REPORT NO. FAA-RD-75-161,II

PULSED ACOUSTIC VORTEX SENSING SYSTEM
Volume II: Studies of Improved PAVSS Processing Techniques

Royal N. Schweiger

Avco Corporation
Systems Division
201 Lowell Street
Wilmington, MA 01887

JUNE 1977

FINAL REPORT

Prepared for
U.S. DEPARTMENT OF TRANSPORTATION
FEDERAL AVIATION ADMINISTRATION
Systems Research and Development Service
Washington DC 20591

Technical Report Documentation Page

1. Report No. FAA-RD-75-161,II	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle PULSED ACOUSTIC VORTEX SENSING SYSTEM Volume II -- Studies of Improved PAVSS Processing Techniques		5. Report Date June 1977
		6. Performing Organization Code
		8. Performing Organization Report No. DOT-TSC-FAA-74-19,II
7. Author(s) Royal N. Schweiger*		
9. Performing Organization Name and Address Avco Corporation Systems Division 201 Lowell Street Wilmington MA 01887		10. Work Unit No. (TRAIS) FA405/R7126
		11. Contract or Grant No. DOT-TSC-620-2
12. Sponsoring Agency Name and Address U. S. Department of Transportation Federal Aviation Administration Systems Research and Development Service Washington DC 20591		13. Type of Report and Period Covered Final Report August 1973 to June 1974
		14. Sponsoring Agency Code
15. Supplementary Notes * Under contract to: U. S. Department of Transportation Transportation Systems Center Kendall Square, Cambridge MA 02142		

16. Abstract

Avco Corporation's Systems Division designed and developed an engineered Pulsed Acoustic Vortex Sensing System (PAVSS). This system is capable of real-time detection, tracking, recording, and graphic display of aircraft trailing vortices.

This volume of the report presents the results of two subcontractor studies directed toward development of improved vortex tracking software techniques for the PAVSS. The volume recommends the incorporation of several improvements in the software. The subcontractor final reports (Scope Electronics, Inc. and Arcon Corporation) are furnished as appendixes to this volume.

Other volumes in this final report are as follows:

Volume I Hardware Design
Volume III PAVSS Operation and Software Documentation
Volume IV PAVSS Program Summary and Recommendations

17. Key Words Pulsed Acoustic Vortex Sensor, Wake Vortex, Trailing Vortex Sensing System, Aircraft Vortex Tracking System	18. Distribution Statement DOCUMENT IS AVAILABLE TO THE U.S. PUBLIC THROUGH THE NATIONAL TECHNICAL INFORMATION SERVICE, SPRINGFIELD, VIRGINIA 22161

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	154	

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

PREFACE

The problems related to aircraft trailing vortices are currently under intensive study for the Federal Aviation Administration (FAA) by the U. S. Department of Transportation. The Transportation Systems Center (TSC) of the U. S. Department of Transportation (DOT) initiated and is carrying out several programs in this area, including programs to develop acoustic systems for detecting, tracking, and measuring the strength of aircraft wake vortices. Avco Corporation's Systems Division (Avco/SD) designed, built, and tested a pulsed acoustic vortex sensing system (PAVSS) under Contract DOT-TSC-620 in support of these DOT/TSC efforts. The software currently used in vortex tracking produces useful, but difficult-to-interpret data tracks. Avco/SD engaged two subcontractors to investigate possible improvements in the area of vortex tracking using pulsed acoustic data. This report presents Avco's recommendations regarding incorporation of various improvements proposed by the subcontractors, and includes the final reports prepared by each of the subcontractors.

This volume of the final report on the PAVSS program describes the approaches made toward improving the software used for vortex tracking. Other aspects of the system are covered in additional volumes (Volumes I, III, and IV).

The work performed under this contract was significantly enhanced by the close cooperation and contributions of Ralph Kodis, David Burnham, and Thomas Sullivan, all of DOT/TSC.

METRIC CONVERSION FACTORS

Approximate Conversions to Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
	LENGTH			
in	inches	2.5	centimeters	cm
ft	feet	30	centimeters	cm
yd	yards	0.9	meters	m
mi	miles	1.6	kilometers	km
	AREA			
in^2	square inches	6.5	square centimeters	cm^2
ft^2	square feet	0.09	square meters	m^2
yd^2	square yards	0.8	square meters	m^2
mi^2	square miles	2.6	square kilometers	km^2
	acres	0.4	hectares	ha
	MASS (weight)			
oz	ounces	28	grams	g
lb	pounds	0.45	kilograms	kg
	short tons (2000 lb)	0.9	tonnes	t
	VOLUME			
tsp	teaspoons	5	milliliters	ml
Tbsp	tablespoons	15	milliliters	ml
fl oz	fluid ounces	30	milliliters	ml
c	cups	0.24	liters	l
pt	pints	0.47	liters	l
qt	quarts	0.95	liters	l
gal	gallons	3.8	liters	l
ft^3	cubic feet	0.03	cubic meters	m^3
yd^3	cubic yards	0.76	cubic meters	m^3
	TEMPERATURE (exact)			
°F	Fahrenheit temperature	5/9 (after subtracting 32)	Celsius temperature	°C

Approximate Conversions from Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
	LENGTH			
mm	millimeters	0.04	inches	in
cm	centimeters	0.4	inches	in
m	meters	3.3	feet	ft
m	meters	1.1	yards	yd
km	kilometers	0.6	miles	mi
	AREA			
cm^2	square centimeters	0.16	square inches	in^2
m^2	square meters	1.2	square yards	yd^2
km^2	square kilometers	0.4	square miles	mi^2
ha	hectares (10,000 m^2)	2.5	acres	
	MASS (weight)			
g	grams	0.035	ounces	oz
kg	kilograms	2.2	pounds	lb
t	tonnes (1000 kg)	1.1	short tons	
	VOLUME			
ml	milliliters	0.03	fluid ounces	fl oz
l	liters	2.1	pints	pt
l	liters	1.06	quarts	qt
l	liters	0.26	gallons	gal
m^3	cubic meters	35	cubic feet	ft^3
m^3	cubic meters	1.3	cubic yards	yd^3
	TEMPERATURE (exact)			
°C	Celsius temperature	9/5 (then add 32)	Fahrenheit temperature	°F

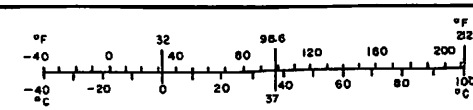


TABLE OF CONTENTS

1. INTRODUCTION

This introduction is presented in two parts. Paragraph 1.1 serves as an introduction to the complete final report; Paragraph 1.2 serves as an introduction to this volume (Volume II) of that report.

1.1 INTRODUCTION TO THE FINAL REPORT

Trailing vortices from heavy jet aircraft represent a currently undefined hazard, particularly during landing and takeoff operations. Considerations of safety and the need to optimize airport operation make it essential to acquire positive information about the presence and locations of vortices generated by heavy aircraft.

The feasibility of using multi-static pulsed acoustic radar to detect and track wake vortices has been demonstrated by the Department of Transportation's Transportation Systems Center (DOT/TSC) in tests at Logan International Airport, Boston, Mass.; at John F. Kennedy International Airport, New York, N. Y.; and at the National Aviation Facilities Experimental Center (NAFEC), Atlantic City, N. J. The hardware used during these tests consisted of laboratory models. The equipment was not engineered for long-term installation in the field, and was incapable of automatic real-time data processing and display.

This report describes a pulsed acoustic vortex sensing system (PAVSS) development program carried out by Avco Systems Division (Avco/SD) for DOT/TSC under Contract DOT-TSC-620. The goal of this program was to develop, build, and test an engineered wake vortex sensing system consisting of acoustic sensors and associated electronics; to acquire and process the sensed data; and to display this data visually in real time.

The complete final report on this program consists of this volume (Volume II, STUDIES OF IMPROVED PAVSS PROCESSING TECHNIQUES) and three additional volumes, as follows:

Volume I	HARDWARE DESIGN
Volume III	PAVSS OPERATION AND SOFTWARE DOCUMENTATION
Volume IV	PAVSS PROGRAM SUMMARY AND RECOMMENDATIONS

1.2 INTRODUCTION TO VOLUME II, STUDIES OF IMPROVED PAVSS PROCESSING TECHNIQUES

This volume of the Final Report on the Pulsed Acoustic Vortex Sensing System describes two studies carried out in support of that program. These were independent studies, carried out by Scope Electronics, Inc. and the Arcon Corporation under Avco/SD subcontracts, to investigate methods of improving automatic vortex delay tracking techniques. The final reports by these subcontractors are included in this volume as Appendixes A and B, respectively.

a. Background

Avco/SD designed and developed a PAVSS capable of detecting and locating trailing (wake) vortices produced by aircraft during landings. In achieving this capability, the system must track a series of acoustic pulses that have been refracted by a vortex (the vortex return). Such returns are slightly delayed relative to direct (line-of-sight, LOS) acoustic pulses. Avco/SD developed a simple tracking algorithm designed to track both the vortex returns and the LOS returns in the presence of noise. This approach, while meeting the goal of simplicity, presented certain problems related to initial acquisition of a vortex return track, to tracking a vortex return close (in time) to an LOS signal, to bridging gaps in a vortex return track, and to stopping the track upon reaching the end of vortex return data.

Avco/SD, therefore, placed subcontracts with two companies (Scope Electronics, Inc., of Reston, Virginia, and Arcon Corporation, of Wakefield, Mass., Avco/SD Purchase Orders 259700 and 259699, respectively) to conduct independent investigations of tracker development. This was done to take full advantage of the current state-of-the-art expertise of these firms in the area of tracker technology.

Appendixes A and B are the final reports by Scope (Report SEI-7063, dated 10 May 1974) and Arcon (Report R74-2W, dated 18 March 1974).

b. Organization of Volume II

The remaining sections of this volume of the final report cover the areas described below:

Section 2	Discusses the major features of the investigation and approach followed by each subcontractor.

Section 3	Presents Avco/SD's recommendations regarding implementation of certain tracker features.
Appendix A	Final Report by Scope Electronics, Inc.
Appendix B	Final Report by Arcon Corporation.

2. DISCUSSION

This section summarizes the features of the approach that each subcontractor employed. Details regarding their efforts are contained in the appropriate appendixes (Appendix A for Scope, Appendix B for Arcon which are presented essentially as received).

2.1 SCOPE ELECTRONICS, INC.

Scope Electronics, Inc. started with the basic "minimum mean square error" tracker developed by Avco/SD for the PAVSS program and improved it in certain specific areas. The modifications consist essentially of a special starting routine, tighter tracking with a non-symmetric tracking window, line-of-sight masking, and a temporary stop routine. These improvements were implemented and tested (using taped data covering 15 different landings). The results indicated that the modified program showed significant improvement in each of the problem areas.

The principal features of the Scope program are:

1. Tracking of the LOS data prior to aircraft arrival with appropriate guard bands.
2. Use of a delay tracker with:
   a. A start routine that employs a bin approach.
   b. Side track avoidance capability.
   c. Provisions for re-starting when track gaps occur.
   d. An effective stop routine.

Except for the LOS guard band provisions, all of the features proposed by Scope are new and are directly applicable to the PAVSS minicomputer software.

2.2 ARCON CORPORATION

The Arcon Corporation developed an entirely new tracking technique, one based upon use of a fixed-point smoothing algorithm. The algorithm represents an implementation of a Kalman-Bucy filter based upon an idealized model of track dynamics. Due to the fixed-point smoothing algorithm's efficiency, multiple tracking is possible, and its use overcomes the various tracking difficulties that were encountered using Avco's single-track technique. A simplified version of the Arcon method has been programmed and checked out. The program has only been given preliminary tests (with promising results). No formal test results were obtained.

The Arcon Corporation program features:

1. Multiple tracking, using:

    a. Parent and trial tracks.

    b. Track selection based on vortex calculations.

2. A smoothing approach based on the Kalman-Bucy filter.

Since Arcon provided no test runs using these features, their actual value in alleviating the present PAVSS software difficulties could not be determined.

3. RECOMMENDATIONS

Avco/SD recommends that consideration be given to implementing several of the tracker improvements developed in the course of the Scope and Arcon investigations.

Initial test results obtained with the present Avco/SD-built PAVSS and its associated software indicate that the most pressing needs are for: (1) improved start, or delay track acquisition, and (2) better smoothing of the track after it is acquired. Track smoothing is also important from the viewpoint of production of more meaningful and more easily interpreted plot presentations. The features incorporated in the program proposed by Scope appear to satisfy all of these needs.

