENT NUMBER      1
TIME PERIOD(SEC)  17.00  26.00

WHEELED VEHICLE   DIRECTION - 40.00

SIMULATED TRKD VEHICLE TARGET FREQUENCY      0.00
AMPLITUDE      0.0000
DIRECTION      0.0000
SIMULATED WILD VEHICLE TARGET FREQUENCY      10.00
AMPLITUDE      3000.0000
DIRECTION      40.0000
SIMULATED HELICOPTER TARGET FREQUENCY      0.00
AMPLITUDE      0.0000
DIRECTION      40.0000
SIMULATED PERSONNEL TARGET FREQUENCY      0.00
AMPLITUDE      0.0000
DIRECTION      0.0000
    
```

Figure 6.8 LMSPP Multiple Target Direction Summary for Event 001

LEAST MEAN SQUARES POLYNOMIAL

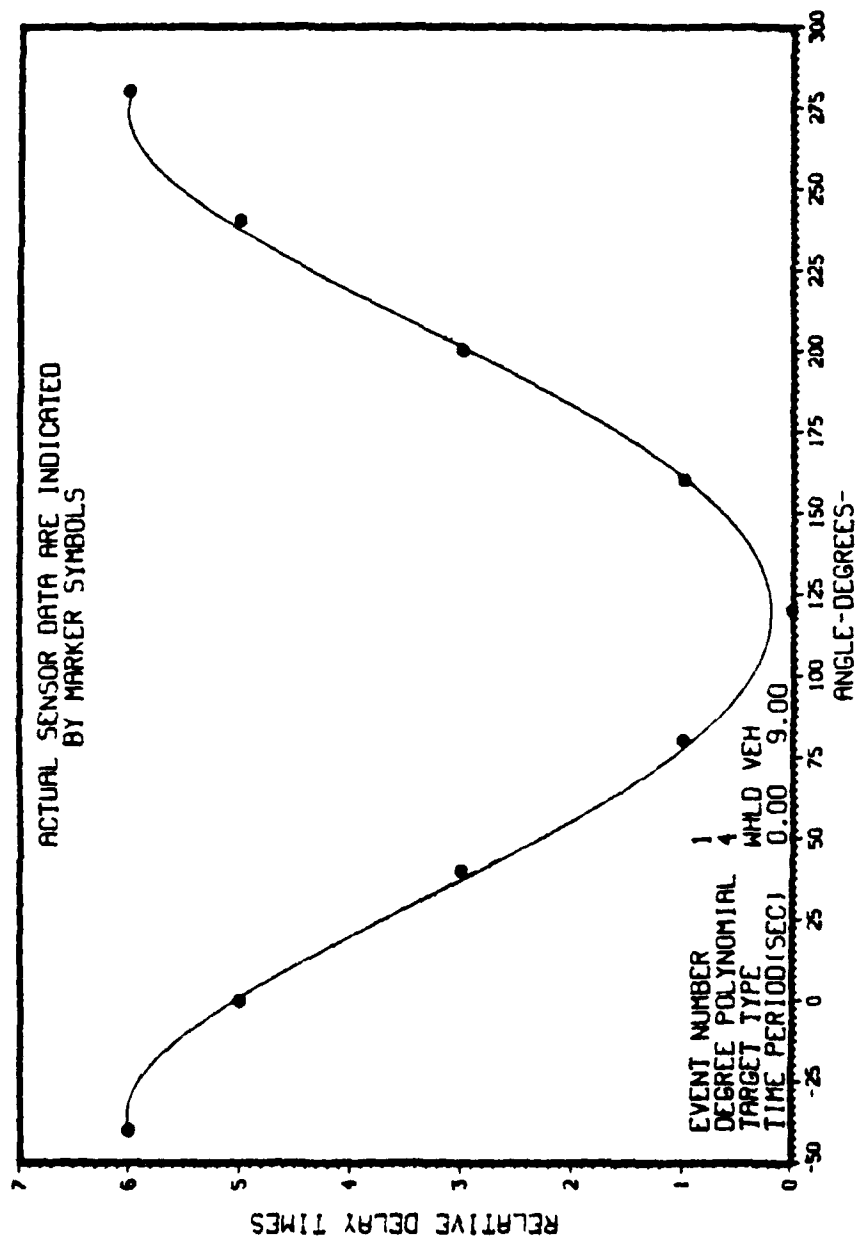


Figure 6.9 LMSP Matched Filter Direction for Event 001

MULTIPLE TARGET - MATCHED FILTER OUTPUT

```

EVENT NUMBER      1
TIME PERIOD(SEC)  0.00  9.00

WHEELED VEHICLE   DIRECTION - 120.00

SIMULATED TRKD VEHICLE TARGET FREQUENCY  10.00
AMPLITUDE 3000.0000
DIRECTION 120.0000
SIMULATED WMLD VEHICLE TARGET FREQUENCY  0.00
AMPLITUDE 0.0000
DIRECTION 0.0000
SIMULATED HELICOPTER TARGET FREQUENCY  0.00
AMPLITUDE 0.0000
DIRECTION 0.0000
SIMULATED PERSONNEL TARGET FREQUENCY  0.00
AMPLITUDE 0.0000
DIRECTION 0.0000
    