The bin approach to the starting routine, which is initiated when aircraft noise has subsided, should provide excellent initial acquisition of the delay track. The side-track avoidance, re-start, and stop routines should insure that the best delay track is maintained while the vortex remains in the receiver/transmitter field of view. This should prevent acquisition of ground clutter during acquisition of weak or intermittent delay track data.

Avco/SD strongly recommends that the PAVSS software be modified immediately to implement the features discussed above, thereby substantially improving the system's tracking capability.

The Kalman-Bucy filter approach investigated by Arcon had previously been considered by both Avco/SD and TSC. No test data has been supplied to evaluate the Arcon technique. It may, however, prove desirable to employ it in several test cases to permit its evaluation.

APPENDIX A

Aircraft Wake Vortex Tracker,
Final Technical Report

Scope Electronics, Inc., SEI Reference 7063,
dated 10 May 1974

AIRCRAFT WAKE VORTEX TRACKER

Final Technical Report

SEI Reference 7063 • 10 May 1974

CONTENTS

ILLUSTRATIONS

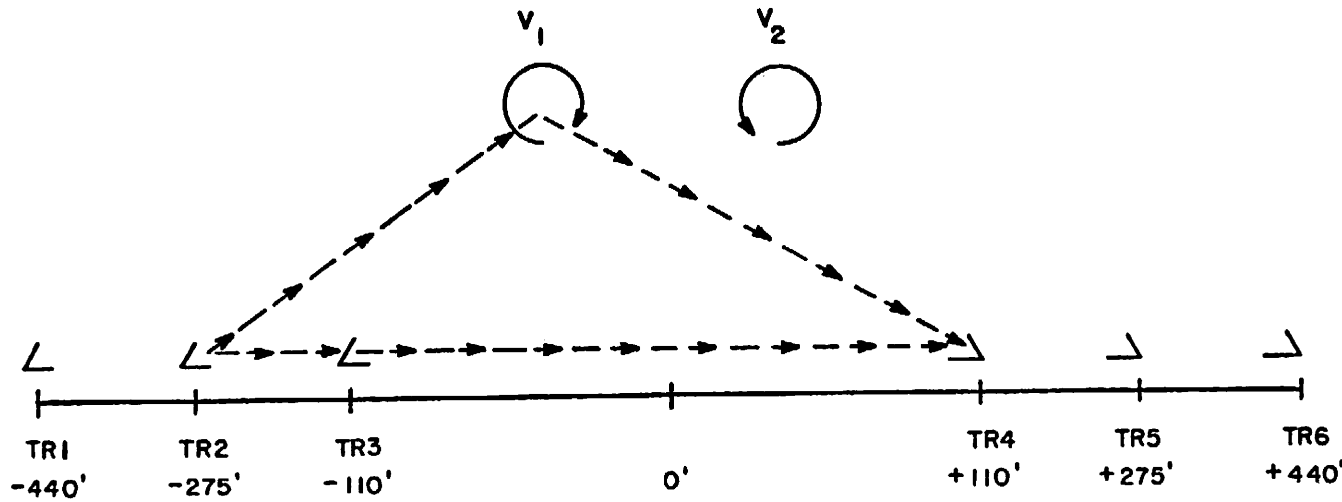


Figure 1. System Configuration

Section 1

INTRODUCTION

This is the final report on the automatic detection and tracking of the pulsed acoustic vortex data after the receiver processing and analog to digital (A/D) conversion. The details of the analog signal processing and A/D conversion prior to the automatic tracking of the vortex from the digital data are dealt with in the AVCO Pulsed Acoustic Vortex Sensing System (PAVSS) Design Evaluation Report (1). Bistatic Acoustic Radar Principles and the experimental results on which the PAVSS is based are well documented in the DOT-TSC Technical Reports of 6/1971, 1/1972 and 12/1972, Ref. (2), (3), (4) respectively.

For the sake of completeness, a brief description of the PAVSS and the front-end receiver processing will be given where applicable to vortex tracking. A typical configuration of the two vortices generated by an aircraft and the six transceivers used to transmit the acoustic pulse and receive the direct line of sight (LOS) return and the scattered delayed return is shown in Figure 1. Vortex V_1 will scatter the pulse from TR1, TR2, and TR3 forwards and vortex V_2 will scatter the pulse from TR4, TR5, and TR6 forwards. Figure 2 shows a typical return for an ideal (noise-free) receiver TR4. The analog processing of the receiver output and A/D conversion

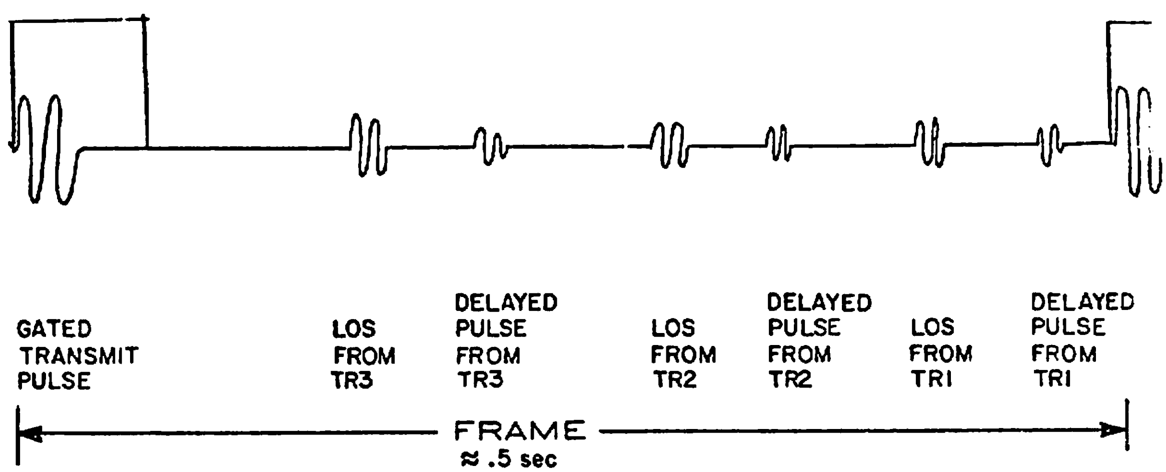


Figure 2. Typical Ideal Receiver Output

are summarized in the following steps. For details, the reader is referred to Ref. (1), pp. 4-35 to 4-44.

1) High-pass prefilter the signal to remove the low-frequency noise without attenuating the signal component.

2) Pass through a synchronous detector which acts as both a narrow-band filter and a detector.

3) Find true derivative and reject negative derivative.

4) Find true integration (of positive derivative only) and reset to zero each time negative derivative is encountered. (The overall effect of 3 and 4 is to restore the signal but without the portions with negative slopes and start all portions with positive slopes from zero baseline.)

5) Compare with preselected thresholds and mark 1 (dot) for crossings with positive slopes. In the ideal receiver, the first crossing would be due to the LOS and the second due to the delayed return from the closest transmitter. (The preselected threshold is chosen on the basis of noise statistics, by computing a weighted sum of the noise mean and noise variation as in (5) pp. 3-79.)

Steps 1 through 5 are shown diagrammatically in Figure 3. This processed and digitized receiver data forms the basis for automatic detection and tracking of the line of sight (LOS) and

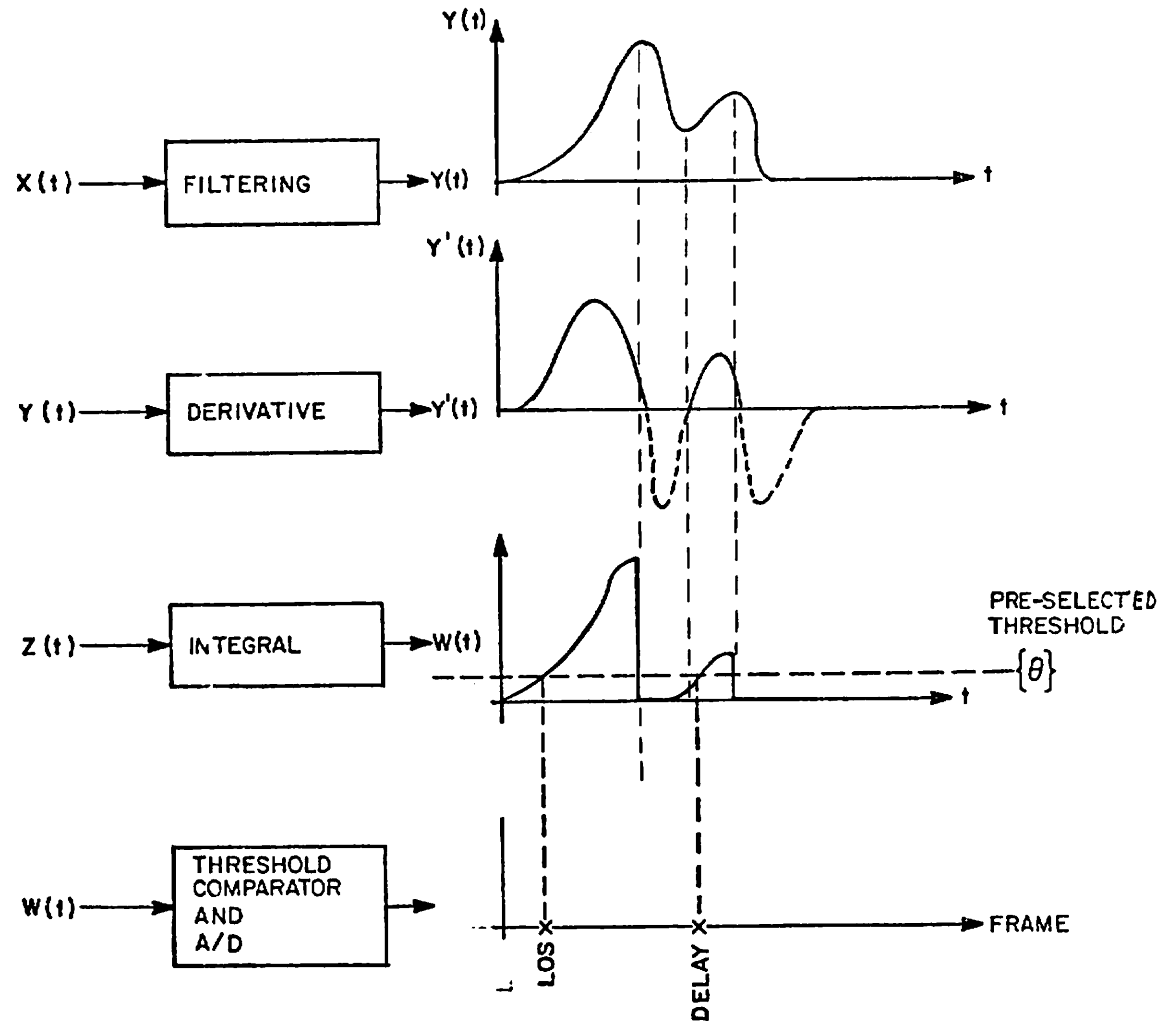


Figure 3. Analog Signal Processing

vortex returns. Knowing the geometry of the transceivers and the speed of sound, the LOS and delay tracker results are then used to compute the vortex position. Redundancy is built into the system because under ideal conditions only 2 receivers (transmitters) and 1 transmitter (receiver) are required to locate the vortex position which is given by the intersection of two ellipses with one common focus at the single transmitter (receiver) and the other foci at the two receivers (transmitters) as shown in Appendix C and D in Ref. (5). A system diagram is shown in Figure 4. This effort was restricted only to the part block diagrammed as "DELAY TRACKER." Magnetic tape consisting of processed and digitized data from 15 aircraft landings were provided for testing the performance of the "DELAY TRACKER".

A few points regarding the analog to digital (A/D) conversion should be noted. The amplitude and pulse width information of the analog return is lost, the threshold level is a constant and time sampling precludes getting more than one dot (1) in a sampling interval (approximately .5 msec.) in a given frame for all receivers. A typical receiver output after A/D conversion is displayed in Figure 5 in the form of a delay (approximately .5 msec. increments) vs. time (approximately .5 sec. increments) plot.

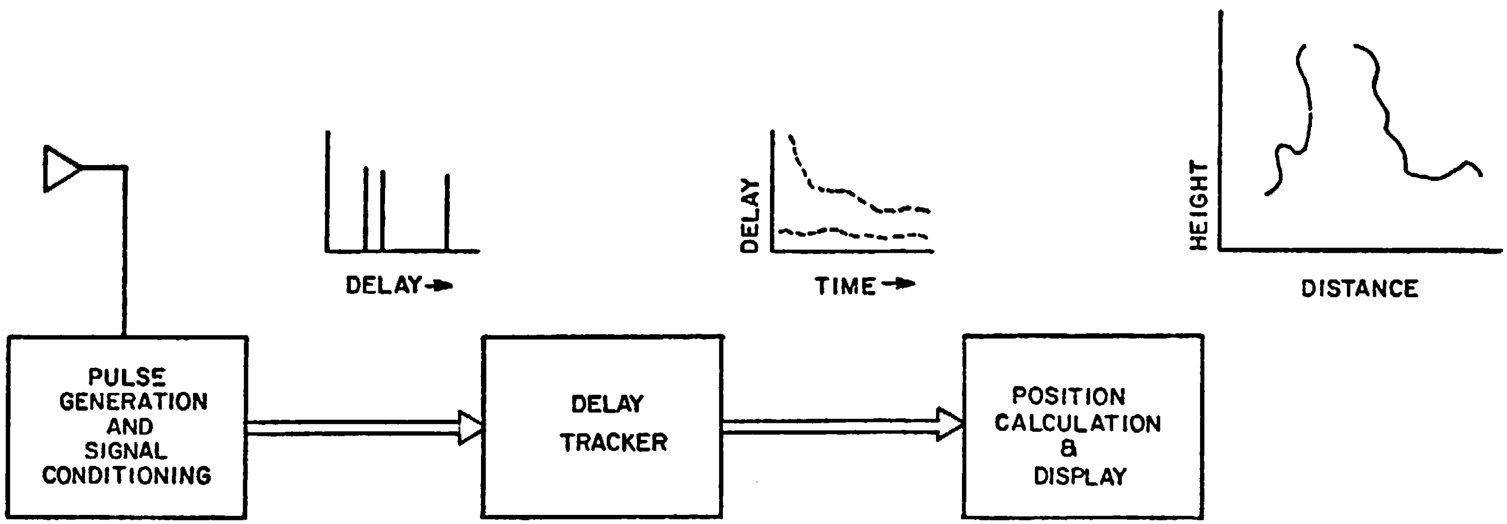


Figure 4. System Block Diagram

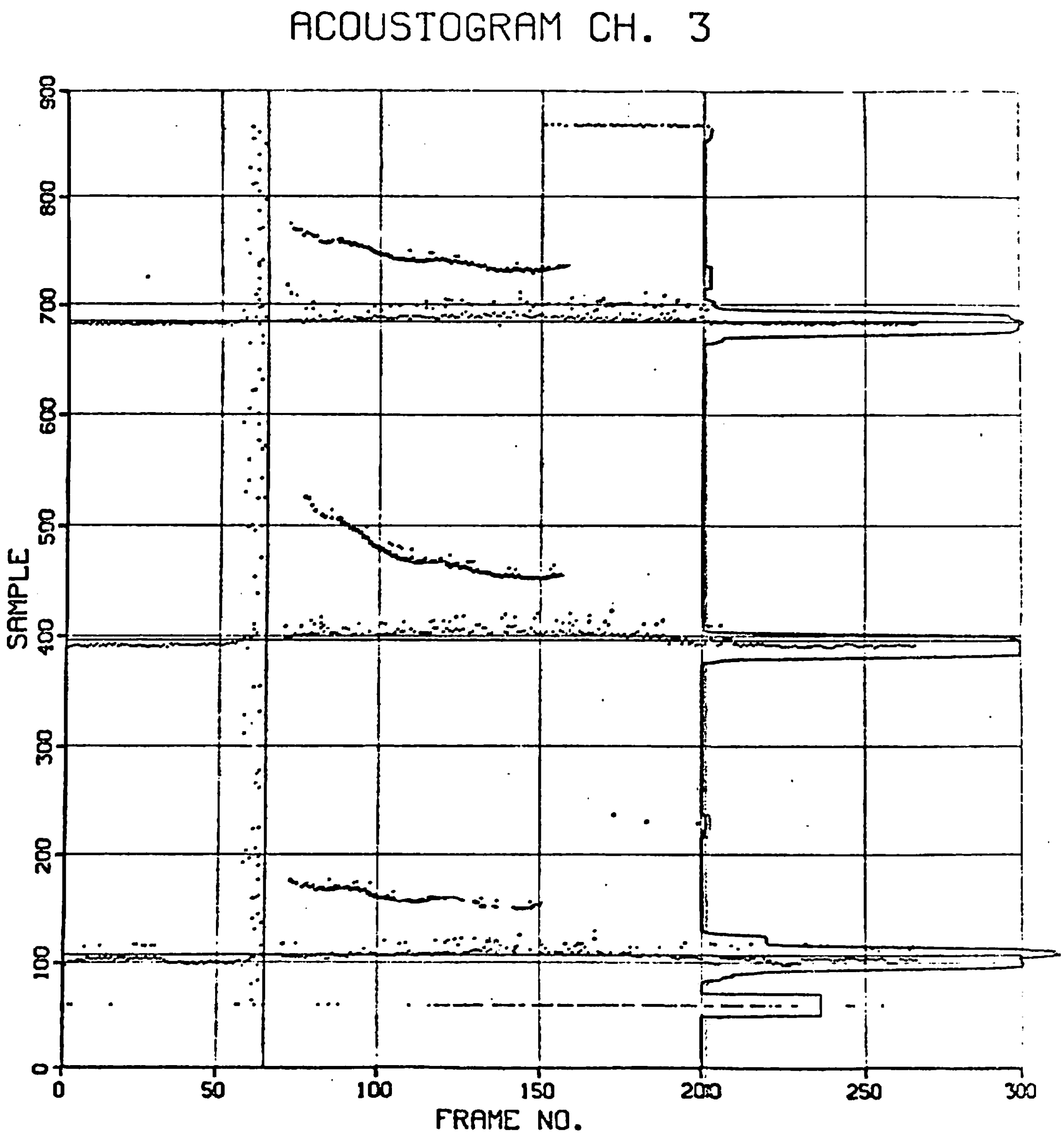


Figure 5. A Typical Receiver Output

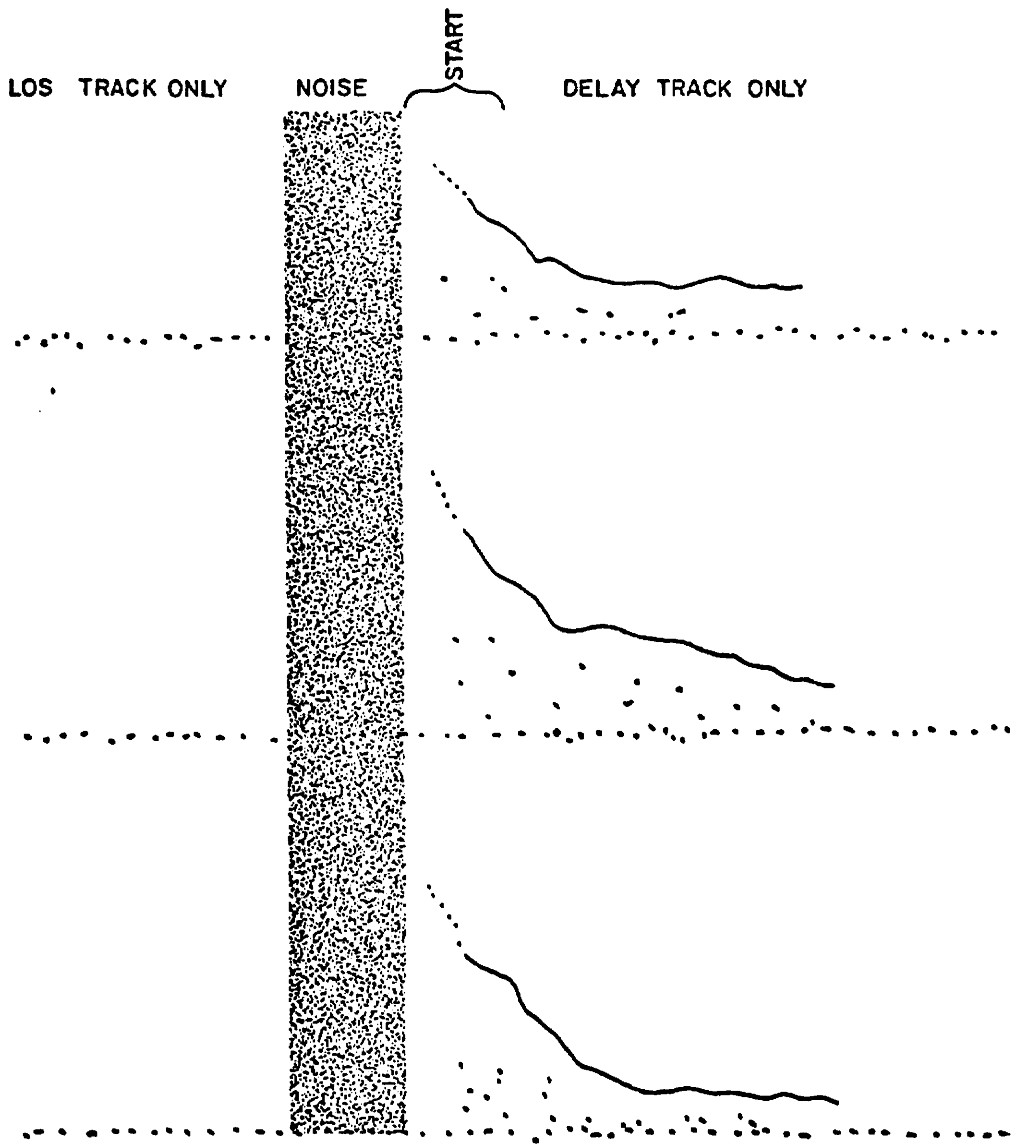


Figure 6. Four Separate Processing Regions

Section 2

THE TRACKER

2.1 The Approach

The receiver output and the analog processing done on it before digitization were briefly discussed in the previous section. A typical receiver output after digitization is shown in Figure 5. as a plot of delay (approximately .5 msec.) vs. time (approximately .5 sec.).

Our approach to tracking was to divide the output into four distinct non-overlapping regions functionally labeled as (1) LOS Track Only, (2) Noise, (3) Start and, (4) Delay Track Only. These regions shown in Figure 6 are categorized by the type of tracking done in them, i.e., LOS tracking only before aircraft arrival in (1), no tracking in aircraft noise, (2) Start routine only for the vortex delay track implemented in (3), and Vortex delay tracking only in (4). These forms of tracking will be discussed below.

2.2 LOS Tracking Only

Before the aircraft arrival, the tracks are relatively noise-free and the regions where the LOS tracks are expected can be closely specified. By restricting the LOS tracking to these tight regions, the problem of tracking clutter can be easily overcome. For each LOS track there are two quantities of interest which must be extracted for later vortex tracking and calculations, i.e., the LOS running average and LOS running variation. Throughout this discussion the term variation will be used to describe a quantity similar to variance but differing in that it uses the absolute value of the error rather than the square of the error as shown in (2.2). This contributes to great savings in computation without degrading the performance.

Since the running average with a time constant of M would require the storage of M values, in general, a slightly different approach yielding a similar answer but requiring one storage element was found to be more appropriate as shown in (2.1) and (2.2). Subscript i represents the ith frame and Y_i the LOS value in ith frame.

LOS Running Average YS_i:

$$YS_i = \frac{M \cdot YS_{i-1} + Y_i}{M + 1} \tag{2.1}$$

LOS Running Variation YSS_i:

$$YSS_i = \frac{M \cdot YSS_{i-1} + |YS_i - Y_i|}{M + 1} \tag{2.2}$$

The LOS tracking is terminated as soon as the noise count in each frame exceeds a preset threshold based on the difference between aircraft noise count and the count in the relatively noise-free region prior to the aircraft arrival. The values of YS_i and YSS_i are accepted as the final values for the LOS and LOS variation respectively for use in the vortex delay tracking and other vortex related calculations. As for the aircraft noise region ideally no tracking at all is done during this interval.