```

Figure 6.10 LESP Multiple Target Direction Summary for Event 001

LEAST MEAN SQUARES POLYNOMIAL

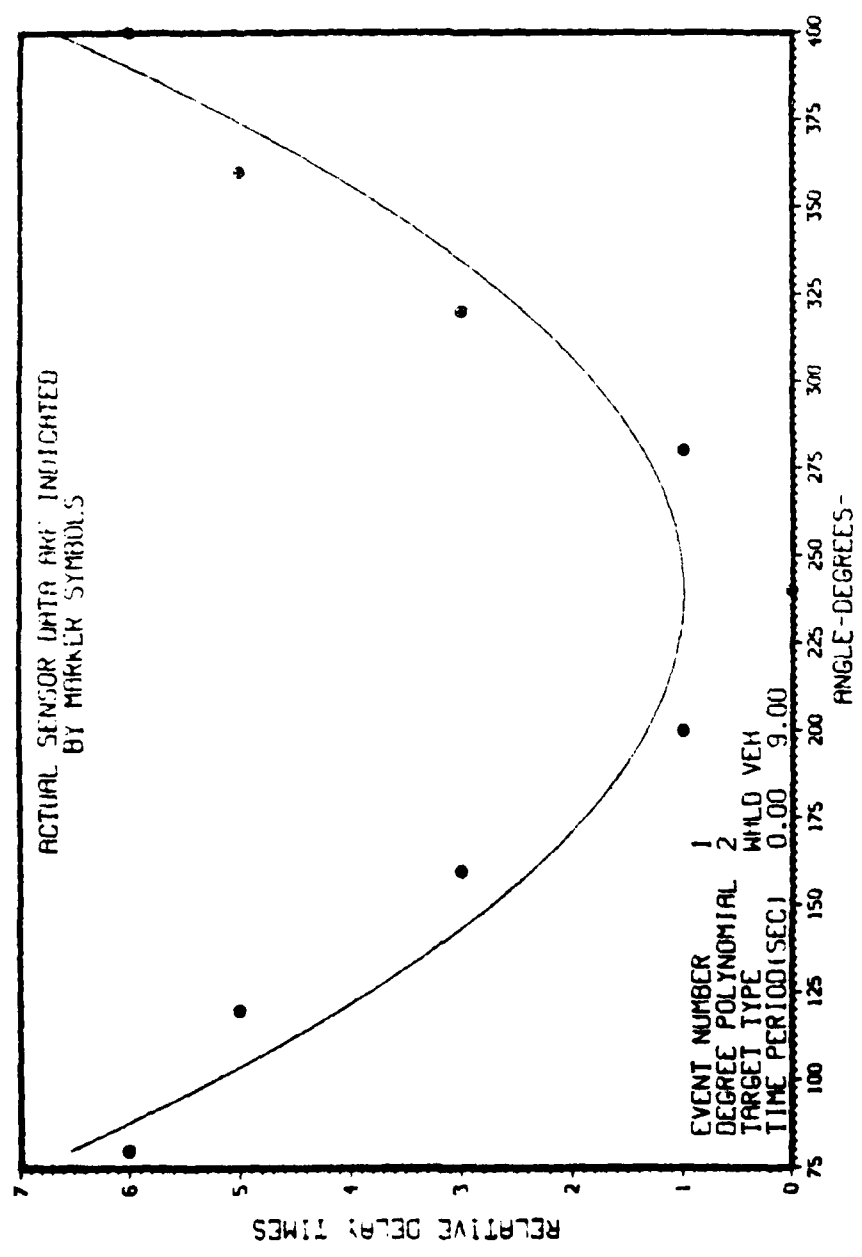


Figure 6.11 LMSP Matched Filter Direction for Event 001

MULTIPLE TARGET - MATCHED FILTER OUTPUT

EVENT NUMBER	1
TIME PERIOD(SEC)	0.00 9.00
WHEELED VEHICLE	DIRECTION - 240.00
SIMULATED TRKD VEHICLE	TARGET FREQUENCY 0.00
AMPLITUDE	0.0000
DIRECTION	0.0000
SIMULATED WMD VEHICLE	TARGET FREQUENCY 10.00
AMPLITUDE	3000.0000
DIRECTION	240.0000
SIMULATED HELICOPTER	TARGET FREQUENCY 0.00
AMPLITUDE	6.0000
DIRECTION	6.0000
SIMULATED PERSONNEL	TARGET FREQUENCY 0.00
AMPLITUDE	0.0000
DIRECTION	0.0000

Figure 6.12 LMS Multiple Target Direction Summary for Event 001

LEAST MEAN SQUARES POLYNOMIAL

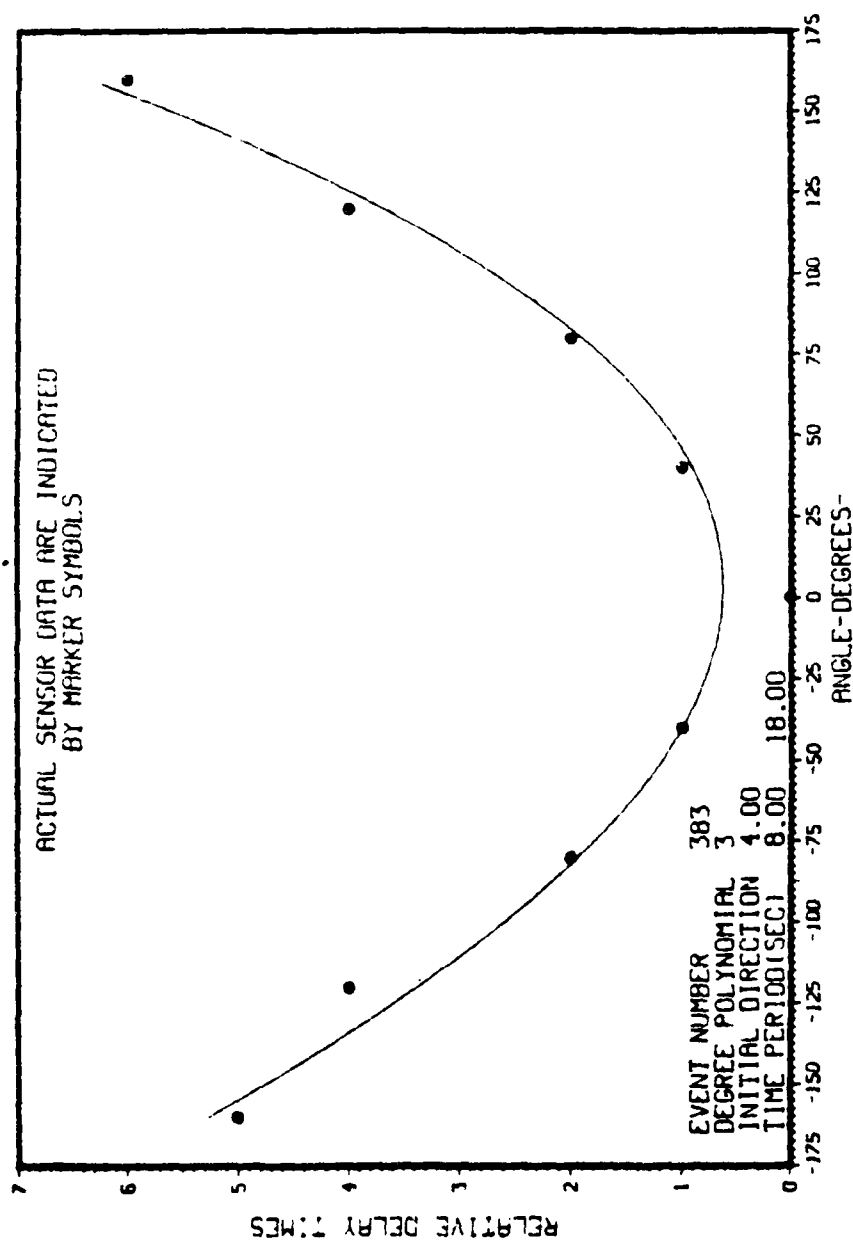


Figure 6.13 LMSP Initial Direction for Event 383

MATCHED FILTER RESPONSE

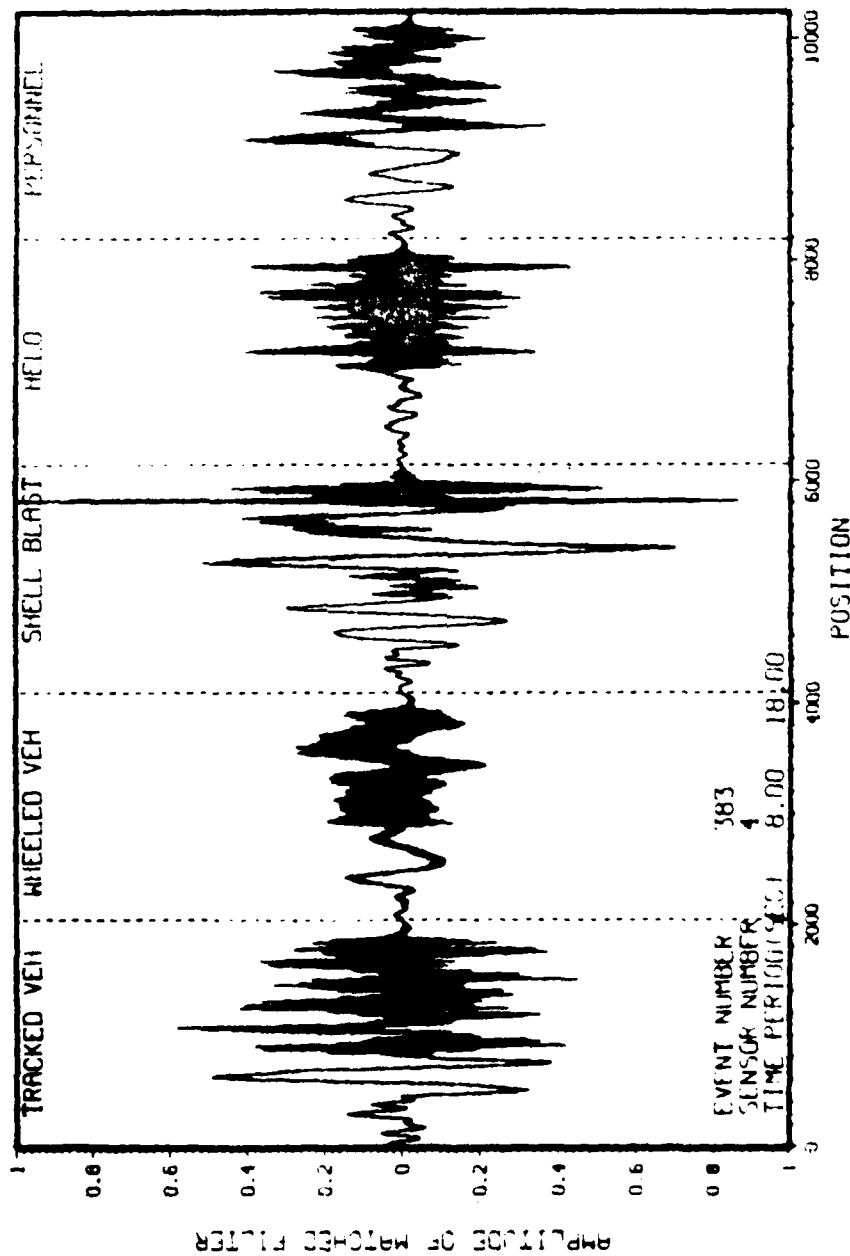


Figure 6.14 Matched Filter Response for Event 383

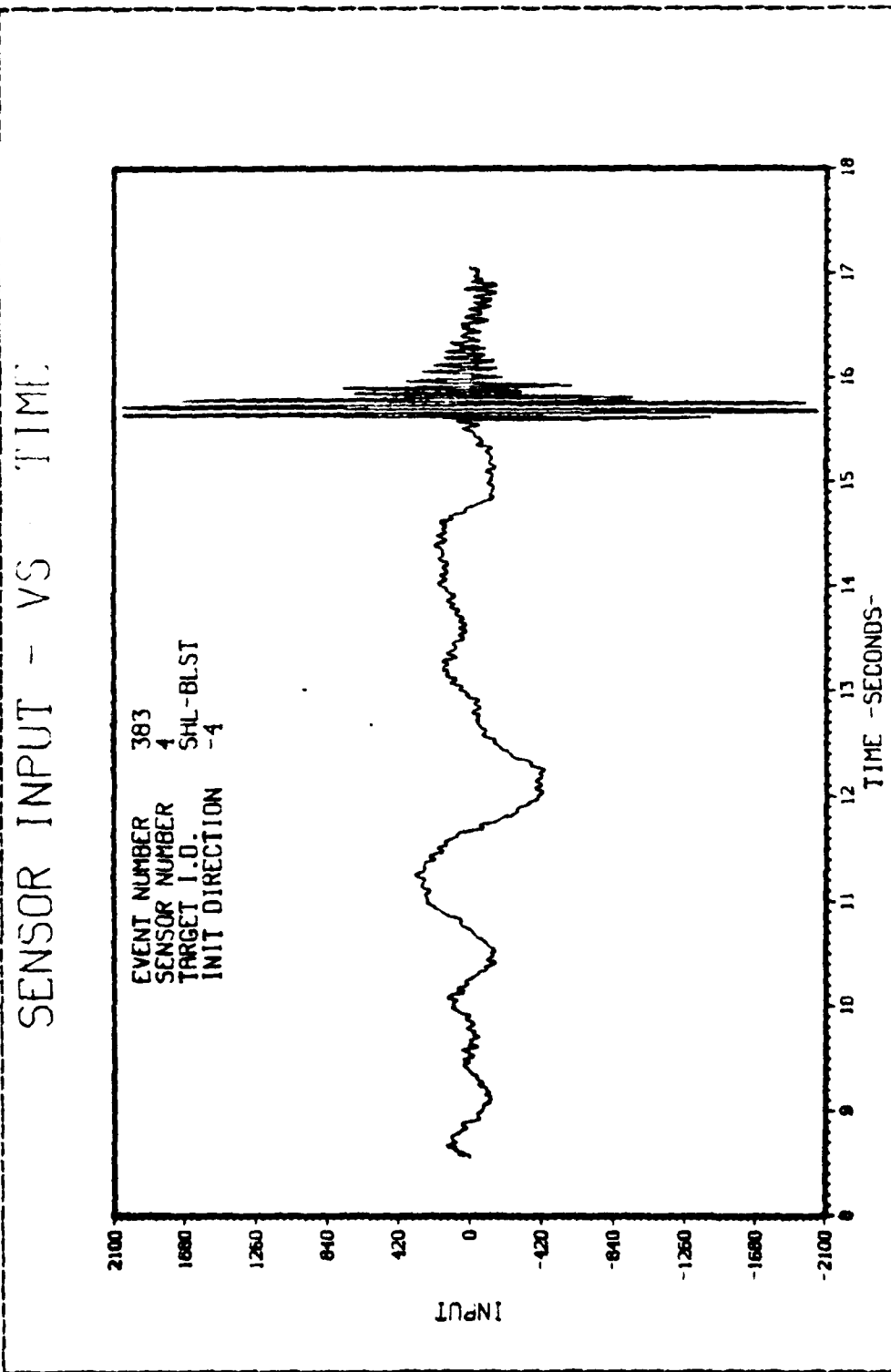


Figure 6.15 Amplitude Response for Event 383

SENSOR POWER -VS- FREQUENCY

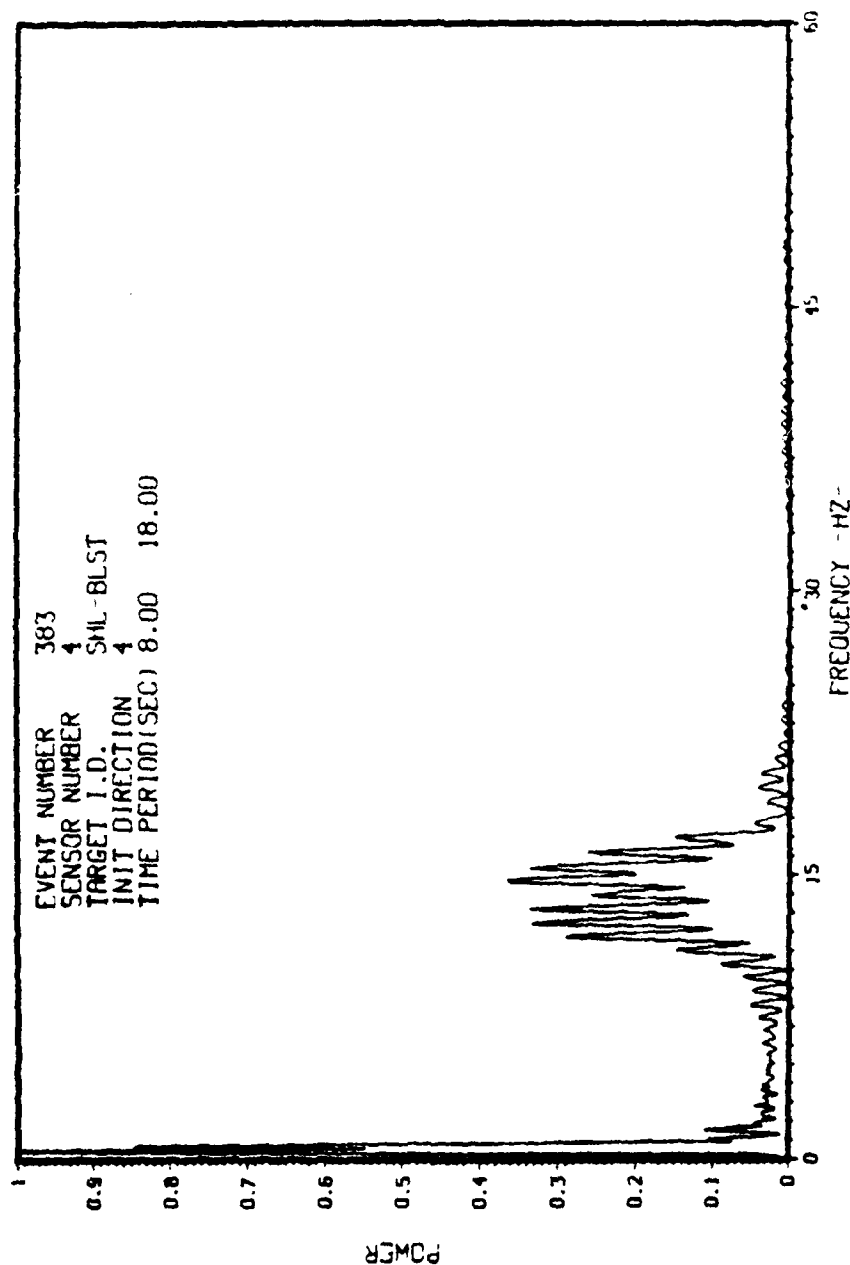


Figure 6.16 Frequency Response for Event 383

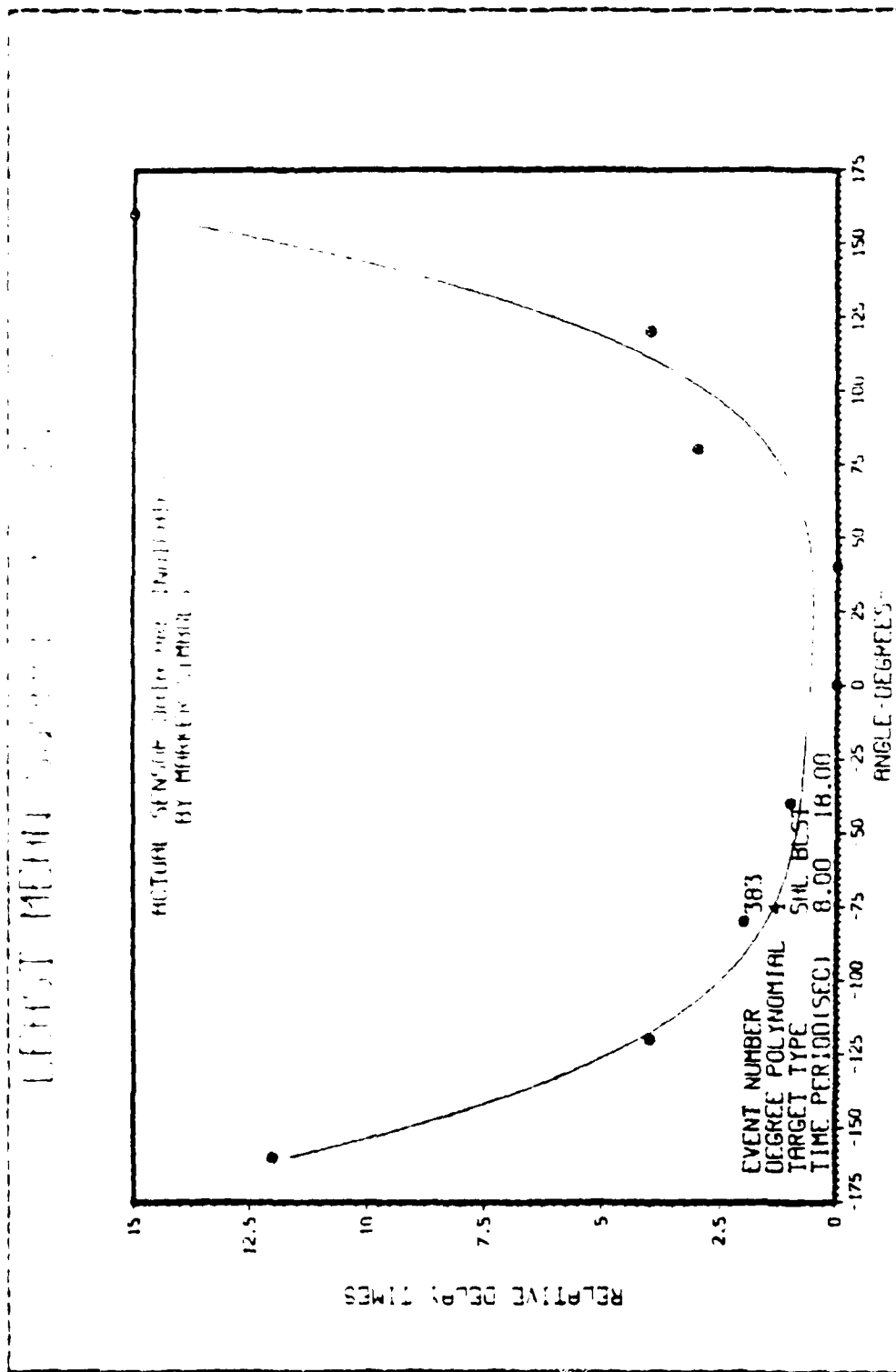


Figure 6.17 Fourth Degree LMSP Matched Filter Direction for Event 383

LEAST MEAN SQUARES POLYNOMIAL

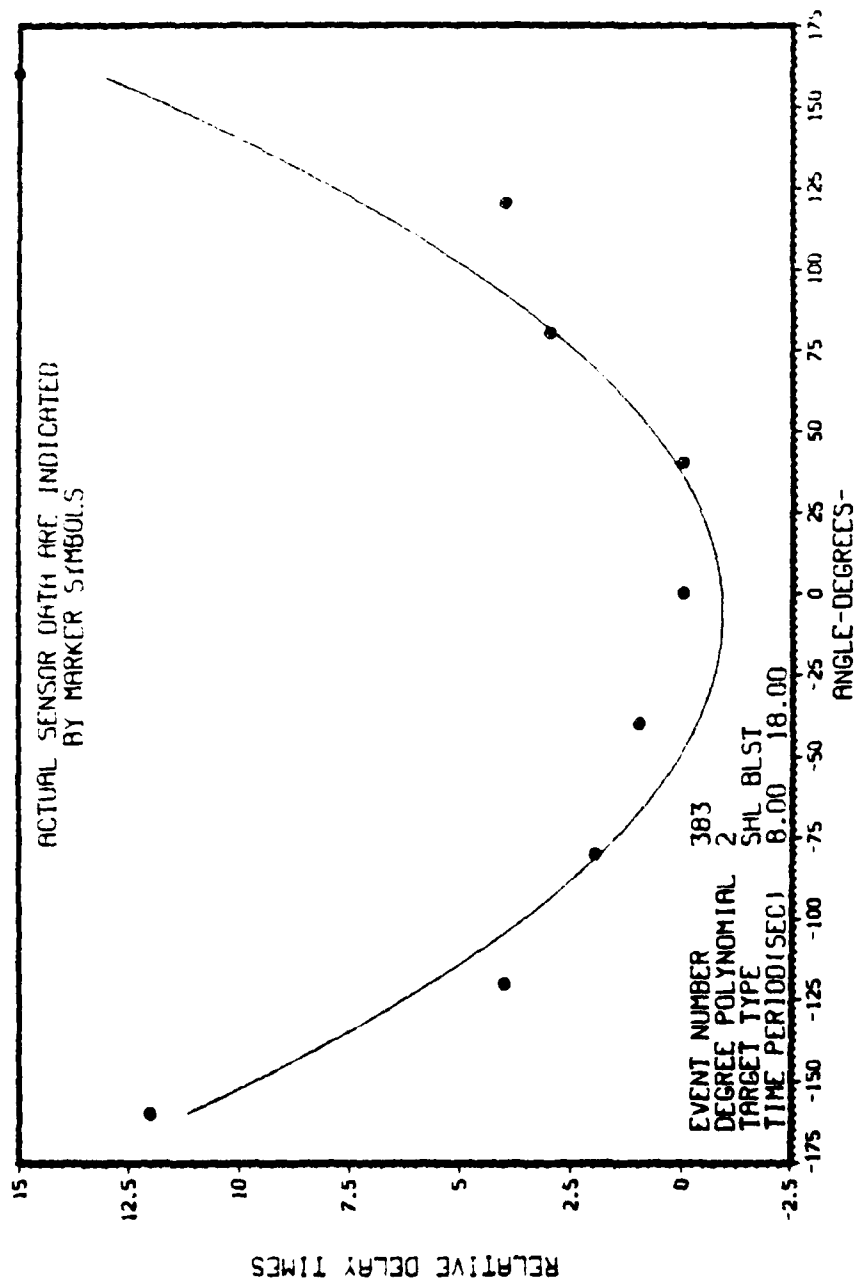


Figure 6.18 Second Degree LMSP Matched Filter Direction for Event 383

MULTIPLE TARGET - MATCHED FILTER OUTPUT

EVENT NUMBER 383

TIME PERIOD(SEC) 8.00 18.00

SHELL BLAST DIRECTION - 136.00

SIMULATED TRKO VEHICLE	TARGET FREQUENCY	0.00
AMPLITUDE	0.0000	
DIRECTION	0.0000	
SIMULATED WILD VEHICLE	TARGET FREQUENCY	0.00
AMPLITUDE	0.0000	
DIRECTION	0.0000	
SIMULATED HELICOPTER	TARGET FREQUENCY	0.00
AMPLITUDE	0.0000	
DIRECTION	0.0000	
SIMULATED PERSONNEL	TARGET FREQUENCY	0.00
AMPLITUDE	0.0000	
DIRECTION	0.0000	

Figure 6.19 LMSP Multiple Target Direction Summary for Event 383

LEAST MEAN SQUARES POLYNOMIAL

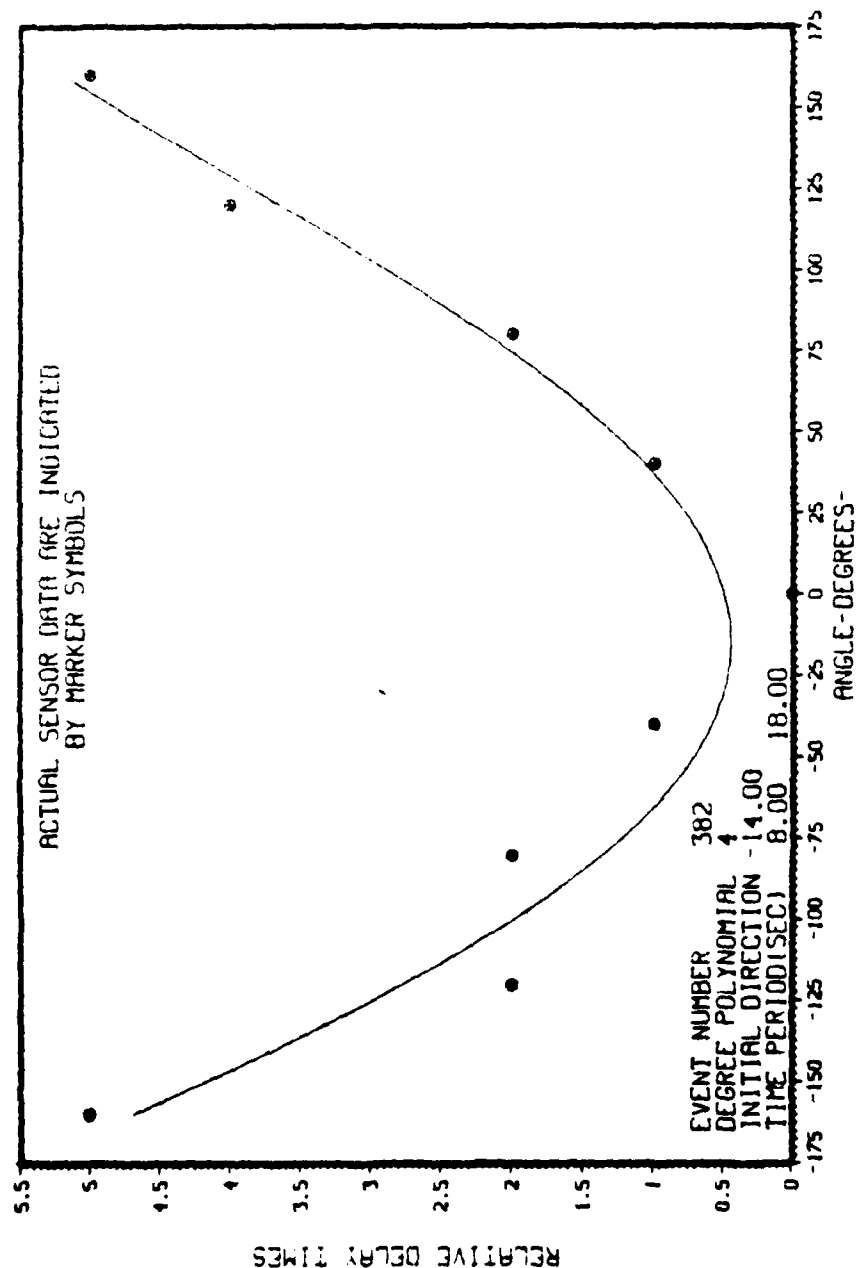


Figure 6.20 LMSP Initial Direction for Event 382

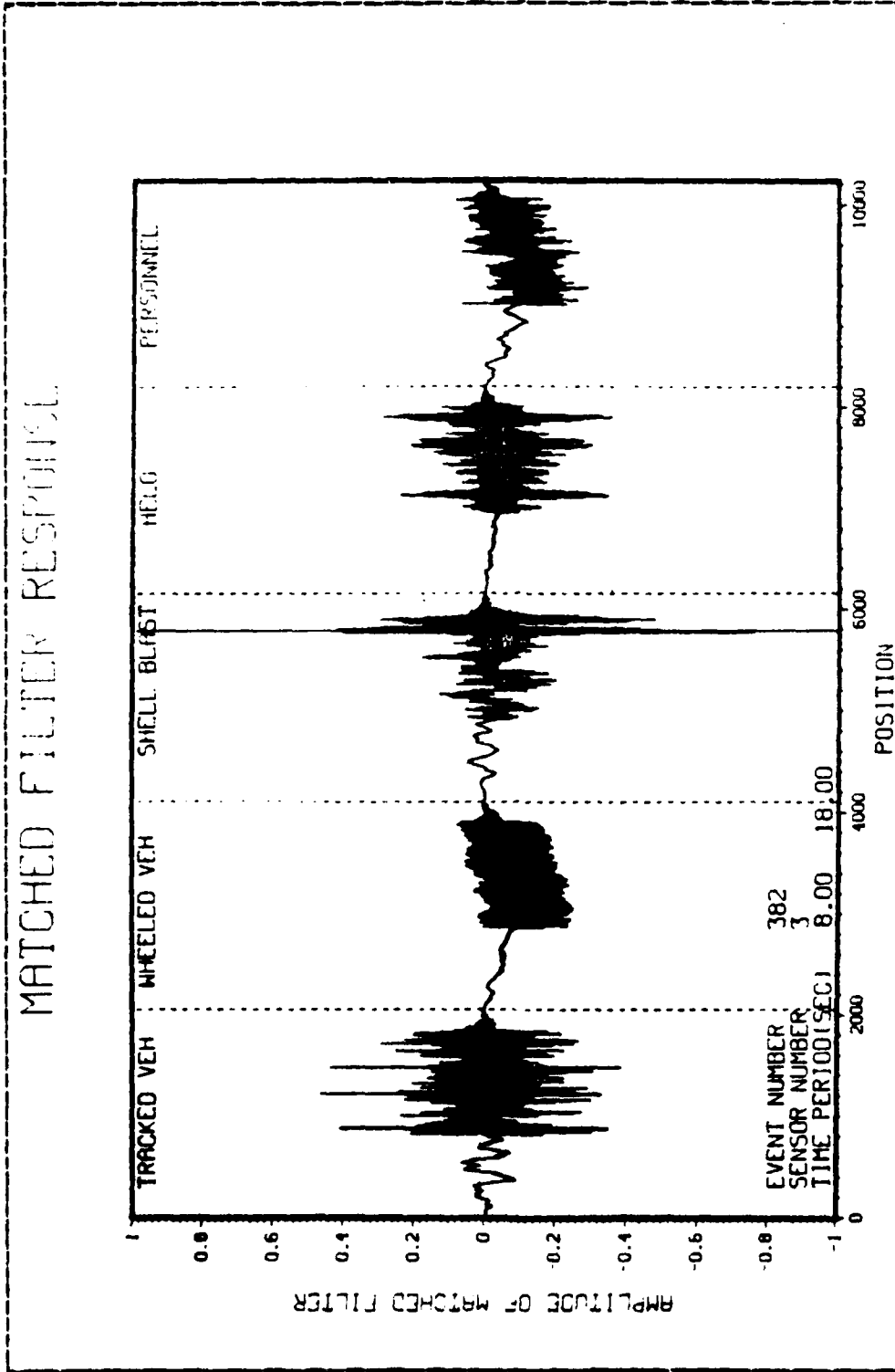


Figure 6.21 Matched Filter Response for Event 382

SENSOR INPUT - WAVEFORM

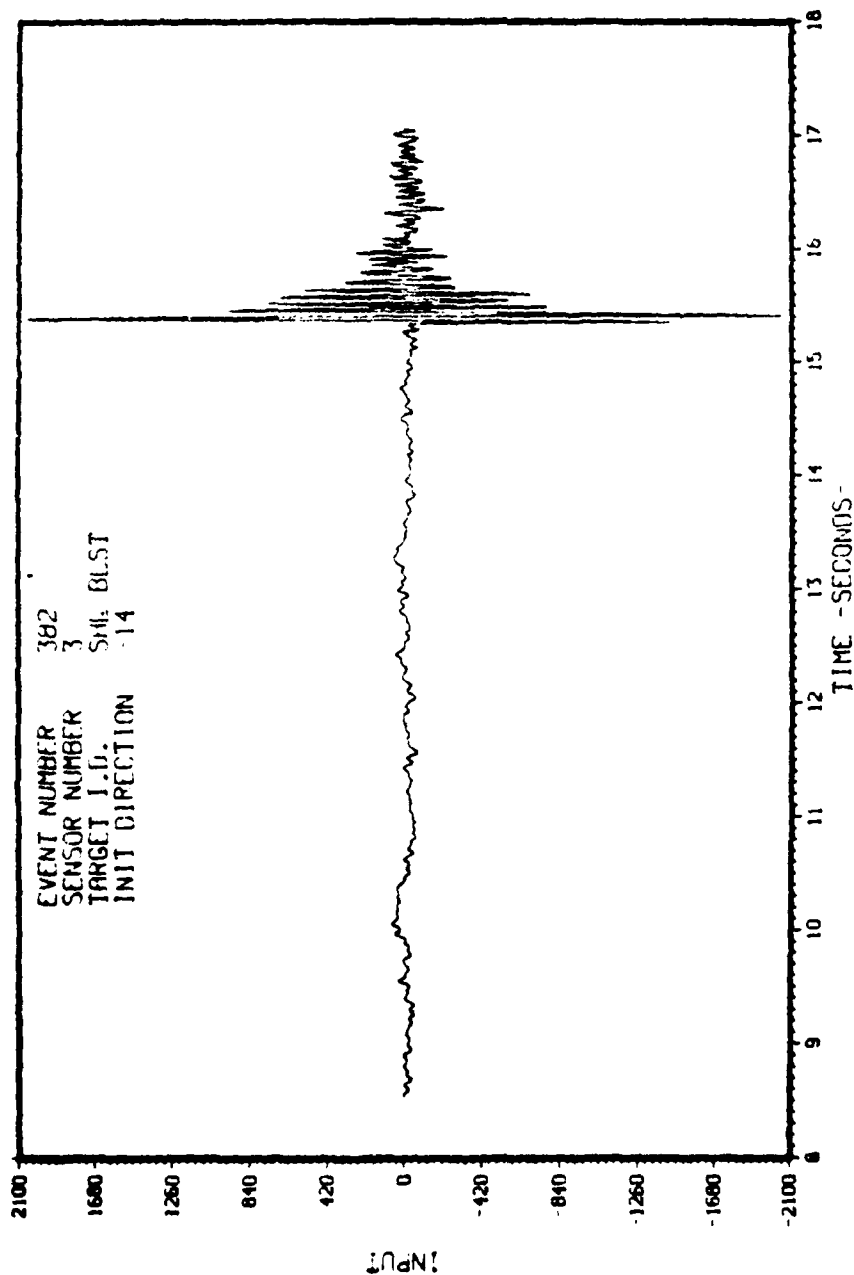


Figure 6.22 Amplitude Response for Event 382

SENSOR POWER - VS- FREQUENCY

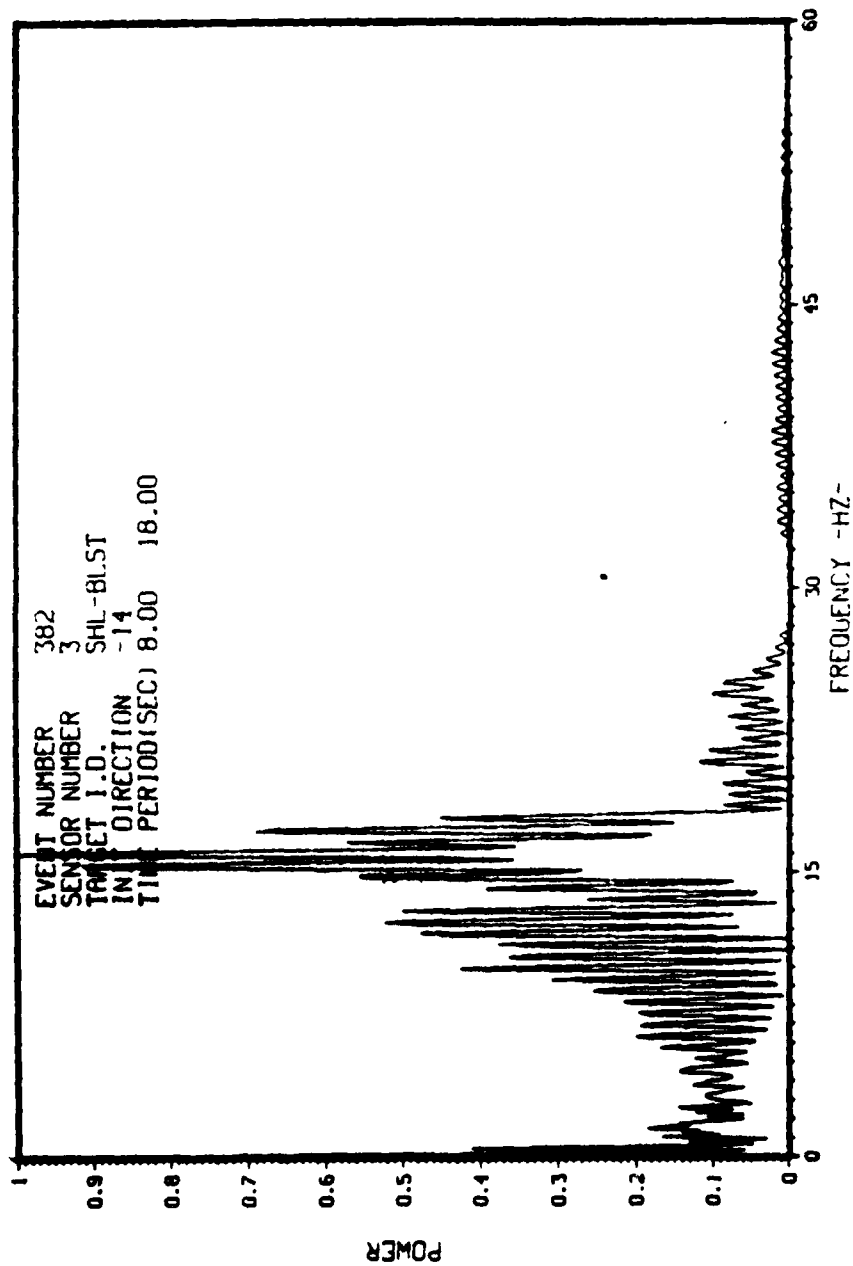


Figure 6.23 Frequency Response for Event 382

LEAST MEAN SQUARES CURVE FIT

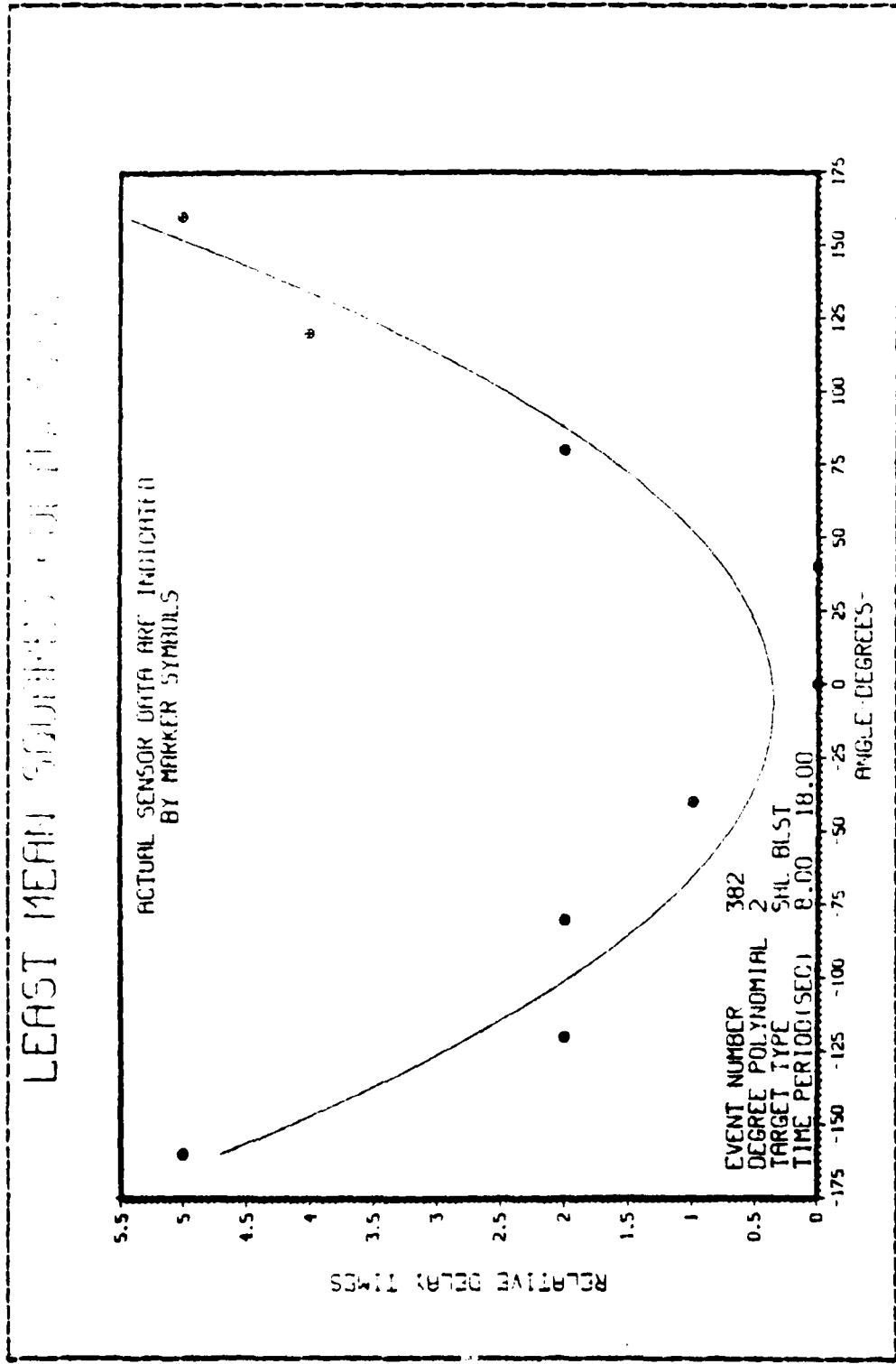


Figure 6.24 LNMP Matched Filter Direction for Event 382

MULTIPLE TARGET - MATCHED FILTER OUTPUT

EVENT NUMBER 382

TIME PERIOD(SEC) 8.00 18.00

SHELL BURST DIRECTION - 6.00

SIMULATED TRK VEHICLE TARGET FREQUENCY	0.00
AMPLITUDE 0.0000	
DIRECTION 0.0000	
SIMULATED WLD VEHICLE TARGET FREQUENCY	0.00
AMPLITUDE 0.0000	
DIRECTION 0.0000	
SIMULATED HELICOPTER TARGET FREQUENCY	0.00
AMPLITUDE 0.0000	
DIRECTION 0.0000	
SIMULATED PERSONNEL TARGET FREQUENCY	0.00
AMPLITUDE 0.0000	
DIRECTION 0.0000	

Figure 6.25 LMSP Multiple Target Direction Summary for Event 382

LEAST MEAN SQUARES POLYNOMIAL

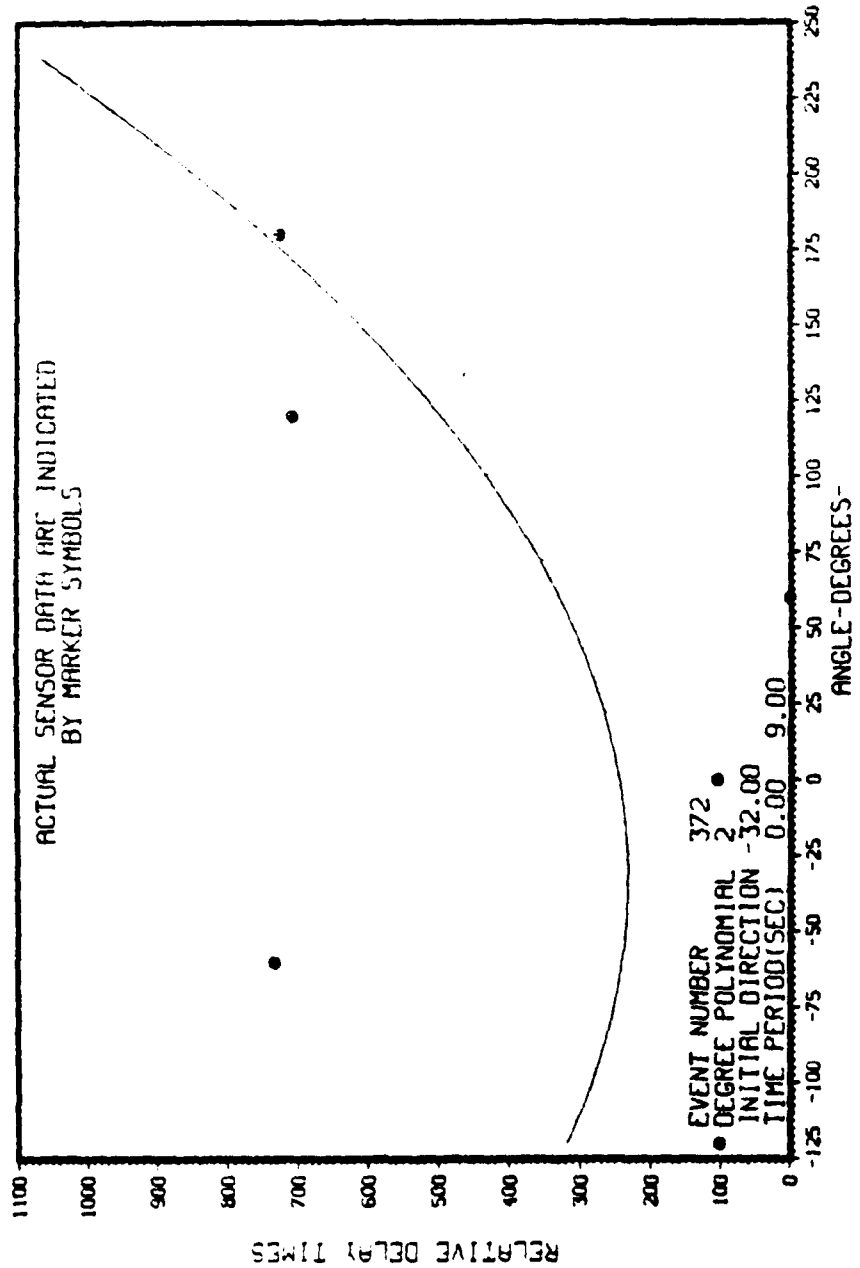


Figure 6.26 LMSP Initial Direction for Event 372

MATCHED FILTER RESPONSE

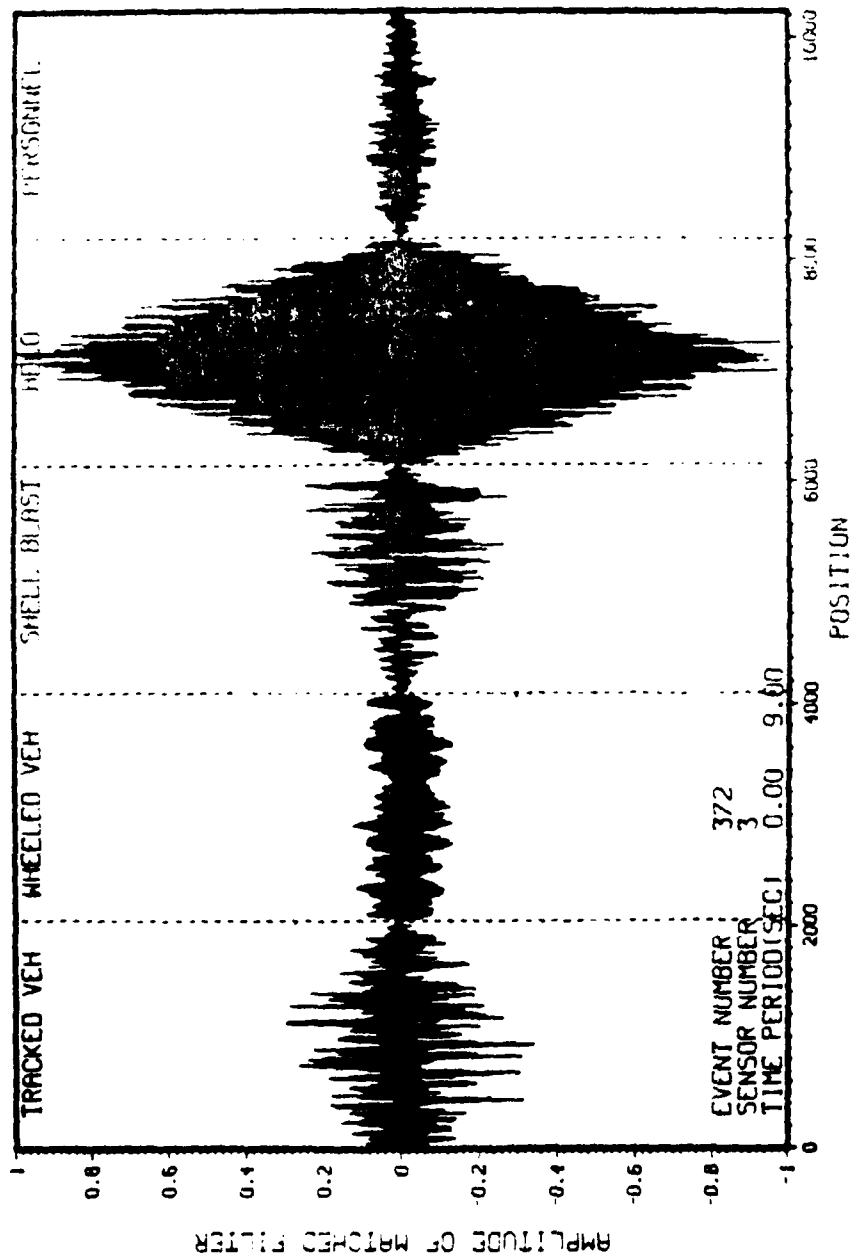


Figure 6.27 Matched Filter Response for Event 372

SENSOR UNIT

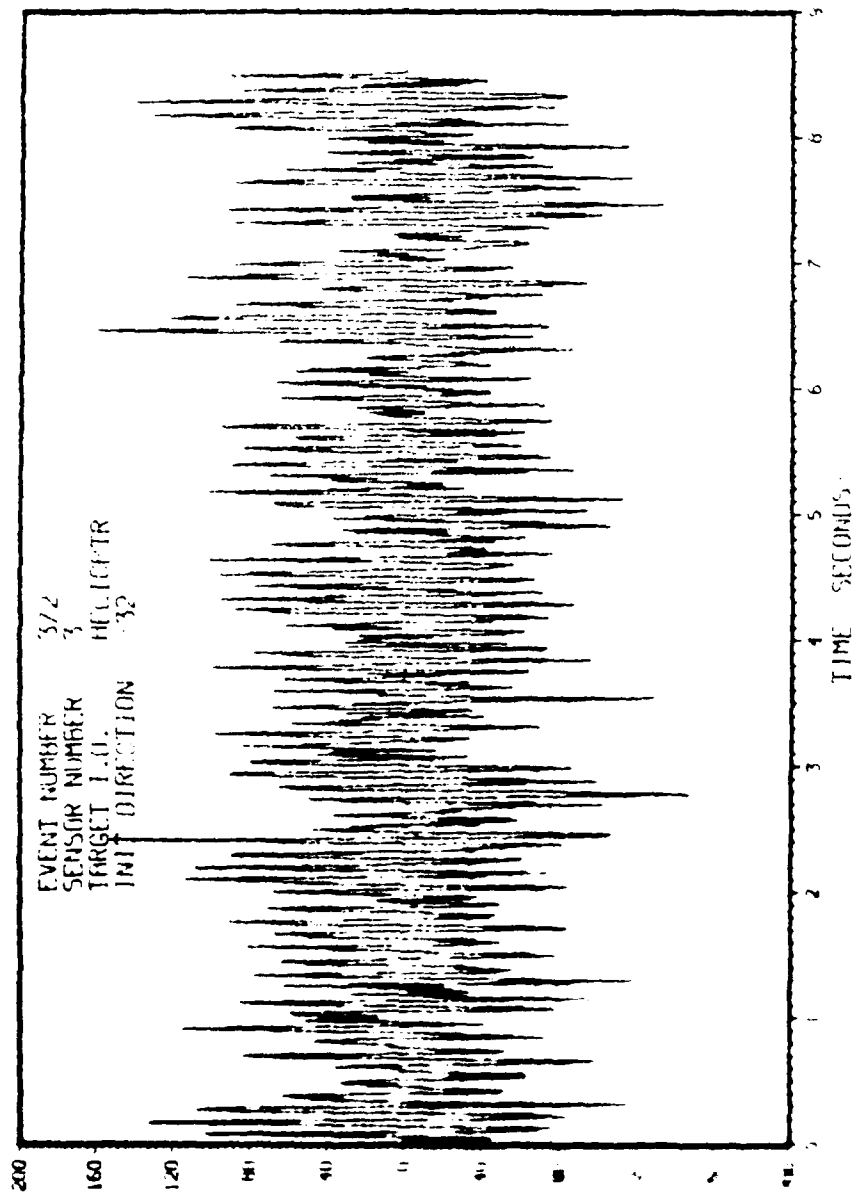


Figure 6.28 Amplitude Response for Event 372

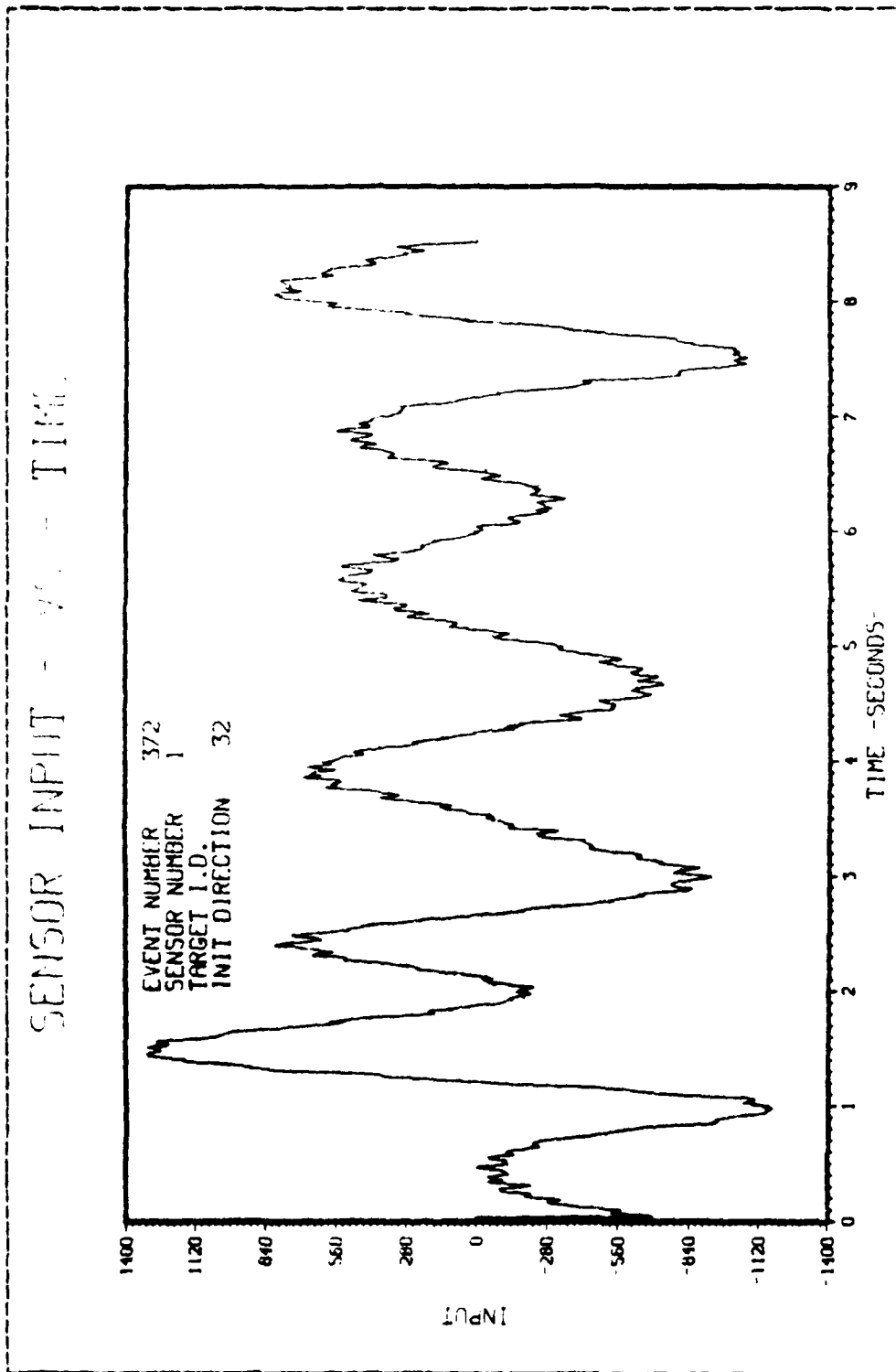


Figure 6.29 Amplitude Response of Malfunctioning Sensor for Event 372

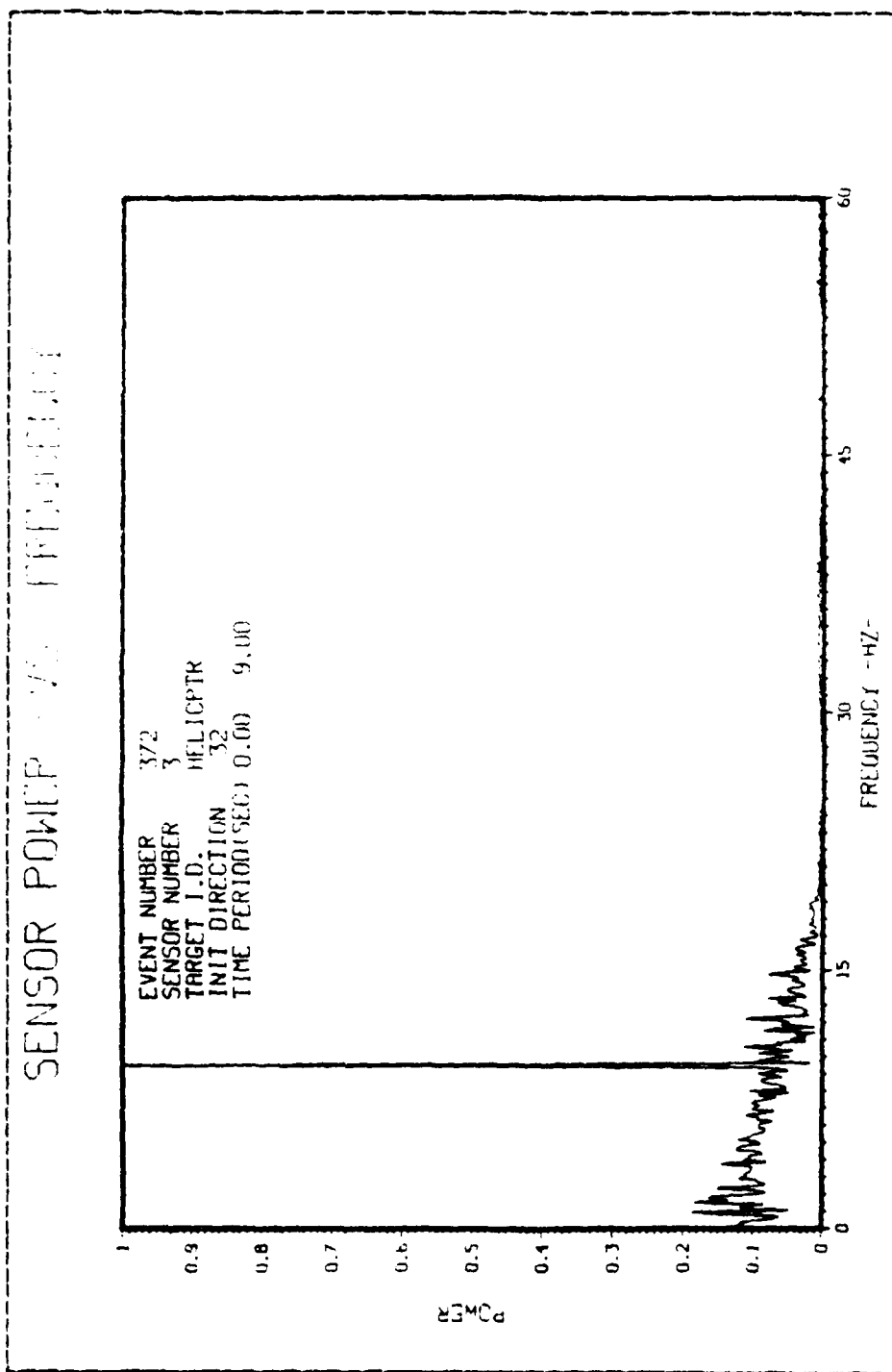


Figure 6.30 Frequency Response for Event 372