Experimentally it was found that a region of 20 increments (approximately 10 msec.) on either side of the expected LOS was more than sufficient for tracking LOS. Also the time constant M = 40 was implemented. It was further found that the results are not degraded even if the tracking is continued into the aircraft noise region. This is true because of the fact that the time constant M = 40 is much larger than the aircraft noise region. As far as the experimental results are concerned this was the procedure that was followed with the above parameters. They are shown in Figures 17 through 31 with suffix b as

NF = -　　　　LOS = -　　　　LOSV = -

where

NF = Total Number of Frames used for averaging
LOS= Final LOS average in approximately .5 msec increments
LOSV = Final LOS Variation in approximately .5 msec. increments

Table 1

RUNNING MMSE DELAY TRACKER

Given:

1) Frame No. = i

2) Delay = Y_i

Want:

1) Estimate Delay = $\hat{Y}_i$

Need:

Running averages of,

1) Frame No. = $\bar{i}$

2) Delay = $\bar{Y}_i$

3) Variance of Frame = σ_i^2

4) Covariance of Frame, Delay = σ_{iY_i}

Additionally:

5) Track-Density = α_i

6) Delay Error = $\overline{|Y_i - \hat{Y}_i|}$

Answer:

$$A = \frac{\sigma_{iY_i}}{\sigma_i^2}$$

$$B = \bar{Y}_i - \bar{i} \cdot A$$

$$\boxed{\hat{Y}_i = B + A \cdot i}$$

2.3 Running MMSE Delay Tracker

Once the aircraft noise has subsided significantly (ascertained by the noise count in a frame) it is possible to do the delay tracking. Since the LOS mean and variation were finalized prior to the aircraft arrival they will be assumed to remain unchanged during delay tracking. This approach not only reduces the number of computations required during delay tracking, (essential for real-time tracking with a minicomputer), but also keeps the LOS mean and variation free of any vortex interference. This is especially true for low-lying vortices close to the LOS.

The quantities required for the running minimum mean square error (MMSE) delay tracker are shown in Table 1 for clarity. The computations required for the MMSE delay tracker are tabulated in Appendix A. Again, as in the LOS calculations, it is convenient to deviate slightly from the exact running average in order to save on the storage requirements. The exact running average would require a storage of as many quantities as the time constant (e.g. N in the case of α_i), however, the modified approach requires only one storage element per running average as shown in Appendix A.

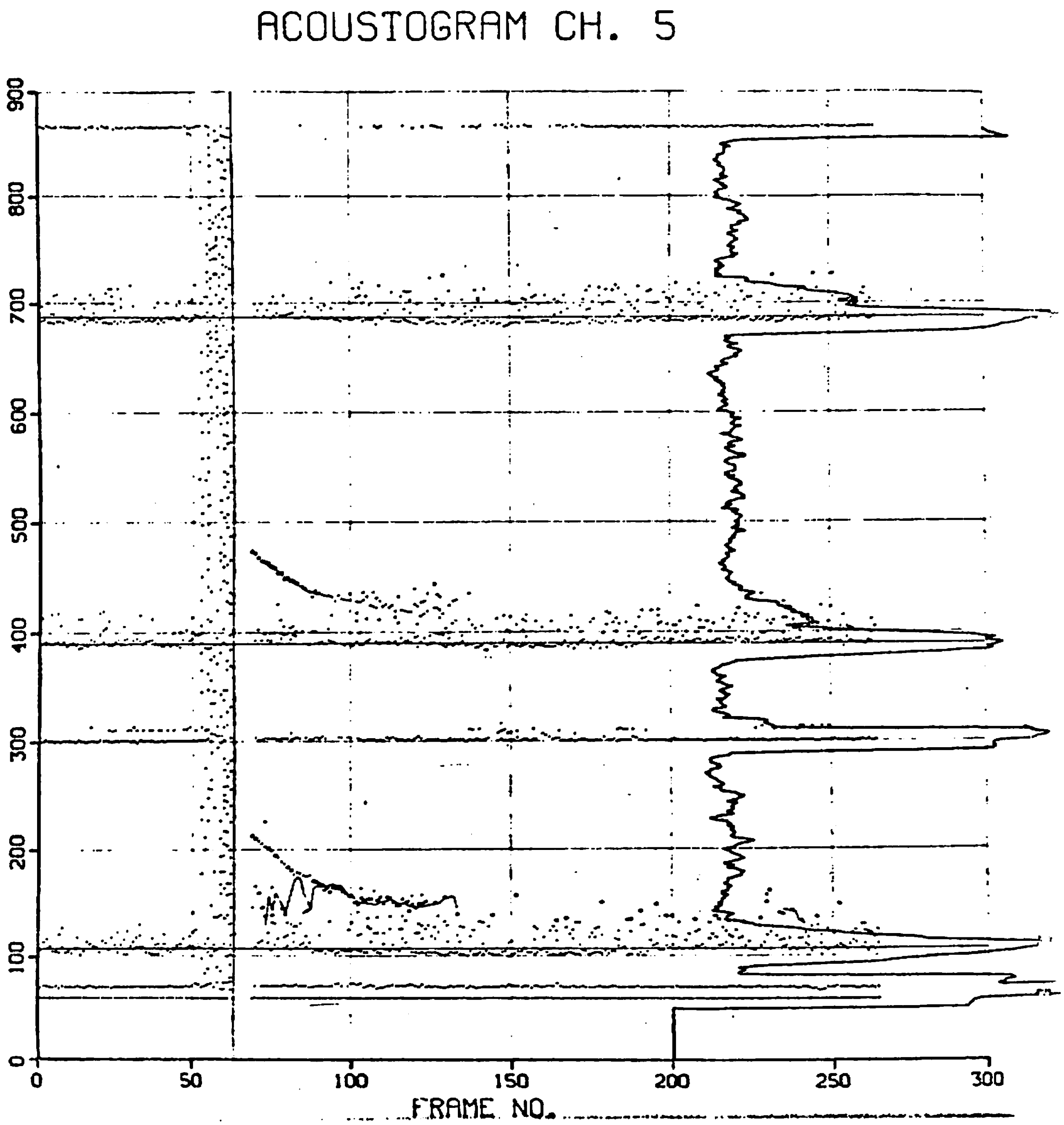


Figure 7. Example of the Start Problem

2.3.1 Basic Running MMSE Tracker Problems

The basic delay tracker as implemented by AVCO was found to suffer from the 5 problems categorized below:

1) Start
2) Side-track
3) LOS Masking
4) Clutter
5) Stop

2.3.1.1 Start

Start refers to the difficulty in acquiring a delay track at the beginning. The twin problem of acquiring the right track and acquiring it as fast as possible was found to be the most important flaw in the basic delay tracker. An example of the start problem. is shown in Figure 7.

The solution to the start problem was to divide the vortex delay into two regions, the first region would be exclusively used for implementing the Start Routine and the following region for delay tracking. The Start Routine was to accomplish the delay track acquisition correctly and as fast as possible with no more computations required than those used for the delay tracking. To

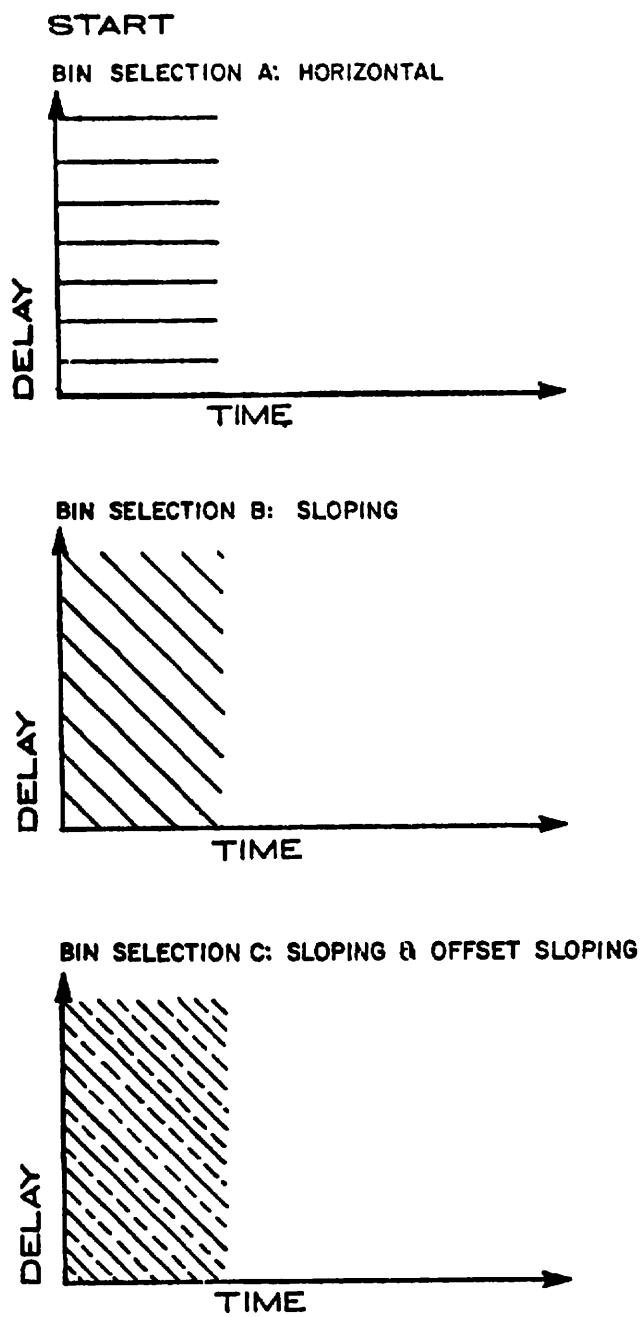


Figure 8. Start Routine Bin Selection

this end the idea selected was to divide the receiver region above the LOS into horizontal, sloping and offset sloping bins as shown in Figure 8.

A score of the number of hits occurring in each bin is kept until one of them crosses a preselected threshold. In case of a draw between the horizontal bin and the sloping bins a preference is given to the sloping bins. When two or more bins with the same slope tie, the uppermost bin is selected.

The delay point within the winning bin and its slope are then used to initiate the delay tracking. All the initial values of the tracking parameters in Appendix A are computed as if the number of hits in the winning bin occurred consecutively and with the slope of the bin as shown in Appendix B. This assures continuity and proper weighting of the past delay data.

As far as the implementation of the Start Routine is concerned, a bin width of 10 increments or about 5 msec was found to be optimum. It was found convenient to ignore the contents of the first 2 horizontal bins and 4 sloping and offset sloping bins immediately above the LOS because of the concentration of noise close to the LOS and the fact that the vortices generally require reasonable amount of time before they descend to such low heights. Because of the ease with which vortices with large delays were acquired and their rapidly descending nature and the available computer storage it was found sufficient to include only 8 horizontal and 16 sloping and 16 offset sloping bins. Also the Start Routine was terminated if the threshold was not crossed after searching through 60 frames (approximately 30 sec). Results of the above implementation are presented in Figures 17 through 21.

All of these parameters may be changed if the situation warrants it. If the storage requirements of the minicomputer necessitate it, the elimination of Offset Sloping Bins would not affect track acquisition significantly, however, it would affect the total time required for track acquisition.

2.3.1.2 Side-track

Side-tracking refers to the problem of getting off the main delay track and tracking a ghost image as shown in Figure 9 or tracking into a false track composed of noise as shown in Figure 10. This difficulty arises due to the fact that when a hit belonging to the correct delay track does not occur, the tracker picks any other closest hit which occurs within the frame and starts tracking along the new path.

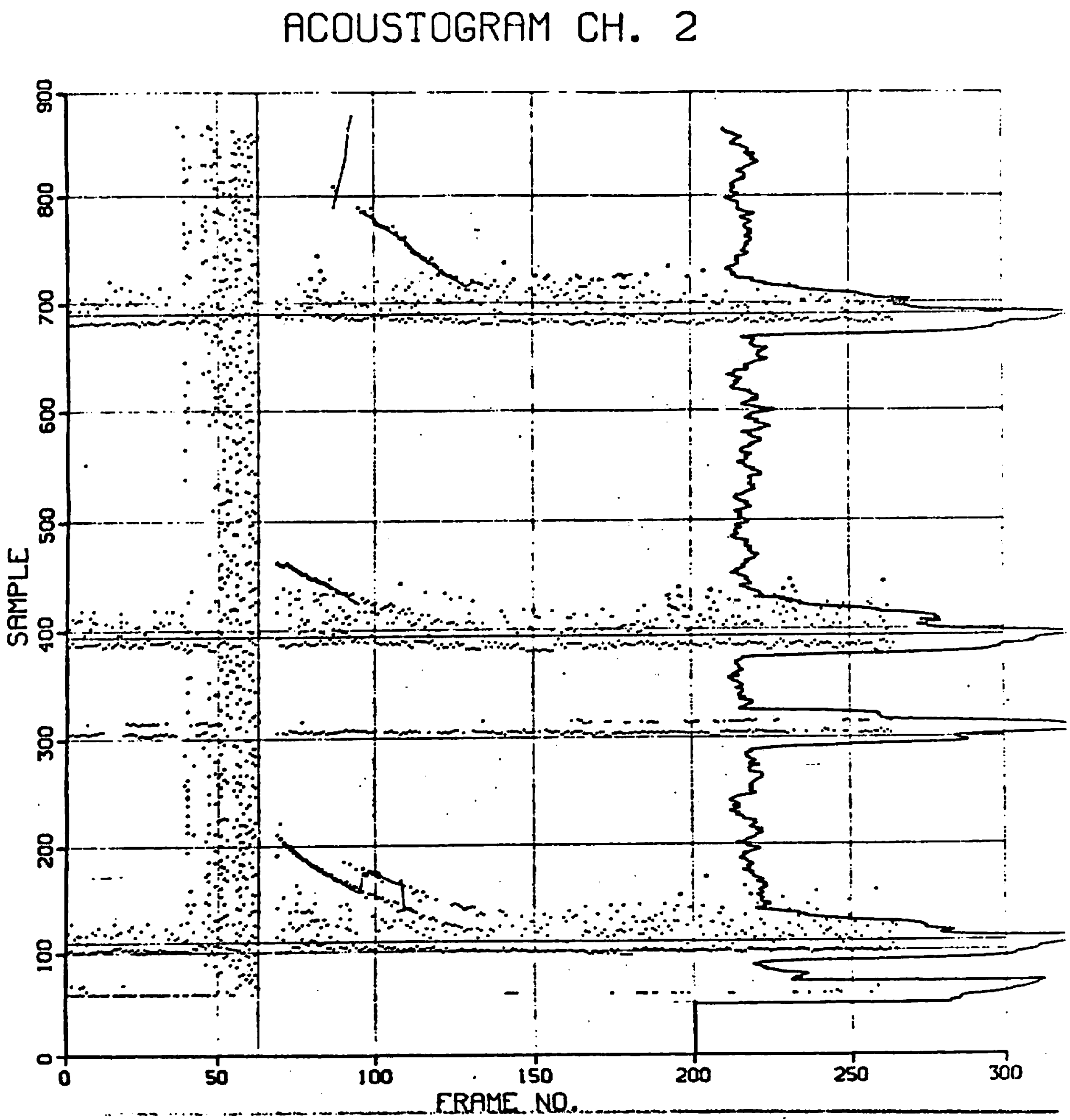


Figure 9. Example of Tracking Ghost Image

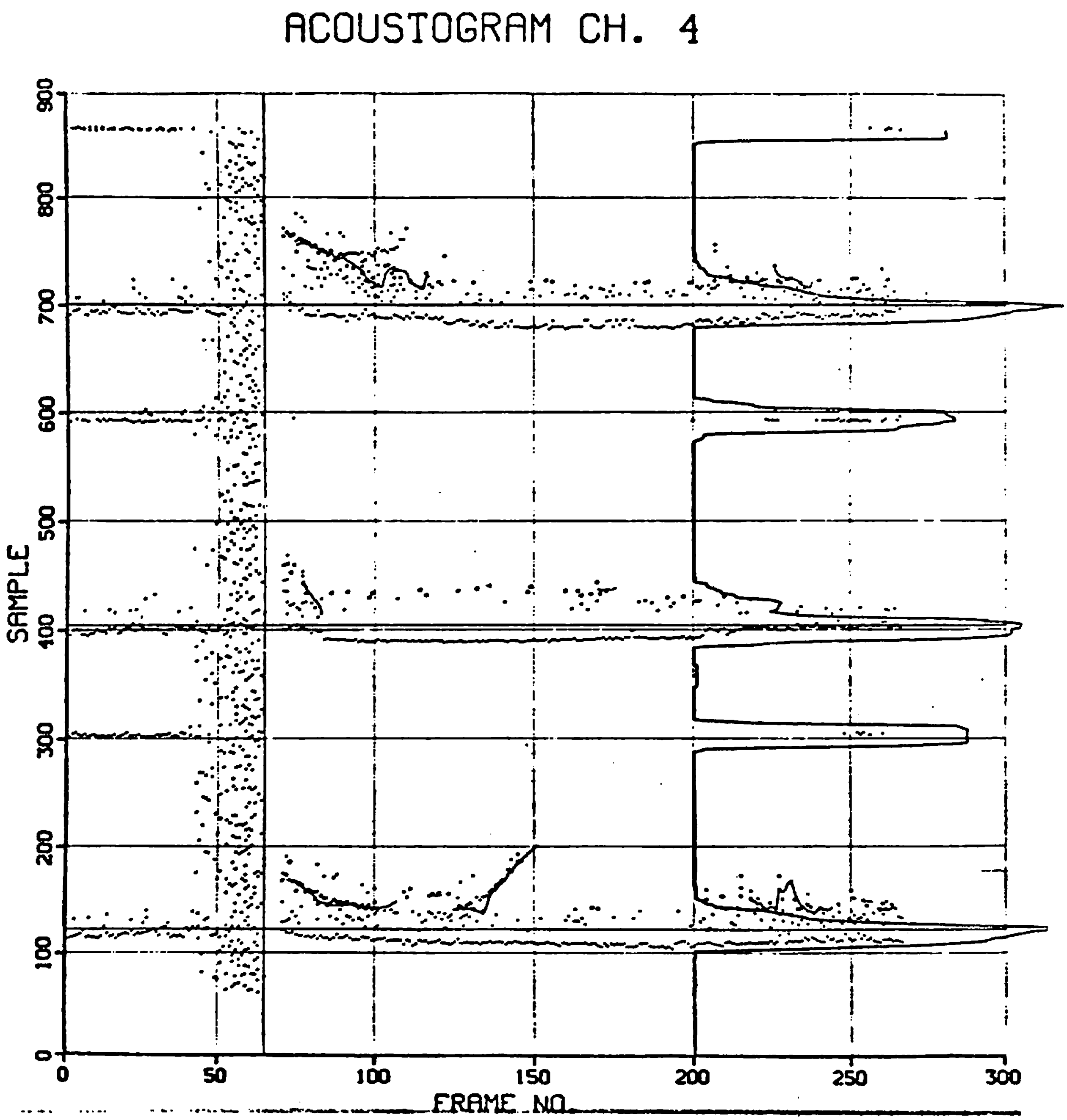


Figure 10. Example of Tracking Noise

The solution to this problem was to reject any hits which occur outside a preselected window around the predicted estimate of the delay. Since the Start Routine enables us to pick the correct track with greater confidence, we can afford to put a tighter bound around the delay tracker. To allow the tracking of upward rising vortices the window is made wider on the upper side as shown in Figure 11. A variable window based on the delay tracker variation was also tried and found to be inappropriate for the type of vortex data provided. The variable window which has a tendency to get very narrow as the delay tracker goes through smooth data quite often loses the track during transition from smooth to rough data and therefore this idea was abandoned.

As far as the implementation is concerned, a window of 7 increments (approximately 3.5 msec) on either side with an extra margin of 7 increments on the upper side was found to give satisfactory results. These values may be changed slightly without affecting the results significantly; however, it was found necessary to retain the asymmetry in the window to track the upward rising vortices without substantially affecting the tracking of normal descending vortices. Some of the results can be seen in Figures 22 through 24.

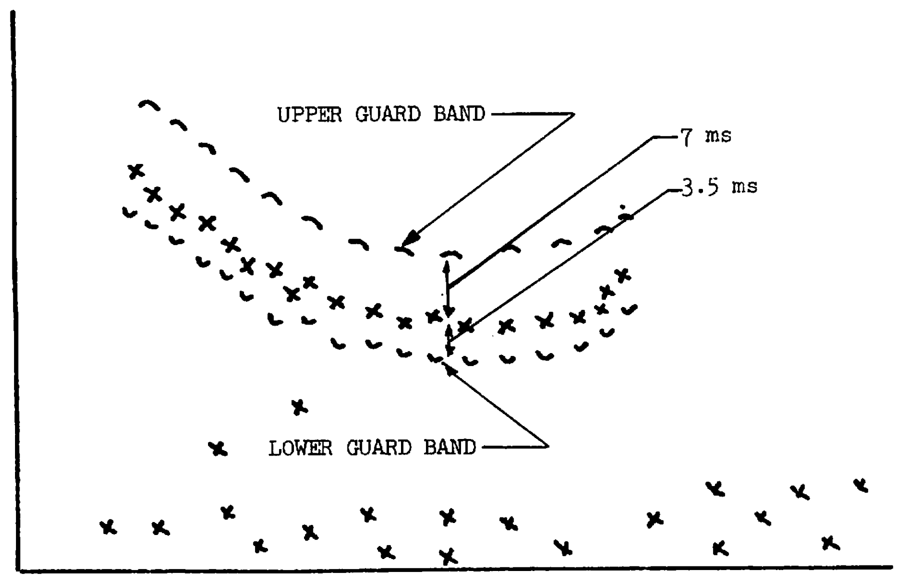


Figure 11. Tighter Window to Prevent Side-Tracking

2.3.1.3 LOS Masking

LOS masking refers to the problem of tracking the vortex delay as close to the line of sight as possible. This is especially true for the low-lying vortices close to ground which persist for a long period of time. The problem of delay tracks which interact with the LOS and disappear into the LOS only to reappear later as distinct delay tracks will be dealt with in the Stop Routine.

The solution,which gave favorable results,was to select a region above the LOS based on the LOS statistics which would be off limits to the delay tracker as shown in Figure 12. The region was set equal to two times the LOS variation or 5 increments (approximately 2.5 msec.) whichever was maximum. The minimum value of 5 increments appears to be a reasonable number to allow for the interaction of the vortex with the LOS.

The above criterion together with the tighter asymmetrical window with a smaller value on the lower side, keeps the delay tracker from getting pulled by the LOS and yet there is enough flexibility to track vortices which may rise again.

In the implementation, the off limits region above the LOS was chosen as the maximum of two times the LOS variation or 5 increments and the lower window was set to 7 increments (approximately

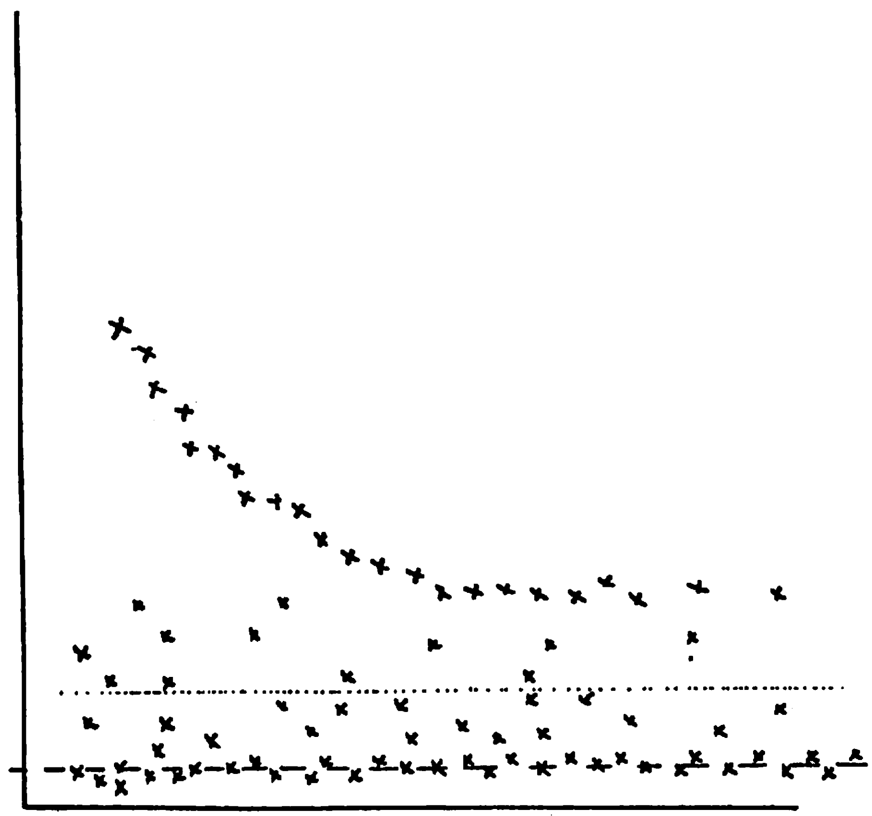


Figure 12. Adaptive LOS Masking

3.5 msec). The results of the LOS masking (some with the STOP routine) are shown in Figures 25 through 28.