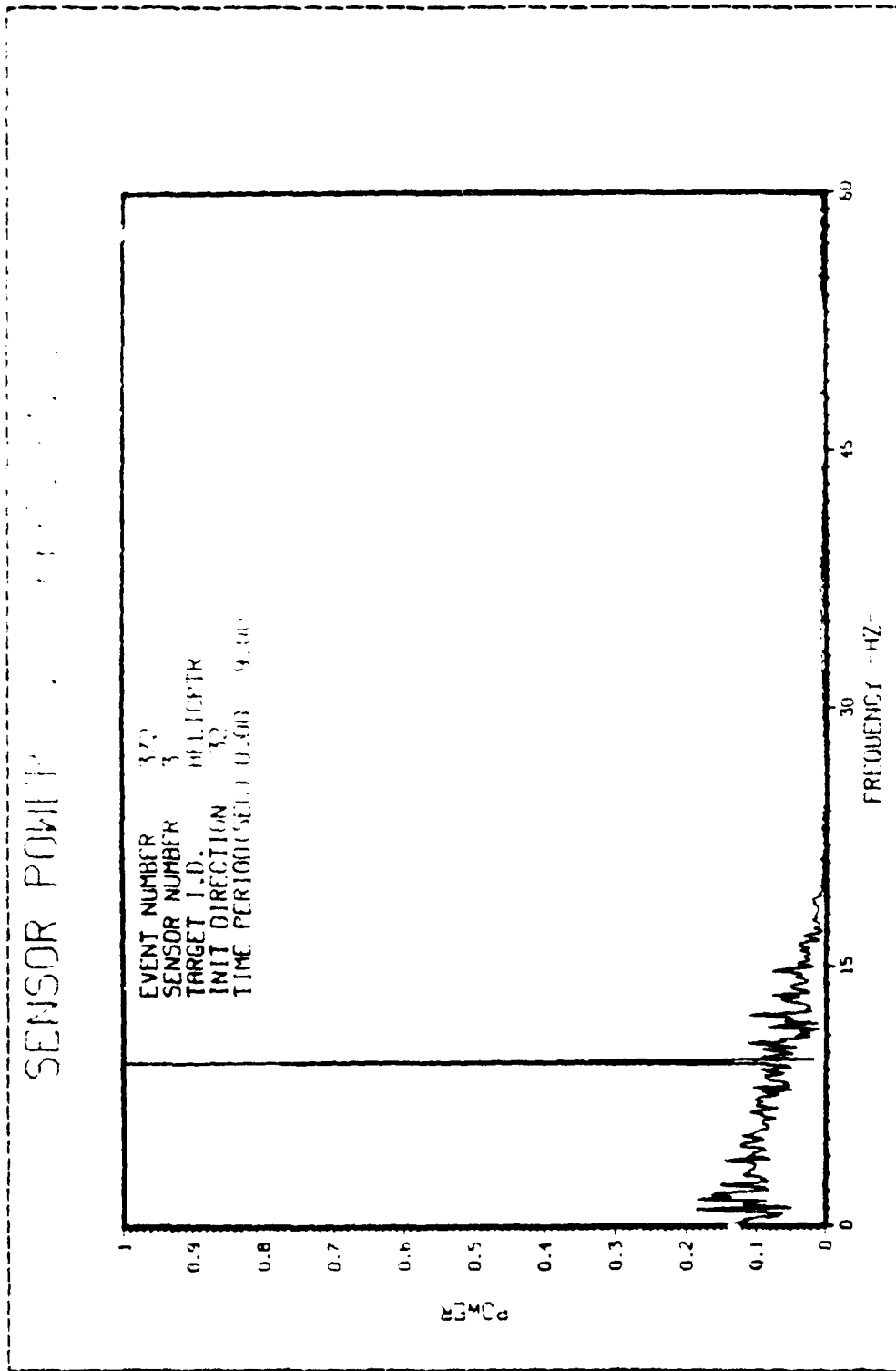


Figure 6.30 Frequency Response for Event 372

LEAST MEAN SQUARES POLYNOMIAL

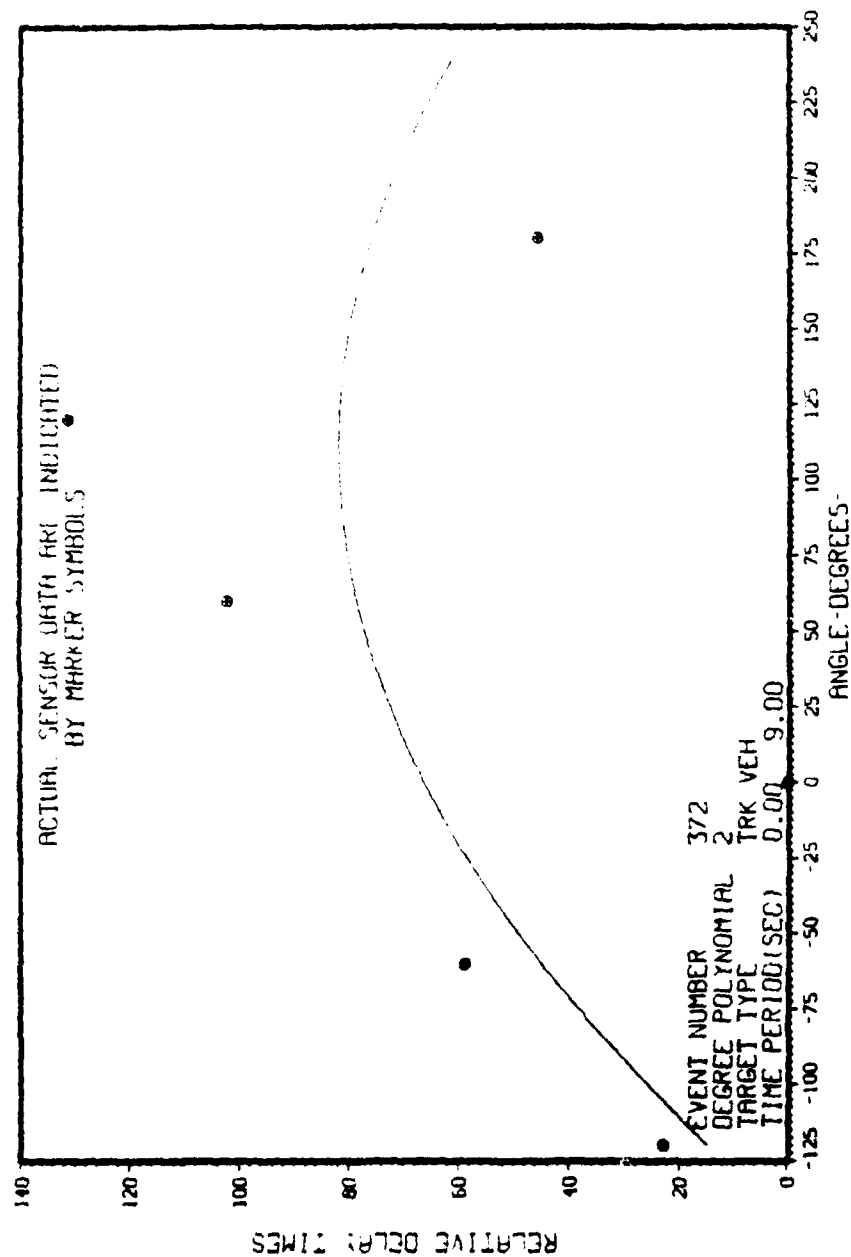


Figure 6.31 LMSP Matched Filter Direction for Event 372

MULTIPLE TARGET - MATCHED FILTER OUTPUT

EVENT NUMBER	372
TIME PERIOD(SEC)	0.00 9.00
TRACKED VEHICLE	DIRECTION - 0.00
HELICOPTER	DIRECTION - 0.00
SIMULATED TRKD VEHICLE	TARGET FREQUENCY 0.00
AMPLITUDE	0.0000
DIRECTION	0.0000
SIMULATED WILD VEHICLE	TARGET FREQUENCY 0.00
AMPLITUDE	0.0000
DIRECTION	0.0000
SIMULATED HELICOPTER	TARGET FREQUENCY 0.00
AMPLITUDE	0.0000
DIRECTION	0.0000
SIMULATED PERSONNEL	TARGET FREQUENCY 0.00
AMPLITUDE	0.0000
DIRECTION	0.0000

Figure 6.32 LMSP Multiple Target Direction Summary for Event 372

LEAST MEAN SQUARES POLYNOMIAL

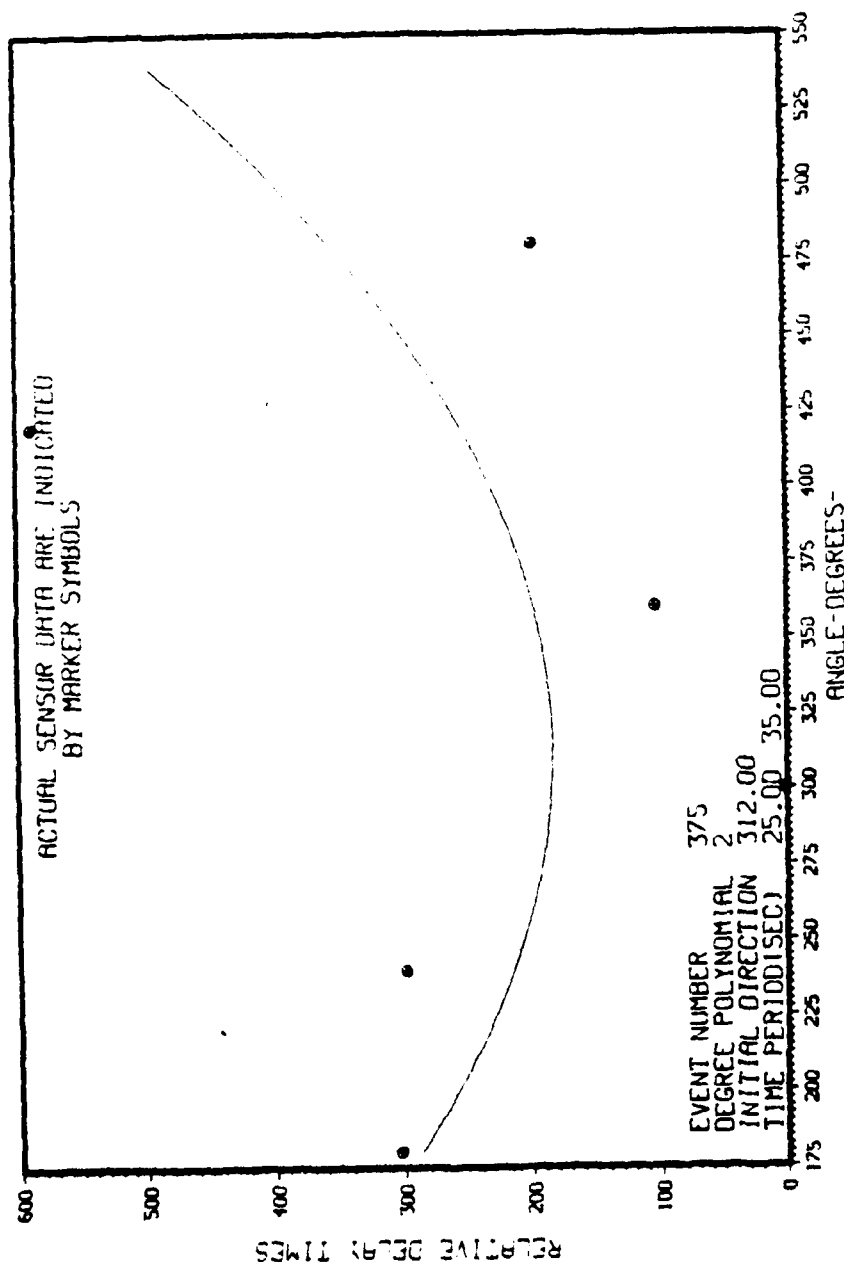


Figure 6.33 LMSP Initial Direction for Event 375

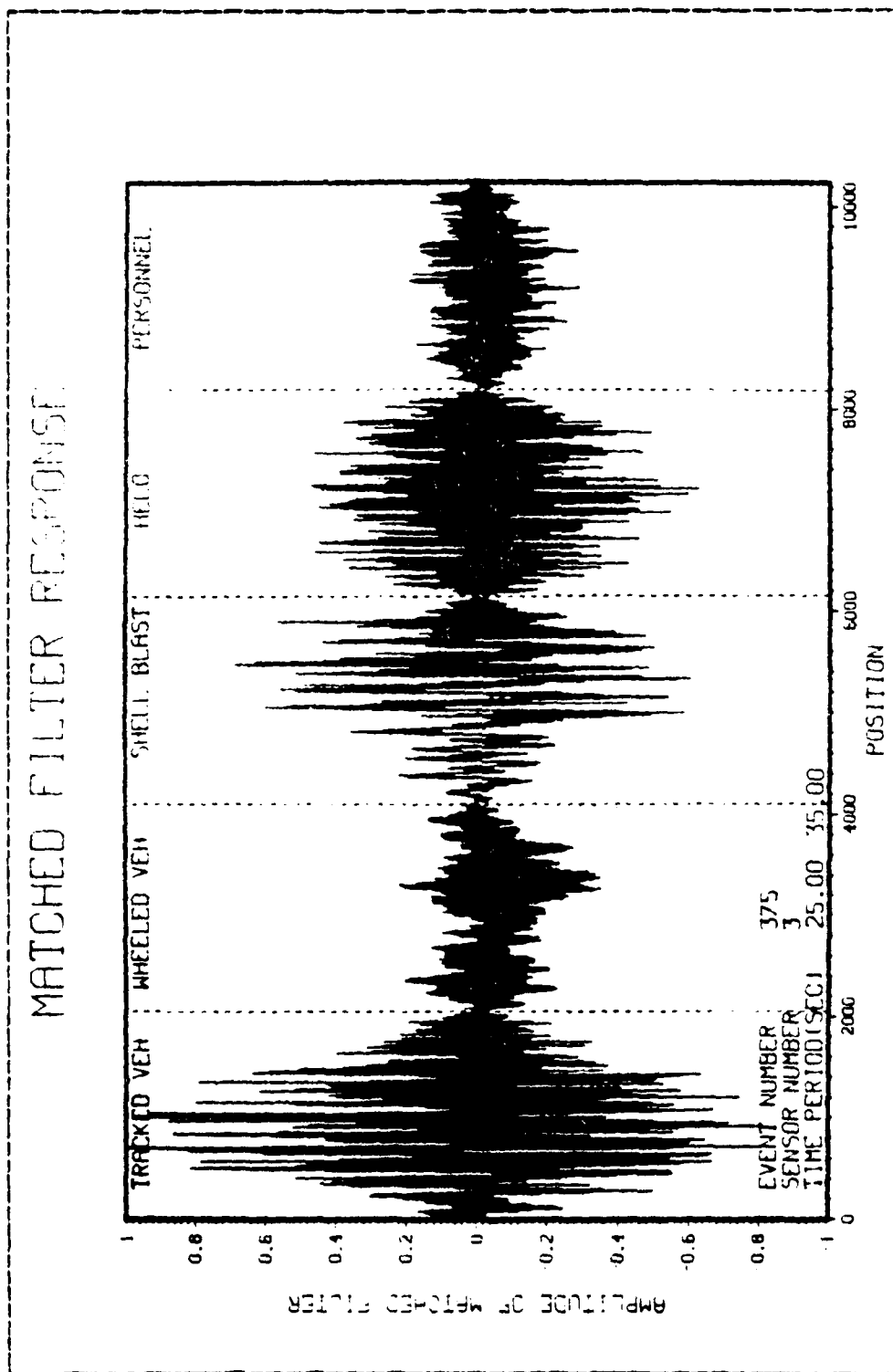


Figure 6.34 Matched Filter Response for Event 375

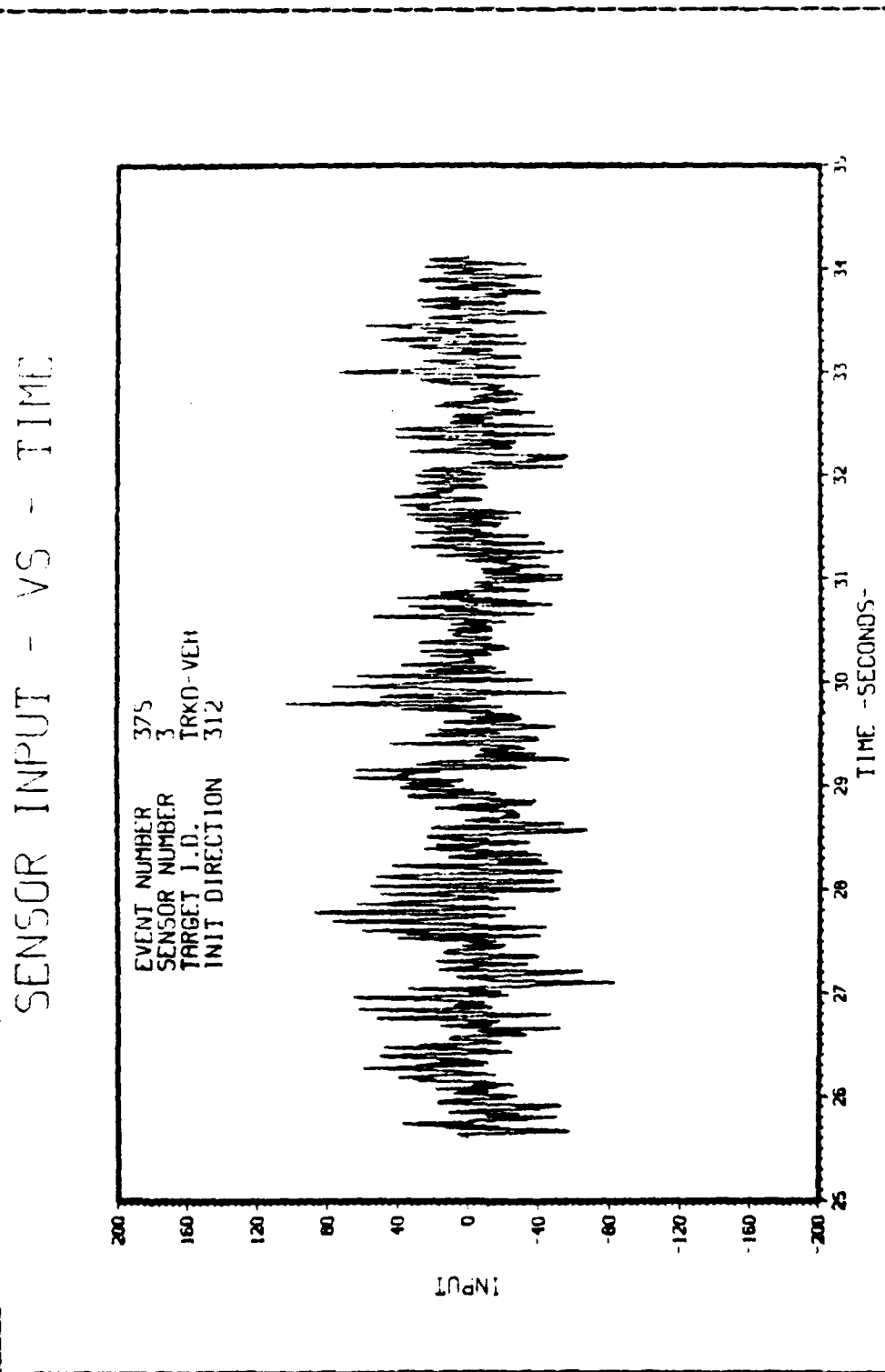


Figure 6.35 Amplitude Response for Event 375

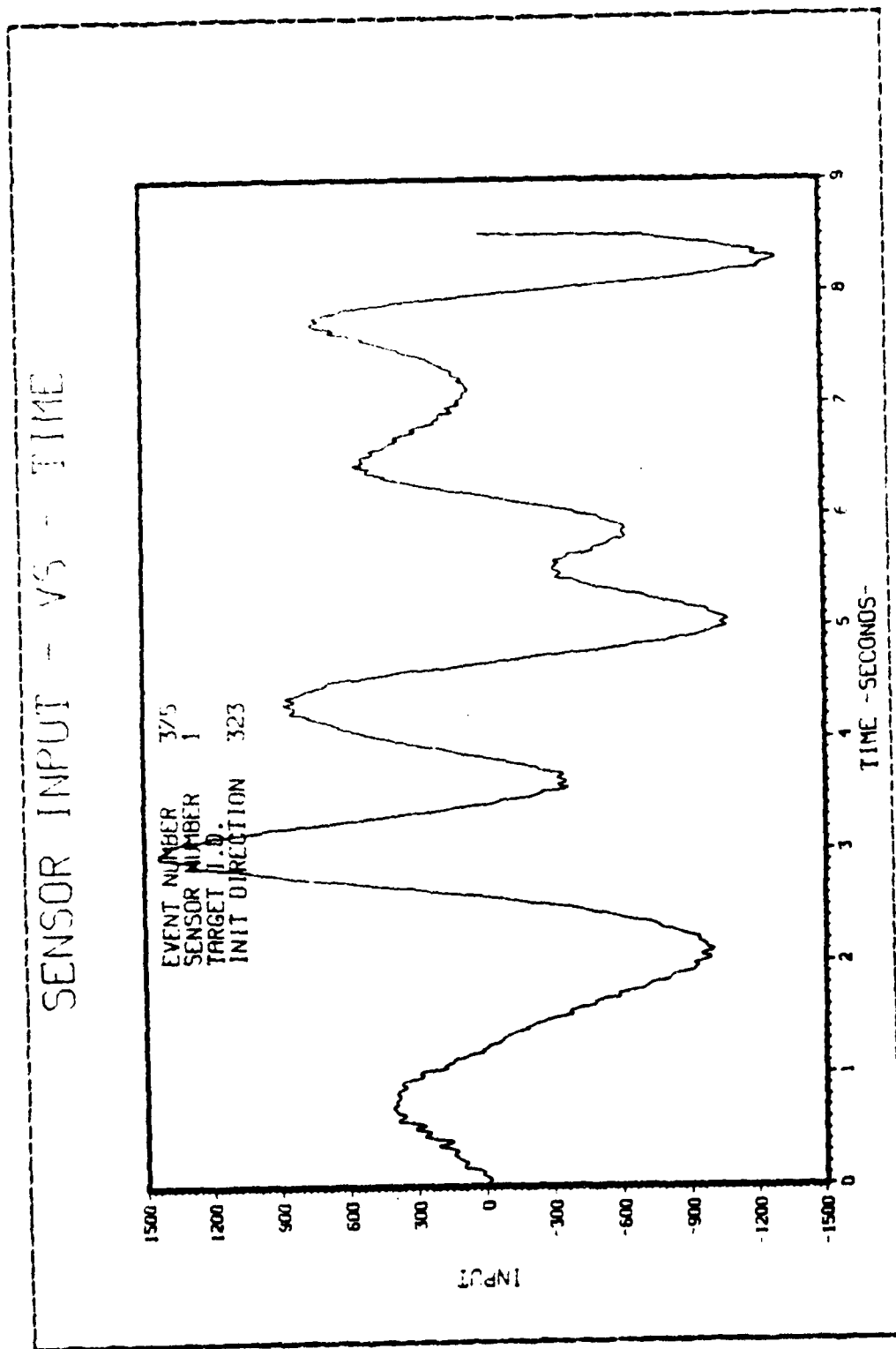


Figure 6.36 Amplitude Response of Malfunctioning Sensor for Event 375

SENSOR POWER -VS- FREQUENCY

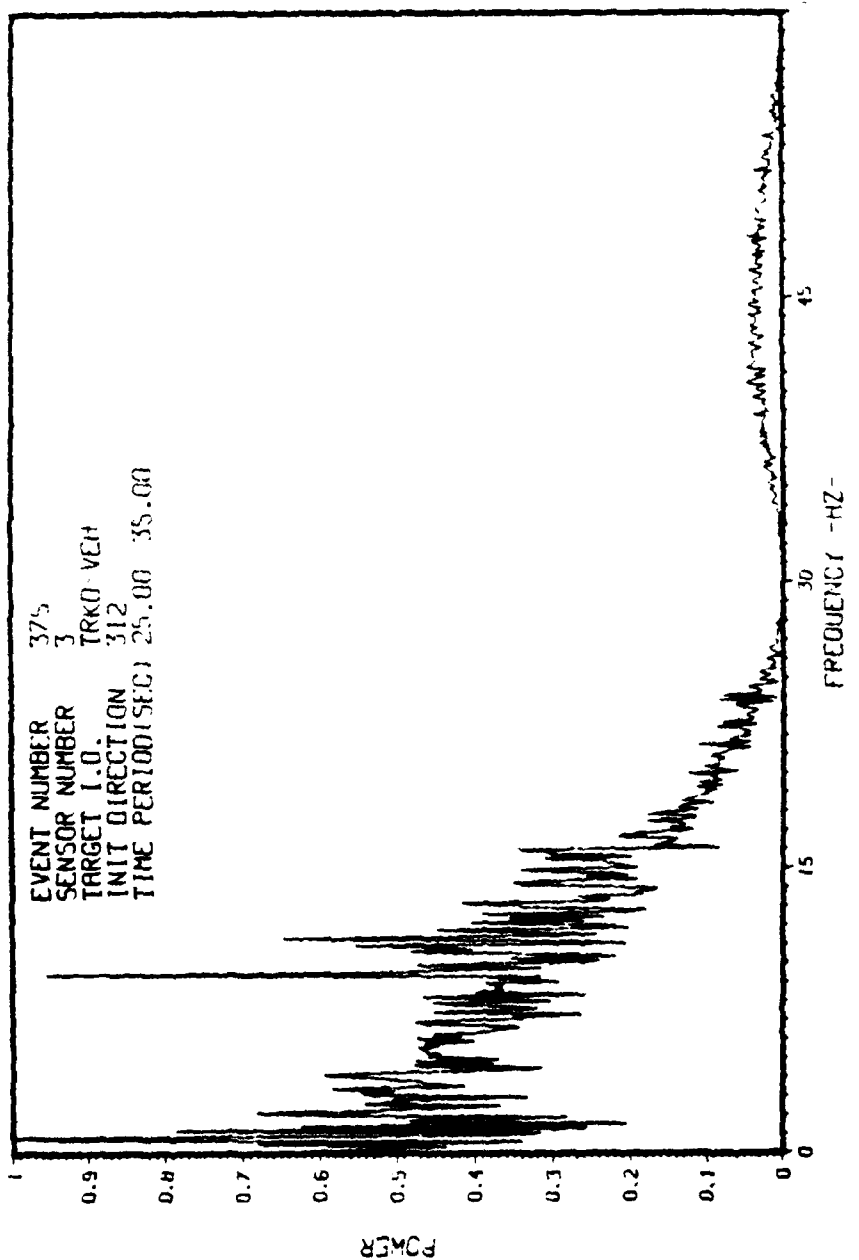


Figure 6.37 Frequency Response for Event 375

LEAST MEAN SQUARES POLYNOMIAL

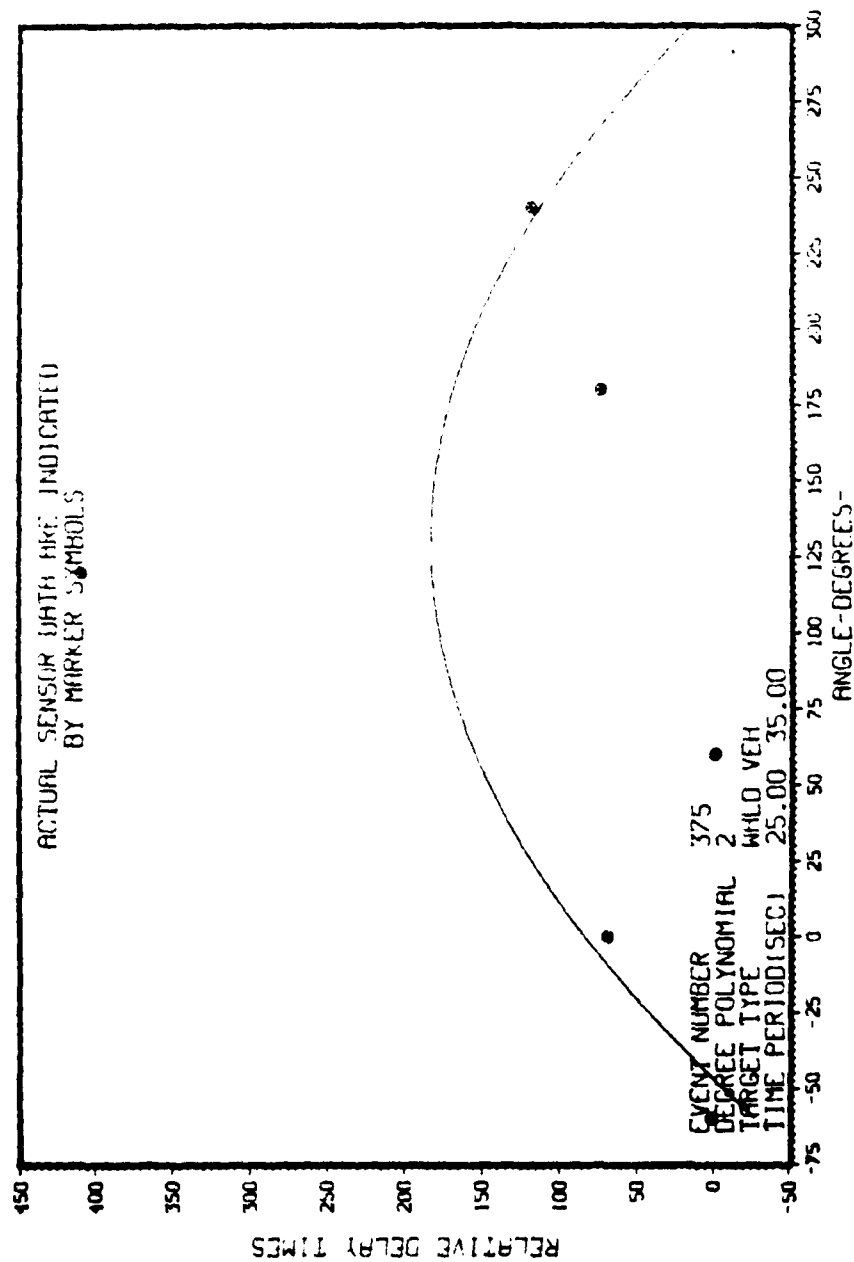


Figure 6.38 LMSP Matched Filter Direction for Event 375

MULTIPLE TARGET - MATCHED FILTER OUTPUT

EVENT NUMBER	375
TIME PERIOD(SEC)	25.00 35.00
TRACKED VEHICLE	DIRECTION - 0.00
SHELL BLAST	DIRECTION - 312.00
HELICOPTER	DIRECTION - 0.00
SIMULATED TRKD VEHICLE	TARGET FREQUENCY 0.00
AMPLITUDE	0.0000
DIRECTION	0.0000
SIMULATED WILD VEHICLE	TARGET FREQUENCY 0.00
AMPLITUDE	0.0000
DIRECTION	0.0000
SIMULATED HELICOPTER	TARGET FREQUENCY 0.00
AMPLITUDE	0.0000
DIRECTION	0.0000
SIMULATED PERSONNEL	TARGET FREQUENCY 0.00
AMPLITUDE	0.0000
DIRECTION	0.0000

Figure 6.39 LAMP Multiple Target Direction Summary for Event 375

LEAST MEAN SQUARES POLYNOMIAL

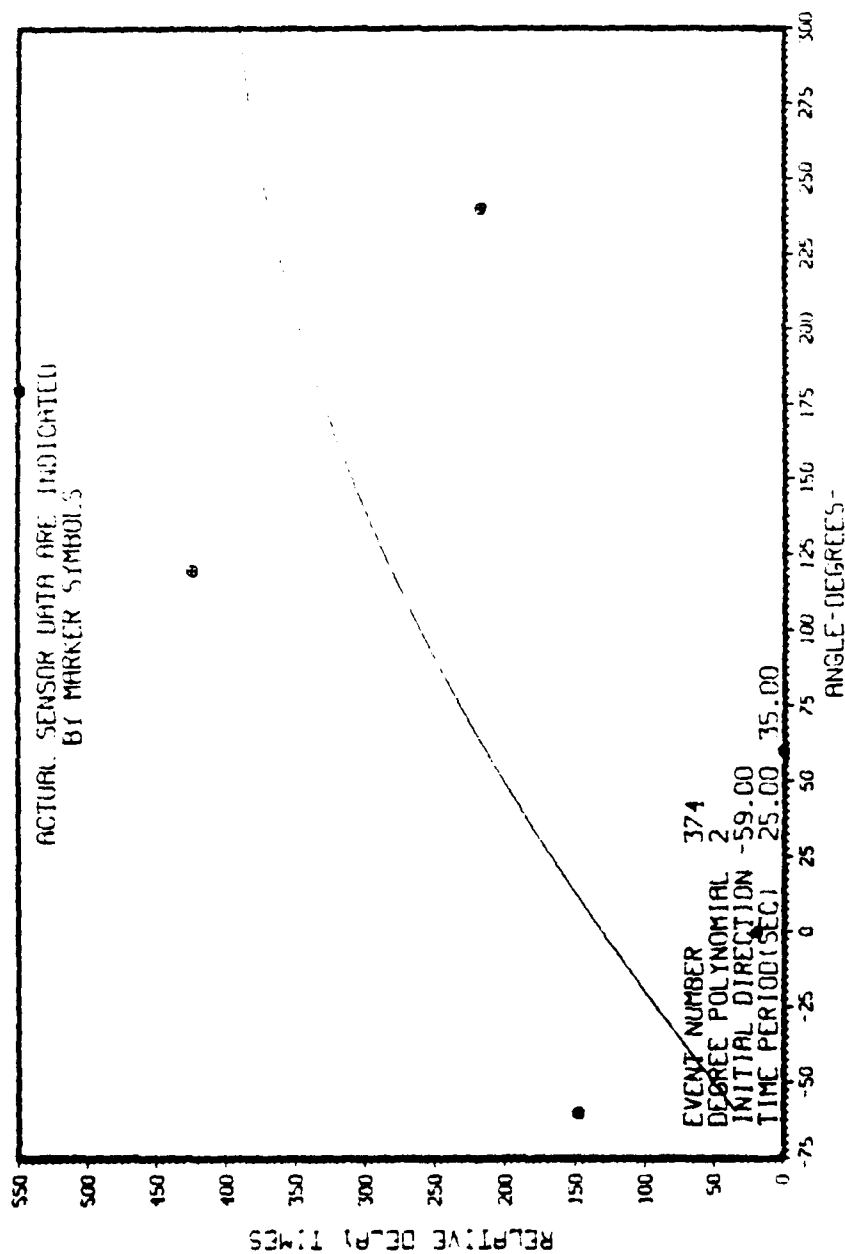


Figure 6.40 LMSP Initial Direction for Event 374

MATCHED FILTER RESPONSE

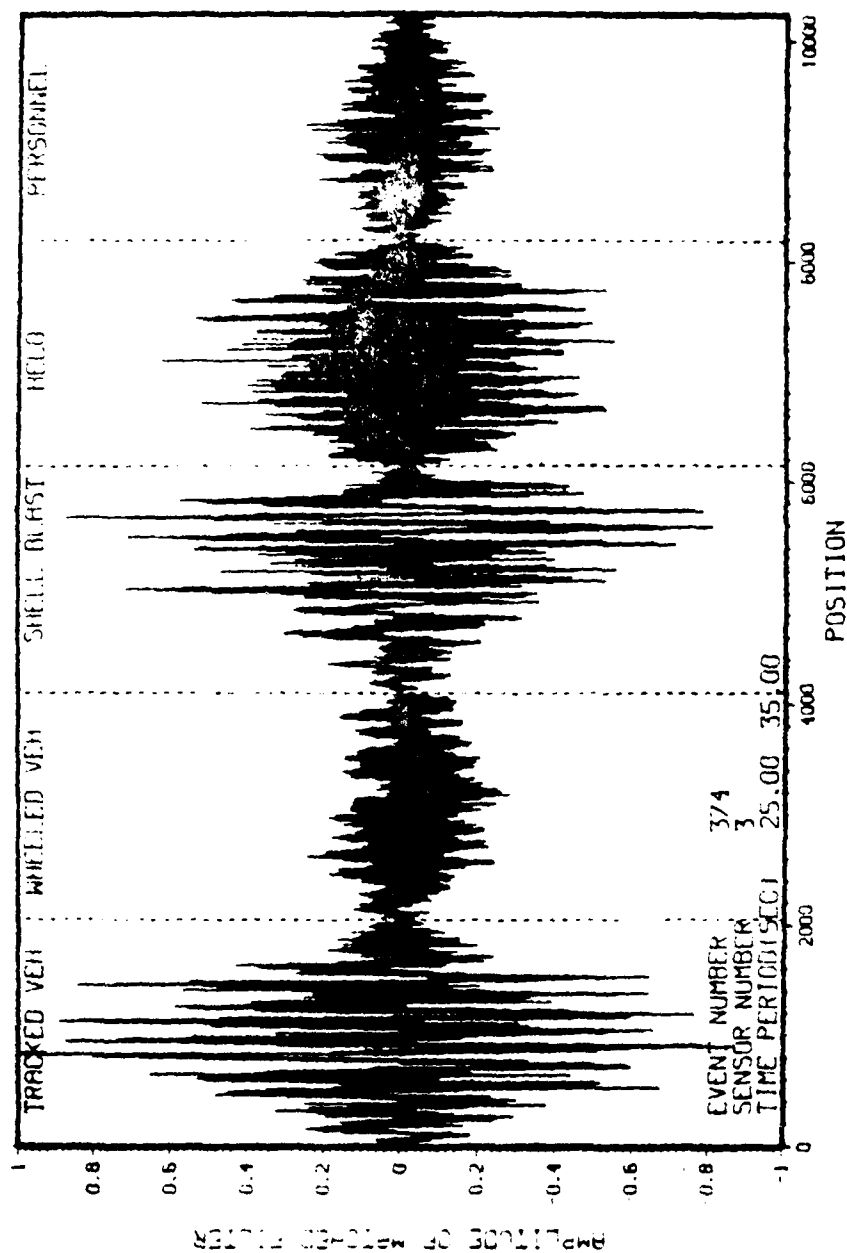


Figure 6.41 Matched Filter Response for Event 374

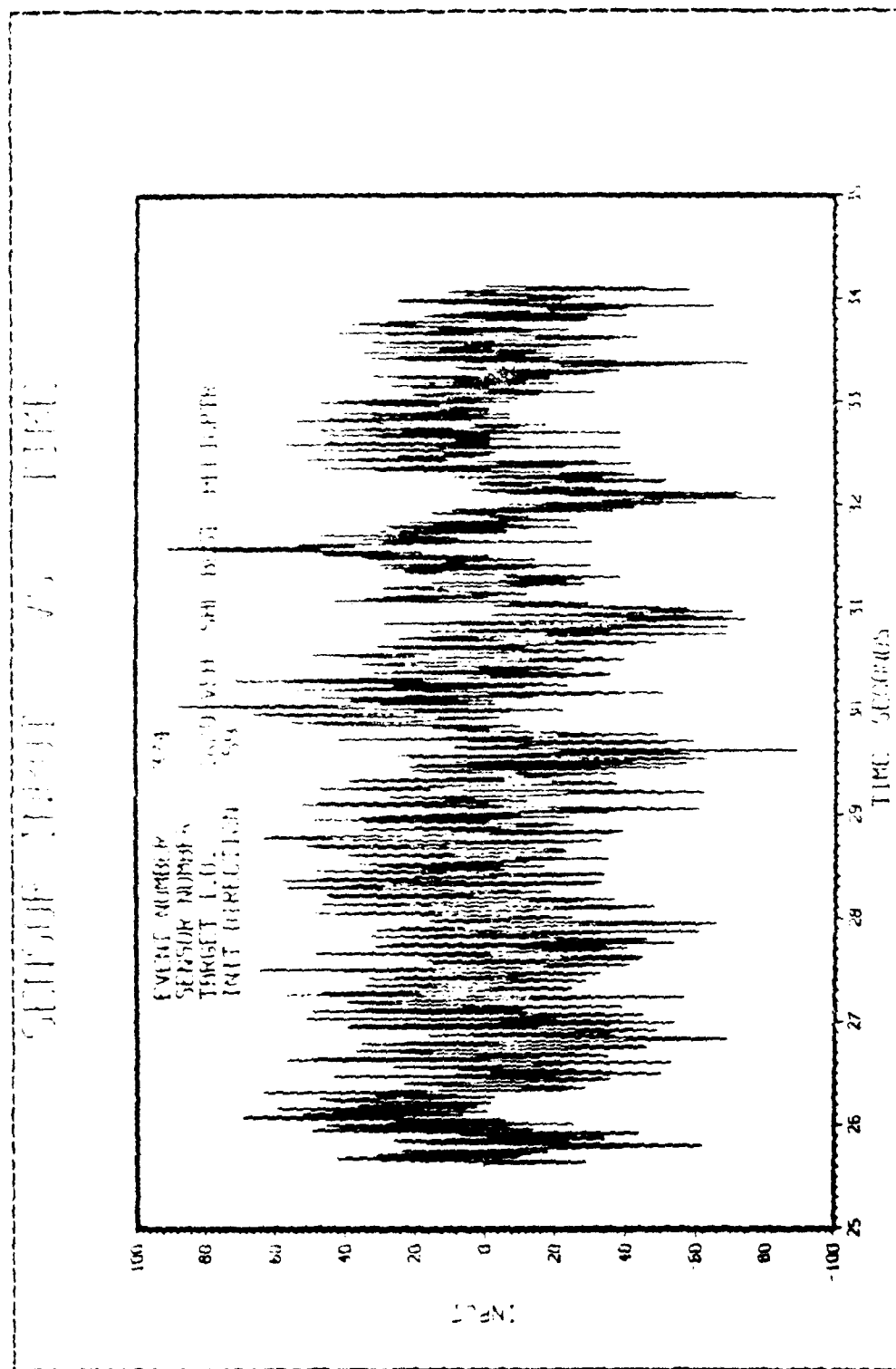


Figure 6.42 Amplitude Response for Event 374

SENSOR INPUT - VS - TIME

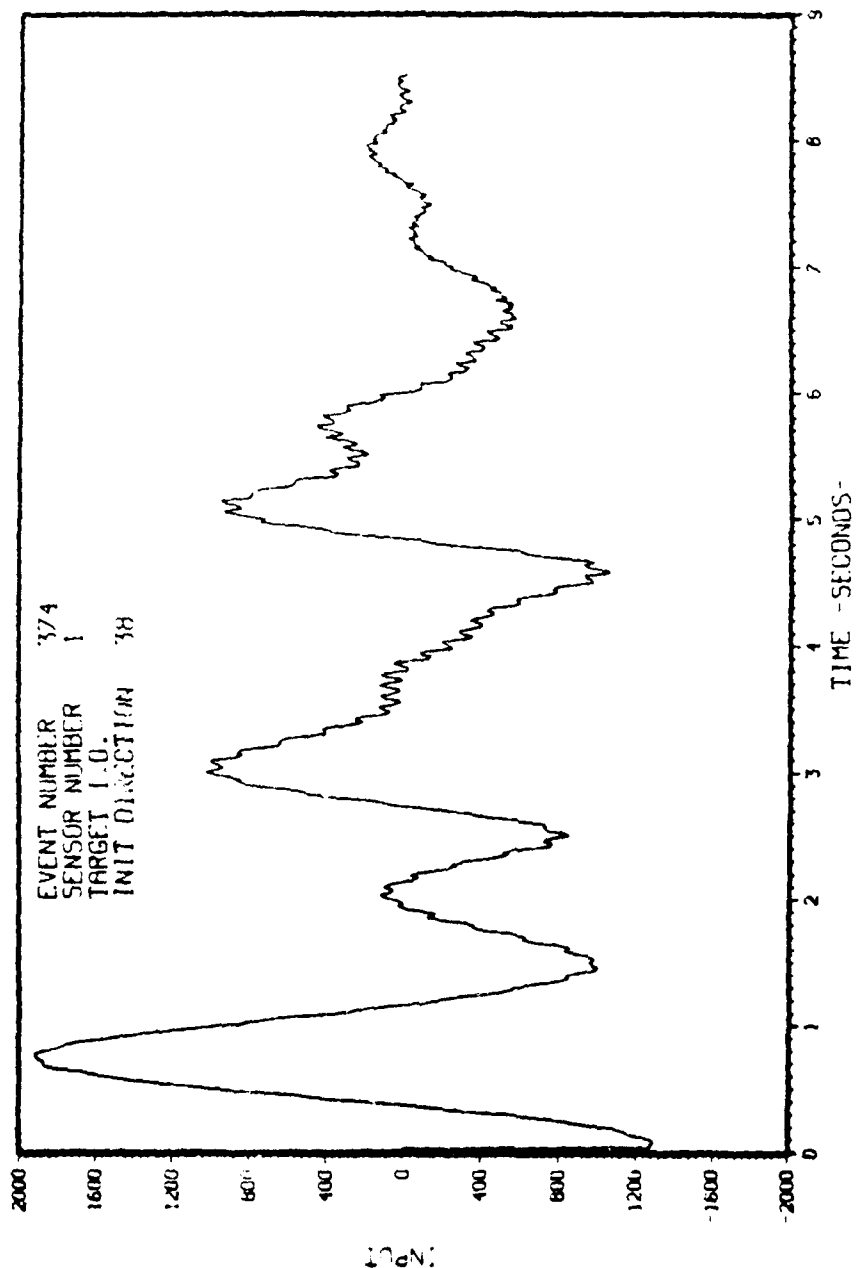


Figure 6.43 Amplitude Response of Malfunctioning Sensor for Event 374

AD-A138 914

MULTIPLE TARGET IDENTIFICATION AND DIRECTION FINDING
USING MATCHED FILTERING TECHNIQUES(U) NAVAL
POSTGRADUATE SCHOOL MONTEREY CA J L JOHNSTON DEC 83

212

UNCLASSIFIED

F/G 17/10

NL

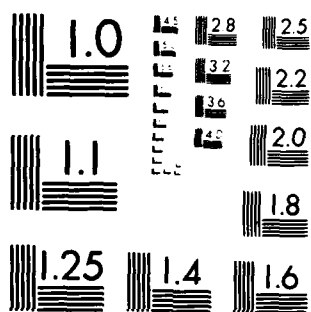
END

DATE

FORMED

4 84

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

SENSOR POWER - VS- FREQUENCY

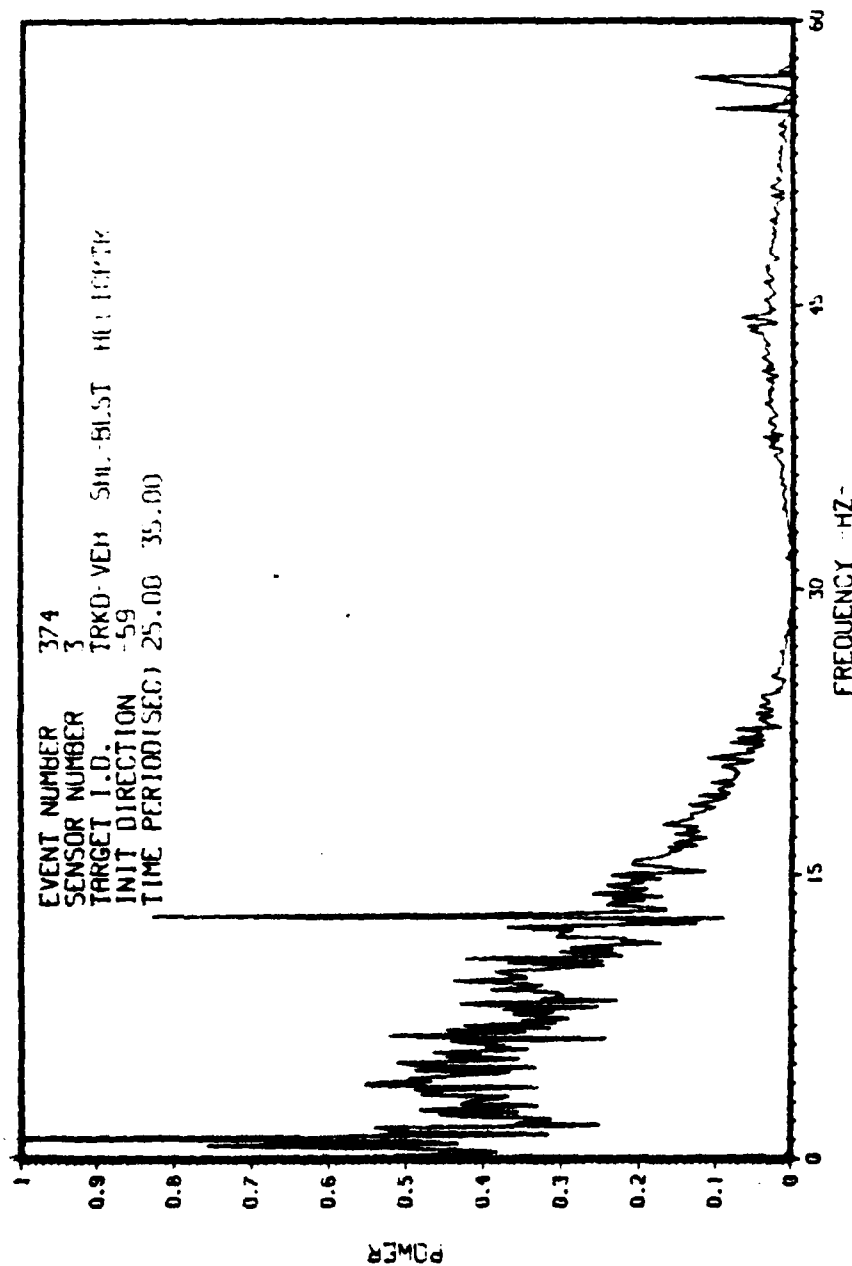


Figure 6.44 Frequency Response for Event 374

LEAST MEAN SQUARES POLYNOMIAL

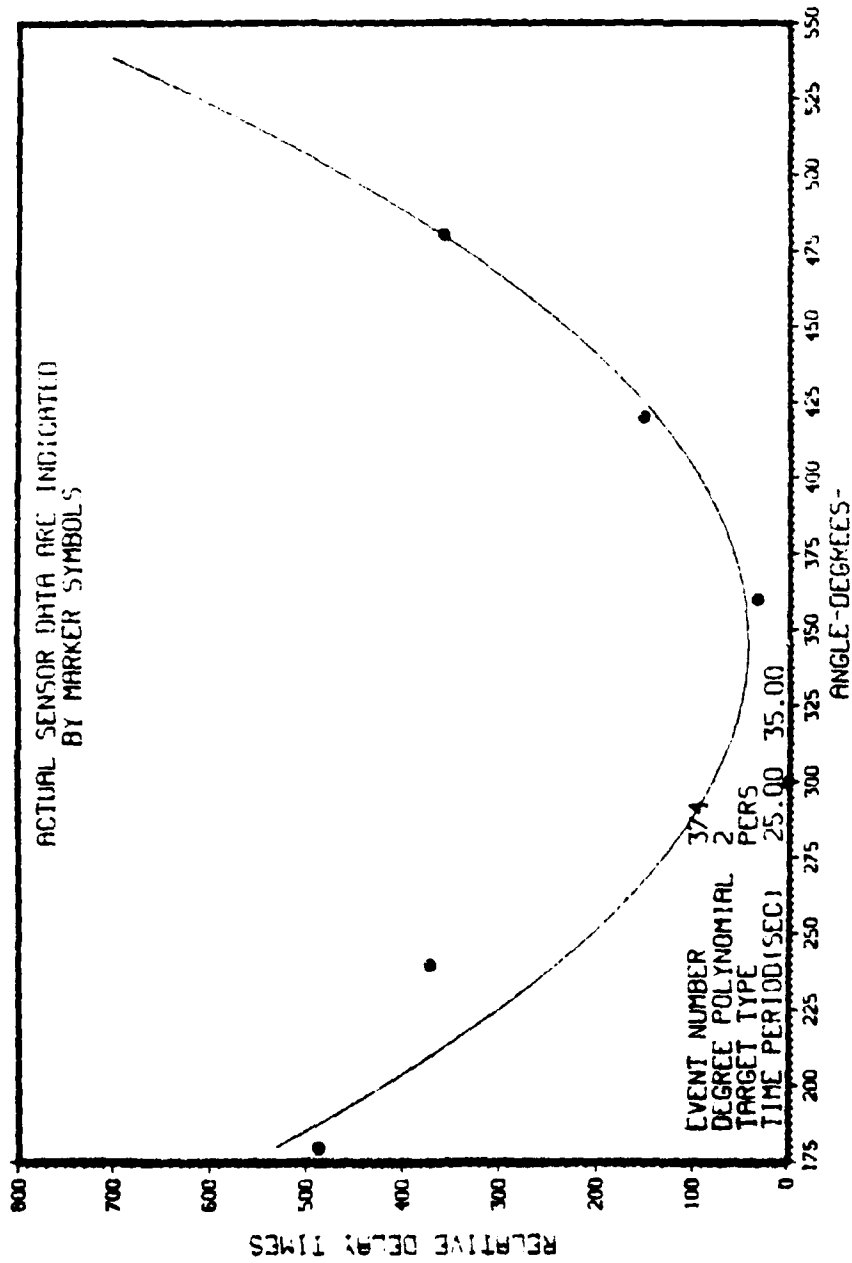


Figure 6.45 LMSP Matched Filter Direction for Event 374

MULTIPLE TARGET - MATCHED FILTER OUTPUT

EVENT NUMBER	374
TIME PERIOD(SEC)	25.00 35.00
TRACKED VEHICLE	DIRECTION - 59.00
SHELL BLAST	DIRECTION - 59.00
PERSONNEL	DIRECTION - 347.00
SIMULATED TRKD VEHICLE	TARGET FREQUENCY 0.00
AMPLITUDE	0.0000
DIRECTION	0.0000
SIMULATED WILD VEHICLE	TARGET FREQUENCY 0.00
AMPLITUDE	0.0000
DIRECTION	0.0000
SIMULATED HELICOPTER	TARGET FREQUENCY 0.00
AMPLITUDE	0.0000
DIRECTION	0.0000
SIMULATED PERSONNEL	TARGET FREQUENCY 0.00
AMPLITUDE	0.0000
DIRECTION	0.0000

Figure 6.46 LMSF Multiple Target Direction Summary for Event 374

LEAST MEAN SQUARES POLYNOMIAL