2.3.1.4 Clutter

Clutter problems can be divided into two categories, clutter within the expected region for delay tracking and clutter outside this region. In the 15 landings that were examined no obvious cases of the former were noticed. A suggested solution for the clutter within the delay tracking region would be to track these before the aircraft arrival in the segment marked LOS Tracking Only in Figure 6. Once the clutter tracks are known they can be used in the Start Routine by eliminating those horizontal bins which contain clutter from consideration. Once the correct start is made, the tighter window around the delay tracker will keep it from being sidetracked by clutter.

As far as the clutter outside the expected delay tracking region is concerned, these can conveniently be accomplished by masking out those regions completely. In the implementation it was found that the horizontal bins above 120 increments (approximately 60 msec) from the LOS were not required, hence clutter in this region would not pose any problems. Figures 31 and 32 show the results.

1. TRACK DENSITY
2. NOISE DENSITY
3. DELAY ERROR
4. COMBINATION

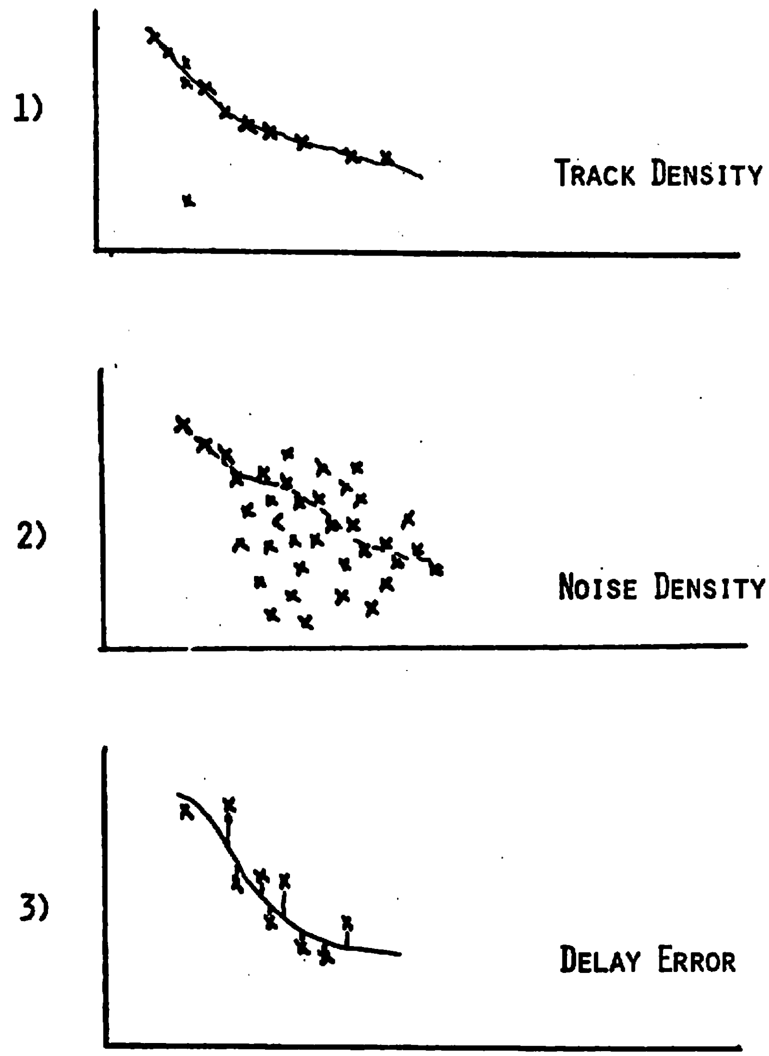


Figure 13. Possible Stop Routine Criteria

2.3.1.5 Stop

Finding suitable criteria for stopping the tracker was another problem encountered in implementing the basic delay tracker. The three criteria shown in Figure 13 as 1) Track Density, 2) Noise Density and 3) Delay Error were examined for their applicability.

1) Track Density refers to the running average of density of hits within the delay tracker window. It is shown as α_i in Appendix A.

2) Noise Density refers to the running average of the density of frames with more than one hit above the LOS masking region per delay track.

3) Delay Error refers to running average of the absolute error between the calculated estimate and the actual delay which is shown as D in Appendix A.

Except for the noise only situations, the track density parameter α_i or a slight variation thereof was found to be quite sufficient for stopping the delay tracker. Since the track density parameter is restricted to a tight window around the delay tracker, the delay error criterion was found to be redundant for this case.

The problem of distinguishing noise only situations from noise plus delay track situations as shown in Figure 14 proved to be

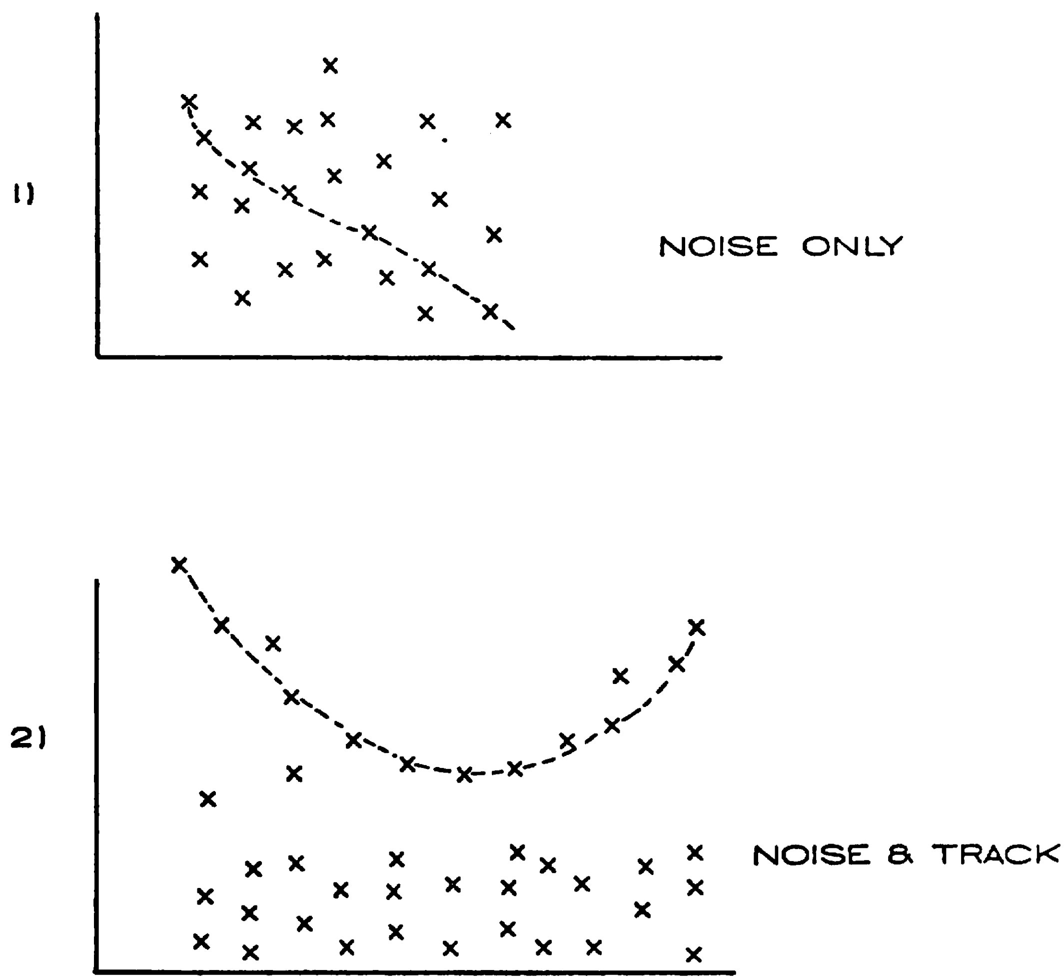


Figure 14. Noise Only and Noise Plus Track

more difficult. Since the noise density criterion as stated earlier (#2 in Figure 13) would give similar results in both these cases and since the tracker always chooses the path with the minimum mean squared error through either of these cases, it would appear that none of the criteria in Figure 13 would be able to distinguish between these two cases.

It seems that the noise density criterion needs to be modified. One promising approach would be to consider the noise density in the tight window around the delay tracker instead of the larger delay frame. This would keep the noise density for the noise only case relatively unchanged but drastically reduce the noise density in the noise plus delay track case.

As far as the situation where the delay track disappears in the LOS and reappears later to hover above the LOS is concerned, it was found convenient to make the first stop a temporary one and initiate a Restart Routine. The Restart Routine consists of looking through the tighter window for a preselected number frames until the number of hits crosses a certain threshold which in general is the same as that in the Start Routine. If the number of hits does not cross the threshold then the delay tracker is terminated permanently. If the number of hits crosses the threshold, then the delay tracking is resumed until the second stop which terminates the delay tracker permanently.

A slight modification of this would be to reinitiate the Start Routine after the first temporary stop. Since the temporary stop generally occurs close to the LOS and noise is also more pronounced close to the LOS, the Start Routine may initiate tracking of noise only. This problem may be alleviated somewhat by 1) choosing only those Start Routine bins which are above the lower bound of the temporarily stopped delay tracker, 2) assigning a different slope to the Sloping Bins, and 3) decreasing the bin width since fewer bins are required.

In Figures 17 through 32 a slight modification of track density criterion was implemented. The delay tracker was temporarily stopped when there were no hits in 8 consecutive frames within the tracker window. The tracker, however, continues looking through the window and updating all the parameters and if the number of hits within the window exceeds 6, restarts tracking until 8 consecutive no hits in the window. Also if the number of hits within the delay tracking window does not exceed the threshold of 6 in the remaining frames, the Restart Routine stops the delay tracker permanently. In general, a number such as 60 consecutive frames (approximately 30 sec) would be more reasonable for the Restart Routine implementation. The results of the Stop Routine implementation can be seen vividly in Figures 26 through 30.

2.3.2 Flow Charts and Computational Requirements

Since the basic tracker is the same as that implemented by AVCO (1), i.e., minimum mean square error delay tracker, its flow chart computational requirements and timing will not be discussed here. The flow chart, time and storage requirements for all the additions and modifications discussed in the previous sections will be the subject of this section.

2.3.2.1 Start Routine Implementation

The flow chart for the Start Routine is shown in Figure 15. The first conditional branch marked "START=1" decides whether the Start Routine or the delay tracker is in effect. START is of course initialized to 0 beforehand. The box computes the bins in which the delay point lies, i.e., J2 for horizontal, J4 for sloping and J6 for offset sloping, and then adds one to the previous number of hits in the appropriate bin, i.e., HITSH for horizontal, HITSS for sloping and HITSO for offset sloping. The following three conditional branches update the maximum number of hits, MAXH if any of the latest bins exceed the previous maximum, otherwise MAXH remains unchanged. The delay and the slope of the bin corresponding to the maximum is also updated or left alone in parallel with MAXH as shown in the three boxes to the right.

At the end of the frame a conditional branch compares the maximum number of hits MAXH thus far with the threshold HTHR to decide whether to continue Start Routine or to initiate delay

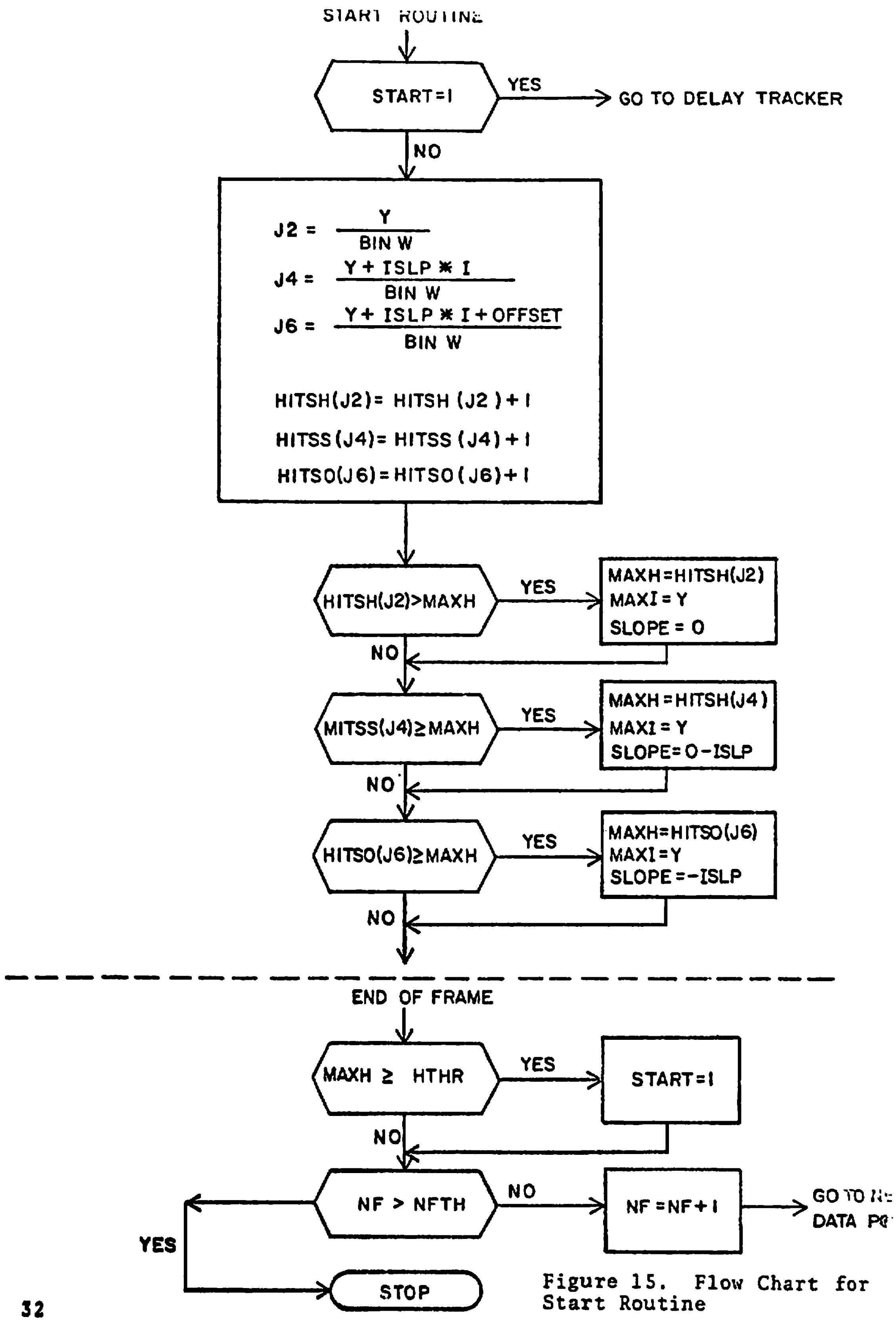


Figure 15. Flow Chart for Start Routine

tracker. The last conditional branch compares the total number of frames gone through so far with the preselected threshold to decide whether to continue the Start Routine or Stop permanently.

The time and storage requirements for the above Start Routine on PDP-11 with the extended Arithmetic Unit (KE11-A) are as follows.

Time:
6 Conditional branches
4 Multiplies/Divides
6 Adds

approximately 45 μsec. per track

Storage:
8 Words HITSH
16 Words HITSS
16 Words HITSO
6 Words Miscellaneous

46 Words total per track

There are a few observations worth noting about the Start Routine.

1) It is inherently fixed point.

2) Although the storage requirement appears to be high compared to the storage requirement for the delay tracker, the time requirement is much lower.

3) Since each word is expected to be less than 4 bits, these words can be packed 4 to 1 in the 16-bit computer word requiring approximately 12 computer words. The time required for scrambling and unscrambling the words is traded for lower storage.

4) Since the Start Routine and delay tracker are operating in mutually exclusive intervals, the storage can overlay that used by the delay tracker.

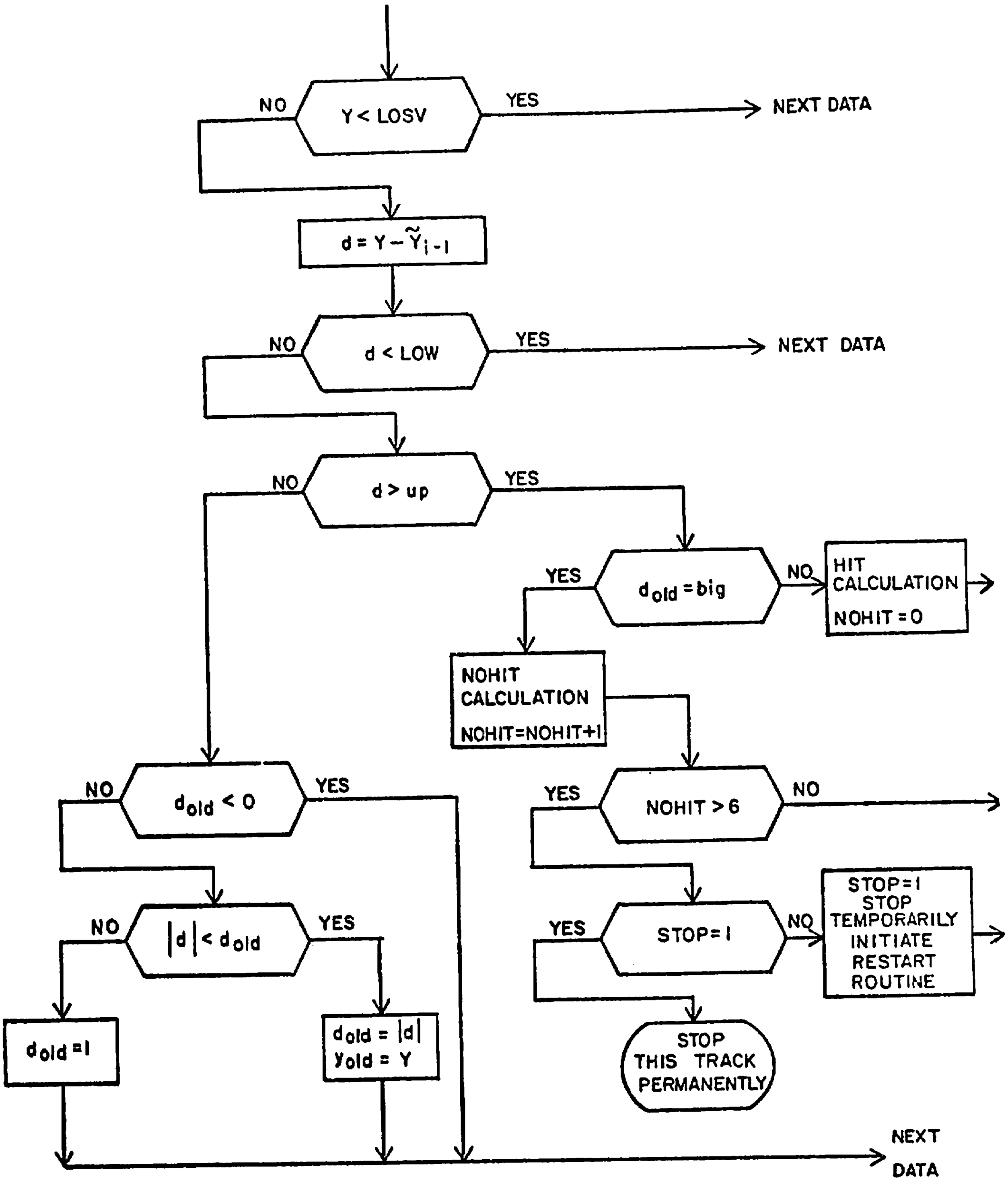


Figure 16. Flow Chart for Remaining Modifications to Delay Tracker

2.3.2.2 Delay Tracker Additions

The flow chart for the remaining modifications under the categories of side-tracking, LOS Masking, Clutter and Stop are shown in Figure 16. The first three-conditional branches combine the solution to side-tracking and LOS Masking. The first branch eliminates data points within LOS variation. The next two branches accept points within the tracker window around the predicted estimate, $\tilde{Y}_{i-1}$. The following two conditional branches on the left choose the point with the lowest error, d_{old}. The following right branch performs all the delay tracker updating when d_{old} has been changed from its initial large value, otherwise it goes through the Stop Routine. The following conditional branch compares the number of consecutive no hits to a threshold and Stops the first time if the threshold is exceeded. The conditional branch "STOP = 1" decides whether it is a first temporary stop or second final stop. STOP is of course initialized to 0 at the beginning.

The maximum time and storage requirements for the above delay tracker modifications on PDP-11 with the Extended Arithmetic Unit (KE11-A) are as follows:

Time:
- 5 Conditional Branches
- 1 Subtract
- 1 Absolute Value
- 2 Stores

Approximately 23 μsec. per track

Storage: $\underset{\sim}{L}OSV$

Y_{i-1}

d_{old}

y_{old}

4 words per track

It is important to note that some of the storage shown in the flow chart i.e., low, up, are common to all tracks and that the path marked "NO HIT CALCULATION" requires an insignificant number of computations compared with "HIT CALCULATION" and also occurs less frequently. Therefore, the above time and storage requirements represent the most frequent and time consuming computational additions to the basic delay tracker. In Ref. (1), the time required for the basic delay tracker calculations was estimated as 300 μsec. per track in fixed point arithmetic, so the above additions which are also in fixed point arithmetic represent an increase of approximately 8%.

2.3.3 Fixed Point Implementation of Tracker

This section deals with the problems involved in using the finite 16-bit word length of the minicomputers such as PDP-11 to do the delay tracker computations. It should again be emphasized that all the additions and modifications done on the basic tracker are inherently fixed point and the 16-bit word size is amply sufficient for accomplishing the Start Routine or the additions to the basic delay tracker. Therefore, we shall be concerned only with the basic tracker computations in fixed point arithmetic.

It can be safely stated that except for the two second order non-central moments in Appendix A, most of the remaining computations can be done in fixed point arithmetic by scaling the quantities up and occasionally rescaling if necessary. Since the 16-bit word allows up to 65536 levels of quantization, this should be quite sufficient for all the quantities in Appendix A except XSS_i and XYS_i as shown below:

$$XSS_i = \frac{M_s \cdot XSS_{i-1} + i^2}{M_s + 1}$$

and,

$$XYS_i = \frac{M_s \cdot XYS_{i-1} + i \cdot Y_i}{M_s + 1}$$

Maximum values of i and Y_i (normalized to LOS) were found to go as high up as 300 frame increments and 120 delay increments respectively, hence i^2 exceeds the 16-bit word size and $i \cdot Y_i$ is very close to the limit. The solution to this problem is to use centralized second order moments as follows:

$$\mathrm{Var}\{X\}_i = \frac{M_s \cdot \mathrm{Var}\{X\}_{i-1} + (i - XS_i)^2}{M_s + 1}$$

and,

$$\mathrm{Cov}\{XY\}_i = \frac{M_s \cdot \mathrm{Cov}\{XY\}_{i-1} + (i - XS_i)(Y_i - YS_i)}{M_s + 1}$$

Maximum values of $(i - XS_i)$ and $(Y_i - YS_i)$ should not go beyond the time constant (=10 frames) and the larger tracker window width (=14 increments) respectively. Besides keeping the values low, it also reduces the number of multiplies by 2 since the slope A is directly given by Cov/Var.

As far as the track density running average, α_i is concerned since it ideally represents an integral number of hits per (N + 1) frames, it should not require any more quantization than (N + 1) levels. In order to avoid the degradation due to the cumulative effect of division every change of frame, the following procedure should work:

$$\alpha_{i-1} = \frac{N \cdot \alpha_{i-2} + \beta_{i-1}}{N + 1} = \frac{(\mathrm{Num})\,i-1}{N + 1}$$

$$\alpha_i = \frac{(Num)_{i-1} - \alpha_{i-1} + \beta_i}{N + 1} = \frac{(Num)_i}{N + 1}$$

For the present frame i, by using mainly the numerator from the previous frame calculation the deterioration from the fixed point division is avoided in the iterative process. This also reduces the number of multiplies by one at the expense of increasing the storage by one.

Assuming M_s is quantized to the same level as α_i i.e., (N + 1) or approximately 4 bits, and XS_{i-1}, YS_{i-1} do not exceed 8 bit word size for integral number of frame and delay increments, that still leaves 4 more bits for finer quantization of XS_i and YS_i.