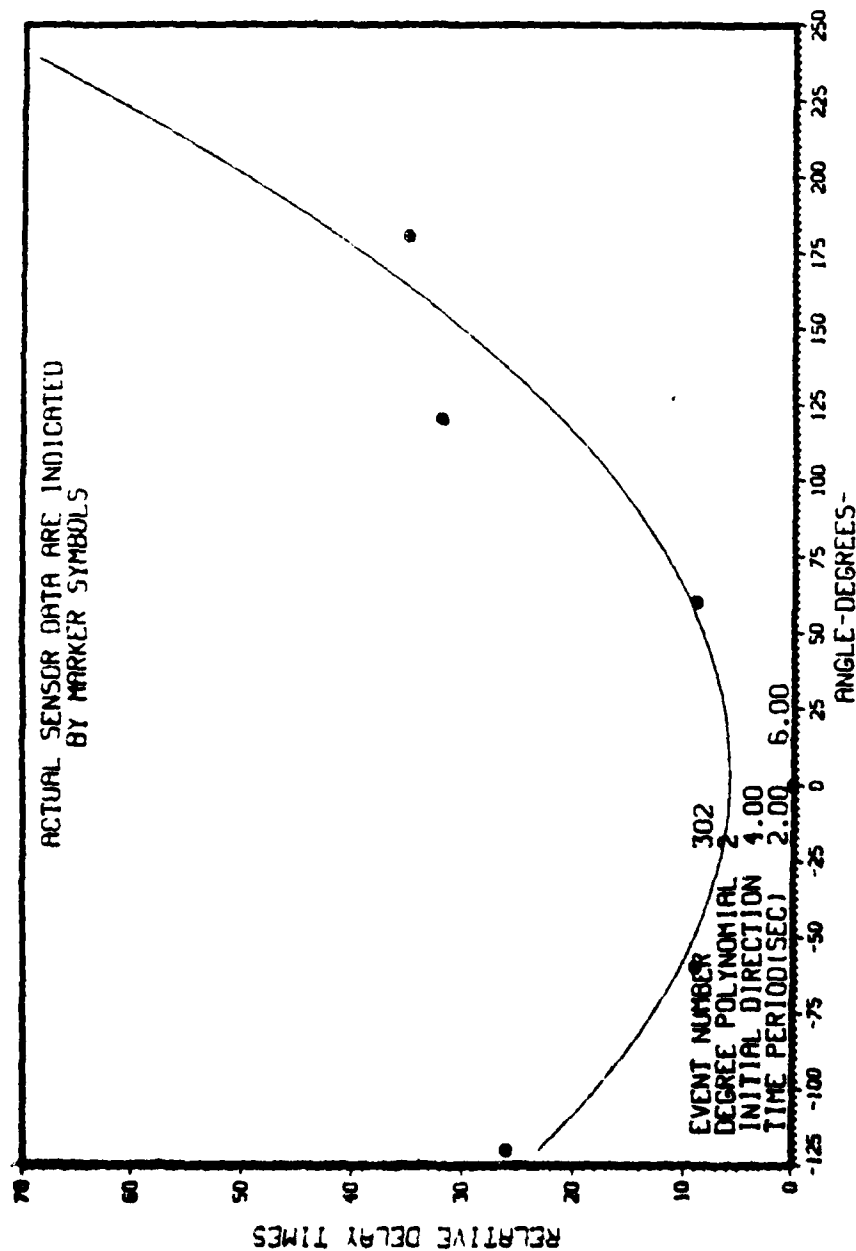


Figure 6.47 LMSP Initial Direction for Event 302

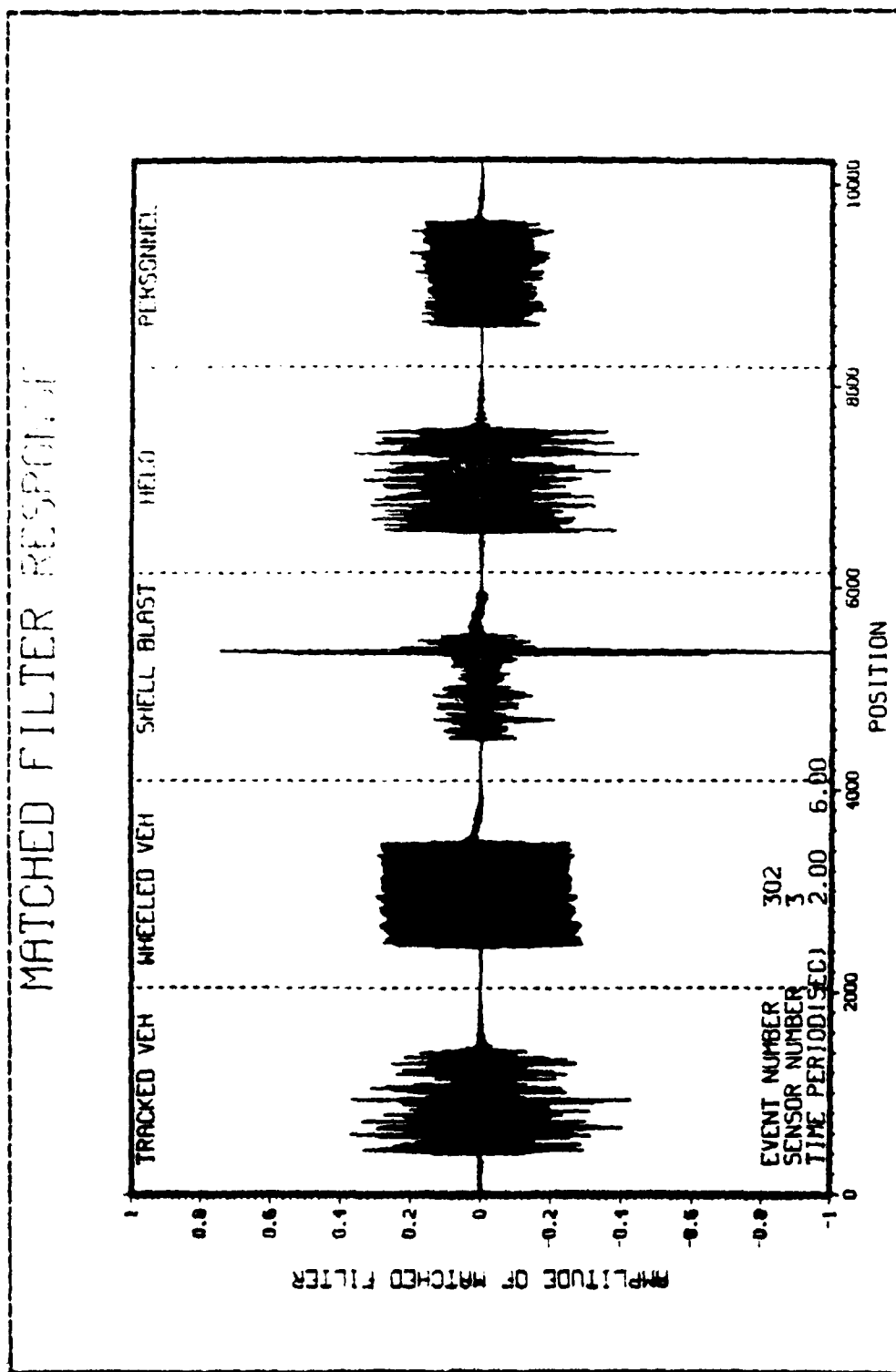


Figure 6.48 Matched Filter Response for Event 302

SENSOR INPUT - VS - TIME

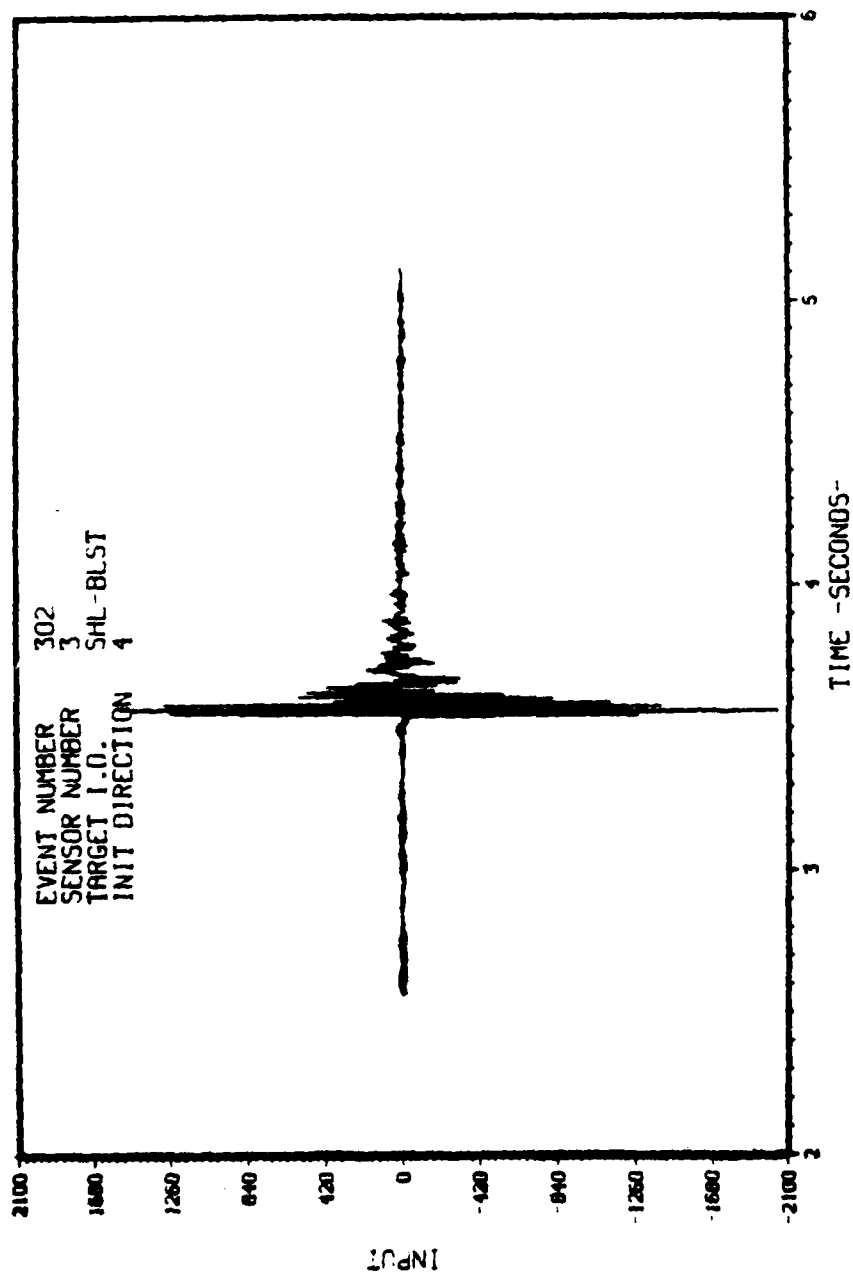


Figure 6.49 Amplitude Response for Event 302

SENSOR POWER -VS- FREQUENCY

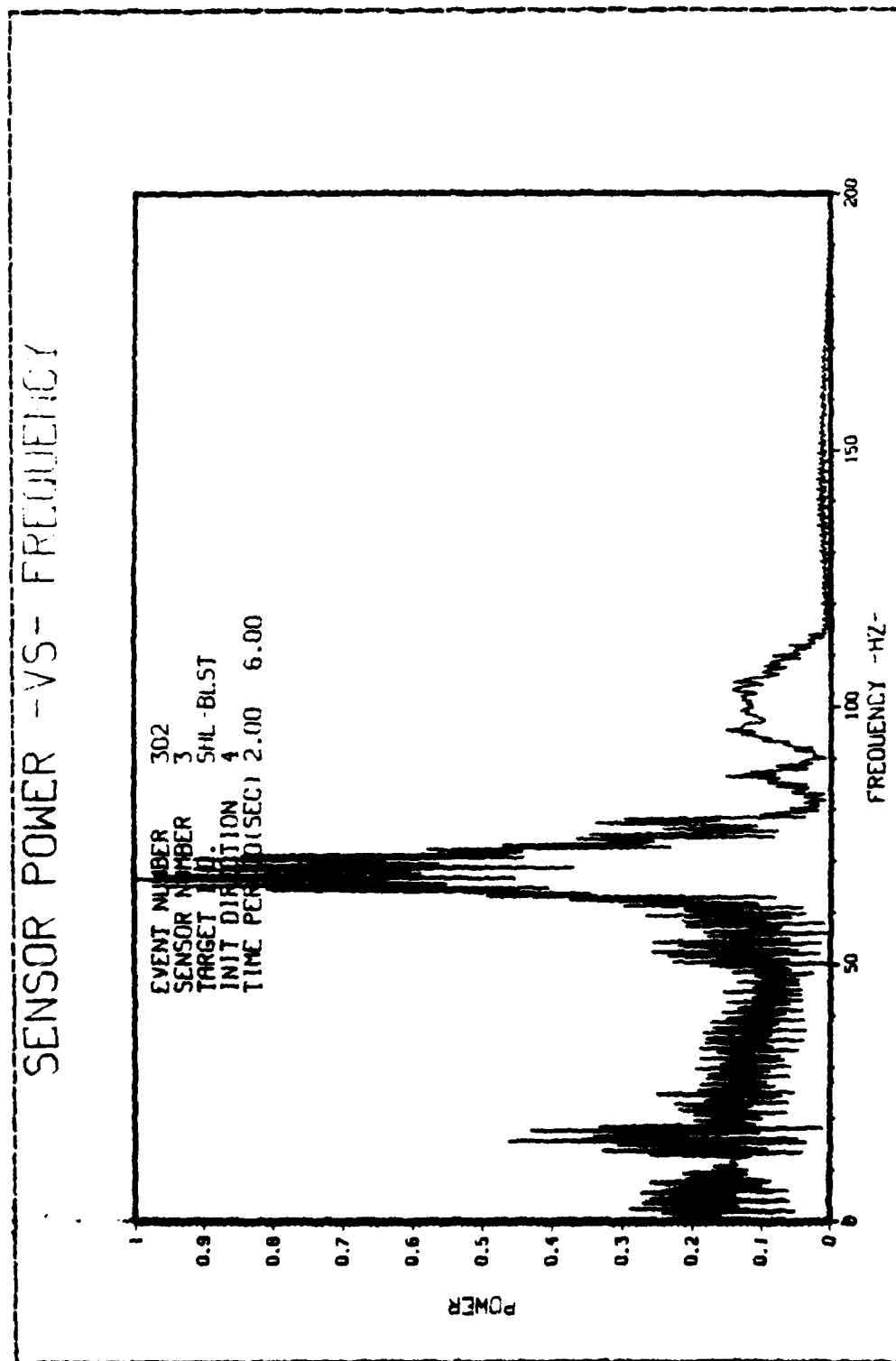


Figure 6.50 Frequency Response for Event 302

LEAST MEAN SQUARES POLYNOMIAL

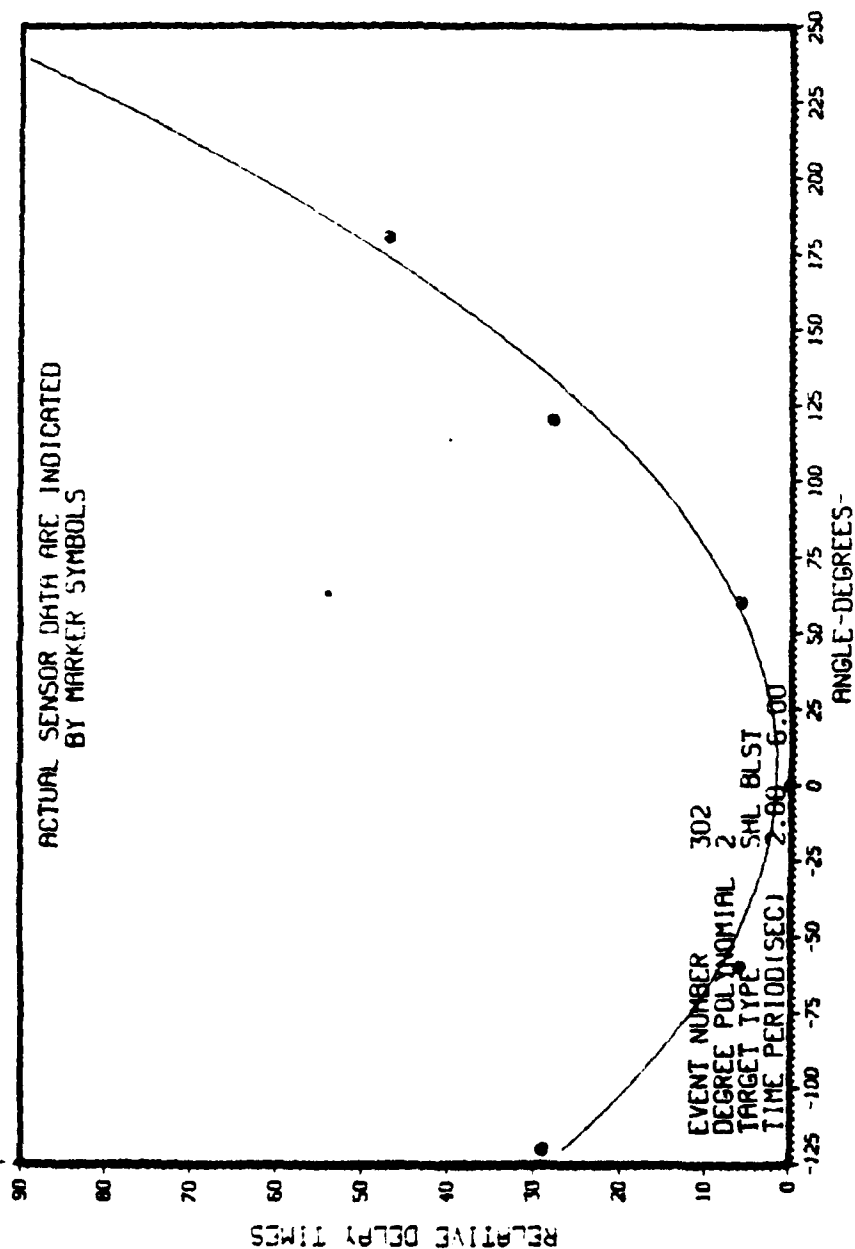


Figure 6.51 LMSP Matched Filter Direction for Event 302

MULTIPLE TARGET - MATCHED FILTER OUTPUT

EVENT NUMBER	302
TIME PERIOD(SEC)	2.00 6.00
SHELL BLAST	DIRECTION - 6.00
SIMULATED TRKO VEHICLE TARGET FREQUENCY	0.00
AMPLITUDE	0.0000
DIRECTION	0.0000
SIMULATED WILD VEHICLE TARGET FREQUENCY	0.00
AMPLITUDE	0.0000
DIRECTION	0.0000
SIMULATED HELICOPTER TARGET FREQUENCY	0.00
AMPLITUDE	0.0000
DIRECTION	0.0000
SIMULATED PERSONNEL TARGET FREQUENCY	0.00
AMPLITUDE	0.0000
DIRECTION	0.0000

Figure 6.52 LNRP Multiple Target Direction Summary for Event 302

LEAST MEAN SQUARES POLYNOMIAL

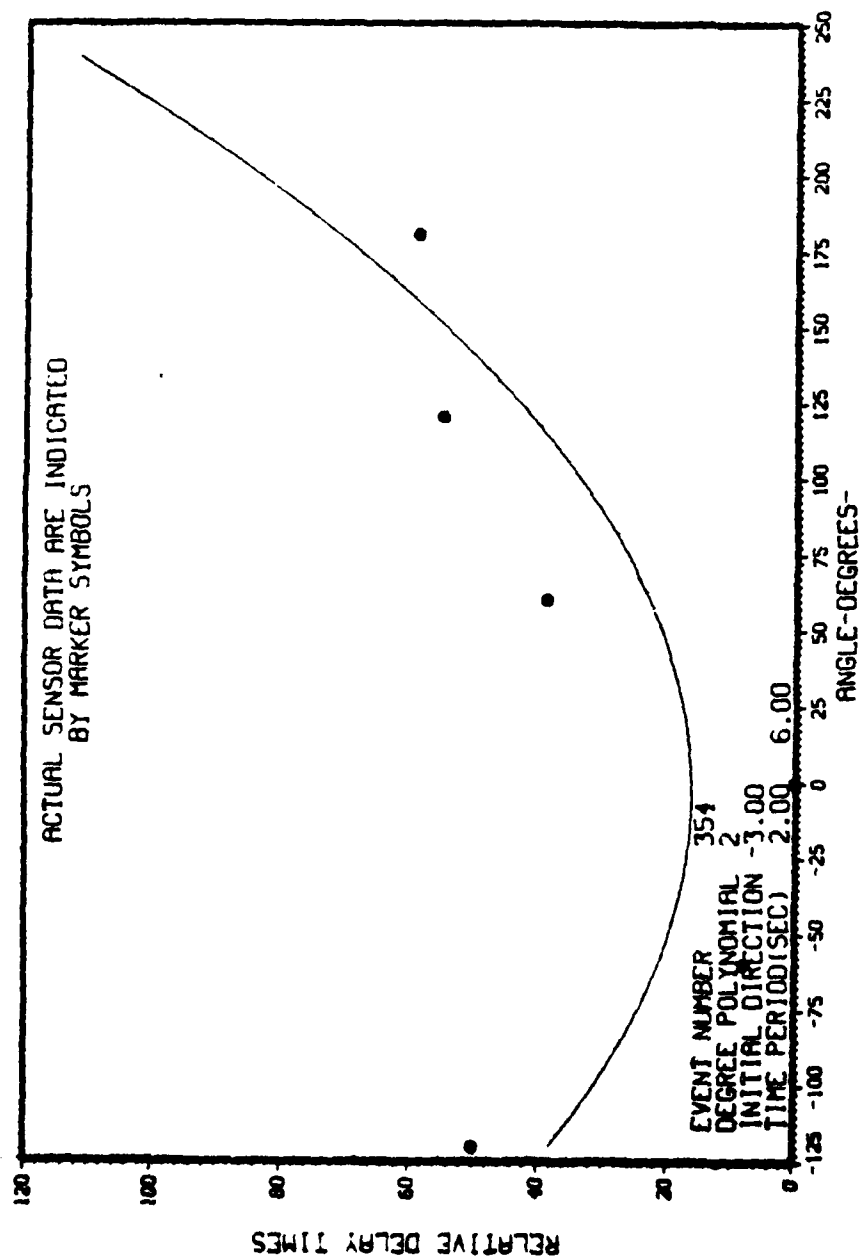


Figure 6.53 LMSP Initial Direction for Event 354 (2 - 6sec)

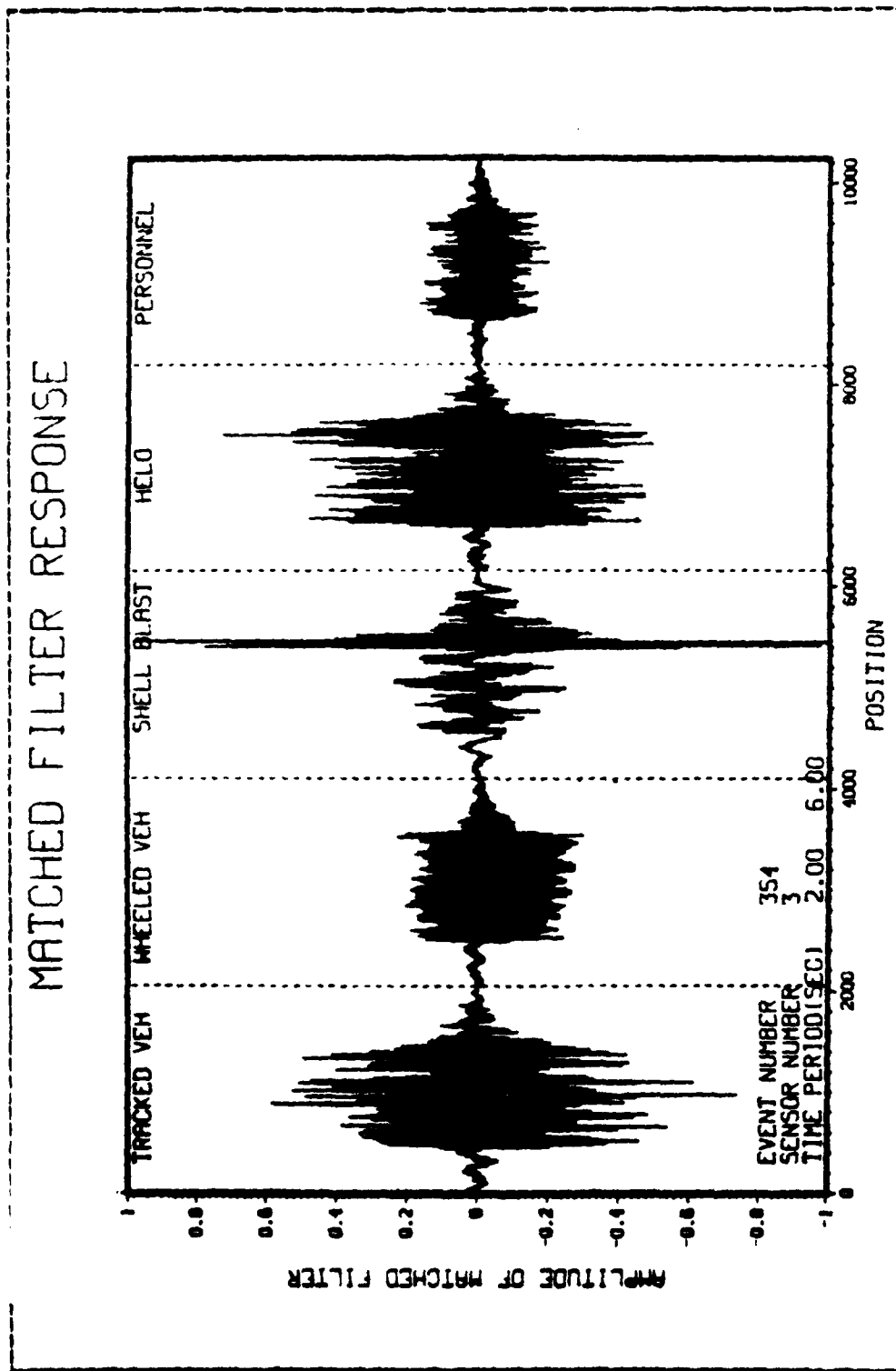


Figure 6.54 Matched Filter Res use for Event 354 (2 - 6sec)

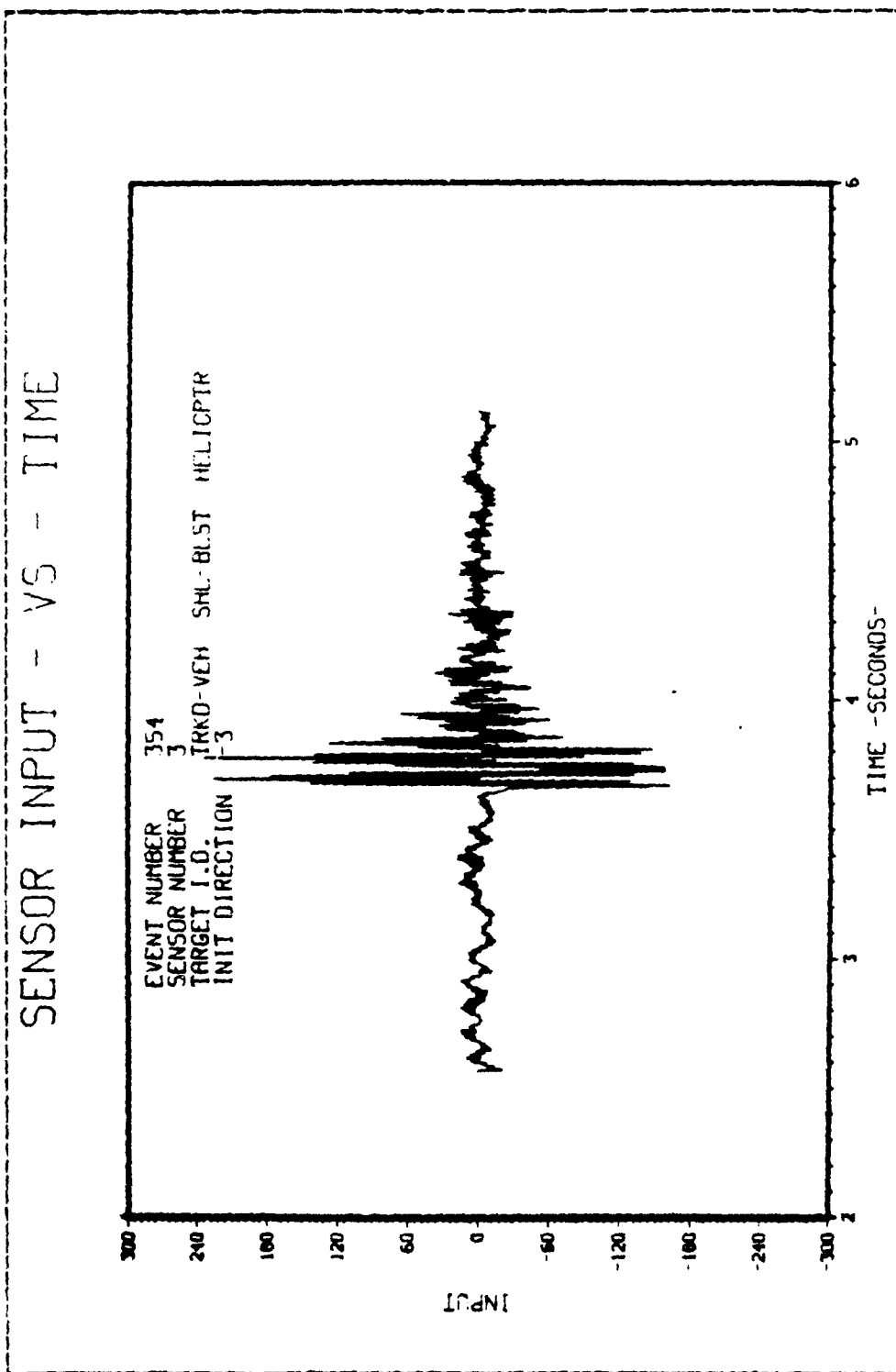


Figure 6.55 Amplitude Response for Event 354 (2 - 6sec)

SENSOR POWER -VS- FREQUENCY

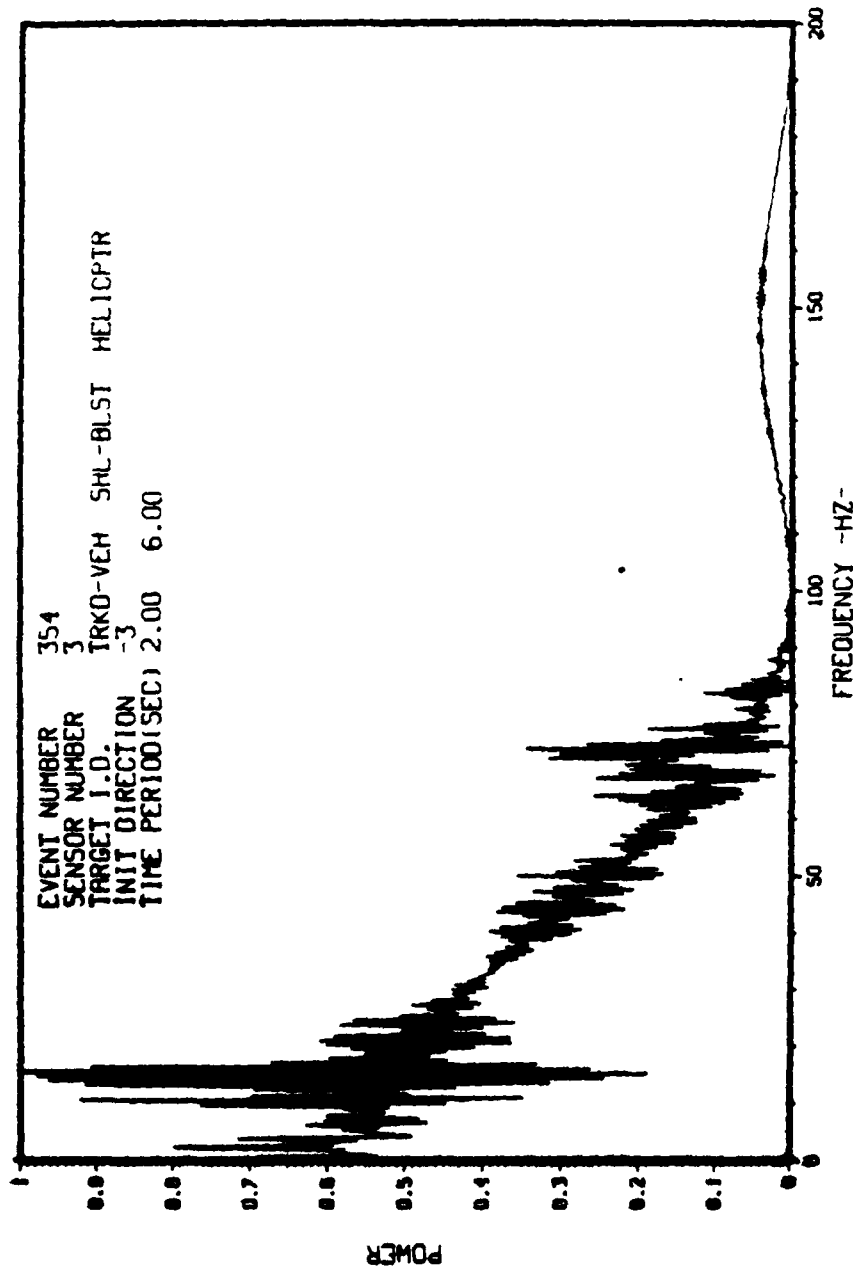


Figure 6.56 Frequency Response for Event 354 (2 - 6sec)

LEAST MEAN SQUARES POLYNOMIAL

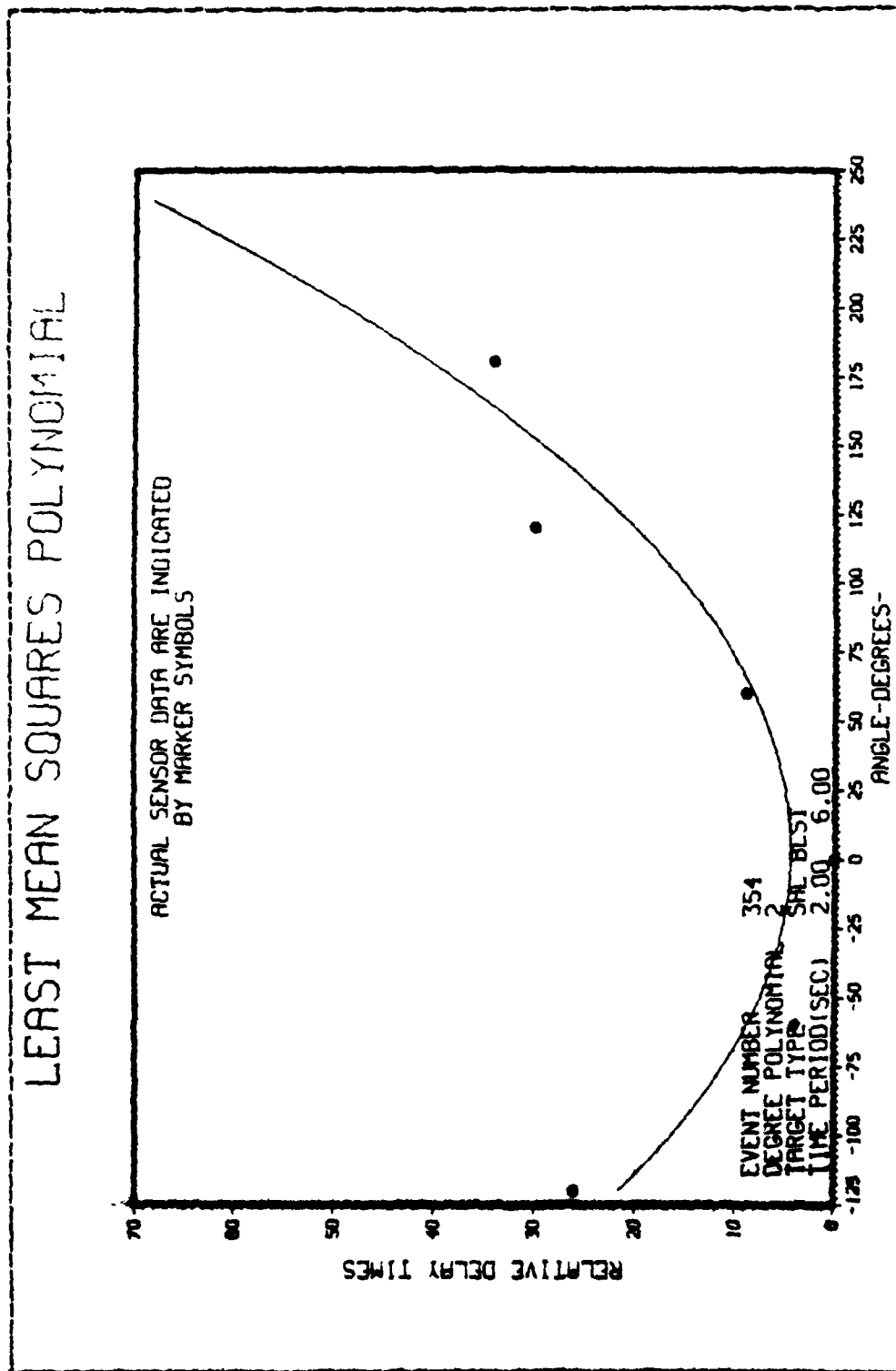


Figure 6.57 LMSP Matched Filter Direction for Event 354 (2 - 6sec)

MULTIPLE TARGET - MATCHED FILTER OUTPUT

EVENT NUMBER	354
TIME PERIOD(SEC)	2.00 6.00
TRACKED VEHICLE	DIRECTION - 332.00
SHELL BLAST	DIRECTION - 3.00
HELICOPTER	DIRECTION - 355.00
SIMULATED TRKD VEHICLE	TARGET FREQUENCY 0.00
AMPLITUDE	0.0000
DIRECTION	0.0000
SIMULATED WILD VEHICLE	TARGET FREQUENCY 0.00
AMPLITUDE	0.0000
DIRECTION	0.0000
SIMULATED HELICOPTER	TARGET FREQUENCY 0.00
AMPLITUDE	0.0000
DIRECTION	0.0000
SIMULATED PERSONNEL	TARGET FREQUENCY 0.00
AMPLITUDE	0.0000
DIRECTION	0.0000

Figure 6.58 LSP Multiple Target Direction Summary Event 354 (2 - 6sec)

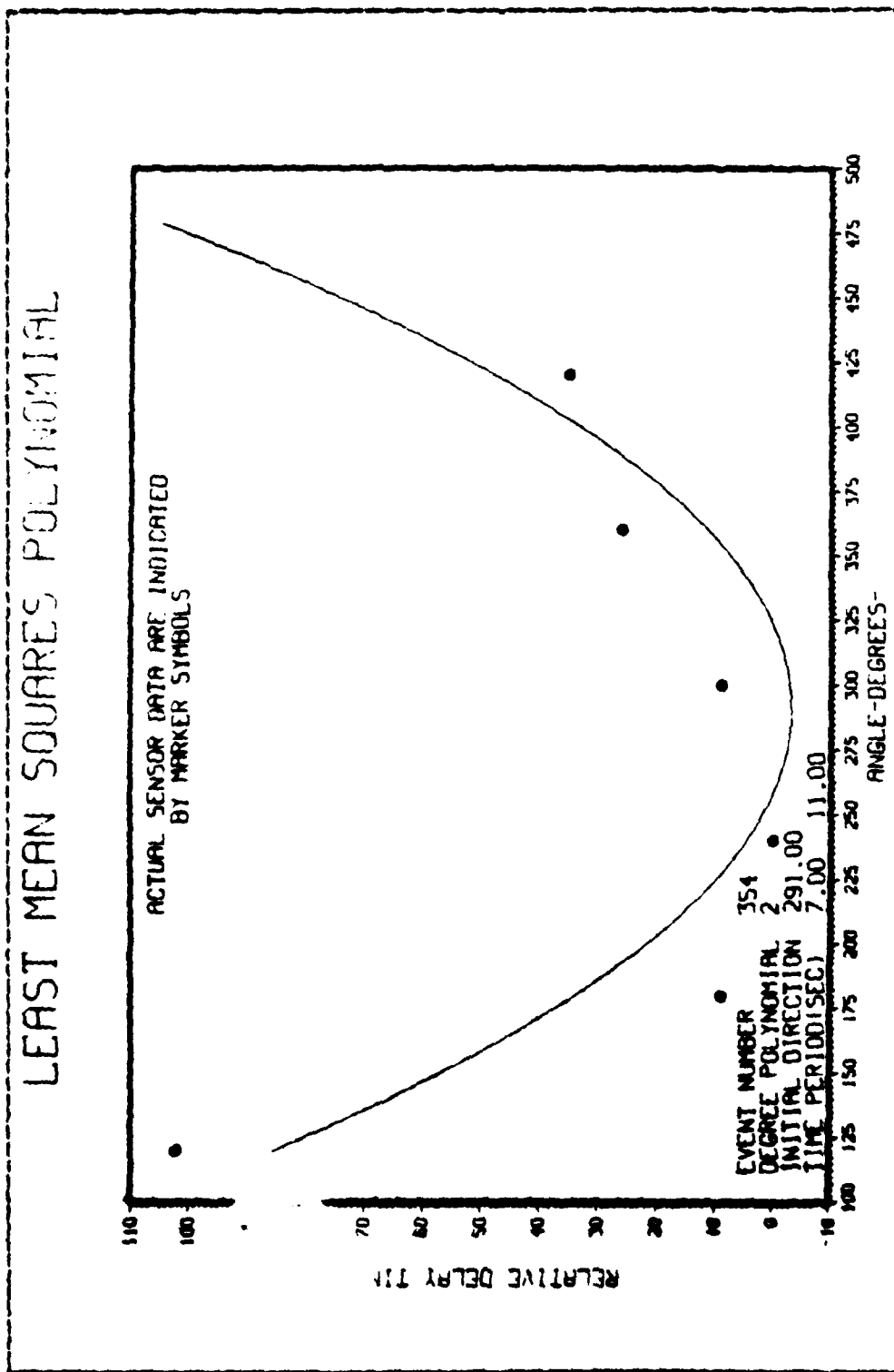


Figure 6.59 LMSP Initial Direction for Event 354 (7 - 11sec)

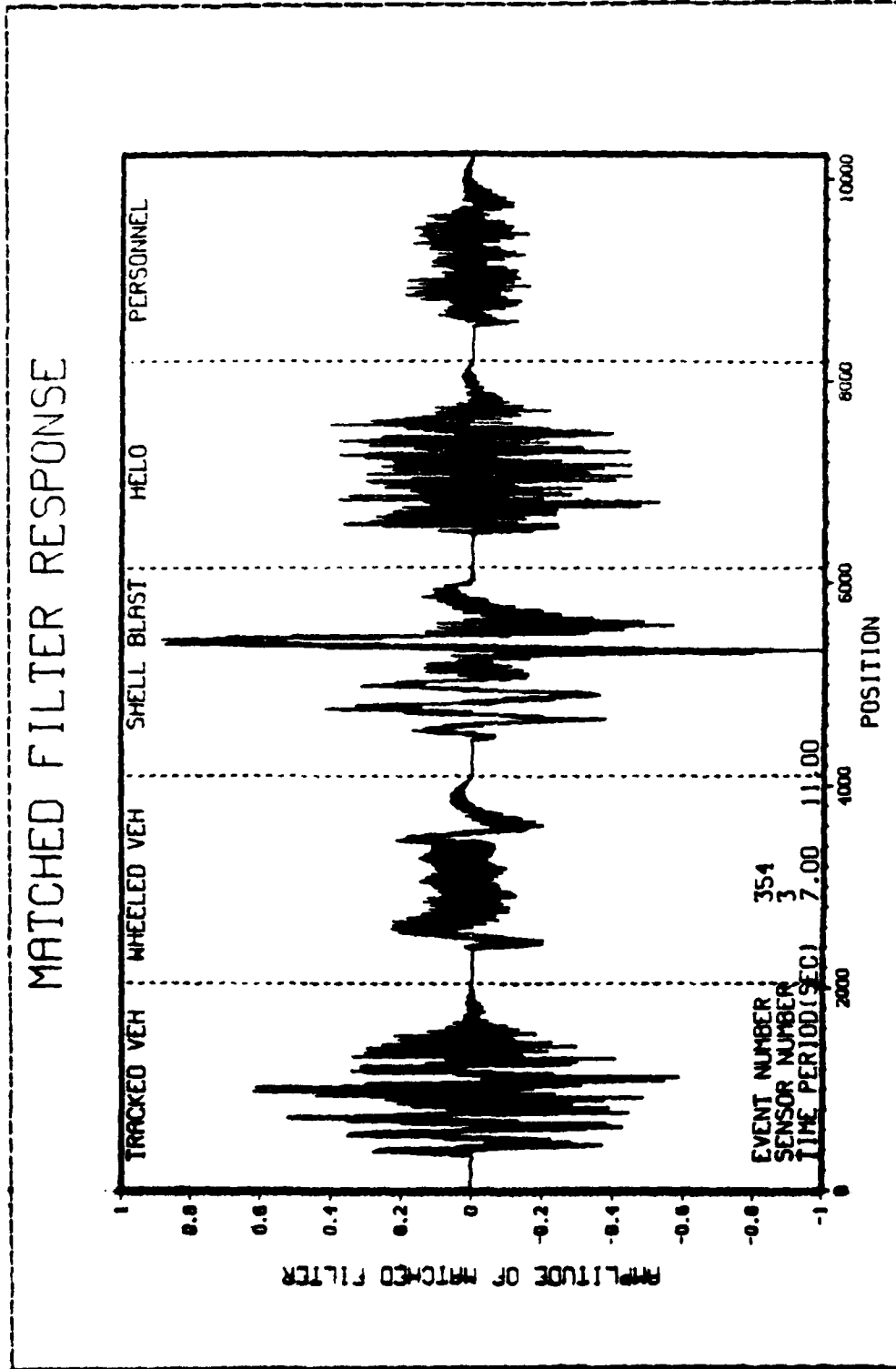


Figure 6.60 Matched Filter Response for Event 354 (7 - 11sec)

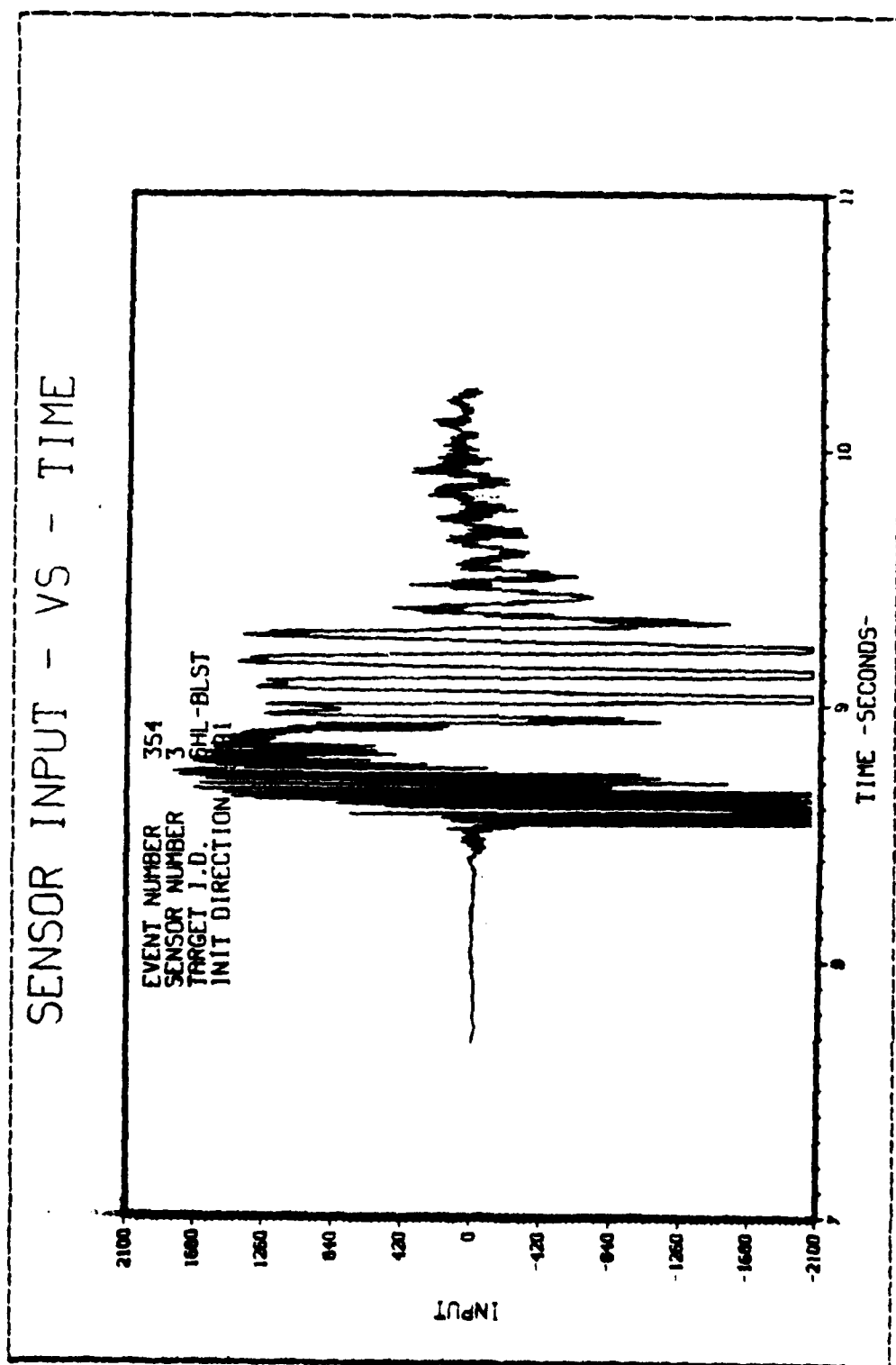


Figure 6.61 Amplitude Response for Event 354 (7 - 11sec)

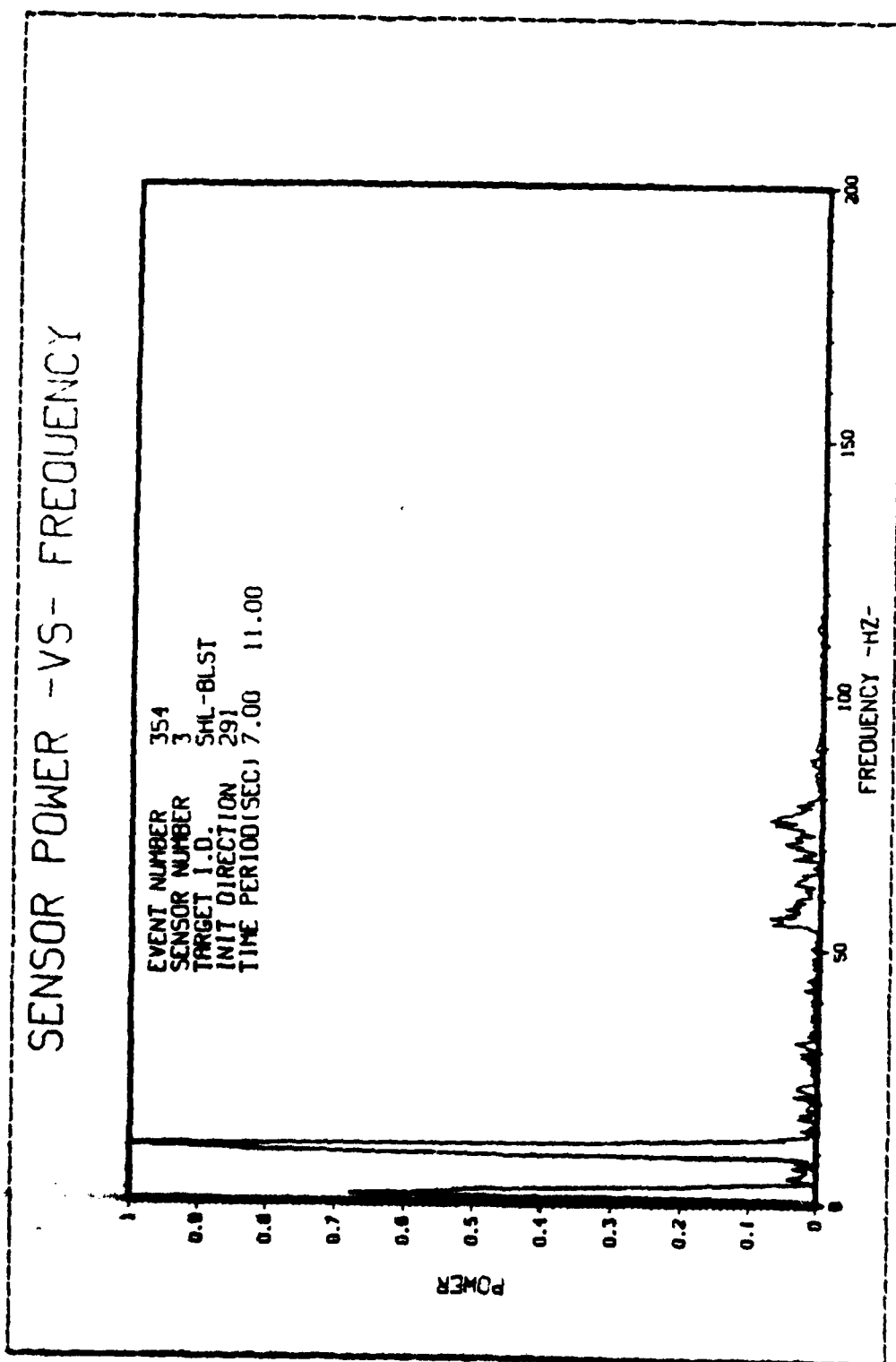


Figure 6.62 Frequency Response for Event 354 (7 - 11sec)

LEAST MEAN SQUARES POLYNOMIAL

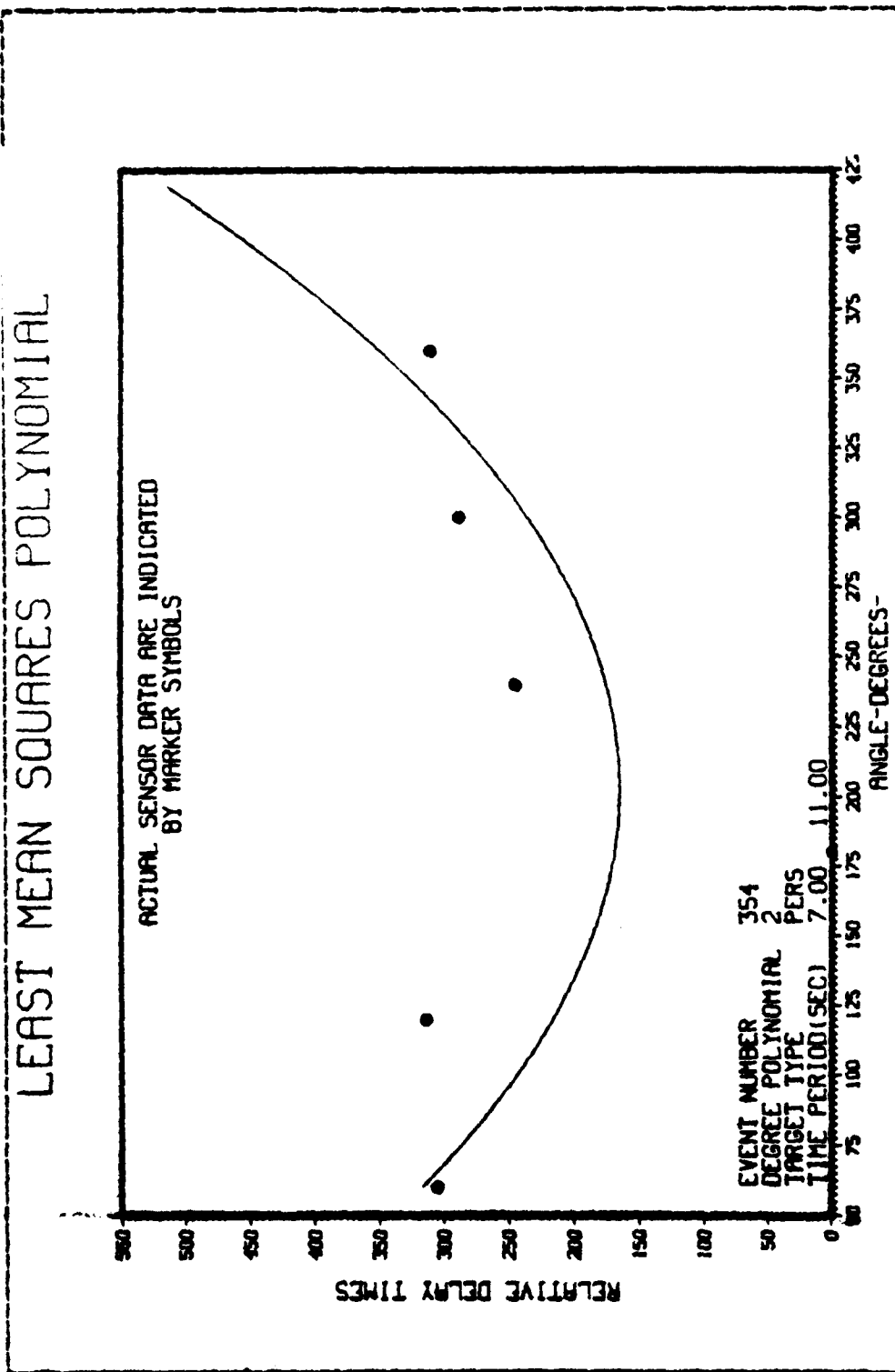


Figure 6.63 LMSP Matched Filter Direction for Event 354 (7 - 11sec)

MULTIPLE TARGET - MATCHED FILTER OUTPUT

EVENT NUMBER	354
TIME PERIOD(SEC)	7.00 11.00
TRACKED VEHICLE	DIRECTION - 291.00
WHEELED VEHICLE	DIRECTION - 291.00
SHELL BLAST	DIRECTION - 291.00
PERSONNEL	DIRECTION - 204.00
SIMULATED TRKD VEHICLE	TARGET FREQUENCY 0.00
AMPLITUDE	0.0000
DIRECTION	0.0000
SIMULATED WHEELED VEHICLE	TARGET FREQUENCY 0.00
AMPLITUDE	0.0000
DIRECTION	0.0000
SIMULATED HELICOPTER	TARGET FREQUENCY 0.00
AMPLITUDE	0.0000
DIRECTION	0.0000
SIMULATED PERSONNEL	TARGET FREQUENCY 0.00
AMPLITUDE	0.0000
DIRECTION	0.0000

Figure 6.64 LMSP Multiple Target Summary Event 354 (7 - 11sec)

VII. CONCLUSIONS AND RECOMMENDATIONS

The ability of the digital matched filter to detect and correctly identify actual discrete single target types was excellent. The adaptive enhancement of the matched filter scheme was found to sharpen the matched filter responses. The complications of multiple targets and continuous targets proved to be less successful. Lack of high signal to noise ratio sample signals for use as filters reduced the ability of the filters to match the signals. As would be expected, the simulated target identification and direction finding operations met with success for both single and multiple targets.

The simulated data validated the usefulness of the least mean squares curve fitting method for target direction finding. This algorithm was noted to be useful in both the relative peak amplitude response method for recoil/blast targets and the matched filter peak position method for all targets. The highest accuracies were found using second degree polynomials. This was due to the reduced noise sensitivity of lower degree polynomials. Conversely, significant errors were found in the directions determined by the phase difference algorithm. These errors were possibly due to round-off error sensitivity in the software/hardware implementation. Experimental data could not be used to effectively crosscheck this finding since most of the experimental data targets were at zero degrees and the phase difference routine seemed to seek zero degrees.

A significant result was the accuracy of the blast/recoil target direction found using only the peak amplitude responses. Directions could be found using only the relative amplitude peak positions for the array sensors and the

least mean squares curve fitting routine. This result indicated a possible counter-fire application using a greatly simplified system. This finding is felt to be significant since artillery type targets are the highest priority target type. The discrete nature of the blast/recoil seismic signals would also allow for ready separation of even a large number of combined hostile and friendly signal sources. This would allow for observerless adjustment of fire onto hostile targets. Artillery and mortar targets may, in fact, be the only target types detectable at the ranges specified for a long range seismic system.

It is recommended that further study be made in to the possible implementation of the least mean square curve fitting of the peak sensor amplitudes responses in a counter-fire system. Digital matched filters for seismic target identification and direction finding may also prove effective after further experimentation with optimum sample signal filters for the various target types. Additionally, the matched filter response may possibly be enhanced by preprocessing the seismic signals through adaptive noise cancellors.

Acoustic vice seismic matched filtering with directional microphones may be useful in target identification. Once the target has been identified, a matched filter/least mean squares based direction finding scheme could then be attempted.

APPENDIX A
USERS MANUAL

The software developed provided for interactive program operation. However, further information must be provided for an initial system setup and correct program operation. To begin with, the seismic data must be transferred from magnetic tape to the IBM 3033's Mass Storage System (MSS). The data must then be transferred to the DISPLA user's disk for analysis of the program. These data transfers may be accomplished using Job Control Language (JCL) procedures.

The magnetic tape volume must first be scanned to determine the storage format of its files. The JCL procedure TSCAN provides this information. A sample TSCAN job follows:

```
//JLJV1677 JOB (3026,0304),'SMC-1677 JOHNSTON',CLASS=F
// EXEC TSCAN,VOLIN=PARK1,DCBIN='DEN=2',UNITIN='3400-4'
// EXEC TSCAN,VOLIN=PARK2,DCBIN='DEN=2',UNITIN='3400-4'
// EXEC TSCAN,VOLIN=PARK3,DCBIN='DEN=2',UNITIN='3400-4'
// EXEC TSCAN,VOLIN=PARK4,DCBIN='DEN=2',UNITIN='3400-4'
//
```

Once the tape scan is completed, the tape files and comments can be transferred to the MSS. Prior to this transfer however, space in the MSS must be made for these files. The procedure IEFBR14 is used for this purpose. A sample job follows:

```
//JLJTN74A JOB (3026,0304),'SMC1677 JOHNSTON',CLASS=A
//*MAIN ORG=HPCGVH1.0131P
// EXEC PGM=IEFBR14
//DD1 DD UNIT=3330V,MSVGP=PUB4B,DISP=(NEW,CATLG),
// SPACE=(CYL,(4,4,3)),DSN=MSS.S3026.P302
```

```

/*
// EXEC PGM=IEFBR14
//DD1 DD UNIT=3330V,MSVGP=PUB4B,DISP=(NEW,CATLG),
//      SPACE=(CYL,(4,4,3)),DSN=MSS.S3026.P314
/*
// EXEC PGM=IEFBR14
//DD1 DD UNIT=3330V,MSVGP=PUB4B,DISP=(NEW,CATLG),
//      SPACE=(CYL,(4,4,3)),DSN=MSS.S3026.P319
/*
//

```

Each event is proceeded by a comment file for that event. These comment files can be identified from the TSCAN output as a file containing only one record. The files containing nine or eighteen records are the sensor data files for the events. Files of nine records in length are events using a circular array of nine sensors designated as a type A33 array. These sensors are all vertical motion-sensing geophones. The eighteen record files contain six sensor groups of three geophones. A circular array is designated type A31 and a linear array is a type A32 array. The three geophones for each group sense either radial, transverse or vertical motion. A JCL routine, using the procedure IEBGENER, transfers the event comment and sensor data. A sample IEBGENER job follows:

```

//JLJ11677 JOB (3026,0304),'JOHNSTON SMC1677',CLASS=P
//*MAIN ORG=NPGVM1.0131P
/*
/* CP/CMS SUBMIT      IEBGENER JCL
/*
/*      COPY TAPE FILES TO MSS.S3026.P319
/*
// EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD UNIT=3400-4,VOL=SER=PARK2,DISP=(,PASS),

```

```

//          LABEL=(1,NL,,IN),
//      DCB=(RECFM=FB,LRECL=64,BLKSIZE=2048,DEN=2,OPTCD=Q)
//SYSUT2 DD  DISP=SHR,DSN=MSS.S3026.COMMENTS(COM319)
//SYSIN DD DUMMY
//*****
//COPY  PROC FILE=,MEM=
//      EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD UNIT=3400-4,VOL=SER=PARK2,DISP=(,PASS),
//          LABEL=(&FILE,NL,,IN),DCB=(RECFM=F,BLKSIZE=2048,DEN=2)
//SYSUT2 DD  DISP=SHR,DSN=MSS.S3026.P319(&MEM)
//SYSIN DD DUMMY
//      PEND
/*
//*****
//      EXEC COPY,FILE=342,MEM=SEN1
//      EXEC COPY,FILE=343,MEM=SEN2
//      EXEC COPY,FILE=344,MEM=SEN3
//      EXEC COPY,FILE=345,MEM=SEN4
//      EXEC COPY,FILE=346,MEM=SEN5
//      EXEC COPY,FILE=347,MEM=SEN6
//      EXEC COPY,FILE=348,MEM=SEN7
//      EXEC COPY,FILE=349,MEM=SEN8
//      EXEC COPY,FILE=350,MEM=SEN9
//      EXEC COPY,FILE=351,MEM=SEN10
//      EXEC COPY,FILE=352,MEM=SEN11
//      EXEC COPY,FILE=353,MEM=SEN12
//      EXEC COPY,FILE=354,MEM=SEN13
//      EXEC COPY,FILE=355,MEM=SEN14
//      EXEC COPY,FILE=356,MEM=SEN15
//      EXEC COPY,FILE=357,MEM=SEN16
//      EXEC COPY,FILE=358,MEM=SEN17
//      EXEC COPY,FILE=359,MEM=SEN18
//      EXEC COPY,FILE=360,MEM=SEN19
/*

```

//

Transfer of the desired event data from MSS to the DISSPLA user's disk may now be performed. A batch fortran job with the appropriate FILEDEFS to denote the various geophone's data is submitted. The RSCS/NET feature is used to send the output of this routine to the user's reader. A sample fortran job for nine sensors follows:

```
//JLJ83026 JOB (3026,0304),'JOHNSTON',CLASS=A
```

```
//*MAIN ORG=NPGVM1.0090P
```

```
// EXEC FOFTXCG,REGION.GO=1024K
```

```
//PORT.SYSIN DD *
```

C

```
LOGICAL*1 INFO1(8),INFO2(8),INFO3(8),INFO4(8),INFO5(8)
```

```
LOGICAL*1 INFO6(8),INFO7(8),INFO8(8),INFO9(8)
```

```
INTEGER*2 DATA1(1020),DATA2(1020),DATA3(1020),DATA4(1020)
```

```
INTEGER*2 DATA5(1020),DATA6(1020),DATA7(1020),DATA8(1020)
```

```
INTEGER*2 DATA9(1020),DATA1(4096)
```

C

```
DO 30 J=1,4
```

```
READ(1,100) INFO1,DATA1
```

```
READ(2,100) INFO2,DATA2
```

```
READ(3,100) INFO3,DATA3
```

```
READ(4,100) INFO4,DATA4
```

```
READ(8,100) INFO5,DATA5
```

```
READ(9,100) INFO6,DATA6
```

```
READ(10,100) INFO7,DATA7
```

```
READ(11,100) INFO8,DATA8
```

```
READ(12,100) INFO9,DATA9
```

```
100 FORMAT(8A1,102(10A2))
```

```
DO 10 I = 10,1020,10
```

```
WRITE(6,101) DATA1(I - 9),DATA1(I - 8),DATA1(I - 7),
```

```
1DATA1(I - 6),DATA1(I - 5),DATA1(I - 4),DATA1(I - 3),
```

```
2DATA1(I - 2),DATA1(I - 1),DATA1(I)
```

```

10  CONTINUE
    DO 20 I = 10,1020,10
      WRITE(6,101) DATA2(I - 9),DATA2(I - 8),DATA2(I - 7),
1DATA2(I - 6),DATA2(I - 5),DATA2(I - 4),DATA2(I - 3),
2DATA2(I - 2),DATA2(I - 1),DATA2(I)
20  CONTINUE
    DO 40 I = 10,1020,10
      WRITE(6,101) DATA3(I - 9),DATA3(I - 8),DATA3(I - 7),
1DATA3(I - 6),DATA3(I - 5),DATA3(I - 4),DATA3(I - 3),
2DATA3(I - 2),DATA3(I - 1),DATA3(I)
40  CONTINUE
    DO 50 I = 10,1020,10
      WRITE(6,101) DATA4(I - 9),DATA4(I - 8),DATA4(I - 7),
1DATA4(I - 6),DATA4(I - 5),DATA4(I - 4),DATA4(I - 3),
2DATA4(I - 2),DATA4(I - 1),DATA4(I)
50  CONTINUE
    DO 60 I = 10,1020,10
      WRITE(6,101) DATA5(I - 9),DATA5(I - 8),DATA5(I - 7),
1DATA5(I - 6),DATA5(I - 5),DATA5(I - 4),DATA5(I - 3),
2DATA5(I - 2),DATA5(I - 1),DATA5(I)
60  CONTINUE
    DO 70 I = 10,1020,10
      WRITE(6,101) DATA6(I - 9),DATA6(I - 8),DATA6(I - 7),
1DATA6(I - 6),DATA6(I - 5),DATA6(I - 4),DATA6(I - 3),
2DATA6(I - 2),DATA6(I - 1),DATA6(I)
70  CONTINUE
    DO 80 I = 10,1020,10
      WRITE(6,101) DATA7(I - 9),DATA7(I - 8),DATA7(I - 7),
1DATA7(I - 6),DATA7(I - 5),DATA7(I - 4),DATA7(I - 3),
2DATA7(I - 2),DATA7(I - 1),DATA7(I)
80  CONTINUE
    DO 90 I = 10,1020,10
      WRITE(6,101) DATA8(I - 9),DATA8(I - 8),DATA8(I - 7),
1DATA8(I - 6),DATA8(I - 5),DATA8(I - 4),DATA8(I - 3),
2DATA8(I - 2),DATA8(I - 1),DATA8(I)

```

```

90    CONTINUE
      DO 91 I = 10,1020,10
        WRITE(6,101) DATA9(I - 9),DATA9(I - 8),DATA9(I - 7),
1DATA9(I - 6),DATA9(I - 5),DATA9(I - 4),DATA9(I - 3),
2DATA9(I - 2),DATA9(I - 1),DATA9(I)
91    CONTINUE
101   FORMAT(I6,I6,I6,I6,I6,I6,I6,I6,I6,I6)
30    CONTINUE
      STOP
      END

```

```

/*
//GO.FT01F001 DD DISP=SHR,DSN=MSS.S3026.P383(SEN1)
//GO.FT02F001 DD DISP=SHR,DSN=MSS.S3026.P383(SEN2)
//GO.FT03F001 DD DISP=SHR,DSN=MSS.S3026.P383(SEN3)
//GO.FT04F001 DD DISP=SHR,DSN=MSS.S3026.P383(SEN4)
//GO.FT08F001 DD DISP=SHR,DSN=MSS.S3026.P383(SEN5)
//GO.FT09F001 DD DISP=SHR,DSN=MSS.S3026.P383(SEN6)
//GO.FT10F001 DD DISP=SHR,DSN=MSS.S3026.P383(SEN7)
//GO.FT11F001 DD DISP=SHR,DSN=MSS.S3026.P383(SEN8)
//GO.FT12F001 DD DISP=SHR,DSN=MSS.S3026.P383(SEN9)
//GO.SYSIN DD *
/*
//

```

A sample job for a six sensor group array follows:

```

//JLJ83026 JOB (3026,0304),'JOHNSTON',CLASS=A
//*MAIN ORG=NPGVM1.0090P
// EXEC PORTXCG,REGION.GO=1024K
//PORT.SYSIN DD *

```

C

```

LOGICAL*1 INFO1(8),INFO2(8),INFO3(8),INFO4(8),INFO5(8)
LOGICAL*1 INFO6(8),INFO7(8),INFO8(8),INFO9(8)
INTEGER*2 DATA1(1020),DATA2(1020),DATA3(1020),DATA4(1020)
INTEGER*2 DATA5(1020),DATA6(1020),DATA7(1020),DATA8(1020)
INTEGER*2 DATA9(1020),DATB1(4096)

```

C

```

DO 30 J=1,4

    READ(1,100) INFO1,DATA1
    READ(2,100) INFO2,DATA2
    READ(3,100) INFO3,DATA3
    READ(4,100) INFO4,DATA4
    READ(8,100) INFO5,DATA5
    READ(9,100) INFO6,DATA6
100  FORMAT(8A1,102(10A2))
    DO 10 I = 10,1020,10
        WRITE(6,101) DATA1(I - 9),DATA1(I - 8),DATA1(I - 7),
1DATA1(I - 6),DATA1(I - 5),DATA1(I - 4),DATA1(I - 3),
2DATA1(I - 2),DATA1(I - 1),DATA1(I)
10  CONTINUE
    DO 20 I = 10,1020,10
        WRITE(6,101) DATA2(I - 9),DATA2(I - 8),DATA2(I - 7),
1DATA2(I - 6),DATA2(I - 5),DATA2(I - 4),DATA2(I - 3),
2DATA2(I - 2),DATA2(I - 1),DATA2(I)
20  CONTINUE
    DO 40 I = 10,1020,10
        WRITE(6,101) DATA3(I - 9),DATA3(I - 8),DATA3(I - 7),
1DATA3(I - 6),DATA3(I - 5),DATA3(I - 4),DATA3(I - 3),
2DATA3(I - 2),DATA3(I - 1),DATA3(I)
40  CONTINUE
    DO 50 I = 10,1020,10
        WRITE(6,101) DATA4(I - 9),DATA4(I - 8),DATA4(I - 7),
1DATA4(I - 6),DATA4(I - 5),DATA4(I - 4),DATA4(I - 3),
2DATA4(I - 2),DATA4(I - 1),DATA4(I)
50  CONTINUE
    DO 60 I = 10,1020,10
        WRITE(6,101) DATA5(I - 9),DATA5(I - 8),DATA5(I - 7),
1DATA5(I - 6),DATA5(I - 5),DATA5(I - 4),DATA5(I - 3),
2DATA5(I - 2),DATA5(I - 1),DATA5(I)
60  CONTINUE

```

```

DO 70 I = 10,1020,10
WRITE(6,101) DATA6(I - 9),DATA5(I - 8),DATA6(I - 7),
1DATA6(I - 6),DATA6(I - 5),DATA6(I - 4),DATA6(I - 3),
2DATA6(I - 2),DATA6(I - 1),DATA6(I)
70    CONTINUE
101    FORMAT(I6,I6,I6,I6,I6,I6,I6,I6,I6,I6)
30    CONTINUE
      STOP
      END

```

```

/*
//GO.FT01F001 DD DISP=SHR,DSN=MSS.S3026.P350(SEN3)
//GO.FT02F001 DD DISP=SHR,DSN=MSS.S3026.P350(SEN6)
//GO.FT03F001 DD DISP=SHR,DSN=MSS.S3026.P350(SEN9)
//GO.FT04F001 DD DISP=SHR,DSN=MSS.S3026.P350(SEN12)
//GO.FT08F001 DD DISP=SHR,DSN=MSS.S3026.P350(SEN15)
//GO.FT09F001 DD DISP=SHR,DSN=MSS.S3026.P350(SEN18)
//GO.SYSIN DD *
/*
//

```

Two files will be returned to user's reader. The first file is the listing and diagnostics file and should be purged. The second file should be named SEN DATA. SEN DATA must now be edited. Delete the first seven lines of the file and issue the command LREC 80 to set the proper file record length.

The interactive program can be run with the complete collection of files listed in the exec MATCH. The MATCH EXEC follows:

```

STRACE OFF
PORTGI MPILTER
GLOBAL TITLIB PORTMOD2 MOD2EEH IMSLSP NONIMSL
FILEDEF 10 TERMINAL
FILEDEF 05 DISK SEN DATA (PERM)
FILEDEF 07 DISK COM DATA (PERM)

```

FI 4 DISK FILTER DATA (RECFM VS PERM
FILEDEF 18 DISK DISSPLA METAFILE T4 (RECFM VBS LRECL
19065 BLOCK 19069
EXEC DISSPLA MFILTER

Entry parameters, such as the number of sensors in the array and the sampling rate, can be found in the NOSC data log for the event under study. All other interactive entries are user selected options or are self explanatory.
[Ref. 12]

APPENDIX B
SAMPLE INTERACTIVE PROGRAM SESSION

match
G1 COMPILER ENTERED
SOURCE ANALYZED
PROGRAM NAME = MAIN
* NO DIAGNOSTICS GENERATED
SOURCE ANALYZED
PROGRAM NAME = ANGLE
* NO DIAGNOSTICS GENERATED
SOURCE ANALYZED
PROGRAM NAME = TIMOUT
* NO DIAGNOSTICS GENERATED
SOURCE ANALYZED
PROGRAM NAME = FREQOT
* NO DIAGNOSTICS GENERATED
SOURCE ANALYZED
PROGRAM NAME = MATCH
* NO DIAGNOSTICS GENERATED
SOURCE ANALYZED
PROGRAM NAME = MYDATA
* NO DIAGNOSTICS GENERATED
SOURCE ANALYZED
PROGRAM NAME = MAXMIN
* NO DIAGNOSTICS GENERATED
SOURCE ANALYZED
PROGRAM NAME = SPCTRM
* NO DIAGNOSTICS GENERATED
SOURCE ANALYZED
PROGRAM NAME = MULTI
* NO DIAGNOSTICS GENERATED
SOURCE ANALYZED

PROGRAM NAME = MLTPLT
* NO DIAGNOSTICS GENERATED
SOURCE ANALYZED
PROGRAM NAME = RMS
* NO DIAGNOSTICS GENERATED
SOURCE ANALYZED
PROGRAM NAME = AVG
* NO DIAGNOSTICS GENERATED
SOURCE ANALYZED
PROGRAM NAME = SIMULT
* NO DIAGNOSTICS GENERATED
SOURCE ANALYZED
PROGRAM NAME = LMS
* NO DIAGNOSTICS GENERATED
SOURCE ANALYZED
PROGRAM NAME = PLT
* NO DIAGNOSTICS GENERATED
SOURCE ANALYZED
PROGRAM NAME = SOLV
* NO DIAGNOSTICS GENERATED
 STATISTICS NO DIAGNOSTICS THIS STEP
DISK 'T' NOT ACCESSED.
B (126) R/O
C (127) R/O
E (128) R/O

... Your Fortran program is now being loaded ...

... execution will soon follow ...

EXECUTION BEGINS...

ENTER EVENT RECORDING NUMBER-13-
383

EVENT NUMBER 383

383. A33. 10 SEPT61 5KN EOD SHOT. SET B. NO DELAY. C141 ON FINAL

AT END OF TAPE."

ENTER SAMPLE RATE IN HERTZ-REAL-
120.

ENTER LOW LOOK ANGLE IN DEGREES-I3-
-100

ENTER HIGH LOOK ANGLE IN DEGREES-I3-
200

ENTER MATCH FILTER THRESHOLD. RANGE OF 0. TO 1.0
.9

ENTER PLOT SCALING FACTOR-REAL-
.85

ENTER NUMBER OF SENSORS IN RING 6 OR 9 ONLY-I1-
9

ENTER SENSOR NUMBER FOR DISPLAY-I1-
4

ENTER NOISE THRESHOLD LEVEL -REAL-
1000.

ENTER DATA WINDOW SIZE FOR DIRECTION FINDING-I4-
0400

FOR COMPRS OUTPUT ENTER -1-. FOR TEK618 ENTER -2-
1

TO CREATE SIMULATED TARGETS ENTER -1- ELSE -2-
1

ENTER THE FOUR SIMULATION FREQUENCIES-REAL-

ENTER FREQUENCY
0

ENTER FREQUENCY
0

ENTER FREQUENCY

0

ENTER FREQUENCY

120.

ENTER AMPLITUDES FOR EACH FREQUENCY-REAL-

ENTER AMPLITUDE

0

ENTER AMPLITUDE

0

ENTER AMPLITUDE

0

ENTER AMPLITUDE

4000.

ENTER TARGET ANGLE FOR EACH FREQUENCY-13-

SIX SENSORS ALLOWABLE ANGLES; 0, 60, 120, 180, 240, 300

FOR NINE SENSORS; 0, 40, 80, 120, 160, 200, 240, 280, 320

ENTER ANGLE

0

ENTER ANGLE

0

ENTER ANGLE

0

ENTER ANGLE

120

TO MODIFY AMPLITUDE OF SIGNAL ABOVE NOISE THRESHOLD

ENTER - 1 -, ELSE ENTER - 2 -

2

ENTER DEGREE OF POLYNOMIAL DESIRED -11-

FOR SIX SENSORS ENTER 2 - 4, FOR NINE ENTER 2 - 7

4

>> USING A PRE-ALLOCATED DATASET FOR UNIT FT17F001.

>> USING A PRE-ALLOCATED DATASET FOR UNIT FT18F001.

CATALOG AS TARGET? IF YES TYPE - 1 -, ELSE - 2 -

2

FOR MATCH FILTER DIREC FINDING ENTER - 1 - ELSE -2-

1

FOR PHASE DELAY METHOD ENTER -1, FOR LMS ENTER -2

2

ENTER DEGREE OF POLYNOMIAL DESIRED -11-

FOR SIX SENSORS ENTER 2 - 4, FOR NINE ENTER 2 - 7

4

TO VIEW OTHER SENSORS ENTER - 1 -

1

ENTER - 2 - TO CONTINUE TO NEXT TIME FRAME

2

PROGRAM LISTING

[illegible]

[illegible]

```

51A(1) V(1260) M(19) I(19) VX(260) I(15)
6MA(1) I(1264) CAT(1024) RMAG(1024) PWR(1024)
7RA(1) I(1264) CAT(1024) RMAG(1024) PWR(1024)
8PLA(1) I(1264) CAT(1024) RMAG(1024) PWR(1024)
2A6(1) I(1024) A(1) I(1024) A(1) I(1024) A(1) I(1024)

```

INTERACTIVE INITIALIZATION OF PROGRAM

```

PROGRAM TESTNO
  ENTER EVENT RECORDING NUMBER-13--

```

```

  FOR I=1 TO 10
    TESTNO = 13
  NEXT I
  ENTER EVENT NUMBER -13-

```

```

  DO WHILE TESTNO = 13
    READ I(13)
    IF I(13) = 13 THEN
      READ I(13)
    ELSE
      READ I(13)
    END IF
  END DO

```

```

  WRITE(6,720) I(13)
  WRITE(6,721) I(13)
  WRITE(6,722) I(13)

```

```

  WRITE(6,723) I(13)
  WRITE(6,724) I(13)
  WRITE(6,725) I(13)

```

```

  ENTER SAMPLE RATE IN HERTZ-REAL--

```

```

  FOR I=1 TO 10
    ENTER LOW LOOK ANGLE IN DEGREES-13--
    ENTER HIGH LOOK ANGLE IN DEGREES-13--
  NEXT I

```

```

  ENTER LOW LOOK ANGLE IN DEGREES-13--
  ENTER HIGH LOOK ANGLE IN DEGREES-13--

```

```

  ENTER MATCH FILTER THRESHOLD. RANGE (F 0. TO 1.0)

```

```

  ENTER PLOT SCALING FACTOR-REAL--

```

```

  ENTER PLOT SCALING FACTOR-REAL--

```

```

PF 0C570
PF 0C580
PF 0C590
PF 0C600
PF 0C610
PF 0C620
PF 0C630
PF 0C640
PF 0C650
PF 0C660
PF 0C670
PF 0C680
PF 0C690
PF 0C700
PF 0C710
PF 0C720
PF 0C730
PF 0C740
PF 0C750
PF 0C760
PF 0C770
PF 0C780
PF 0C790
PF 0C800
PF 0C810
PF 0C820
PF 0C830
PF 0C840
PF 0C850
PF 0C860
PF 0C870
PF 0C880
PF 0C890
PF 0C900
PF 0C910
PF 0C920
PF 0C930
PF 0C940
PF 0C950
PF 0C960
PF 0C970
PF 0C980
PF 0C990
PF 0D000
PF 0D010
PF 0D020
PF 0D030
PF 0D040
PF 0D050
PF 0D060
PF 0D070
PF 0D080
PF 0D090
PF 0D100
PF 0D110
PF 0D120
PF 0D130
PF 0D140
PF 0D150
PF 0D160
PF 0D170
PF 0D180
PF 0D190
PF 0D200
PF 0D210
PF 0D220
PF 0D230
PF 0D240
PF 0D250
PF 0D260
PF 0D270
PF 0D280
PF 0D290
PF 0D300
PF 0D310
PF 0D320
PF 0D330
PF 0D340
PF 0D350
PF 0D360
PF 0D370
PF 0D380
PF 0D390
PF 0D400
PF 0D410
PF 0D420
PF 0D430
PF 0D440

```

CC 79 80 552 560 559 720 727 47 78 81 83 82 19 17 11 22

[illegible]

[illegible]

[illegible]

```

837 WRITE(6,837) 'FOR MATCH FILTER DIREC FINCING ENTER - 1 - ELSE -2-'
      REACT(1,15) = MNY
      IF ((MNY).EQ.(1)) CALL MULTI(RA1,RA2,RA3,RA4,RA5,RA6,RA7,RA8,
1RA9,NTYP,SCALE,DIR,THRES,BCAT,TESTNO,JL,C,IS,TF,V,A,ND,
2NW,NUMSEN,TIME,ITARGET,DIRECT,DIR,DIRC)
      IF ((MNY).NE.(1)) GC TC 454
      WRITE(6,44) 'FOR PHASE DELAY METHOD ENTER -1, FOR LMS ENTER -2-'
44      FORMAT(1,35) INGL
      REACT(1,15) = 1.5
393      DO 666 I21 = 1,5
          DO 666 I21 = 1,5
666      CONTINUE
          IF ((INGL).EQ.(1)) GO TO 101
          DO 678 I1 = 1,5
              NTAR(I1) = ITRGET(1,I1)
              NTAR(I2) = ITRGET(2,I1)
              NTAR(I3) = ITRGET(3,I1)
              NTAR(I4) = ITRGET(4,I1)
              NTAR(I5) = ITRGET(5,I1)
              NTAR(I6) = ITRGET(6,I1)
              IF ((NUMSEN).EQ.(6)) GC TC 646
646          IF ((NTAR(I1)).EQ.(0)) GC TC 678
              NTAR(I1) = ITRGET(1,I1)
              NTAR(I2) = ITRGET(2,I1)
              NTAR(I3) = ITRGET(3,I1)
              NTAR(I4) = ITRGET(4,I1)
              NTAR(I5) = ITRGET(5,I1)
              NG = 1
              IF ((NTAR(I1)).EQ.(0)) GC TC 678
              CALL LMS (NTAR,DIRC,NUMSEN,C,SCALE,TESTNO,TIME,NG,Y,
1VX,X,T,INDEX,NCR,TRIX,B,WKAREA)
              CALL PLT(C,SCALE,TESTNO,NUMSEN,TIME,Y,VX,X,T,INDEX,NOR,DIRC,
1NG)
              DIR(NG) = DIRC
              CCNT INUE
              GO TO 601
C      PHASE DELAY METHOD
C
101      DO 60 J3 = 1,5
          TC = C
          DC TC K3 = 1,NUMSEN
          TCEN(K3) = FLOAT(ITARGET(K3,J3))
          TC = FLCAT(ITARGET(K3,J3)) + TC
          CCNT INUE
          TC = TC/FLOAT(NUMSEN)
          CE = C
          DIVIDE = C
          CO 84C N = 1,NUMSEN

```



```
CONTINUE
CALL MAXMIN(R1,1024,M(1),RMAX,L,RMIN)
CALL MAXMIN(R2,1024,M(2),RMAX,L,RMIN)
CALL MAXMIN(R3,1024,M(3),RMAX,L,RMIN)
CALL MAXMIN(R4,1024,M(4),RMAX,L,RMIN)
CALL MAXMIN(R5,1024,M(5),RMAX,L,RMIN)
CALL MAXMIN(R6,1024,M(6),RMAX,L,RMIN)
IF(I,NUMSEN,EQ,16) GO TO 10
CALL MAXMIN(R7,1024,M(7),RMAX,L,RMIN)
CALL MAXMIN(R8,1024,M(8),RMAX,L,RMIN)
CALL MAXMIN(R9,1024,M(9),RMAX,L,RMIN)
DO 11 I2 = 1,NUMSEN
    NTAR(I2) = M(I2)
CONTINUE
RETURN
END

SUBROUTINE TIMOUT PLCTS AMPLITUDE VERSUS
ENSCR
1C) SUBROUTINE TIMOUT (TIME,K,A,DIRECT,SCALE)
RECAL RZ(1024), TIME(1024)
INTEGER NTYP(6), C,NV,DIRECT,TESTNO
IF((C).EQ.(1)) CALL COMPRS
IF((C).NE.(1)) CALL TEK618
CALL PAGECR(11.0,E.5)
CALL NCRCR
CALL BLCKUP(SCALE)
CALL AREA2C(9,C.6,C)
CALL FRAME
CALL XNAME('TIME-SECONDS-',14)
CALL YNAME('INPLT.',5)
CALL HESSAG('SENSOR NUMBER ',14,2.0,5.7)
CALL INMSG('ORR',4,C,5.5)
CALL INMSG('EVENT NUMBER',12,2.0,5.7)
CALL INMSG('TEST NO',4,0,5.7)
CALL INMSG('TARGET I.D.',11,2.0,5.3)
XPCS = 80
IF(INTYF(1)) NE.(0)) XPCS = XPCS + 1.
IF(INTYF(1)) NE.(0)) XPOS = XPOS + 1.
1 IF(INTYF(2)) NE.(0)) XPCS = XPCS + 1.
1 IF(INTYF(2)) NE.(0)) XPOS = XPOS + 1.
1 IF(INTYF(3)) NE.(0)) XPCS = XPCS + 1.
1 IF(INTYF(3)) NE.(0)) XPOS = XPOS + 1.
```

12

10

11

3

00

U

```

1CALL MESSAG('SHL-BLST',8,XPCS,5.3)
  IF((INTYP(4)).NE.(0)) XPCS = XPCS + 1.2
  IF((INTYP(4)).NE.(0))
1CALL MESSAG('HELICPTR',8,XPOS,5.3)
  IF((INTYP(5)).NE.(0)) XPOS = XPOS + 1.2
  IF((INTYP(5)).NE.(0))
1CALL MESSAG('PERSONNEL',9,XPOS,5.3)
1CALL MESSAG('INIT DIRECT',14,2.0,5.1)
  CALL INTAC(DIRECT,4.0,5.1)
  CALL INTAXS(0.)
  CALL VAXANG(0.)
  CALL XTICKS(10)
  CALL YTICKS(10)
  TS = FLCAT(IFIX(TIME(1024)) + 1.0)
  TF = FLCAT(IFIX(TIME(1024)) + 1.0)
  CALL MAXMIN(RA, 1024,K,RAMAX,L,RAMIN)
  IAMP = 1
DO 337 N1 = 10C,50CC,10C
  IF((IAMP).NE.(1)) GO TO 337
  IF((N1).GE.(ABS(RAMAX))) IAMP = N1
  CONTINUE
  STEP = IAMP/5.0
  CALL GRAF(TS,1,IF,FLCAT(-IAMP),STEP,FLCAT(IAMP))
  CALL CURVE(TIME,RA,1024,0.)
  CALL ENCP(0)
  WRITE(6,70)
  FORMAT(10,'CATALOG AS TARGET? IF YES TYPE - 1 -, ELSE - 2 -')
  READ(1,71)N
  FORMAT(1)
  IF((N).NE.(1)) GO TO 45
DO 44 J = 4,1024,4
  IF((ABS(RA(JJ) - 1)).LT.(.001)) RA(JJ) = 0
  IF((ABS(RA(JJ) - 2)).LT.(.001)) RA(JJ) = 0
  IF((ABS(RA(JJ) - 3)).LT.(.001)) RA(JJ) = 0
  WRITE(4,3)RA(JJ-3),RA(JJ-2),RA(JJ-1),RA(JJ)
  FORMAT(1E)
  CONTINUE
  RETURN
END
SUBROUTINE FREQUOT PLCTS SPECTRAL POWER VERSUS FREQUENCY
SUBROUTINE FREQUOT(FREQ,PWR,DIRECT,SCALE,IORF,NTYP,TESTNC,TS,TF,

```

```

1C) REAL PWR(1024), FREQ(1024)
   INTENT(NTYP(6)), C, DIRECT, IESTING
   IF((C).EQ.(1)) CALL CLMPARS
   IF((C).NE.(1)) CALL TEK618
   CALL PAGERD(11.0, 8.5)
   CALL NCRDR
   CALL BLCDUP(SCALE)
   CALL AFREQ(9.0, 6.0)
   CALL XNAME('FREQUENCY -HZ-', 14)
   CALL YNAME('POWER', 5)
   CALL HEADING('SENSOR NUMBER', 13, 2.0, 5.7)
   CALL MESSAGE('EVENT NUMBER', 14, 2.0, 5.5)
   CALL INTC('TESTING', 4.0, 5.7)
   CALL MESSAGE('SENSOR NUMBER', 14, 2.0, 5.5)
   CALL INTC('CURR', 4.0, 5.5)
   CALL MESSAGE('TARGET', 1.0, 11, 2.0, 5.3)
   XPCS = 2.80
   IF((NTYP(1)).NE.(0)) XPCS = XPCS + 1.2
   IF((NTYP(1)).NE.(0))
   1CALL MESSAGE('TRKD-VEH', 8, XPCS, 5.3)
   IF((NTYP(2)).NE.(0)) XPCS = XPCS + 1.2
   IF((NTYP(2)).NE.(0))
   1CALL MESSAGE('WHLD-VEH', 8, XPCS, 5.3)
   IF((NTYP(3)).NE.(0)) XPCS = XPCS + 1.2
   IF((NTYP(3)).NE.(0))
   1CALL MESSAGE('SHL-BLST', 8, XPCS, 5.3)
   IF((NTYP(4)).NE.(0)) XPCS = XPCS + 1.2
   IF((NTYP(4)).NE.(0))
   1CALL MESSAGE('HELICFTR', 8, XPCS, 5.3)
   IF((NTYP(5)).NE.(0)) XPCS = XPCS + 1.2
   IF((NTYP(5)).NE.(0))
   1CALL MESSAGE('PERSONNEL', 9, XPCS, 5.3)
   CALL MESSAGE('INIT DIRECTION', 14, 2.0, 5.1)
   CALL INTC('DIRECT', 4.0, 5.1)
   CALL MESSAGE('TIME PERIOD(SEC)', 16, 2.0, 4.9)
   CALL REALNCTS(2.3, 9.4, 9)
   CALL REALNCTF(2.4, 7.4, 9)
   XMAX = 1.0
   DO 39 I = 2, 500, 20
   IF((XMAX).NE.(1.0)) GO TO 39
   IF((I).GE.(IFIX(FREQ(512)))) XMAX = FLUAT(I)
   CONTINUE
   XSTEP = XMAX/4.00
   CALL MAXMIN(PWR, 1024, KP, PWRMAX, LP, PWRMIN)
   NS = IFIX(XSTEP)
   IF((XSTEP).GT.(20.0)) NS = IFIX(FLCAT(NS)/10.0)

```

MF106250
 MF106260
 MF106270
 MF106280
 MF106290
 MF106300
 MF106310
 MF106320
 MF106330
 MF106340
 MF106350
 MF106360
 MF106370
 MF106380
 MF106390
 MF106400
 MF106410
 MF106420
 MF106430
 MF106440
 MF106450
 MF106460
 MF106470
 MF106480
 MF106490
 MF106500
 MF106510
 MF106520
 MF106530
 MF106540
 MF106550
 MF106560
 MF106570
 MF106580
 MF106590
 MF106600
 MF106610
 MF106620
 MF106630
 MF106640
 MF106650
 MF106660
 MF106670
 MF106680
 MF106690
 MF106700
 MF106710
 MF106720

NMF	06730
NMF	06740
NMF	06750
NMF	06760
NMF	06770
NMF	06780
NMF	06790
NMF	06800
NMF	06810
NMF	06820
NMF	06830
NMF	06840
NMF	06850
NMF	06860
NMF	06870
NMF	06880
NMF	06890
NMF	06900
NMF	06910
NMF	06920
NMF	06930
NMF	06940
NMF	06950
NMF	06960
NMF	06970
NMF	06980
NMF	06990
NMF	07000
NMF	07010
NMF	07020
NMF	07030
NMF	07040
NMF	07050
NMF	07060
NMF	07070
NMF	07080
NMF	07090
NMF	07100
NMF	07110
NMF	07120
NMF	07130
NMF	07140
NMF	07150
NMF	07160
NMF	07170
NMF	07180
NMF	07190
NMF	07200

```
CALL TAXS (0.)
CALL YAXXS (NS)
CALL YXICKS (IO)
CALL XTICKS (IO)
CALL GRAFVE (XEQ, PWR, XMAX, O, -1, PWRMAX)
CALL CLEVEL ('ALL')
CALL RENDPL (0)
```

337
38

SUBROUTINE MATCH PERFORMS THE MATCHED FILTERING OF THE SELECTED SENSORS CAIATA TARGET CLASSIFICATIONS FOUND ARE RETURNED, AS ARE THEIR RESPECTIVE PEAK SIGNAL DETECTION POSITIONS.

```

SUBROUTINE MATCHTIME
      J=1,C,NP,TS,TF,CATIME)
      1JAL,C,NP,LEG(NT),CATC,I024,IUNIT(I1264),PLATE(2C48),SCAT(I024),
      1RMAT(I11264),THRES,XRAY4(2),YRAY(2),TIME(I024),
      2INTEGR(12),XRAY2(2),XRAY3(2),ICAT(512C)
      2INTEGR(12),NTPV(6),C,LCRECT,JESTNG

```

U

```
DATA XRAY1/2048*.2C48./
DATA XRAY2/4096*.4C56./
DATA XRAY3/6144*.6I44./
DATA XRAY4/8192*.8I52./
DATA XRAY/-1.0/1.0/
DO DD J=1,1024
PLATE(J)=0
PLATE(J+1024)=0.
```

```

33
600
CN=NI*2
NH=NI*3
N2=NI*4
N4=NI*6
N8=NI*10
N10=NI*11
DO600PM=1,N11
RMAT(MM)=0.0
UNIT(MNLE)=MM
DO201K=NT,N10,N2
NMDO202KK=1,NT

```

600

```

202 RMAT(NM + KK) = RMAG(KK)
201 CONTINUE
C CONTINUE
C NTYP(1) = 0
C NTYP(2) = 0
C NTYP(3) = 0
C NTYP(4) = 0
C NTYP(5) = 0
C
C NEW FILTER SAMPLE SIGNALS ARE LOADED AFTER EACH SUCCESSIVE 2048
C ELEMENTS (F THE WORKING ARRAY 'RPAT' ARE CONVOLVED
C
111 DO 400 I = 1, N10
C IF(((1).EQ.(1)).OR.((1).EQ.(N2)).OR.((1).EQ.(N4)).OR.
C 1((1).EQ.(N6)).OR.((1).EQ.(N8)))
C 2GO TO 55
C GO TO 55
C IF(((1).EQ.(1)).OR.((1).EQ.(N2)).OR.((1).EQ.(N4)).OR.
C 98 IF(((1).EQ.(N6)).OR.((1).EQ.(N8))) MARK = 1025
C IF(((1).EQ.(N2)).OR.((1).EQ.(N4)).OR.((1).EQ.(N6)).OR.
C IF(((1).EQ.(N8))) MARK = 3073
C IF(((1).EQ.(N6)).OR.((1).EQ.(N8))) MARK = 4097
C IF(((1).EQ.(N8))) MARK = 5121
C
C FILTER DATA REVERSED
C
C DO 777 LL = 1, NT
C CAT(LL) = TCAT(-LL + MARK)
C CONTINUE
777
C
C EXECUTION TIME MAY BE REDUCED BY SELECTING SMALLER WINDOWS OR
C SEGMENTS AROUND THE PEAK AMPLITUDES OF THE FILTER AND SEISMIC DATA
C
C CALL MAXMINCAT, 1024, LL, CATM, KK, CATMIN)
C DO 10 11 = 1, NT
C IF((LL - NH).LE.(0)) LL = NH
C IF((LL).GT.(1024 - NH)) LL = 1024 - NH
C SCAT(11) = CAT(11 - NH + LL)
C CONTINUE
C CALL RMS(SCAT, NT, CATMAX)
C CALL RMS(RMAG, NT, RAGMAX)
C CALL AVG(SCAT, NT, ACAT)
C RMAT(11) = 0
10
99
C
C REMOVAL OF CC EIAS AND NORMALIZATION OF THE FILTER AND SEISMIC DATA
C
C CC 2CC KN = 1, NT
C IF(((CATMAX).EQ.(0)).OR.((RAGMAX).EQ.(0))) GO TO 200
C

```

```

MF107210
MF107220
MF107230
MF107240
MF107250
MF107260
MF107270
MF107280
MF107290
MF107300
MF107310
MF107320
MF107330
MF107340
MF107350
MF107360
MF107370
MF107380
MF107390
MF107400
MF107410
MF107420
MF107430
MF107440
MF107450
MF107460
MF107470
MF107480
MF107490
MF107500
MF107510
MF107520
MF107530
MF107540
MF107550
MF107560
MF107570
MF107580
MF107590
MF107600
MF107610
MF107620
MF107630
MF107640
MF107650
MF107660
MF107670
MF107680

```

MF107690
MF107700
MF107710
MF107720
MF107730
MF107740
MF107750
MF107760
MF107770
MF107780
MF107790
MF107800
MF107810
MF107820
MF107830
MF107840
MF107850
MF107860
MF107870
MF107880
MF107890
MF107900
MF107910
MF107920
MF107930
MF107940
MF107950
MF107960
MF107970
MF107980
MF107990
MF108000
MF108010
MF108020
MF108030
MF108040
MF108050
MF108060
MF108070
MF108080
MF108090
MF108100
MF108110
MF108120
MF108130
MF108140
MF108150
MF108160

```

C C DISCRETE CONVOLUTION PERFORMED
      RMAT(I) = ((RMAT(I) + KN1)/(ABS(RAGMAX))) *
      1((SCAT(KN) - ACAT)/(CATMAX)) + RMAT(I)
      CONTINUE
      CONTINUE
      CALL MAXMIN(RMAT,N11,L2,KATMAX,K2,RATMIN)
      DO 390 NN = 1,N11
      IF ((RATMAX).EQ.(0)) GO TO 390
      RMAT(NN) = RMAT(NN)/(ABS(RATMAX))
      CONTINUE
C C CLASSIFICATION OF TARGETS BY TARGET TYPE
      DO 302 J1 = N2,N10,N2
      DO 301 I1 = 1,N2
      PLATE(I1) = RPAT(J1 - N2 + 11)
      CONTINUE
      CALL MAXMIN(PLATE,N2,L2,PMAX,LS,PMIN)
      IF ((ABS(PMAX)).GE.(THRES)) NTYP(J1/N2) = LP
      CONTINUE
C C MATCHED FILTER OUTPUT PLCTING
      IF ((IOPF).EQ.(0)) GC TC 391
      IF ((C).EQ.(1)) CALL CCMPS
      IF ((C).NE.(1)) CALL TEK618
      CALL PAGE(11.0,E.5)
      CALL NCERDR
      CALL BLCWUP(SCALE)
      CALL AREA2D(9,C.6,C)
      CALL FRAME
      CALL XNAME('POSITION',8)
      CALL YNAME('AMPLITUDE',8)
      CALL HEADIN('MATCHED FILTER RESPONSE',23.2,C,1)
      CALL MESSAG('TRACKED VEH',11.25,5.8)
      CALL MESSAG('WHEEL VEH',11.2,0.5,8)
      CALL MESSAG('SHELL BLAST',11,4.00,5.8)
      CALL MESSAG('SHELL',4,6.0,5.8)
      CALL MESSAG('PERSONNEL',9,7.55,5.8)
      CALL MESSAG('PERSON NUMBER',13.25,25.25)
      CALL MESSAG('SERIAL',2.5,25)
      CALL MESSAG('EVENT',12.25,45)
      CALL MESSAG('ESTIMATED TIME PERIOD',16.25,C5)
      TS = FLCAT(IFIX(TIME(1)))
      TF = FLCAT(IFIX(TIME(1024)) + 1.1)
      CALL REALND(TS,2.2,4.05)

```



```

C
C
C
C
C
SUBROUTINE MAXMIN FINDS THE MAXIMUM AND MINIMUM VALUES FOR THE ARRAY
PASSED TO IT. IT ALSO RETURNS THE ELEMENT NUMBERS ASSOCIATED WITH
THE MAXIMUM AND MINIMUM VALUES.
      SUBROUTINE MAXMIN(A,N,K,AMAX,L,AMIN)
      REAL A(N)
      AMAX = A(1)
      AMIN = A(1)
      K=1
      L=1
      DO 10 I=2,N
      IF ((A(I)).LE.(AMAX)) GO TO 20
      AMAX=A(I)
      K=I
      IF ((A(I)).GE.(AMIN)) GO TO 10
      AMIN=A(I)
      L=I
      CONTINUE
      IF ((ABS(AMIN)).GT.(ABS(AMAX))) AMAX = AMIN
      RETURN
      END
C
C
C
SUBROUTINE SPCTRM COMPUTES THE NORMALIZED POWER FOR THE COMPLEX ARRAY
PASSED TO IT.
      SUBROUTINE SPCTRM(T,RA,F,A,N,PWR,PWRDB,PHASE)
      REAL T(N),RA(N),F(N),PWR(N),PHASE(N)
      COMPLEX A(N)
      DO 10 K=1,N
      PWR(K)=CABS(A(K))**2
      AR=REAL(A(K))
      AI=AIMAG(A(K))
      IF (AR.EQ.C.O.AND.AI.EC.0.0) GC TO 40
      IF (PHASE(K)=ATAN2(AI,AR)
      GC TO 50
      PHASE(K)=0.0
      CONTINUE
      CONTINUE
      N2=N/2
      CALL MAXMIN(PWR,N,KF,PMAX,LP,PMIN)
      CALL NCRMLIZE POWER SPECTRUM
      DO 60 J=1,N
      PWR(J)=PWR(J)/AES(PMAX)
      CONTINUE
      RETURN
      END
C
C

```

```

MF108650
MF108660
MF108670
MF108680
MF108690
MF108700
MF108710
MF108720
MF108730
MF108740
MF108750
MF108760
MF108770
MF108780
MF108790
MF108800
MF108810
MF108820
MF108830
MF108840
MF108850
MF108860
MF108870
MF108880
MF108890
MF108900
MF108910
MF108920
MF108930
MF108940
MF108950
MF108960
MF108970
MF108980
MF108990
MF109000
MF109010
MF109020
MF109030
MF109040
MF109050
MF109060
MF109070
MF109080
MF109090
MF109100
MF109110
MF109120

```

```
C SUBROUTINE MULT I IS THE MULTIPLE TARGET DIRECTION ROUTINE.  
C DIRECTIONS ARE COMPUTED FOR UP TO FIVE TARGET CLASSES. FOR SINGLE  
C TARGETS, ADAPTIVE MATCHED FILTERING IS PERFORMED ALLOWING FOR  
C IMPROVED ACCURACY.  
C  
C SUPEROUTINE MULTI(RMAG1,RMAG2,RMAG3,RMAG4,RMAG5,RMAG6,RMAG7,  
C 1RMAG8,APAG9,NTRYF,SCALE,TORRT,HRES,ECAJ,TESTAD,JL,C,TS,TF,  
C 2V,A,ND,NUMSEN,C,NTRYF,DIRECT,CIR,DIRC)  
C INTR,ND(4),TESTINC,NLCSEN  
C 1REAL,RMAG1(1024),RMAG2(1024),RMAG3(1024),RMAG4(1024),  
C 1RMAG5(1024),RMAG6(1024),RMAG7(1024),RMAG8(1024),  
C 2BCAT(512C),CCAT(512C),TIME(1024),DIR(5),DIRC,  
C 3X(S),Y(S),THETA(9),T(9),V(4),A(4),RMAG(1024)  
C  
C NSEN = 1  
C NH = NM/2  
C DO 6 I1 = 1,512C  
C CCAT(I1) = BCAT(I1)  
C CONTINUE  
C  
C ADAPTIVE MATCHED FILTERING FOR A SINGLE TARGET IS ACCOMPLISHED  
C BY USING THE SELECTED SENSOR'S SEISMIC DATA AS THE MATCHED FILTER  
C WHEN COMPUTING THE IDENTIFIED TARGET'S DIRECTION.  
C  
C NTR = C  
C DO 888 I14 = 1,4  
C IF((V(I14)).NE.(O)) GC TC 5  
C CONTINUE  
C DO 377 JV = 1,5  
C IF((NTRY(JV)).NE.(O)) NTR = NTR + 1  
C CONTINUE  
C IF((NTRYF(1)),EQ.(O)) GC TU 1  
C IF((NTRY).NE.(1)) GC TC 1  
C NTR = NTR + 1  
C DC 154 I2 = 1,1024  
C CCAT(I2) = RMAG1(I2)  
C CONTINUE  
C IF((NTRYF(2)),EQ.(O)) GC TO 2  
C IF((NTRY).NE.(1)) GC TO 2  
C NTR = 1024 I3 = 1,1024  
C CCAT(I3) = RMAG2(I3)  
C CONTINUE  
C IF((NTRYF(3)),EQ.(O)) GC TO 3  
C IF((NTRY).NE.(1)) GC TO 3  
C NTR = NTR + 1
```

```

DC 103 I4 = 1,1024
CCAT(I4 + 2048) = RMAG3(I4)
CONTINUE
IF((INTR).EQ.(0)) GC TO 4
IF((INTR).NE.(1)) GC TC 4
NTR = NTR + 1
DC 104 I5 = 1,1024
CCAT(I5 + 3072) = RMAG4(I5)
CONTINUE
IF((INTR).EQ.(0)) GC TO 5
IF((INTR).NE.(1)) GC TC 5
DC 105 I6 = 1,1024
CCAT(I6 + 4096) = RMAG5(I6)
CONTINUE
DO 127 J2 = 1,1024
RMAG(J2) = 0.
CONTINUE
II = 0
FOR WINDING OF DATA, FIND THE POSITION OF THE PEAK AMPLITUDE
IF((IORF).EQ.(1)) CALL MAXMIN(RMAG1,1024,LIM,RMAX,LIM,KMIN)
IF((IORF).EQ.(2)) CALL MAXMIN(RMAG2,1024,LIM,RMAX,LIM,KMIN)
IF((IORF).EQ.(3)) CALL MAXMIN(RMAG3,1024,LIM,RMAX,LIM,KMIN)
IF((IORF).EQ.(4)) CALL MAXMIN(RMAG4,1024,LIM,RMAX,LIM,KMIN)
IF((IORF).EQ.(5)) CALL MAXMIN(RMAG5,1024,LIM,RMAX,LIM,KMIN)
IF((IORF).EQ.(6)) CALL MAXMIN(RMAG6,1024,LIM,RMAX,LIM,KMIN)
IF((IORF).EQ.(7)) CALL MAXMIN(RMAG7,1024,LIM,RMAX,LIM,KMIN)
IF((IORF).EQ.(8)) CALL MAXMIN(RMAG8,1024,LIM,RMAX,LIM,KMIN)
IF((IORF).EQ.(9)) CALL MAXMIN(RMAG9,1024,LIM,RMAX,LIM,KMIN)
REPEATED CALLS TO THE MATCHED FILTER ROUTINE ARE MADE WITH WINDOWED
DATA FROM ALL SENSORS.
DO 555 IX = 1,5
CC 666 IV = 1,5
ITRGET(IX,IV) = 0
CONTINUE
IF((LM - NH).LE.(0)) LM = NH
IF((LM).GT.(1024 - NH)) LM = 1024 - NH
DO 111 I7 = 1,NW
CC RMAG(I7) = RMAG1(LM - NH + I7)
CONTINUE
NTYP(1) = 0
NTYP(2) = 0
NTYP(3) = 0
NTYP(4) = 0

```

PF11C090
 PF11C100
 PF11C110
 PF11C120
 PF11C130
 PF11C140
 PF11C150
 PF11C160
 PF11C170
 PF11C180
 PF11C190
 PF11C200
 PF11C210
 PF11C220
 PF11C230
 PF11C240
 PF11C250
 PF11C260
 PF11C270
 PF11C280
 PF11C290
 PF11C300
 PF11C310
 PF11C320
 PF11C330
 PF11C340
 PF11C350
 PF11C360
 PF11C370
 PF11C380
 PF11C390
 PF11C400
 PF11C410
 PF11C420
 PF11C430
 PF11C440
 PF11C450
 PF11C460
 PF11C470
 PF11C480
 PF11C490
 PF11C500
 PF11C510
 PF11C520
 PF11C530
 PF11C540
 PF11C550
 PF11C560

```

NTYP(5) = 0
CALL MATCHE(RMAG,NTYP,SCALE,II,DIRECT,THRES,CCAT,TESTNO,JL,C,NW,
1ITS,TF,TIME)
ITARGET(1,1,1) = NTYP(1)
ITARGET(1,1,2) = NTYP(2)
ITARGET(1,1,3) = NTYP(3)
ITARGET(1,1,4) = NTYP(4)
ITARGET(1,1,5) = NTYP(5)
DO 112 18 = 1,NH
  RMAG(18) = RMAG2(LM - NH + 18)
CONTINUE = 0
NTYP(1) = 0
NTYP(2) = 0
NTYP(3) = 0
NTYP(4) = 0
NTYP(5) = 0
CALL MATCHE(RMAG,NTYP,SCALE,II,DIRECT,THRES,CCAT,TESTNO,JL,C,NW,
1ITS,TF,TIME)
ITARGET(2,2,1) = NTYP(1)
ITARGET(2,2,2) = NTYP(2)
ITARGET(2,2,3) = NTYP(3)
ITARGET(2,2,4) = NTYP(4)
ITARGET(2,2,5) = NTYP(5)
DO 113 19 = 1,NH
  RMAG(19) = RMAG3(LM - NH + 19)
CONTINUE = 0
NTYP(1) = 0
NTYP(2) = 0
NTYP(3) = 0
NTYP(4) = 0
NTYP(5) = 0
CALL MATCHE(RMAG,NTYP,SCALE,II,DIRECT,THRES,CCAT,TESTNO,JL,C,NW,
1ITS,TF,TIME)
ITARGET(3,3,1) = NTYP(1)
ITARGET(3,3,2) = NTYP(2)
ITARGET(3,3,3) = NTYP(3)
ITARGET(3,3,4) = NTYP(4)
ITARGET(3,3,5) = NTYP(5)
DO 114 1A = 1,NH
  RMAG(1A) = RMAG4(LM - NH + 1A)
CONTINUE = 0
NTYP(1) = 0
NTYP(2) = 0
NTYP(3) = 0
NTYP(4) = 0
NTYP(5) = 0
CALL MATCHE(RMAG,NTYP,SCALE,II,DIRECT,THRES,CCAT,TESTNO,JL,C,NW,
1ITS,TF,TIME)

```

112

113

114

MF11C570
MF11C580
MF11C590
MF11C600
MF11C610
MF11C620
MF11C630
MF11C640
MF11C650
MF11C660
MF11C670
MF11C680
MF11C690
MF11C700
MF11C710
MF11C720
MF11C730
MF11C740
MF11C750
MF11C760
MF11C770
MF11C780
MF11C790
MF11C800
MF11C810
MF11C820
MF11C830
MF11C840
MF11C850
MF11C860
MF11C870
MF11C880
MF11C890
MF11C900
MF11C910
MF11C920
MF11C930
MF11C940
MF11C950
MF11C960
MF11C970
MF11C980
MF11C990
MF11C100
MF11C101
MF11C102
MF11C103
MF11C104
MF11C105

```

115      ITRGETI(4,1) = NIYP(1)
      ITRGETI(4,2) = NIYP(2)
      ITRGETI(4,3) = NIYP(3)
      ITRGETI(4,4) = NIYP(4)
      ITRGETI(4,5) = NIYP(5)
      DO 115 I8 = 1, NW
      RMAG(I8) = RMAG5(LM - NH + I8)
      CONTINUE
      NTYP(1) = 0
      NTYP(2) = 0
      NTYP(3) = 0
      NTYP(4) = 0
      NTYP(5) = 0
      CALL MATCH(RMAG,NTYP,SCALE,II,DIRECT,THRES,CCAT,TESTNO,JL,C,NW,
      1TS,TIME)
      ITRGETI(5,1) = NIYP(1)
      ITRGETI(5,2) = NIYP(2)
      ITRGETI(5,3) = NIYP(3)
      ITRGETI(5,4) = NIYP(4)
      ITRGETI(5,5) = NIYP(5)
      DO 116 I16 = 1, NW
      RMAG(I16) = RMAG6(LM - NH + I16)
      CONTINUE
      NTYP(1) = 0
      NTYP(2) = 0
      NTYP(3) = 0
      NTYP(4) = 0
      NTYP(5) = 0
      CALL MATCH(RMAG,NTYP,SCALE,II,DIRECT,THRES,CCAT,TESTNO,JL,C,NW,
      1TS,TIME)
      ITRGETI(6,1) = NIYP(1)
      ITRGETI(6,2) = NIYP(2)
      ITRGETI(6,3) = NIYP(3)
      ITRGETI(6,4) = NIYP(4)
      ITRGETI(6,5) = NIYP(5)
      IF((NUMSEN).EQ.16) GO TO 101
      DO 117 I17 = 1, NW
      RMAG(I17) = RMAG7(LM - NH + I17)
      CONTINUE
      NTYP(1) = 0
      NTYP(2) = 0
      NTYP(3) = 0
      NTYP(4) = 0
      NTYP(5) = 0
      CALL MATCH(RMAG,NTYP,SCALE,II,DIRECT,THRES,CCAT,TESTNO,JL,C,NW,
      1TS,TIME)
      ITRGETI(7,1) = NIYP(1)
      ITRGETI(7,2) = NIYP(2)

```

MF111C50
MF111C60
MF111C70
MF111C80
MF111C90
MF111100
MF111110
MF111120
MF111130
MF111140
MF111150
MF111160
MF111170
MF111180
MF111190
MF111200
MF111210
MF111220
MF111230
MF111240
MF111250
MF111260
MF111270
MF111280
MF111290
MF111300
MF111310
MF111320
MF111330
MF111340
MF111350
MF111360
MF111370
MF111380
MF111390
MF111400
MF111410
MF111420
MF111430
MF111440
MF111450
MF111460
MF111470
MF111480
MF111490
MF111500
MF111510
MF111520

```

118 IIRGET(7,3) = NIYP(3)
IIRGET(7,4) = NIYP(4)
IIRGET(7,5) = NIYP(5)
DO 118 IE = 1,NA
RMAG(IE) = RMAG(LM - NH + IE)
CONTINUE
NTYP(1) = 0
NTYP(2) = 0
NTYP(3) = 0
NTYP(4) = 0
NTYP(5) = 0
CALL MATCH(RMAG,NTYP,SCALE,II,DIRECT,THRES,CCAT,TESTNO,JL,C,NW,
1ITS,TIME)
IIRGET(8,1) = NIYP(1)
IIRGET(8,2) = NIYP(2)
IIRGET(8,3) = NIYP(3)
IIRGET(8,4) = NIYP(4)
IIRGET(8,5) = NIYP(5)
DO 119 IF = 1,NA
RMAG(IF) = RMAG9(LM - NH + IF)
CONTINUE
NTYP(1) = 0
NTYP(2) = 0
NTYP(3) = 0
NTYP(4) = 0
NTYP(5) = 0
CALL MATCH(RMAG,NTYP,SCALE,II,DIRECT,THRES,CCAT,TESTNO,JL,C,NW,
1ITS,TIME)
IIRGET(9,1) = NIYP(1)
IIRGET(9,2) = NIYP(2)
IIRGET(9,3) = NIYP(3)
IIRGET(9,4) = NIYP(4)
IIRGET(9,5) = NIYP(5)
NTYP(1) = 0
NTYP(2) = 0
NTYP(3) = 0
NTYP(4) = 0
NTYP(5) = 0
DC 675 IF = 1,5
DIR(IN) = 955.
CCNTINCE
CCRETURN
END
679
101
C
C
C
MULTIPLE DIRECTIONAL OUTPUT PLACING
SUBROUTINE MULTPLTIC(SCALE,TESTNO,IS,IF,CIR,V,A,WD,IIRGET)
INTEGER C,NC(4),IIRGET(5,5)

```

```

REAL CIF(5),V(4),A(4)
IF((C).EQ.(1)) CALL CCMPRS
IF((C).NE.(1)) CALL TEK618
CALL PACERDR
CALL BLCDRUP(SCALE)
CALL AREA2D(10,C,6,C)
CALL FRAME
CALL HEADSAG(,EVENT NUMBER,35,2.0,1)
CALL INTNG(,TESTING,5.0,5.8)
CALL MESSAG(,TIME,PERIOD(SEC),16,3.0,5.4)
CALL REALNG(ITS,2.4,5.5,4)
CALL REALNG(ITS,2.5,7.5,4)
IF(((ITARGET(1,1)).NE.(0)).AND.((DIR(1,1)).NE.(999.1)))
1 IF(((ITARGET(1,1)).NE.(0)).AND.((DIR(1,1)).NE.(999.1)))
1 IF(((ITARGET(1,2)).NE.(0)).AND.((DIR(2,1)).NE.(999.1)))
1 IF(((ITARGET(1,2)).NE.(0)).AND.((DIR(2,1)).NE.(999.1)))
1 IF(((ITARGET(1,3)).NE.(0)).AND.((DIR(3,1)).NE.(999.1)))
1 IF(((ITARGET(1,3)).NE.(0)).AND.((DIR(3,1)).NE.(999.1)))
1 IF(((ITARGET(1,4)).NE.(0)).AND.((DIR(4,1)).NE.(999.1)))
1 IF(((ITARGET(1,4)).NE.(0)).AND.((DIR(4,1)).NE.(999.1)))
1 IF(((ITARGET(1,5)).NE.(0)).AND.((DIR(5,1)).NE.(999.1)))
1 IF(((ITARGET(1,5)).NE.(0)).AND.((DIR(5,1)).NE.(999.1)))
1 CALL MESSAG(,SIMULATED TRK VEHICLE TARGET FREQUENCY,35,3.0,2.8)
CALL REALNG(ITS,2.6,0.2,8)
CALL REALNG(ITS,2.7,0.2,8)
CALL REALNG(ITS,2.8,0.2,8)
CALL REALNG(ITS,2.9,0.2,8)
CALL REALNG(ITS,3.0,0.2,8)
CALL REALNG(ITS,3.1,0.2,8)
CALL REALNG(ITS,3.2,0.2,8)
CALL REALNG(ITS,3.3,0.2,8)
CALL REALNG(ITS,3.4,0.2,8)
CALL REALNG(ITS,3.5,0.2,8)
CALL REALNG(ITS,3.6,0.2,8)
CALL REALNG(ITS,3.7,0.2,8)
CALL REALNG(ITS,3.8,0.2,8)
CALL REALNG(ITS,3.9,0.2,8)
CALL REALNG(ITS,4.0,0.2,8)
CALL REALNG(ITS,4.1,0.2,8)
CALL REALNG(ITS,4.2,0.2,8)
CALL REALNG(ITS,4.3,0.2,8)
CALL REALNG(ITS,4.4,0.2,8)
CALL REALNG(ITS,4.5,0.2,8)
CALL REALNG(ITS,4.6,0.2,8)
CALL REALNG(ITS,4.7,0.2,8)
CALL REALNG(ITS,4.8,0.2,8)
CALL REALNG(ITS,4.9,0.2,8)
CALL REALNG(ITS,5.0,0.2,8)
CALL REALNG(ITS,5.1,0.2,8)
CALL REALNG(ITS,5.2,0.2,8)
CALL REALNG(ITS,5.3,0.2,8)
CALL REALNG(ITS,5.4,0.2,8)
CALL REALNG(ITS,5.5,0.2,8)
CALL REALNG(ITS,5.6,0.2,8)
CALL REALNG(ITS,5.7,0.2,8)
CALL REALNG(ITS,5.8,0.2,8)
CALL REALNG(ITS,5.9,0.2,8)
CALL REALNG(ITS,6.0,0.2,8)
CALL REALNG(ITS,6.1,0.2,8)
CALL REALNG(ITS,6.2,0.2,8)
CALL REALNG(ITS,6.3,0.2,8)
CALL REALNG(ITS,6.4,0.2,8)
CALL REALNG(ITS,6.5,0.2,8)
CALL REALNG(ITS,6.6,0.2,8)
CALL REALNG(ITS,6.7,0.2,8)
CALL REALNG(ITS,6.8,0.2,8)
CALL REALNG(ITS,6.9,0.2,8)
CALL REALNG(ITS,7.0,0.2,8)
CALL REALNG(ITS,7.1,0.2,8)
CALL REALNG(ITS,7.2,0.2,8)
CALL REALNG(ITS,7.3,0.2,8)
CALL REALNG(ITS,7.4,0.2,8)
CALL REALNG(ITS,7.5,0.2,8)
CALL REALNG(ITS,7.6,0.2,8)
CALL REALNG(ITS,7.7,0.2,8)
CALL REALNG(ITS,7.8,0.2,8)
CALL REALNG(ITS,7.9,0.2,8)
CALL REALNG(ITS,8.0,0.2,8)
CALL REALNG(ITS,8.1,0.2,8)
CALL REALNG(ITS,8.2,0.2,8)
CALL REALNG(ITS,8.3,0.2,8)
CALL REALNG(ITS,8.4,0.2,8)
CALL REALNG(ITS,8.5,0.2,8)
CALL REALNG(ITS,8.6,0.2,8)
CALL REALNG(ITS,8.7,0.2,8)
CALL REALNG(ITS,8.8,0.2,8)
CALL REALNG(ITS,8.9,0.2,8)
CALL REALNG(ITS,9.0,0.2,8)
CALL REALNG(ITS,9.1,0.2,8)
CALL REALNG(ITS,9.2,0.2,8)
CALL REALNG(ITS,9.3,0.2,8)
CALL REALNG(ITS,9.4,0.2,8)
CALL REALNG(ITS,9.5,0.2,8)
CALL REALNG(ITS,9.6,0.2,8)
CALL REALNG(ITS,9.7,0.2,8)
CALL REALNG(ITS,9.8,0.2,8)
CALL REALNG(ITS,9.9,0.2,8)
CALL REALNG(ITS,10.0,0.2,8)

```

```

C C C
CALL MESSAG('SIMULATED HELICOPTER TARGET FREQUENCY',37,3.0,1.0)
CALL REALN(3,2,8.0,1.6)
CALL MESSAG('AMPLITUDE',9,4.0,1.4)
CALL MESSAG('DIRECTION',9,4.0,1.2)
CALL REALN(3,4,5.5,1.4)
D = FLOAT(NC(3))
CALL REALN(4,5,5.1,2)
CALL MESSAG('SIMULATED PERSONNEL TARGET FREQUENCY',36,3.0,1.0)
CALL REALN(4,2,8.0,1.0)
CALL MESSAG('AMPLITUDE',9,4.0,0.8)
CALL MESSAG('DIRECTION',9,4.0,0.6)
CALL REALN(4,4,5.5,0.8)
D = FLOAT(NC(4))
CALL REALN(4,5,5.0,6)
CALL ENCLPL(0)
RETURN
END

C C C
SUBROUTINE RMS(COMPUTES THE ROOT MEAN SQUARE VALUE OF THE N-ARRAY
DATA PASSED TO IT)
  SUBROUTINE RMS(A,N,TMAX)
    REAL A(N)
    TMAX = C
    DO 200 I = 1,N
      TMAX = (A(I))**2 + TMAX
    CONTINUE
    TMAX = SQRT(TMAX)
    RETURN
  END

200

C C C
SUBROUTINE AVG(COMPUTES THE AVERAGE VALUE OF THE N-ARRAY DATA PASSED
TO IT)
  SUBROUTINE AVG(A,N,AAVG)
    REAL A(N)
    AAVG = C
    DO 100 I = 1,N
      AAVG = A(I) + AAVG
    CONTINUE
    AAVG = AAVG/FLOAT(N)
    RETURN
  END

100

C C C
SUBROUTINE SIMULATE ALLOWS FOR THE MODIFICATION OF EXPERIMENTAL
DATA TO INCLUDE UP TO FOUR SIMULATED TARGETS, AND MODIFICATION OF
SEISMIC SIGNAL AMPLITUDES ABOVE THE SELECTED NOISE LEVEL.

```

```

MF1122C10
MF1122C20
MF1122C30
MF1122C40
MF1122C50
MF1122C60
MF1122C70
MF1122C80
MF1122C90
MF1122100
MF1122110
MF1122120
MF1122130
MF1122140
MF1122150
MF1122160
MF1122170
MF1122180
MF1122190
MF1122200
MF1122210
MF1122220
MF1122230
MF1122240
MF1122250
MF1122260
MF1122270
MF1122280
MF1122290
MF1122300
MF1122310
MF1122320
MF1122330
MF1122340
MF1122350
MF1122360
MF1122370
MF1122380
MF1122390
MF1122400
MF1122410
MF1122420
MF1122430
MF1122440
MF1122450
MF1122460
MF1122470
MF1122480

```

NF11122490
 NF11122500
 NF11122510
 NF11122520
 NF11122530
 NF11122540
 NF11122550
 NF11122560
 NF11122570
 NF11122580
 NF11122590
 NF11122600
 NF11122610
 NF11122620
 NF11122630
 NF11122640
 NF11122650
 NF11122660
 NF11122670
 NF11122680
 NF11122690
 NF11122700
 NF11122710
 NF11122720
 NF11122730
 NF11122740
 NF11122750
 NF11122760
 NF11122770
 NF11122780
 NF11122790
 NF11122800
 NF11122810
 NF11122820
 NF11122830
 NF11122840
 NF11122850
 NF11122860
 NF11122870
 NF11122880
 NF11122890
 NF11122900
 NF11122910
 NF11122920
 NF11122930
 NF11122940
 NF11122950
 NF11122960
 NF11122970
 NF11122980
 NF11122990
 NF11123000

SUBROUTINE SIMULT(RA1, RA2, RA3, RA4, RA5, RA6, RA7, RA8, RA9,
 1A1, A2, A3, A4, A5, A6, A7, A8, A9,
 2V, A, INC, TIME, JL, P, REDUCE, NUMSEN)
 IN, TEG, RND(4), NUMSEN
 1RA7(1024), RA8(1024), RA2(1024), RA3(1024), RA4(1024), RA5(1024), RA6(1024),
 2V(4), A(4), TIME(1024), P(9,4), REDUCE
 COMPLEX A1(1024), A2(1024), A3(1024), A4(1024), A5(1024),
 1A6(1024), A7(1024), A8(1024), A9(1024)
 WRITE(6,10) *ENTER THE FOUR SIMULATION FREQUENCIES-REAL-*)
 10 FORMAT(10,1) *ENTER FREQUENCY*)
 WRITE(6,11) *ENTER FREQUENCY*)
 11 FORMAT(10,12)V(1)
 READ(6,12)V(2)
 WRITE(6,13)V(2)
 READ(6,14)V(3)
 WRITE(6,15)V(3)
 READ(6,16)V(4)
 WRITE(6,17)V(4)
 12 FORMAT(10,12)V(4)
 WRITE(6,18) *ENTER AMPLITUDES FOR EACH FREQUENCY-REAL-*)
 13 FORMAT(10,14) *ENTER AMPLITUDE*)
 14 FORMAT(10,15)A(1)
 READ(6,16)A(2)
 WRITE(6,17)A(2)
 READ(6,18)A(3)
 WRITE(6,19)A(3)
 READ(6,20)A(4)
 WRITE(6,21)A(4)
 15 FORMAT(10,15)A(4)
 WRITE(6,22) *ENTER TARGET ANGLE FOR EACH FREQUENCY-13-*)
 16 FORMAT(10,17) *ENTER ANGLE*)
 18 FORMAT(10,17) *SIX SENSORS ALLOWABLE ANGLES:0,60,120,180,240,300,*)
 19 FORMAT(10,17) *FOR NINE SENSORS:0,40,80,120,160,200,240,280,320,*)
 17 FORMAT(10,17) *ENTER ANGLE*)
 READ(6,25)ND(1)
 WRITE(6,26)ND(1)
 READ(6,27)ND(2)
 WRITE(6,28)ND(2)
 READ(6,29)ND(3)
 WRITE(6,30)ND(3)
 READ(6,31)ND(4)
 WRITE(6,32)ND(4)

[illegible]

[illegible]

```

MF114410
MF114420
MF114430
MF114440
MF114450
MF114460
MF114470
MF114480
MF114490
MF114500
MF114510
MF114520
MF114530
MF114540
MF114550
MF114560
MF114570
MF114580
MF114590
MF114600
MF114610
MF114620
MF114630
MF114640
MF114650
MF114660
MF114670
MF114680
MF114690
MF114700
MF114710
MF114720
MF114730
MF114740
MF114750
MF114760
MF114770
MF114780
MF114790
MF114800
MF114810
MF114820
MF114830
MF114840
MF114850
MF114860
MF114870
MF114880

```

```

98      IF(AC(I)).NE.(320)) GO TO 41
        P(1:I) = -21.3
        P(2:I) = -76.7
        P(3:I) = -138.7
        P(4:I) = -180.
        P(5:I) = -180.
        P(6:I) = -138.7
        P(7:I) = -76.
        P(8:I) = -21.3
        P(9:I) = 0.

C      ADDITION OF THE SIMULATED SIGNALS TO EACH SENSORS DATA
C41
DO 2 J = 1,1024
  RA1(J) = 1.024
  RA2(J) = RA1(J) + (A(I)*COS(2*PI*V(I))*TIME(J) + PP*P(1,I))
  RA3(J) = RA2(J) + (A(I)*COS(2*PI*V(I))*TIME(J) + PP*P(2,I))
  RA4(J) = RA3(J) + (A(I)*COS(2*PI*V(I))*TIME(J) + PP*P(3,I))
  RA5(J) = RA4(J) + (A(I)*COS(2*PI*V(I))*TIME(J) + PP*P(4,I))
  RA6(J) = RA5(J) + (A(I)*COS(2*PI*V(I))*TIME(J) + PP*P(5,I))
  IF(ALMSEN).EQ.(6)) GO TO 2
  RA7(J) = RA6(J) + (A(I)*COS(2*PI*V(I))*TIME(J) + PP*P(6,I))
  RA8(J) = RA7(J) + (A(I)*COS(2*PI*V(I))*TIME(J) + PP*P(7,I))
  RA9(J) = RA8(J) + (A(I)*COS(2*PI*V(I))*TIME(J) + PP*P(8,I))
CONTINUE
CONTINUE

21000  SUPPRESSION OR ENHANCEMENT OF SEISMIC SIGNALS ABOVE THE NOISE LEVEL
C
DO 3 K = 1,1024
  IF(AES(RA1(K))) .GT. (TNOISE)) RA1(K) = RA1(K)/REDUCE
  IF(AES(RA2(K))) .GT. (TNOISE)) RA2(K) = RA2(K)/REDUCE
  IF(AES(RA3(K))) .GT. (TNOISE)) RA3(K) = RA3(K)/REDUCE
  IF(AES(RA4(K))) .GT. (TNOISE)) RA4(K) = RA4(K)/REDUCE
  IF(AES(RA5(K))) .GT. (TNOISE)) RA5(K) = RA5(K)/REDUCE
  IF(ALMSEN).EQ.(6)) GO TO 3
  IF(AES(RA6(K))) .GT. (TNOISE)) RA6(K) = RA6(K)/REDUCE
  IF(AES(RA7(K))) .GT. (TNOISE)) RA7(K) = RA7(K)/REDUCE
  IF(AES(RA8(K))) .GT. (TNOISE)) RA8(K) = RA8(K)/REDUCE
  IF(AES(RA9(K))) .GT. (TNOISE)) RA9(K) = RA9(K)/REDUCE
CONTINUE
DO 20 KI = 1,1024
  ABC = 0.
  A1(KI) = CMPLX(RA1(KI),ABC)
  A2(KI) = CMPLX(RA2(KI),ABC)
  A3(KI) = CMPLX(RA3(KI),ABC)
  A4(KI) = CMPLX(RA4(KI),ABC)
  A5(KI) = CMPLX(RA5(KI),ABC)
3

```

MF114490
MF114500
MF114510
MF114520
MF114530
MF114540
MF114550
MF114560
MF114570
MF114580
MF114590
MF114600
MF114610
MF114620
MF114630
MF114640
MF114650
MF114660
MF114670
MF114680
MF114690
MF114700
MF114710
MF114720
MF114730
MF114740
MF114750
MF114760
MF114770
MF114780
MF114790
MF114800
MF114810
MF114820
MF114830
MF114840
MF114850
MF114860

```

20      A6(K1) = CMPLX(RA6(K1),AEC)
30      IF((ALMSEN).EQ.(6)) GO TO 20
      A7(K1) = CMPLX(RA7(K1),AEC)
      A8(K1) = CMPLX(RA8(K1),AEC)
      A9(K1) = CMPLX(RA9(K1),AEC)
      CONTINUE
      RETURN
      ENC
C
C      LEAST MEAN SQUARES POLYNOMIAL CURVE FITTING ALGORITHM
C
      SUBROUTINE LMS (NTAR,DIRC,NUMSEN,C,SCALE,TESTINC,TIME,NFRM,Y,
      1 VX,X,T,INDEX,NCR,TRIX(5,9),B(5),WKAREA(108)
      DIMENSION NTRIX(9),T(9),C,NHALF,MIN,NUMSEN,TESTINC,NFRM
      INTEGER NTRIX(9),T(9),X(9),A(9),Y(360),
      REAL P,FMIN,S,XS(16),X(9),Y(360),
      1 VX(360),T(9),TIME(1024),DIRC,SCALE,C
C
C      FIND FIRST SENSOR TO MATCH THE INPUT SIGNAL
C
      MIN = NTRIX(1)
      MARK = 1
      IF((NTRIX(1)).LT.(1)) GO TO 66
      DO 51 I17 = 1,9
      1(I17) = 0
      IF(1(I17) = 0)
      WRITE(6,53) NTRIX(117)
      CONTINUE
      FCFORMAT(17)
      DO 11 I12 = 2,NUMSEN
      IF((NTRIX(12)).GE.(MIN)) GO TO 11
      MIN = NTRIX(12)
      MARK = 12
      CONTINUE
      DO 46 IE = 1,360
      Y(IE) = C
      VX(IE) = 0.
      CONTINUE
      CENTER THE PARABOLA
      IF((MIN).EQ.(0)) GO TO 66
      DO 21 K1 = 1,NUMSEN
      INA = 1
      IF((NUMSEN).EQ.(6)) INA = 0
      N = NUMSEN/2 - MARK + K1 + INA
      IF((N).GT.(NUMSEN)) N = N - NUMSEN
      IF((N).LT.(1)) N = N + NUMSEN

```

```

21      T(N) = NTAR(K1)
      CONTINUE
      DO 41 L = 1, NUMSEN
        TT(L) = T(L) - MIN
      CONTINUE
      DO 10 I = 1, 16
        XS(I) = 0.
      CONTINUE
      LN = NUMSEN - 1
      DO 130 I4 = 1, LN
        IF((NUMSEN).EQ.(6)) X(I4 + 1) = (FLCAT(14*60))/360.
        IF((NUMSEN).EQ.(9)) X(I4 + 1) = (FLCAT(14*40))/320.
      CONTINUE
130      C
      C
      C      LOAD THE MATRICES FOR THE LEAST MEAN SQUARES POLYNOMIAL SOLUTION
      C
      IF (NUMSEN.EQ.6) X(7) = 0
      IF (NUMSEN.EQ.6) X(8) = 0
      IF (NUMSEN.EQ.6) X(9) = 0
      NTC = 2 + NUMSEN - 2
      DO 20 N1 = 1, NTC
        DO 30 K3 = 1, NUMSEN
          XS(N1) = X(K3)*N1 + XS(N1)
        CONTINUE
      CONTINUE
      DO 47 I2A = 1, 9
        CO 45 I3A = 1, 9
        TRIXX(I2A, I3A) = 0.
      CONTINUE
      CONTINUE
      CONTINUE
      TRIX(1, 1) = NUMSEN
      DO 50 N2 = 2, NUMSEN
        TRIX(1, N2) = XS(N2 - 1)
      CONTINUE
      CO 40 I12 = 1, NUMSEN
      TRIXX(I2, I5) = XS(I5)
      TRIXX(I3, I5) = XS(I5) + 1
      TRIXX(I4, I5) = XS(I5) + 2
      TRIXX(I5, I5) = XS(I5) + 3
      TRIXX(I6, I5) = XS(I5) + 4
      IF((ALHSEN).EQ.(6)) GC TC 40
      TRIXX(I7, I5) = XS(I5) + 5
      TRIXX(I8, I5) = XS(I5) + 6
      TRIXX(I9, I5) = XS(I5) + 7
      CONTINUE
      DO 60 I6 = 1, 9

```



```

120      PMIN = F
CCONTINLE = NUMSEN/2
NHALF = NINI + (MARK - INAI)*DIVIDE
DO 77 IB = 1,NUMSEN
    X(IB) = X(IB)*FLOAT(INDEX) + FLOAT(MARK - NUMSEN/2 - INAI)*DIVIDE
CONTINUE
RETURN
END
77
66
CC C
      OUTPUT PLOTTING FOR THE LEAST MEAN SQUARES SUBROUTINE
SUBROUTINE PLT (C,SCALE,TESTNO,NUMSEN,TIME,VY,VXR,BX,TTR,INDEX,
1NOR,DIRC,NFRM)
INTEGER C,TESTNC,NUMSEN,INDEX,NCR,NFRM
REAL VY(INDEX),TTR(NUMSEN),TIME(1024),SCALE
IF((C).EQ.(1)) CALL COMPRS
IF((C).NE.(1)) CALL TEK618
CALL PACER(11.0,E.5)
CALL NCORDER
CALL BLCHUP(SCALE)
CALL AREAZ(9.0,G.6,C)
CALL FRAME
CALL XNAME('ANGLE-DEGREES-',14)
CALL YNAME('RELATIVE DELAY TIMES',20)
CALL HXACIN('LEAST MEAN SQUARES POLYNOMIAL',25,2.0,1)
CALL MESSAGE('EVENT NUMBER',12,.25,.65)
CALL INTNG(TESTNO,2.5,.65)
CALL MESSAGE('DEGREE POLYNOMIAL',17,.25,.45)
CALL INTINC(NOR,2.5,.45)
IF((NFRM).EQ.(0))CALL MESSAGE('INITIAL DIRECTION',17,.25,.25)
IF((NFRM).EQ.(0))CALL MESSAGE(DIRC,2,2.4,.25)
IF((NFRM).NE.(0))CALL MESSAGE('TARGET TYPE',11,.25,.25)
IF((NFRM).EQ.(1))CALL MESSAGE('TRK VEH',7,2.2,1,.25)
IF((NFRM).EQ.(2))CALL MESSAGE('SHLD VEH',8,2.2,1,.25)
IF((NFRM).EQ.(3))CALL MESSAGE('SHL BLST',8,2.2,1,.25)
IF((NFRM).EQ.(4))CALL MESSAGE('HELD',4,2.5,.25)
CALL MESSAGE('PERIOD',16,.25,.05)
S = FLCLAT(FIX(TIME(1)))
F = FLCLAT(FIX(TIME(1024) + 1.))
CALL REALNO(S,2,2.4,.05)
CALL REALNG(F,2,3.2,.05)
CALL MESSAGE('ACTUAL SENSOR DATA ARE INDICATED',32,2.5,5.7)
CALL MESSAGE('BY MARKER SYMBOLS',17,3.3,5.5)
CALL INTAXS

```

```

CALL YAXANG(0.)
CALL XTICKS(10)
CALL YTICKS(10)
TS = VXR(1)
TF = VXR(INDEX)
CALL MAXMIN(Y, INDEX, K, YMAX, L, YMIN)
IF((YMAX).LE.(YMIN)) YMAX = - YMAX
CALL MAXMIN(TTB, NUMSEN, K, TMAX, L, TMIN)
IF((TMAX).LE.(TMIN)) TMAX = - TMIN
IF((TMAX).GT.(YMAX)) YMAX = TMAX
IF((TMIN).LT.(YMIN)) YMIN = TMIN
CALL GRAPT(TS, SCALE, TF, YMIN, SCALE, YMAX)
CALL RESET(MARKER)
CALL CLEVER(VXR, Y, INDEX, 0.)
CALL MARKER(10)
CALL CLEVE(BX, TTB, NUMSEN, -1)
CALL RESET(MARKER)
CALL ENDPL(0)
RETURN
END

```

CC

MATRIX SOLVING ROUTINE

```

SUBROUTINE SOLV(MATRIX, NZ, B)
DIMENSION TRIX(5,9), B(5), WKAREA(108)
CALL LECT2F(TRIX, 1, NZ, 5, B, 0, WKAREA, IER)
RETURN
END

```

```

MF116810
MF116820
MF116830
MF116840
MF116850
MF116860
MF116870
MF116880
MF116890
MF116900
MF116910
MF116920
MF116930
MF116940
MF116950
MF116960
MF116970
MF116980
MF116990
MF117000
MF117010
MF117020
MF117030
MF117040
MF117050
MF117060
MF117070
MF117080

```

LIST OF REFERENCES

1. Naval Ocean Systems Command Technical Report 747, Seismic/Acoustic Field Measurements and Analysis, by R. G. Hock, J. G. Hoffman, A. P. Roone, L. H. Bossert and J. H. Brown, 9 November 1981.
2. Grant, F. S., Interpretation Theory in Applied Geophysics, pp.5 - 11, McGraw Hill, 1965.
3. Orincon Corporation, Signal Processing for Seismic Surveillance, by H. W. Sorensen, pp.1 - 35, July 1981.
4. Shelef, S. C., Phase Difference/Time Delay Trace Functions and their Application to Seismic Data in Arrays, Ph.d. Thesis, Naval Postgraduate School, 1979.
5. Dublin, C. P., Marine Corps Passive Surveillance Requirements, Masters Thesis, Naval Postgraduate School, Monterey, 1980.
6. Skolnik, M. I., Introduction to Radar Systems. pp. 373 - 376, 428 - 431, McGraw-Hill, 1980.
7. Cooper, G. R., and Mc Gillen C. D., Probabilistic Methods of Signal and System Analysis, pp 205 - 210, Holt, Rinehart, and Winston, 1971.
8. Oppenheim, A. V., and Schaffer, R. W., Digital Signal Processing, pp. 111 - 115, Prentice - Hall, 1975.
9. Park, S., Detection, Classification and Locating via Seismic/Acoustic Sensors, Masters Thesis, Naval Postgraduate School, Monterey, 1982.
10. Gerald, C. F., Applied Numerical Analysis, pp.465 - 474, Addison-Wesley, 1980.
11. Salvadori, M. G., and Baron, M. L., Numerical Methods in Engineering, pp.65 - 87, Prentice-Hall, 1961.
12. Naval Postgraduate School Technical Note MVS - 01, Users Guide to MVS at NPS, by June Favorite, pp.1 - 154, July 1983.

INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2. Library, Code 0142 Naval Postgraduate School Monterey, California 93943	2
3. Commanding Officer Naval Ocean System Command Code: 8105 Attn: Dan Leonard San Diego, California 92152	3
4. Professor H. A. Titus, code 62TS Department of Electrical Engineering Naval Postgraduate School Monterey, California 93943	12
5. GTE Systems Attn: Bruce P. Marion P.O. Box 7188 100 Ferguson Drive Mountain View, California 94039	1
6. Lt. Constantinos Manikas 428 Delavina #122 Monterey, California 93940	1
7. Commander JITF, JINRACCS Attn: Captain Robert R. Logan Ft Monmouth, New Jersey 07703	1
8. Ken Duval 250 Polaris San Jose, California 95126	1
9. Captain James L. Johnston USMC 17410 N.E. 36th St. Redmond, Washington 98052	6