Although the fixed point version of the basic tracker was not implemented as such the results of the modified delay tracker still hold since the changes applied to the basic tracker are all in fixed point arithmetic.

2.4 Results

This section discusses the effect of the various modifications to the basic delay tracker on actual field data after analog processing and A/D conversion by AVCO. The following data were examined.

No. of Landings	15
No. of Receivers/Landing	6
No. of Delay Tracks/Receiver	3
Total no. of Tracks examined	270

For the sake of brevity,of the 270 delay tracks only those which gave problems with the basic tracker and the corresponding improvement from the modifications are shown in Figures 17 through 32. From the figures it can be seen that:

1) The Start Routine with the Horizontal, Sloping and Offset Sloping Bins significantly improves the wrong start and late start problem.

2) Tighter delay tracking eliminates sharp discontinuities, wrong tracks and tracking the LOS.

3) Non-symmetric window around delay tracker enables tracking of upward rising vortices and prevents tracking LOS.

4) Masking of a region 2 times LOS variation.allows tracking as close to LOS as statistically possible.

5) Temporary stop allows for tracking of delay tracks which disappear into LOS only to reappear later as low-lying tracks which tend to persist.

From the visual observation of the modified delay tracker results of the 270 tracks it can be stated that the miss rate (or tracking wrong tracks) was 2 to 3%. An example of miss is shown in Figure 26b. Most of the misses were due to situations similar to those shown in Figure 26b, where there are two equally strong delay tracks. Some misses occurred due to wrong start through noise.

NOT USED

Noisy situations where there is no visible delay track or the delay track is buried in too much noise accounted for 12% of the total 270 cases. Although the modified noise density criterion was not implemented, it is believed that its use would significantly improve the rejection rate of noise only situations without worsening the miss rate of 2 to 3%. Since there is a built-in redundancy in the PAVSS, both the miss rate and the false alarm rate would be drastically reduced when the redundant tracks are omitted from consideration. Theoretically, since only 2 tracks out of the possible 18 are nonredundant, the above percentages could be reduced by a factor of as much as 9, assuming uniform distribution.

ACOUSTOGRAM CH. 1

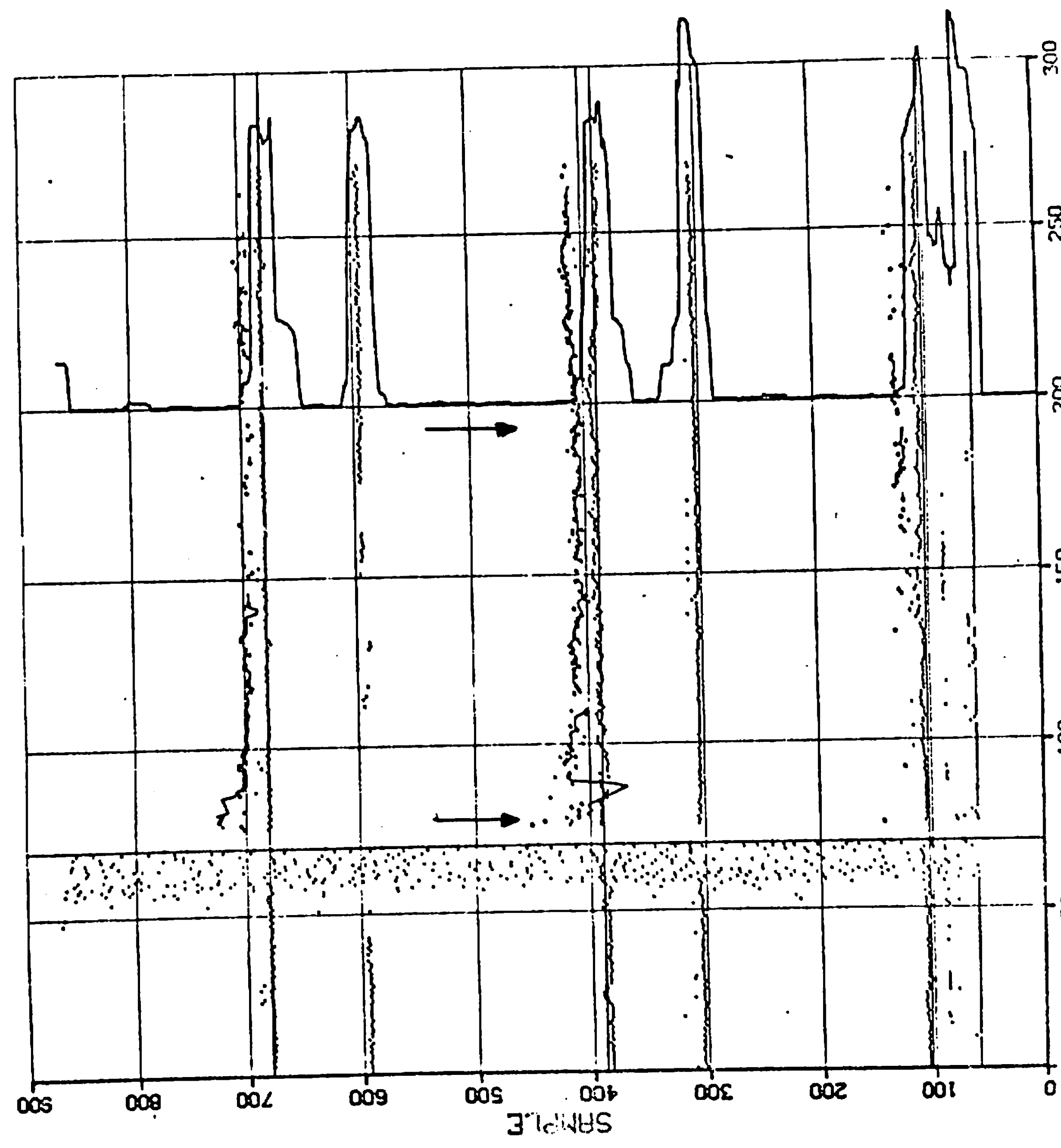
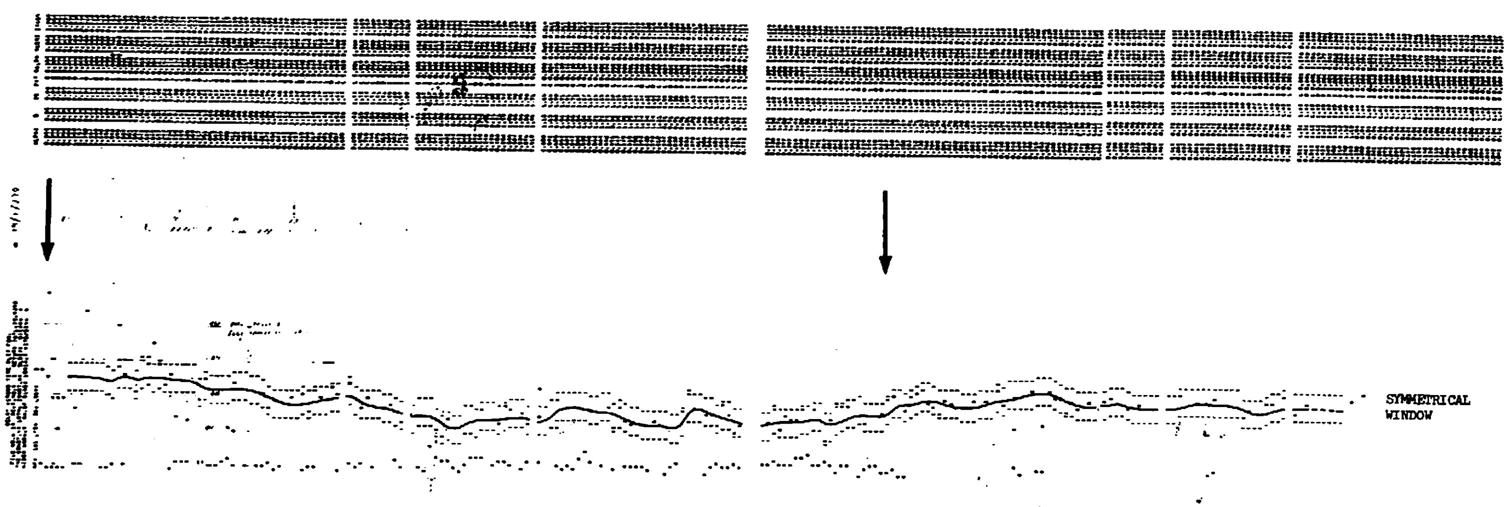


Figure 17a



Improved Track from Figure 17a (~400 samples into frame)

Figure 17b

ACOUSTOGRPM CH. 5

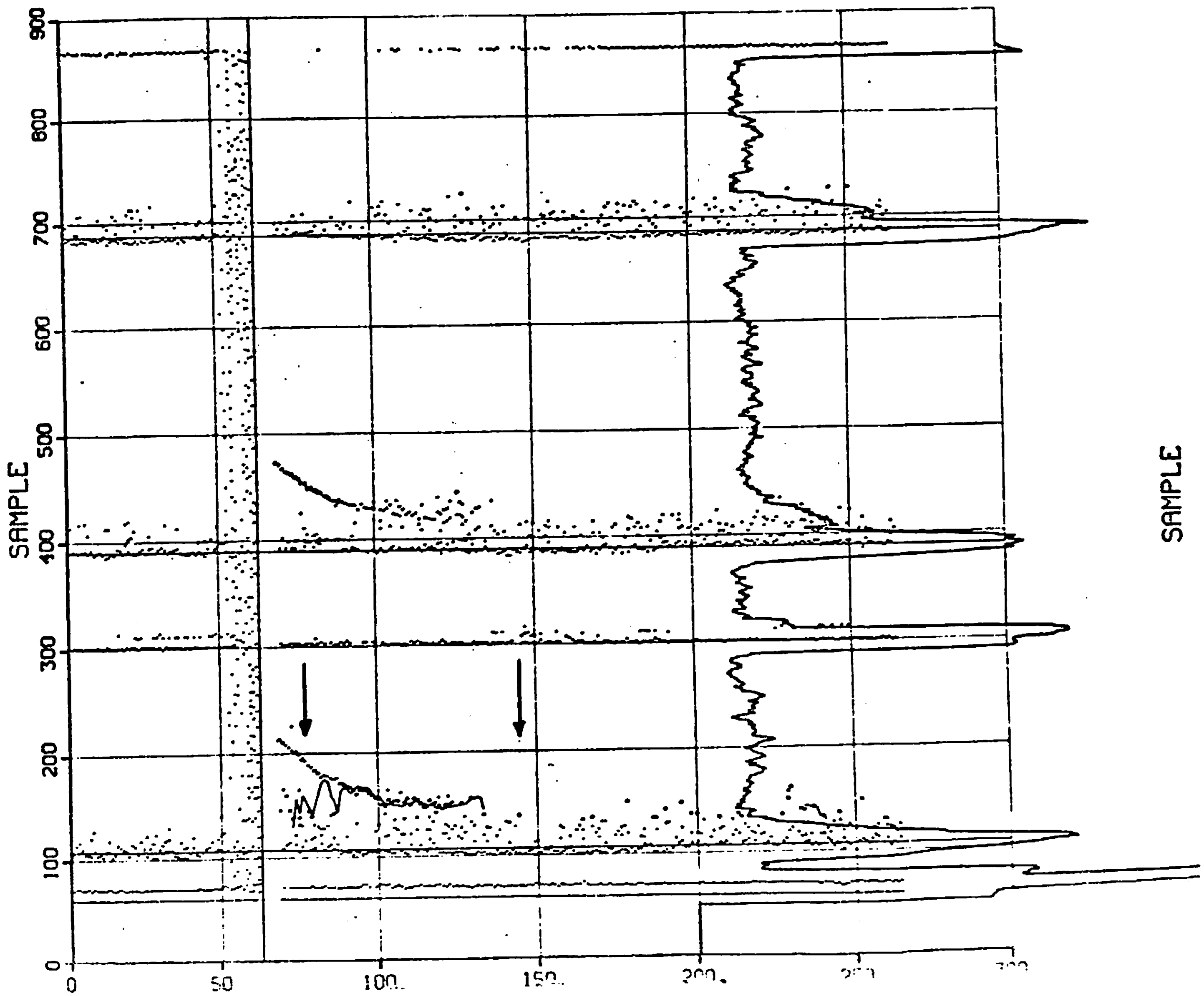
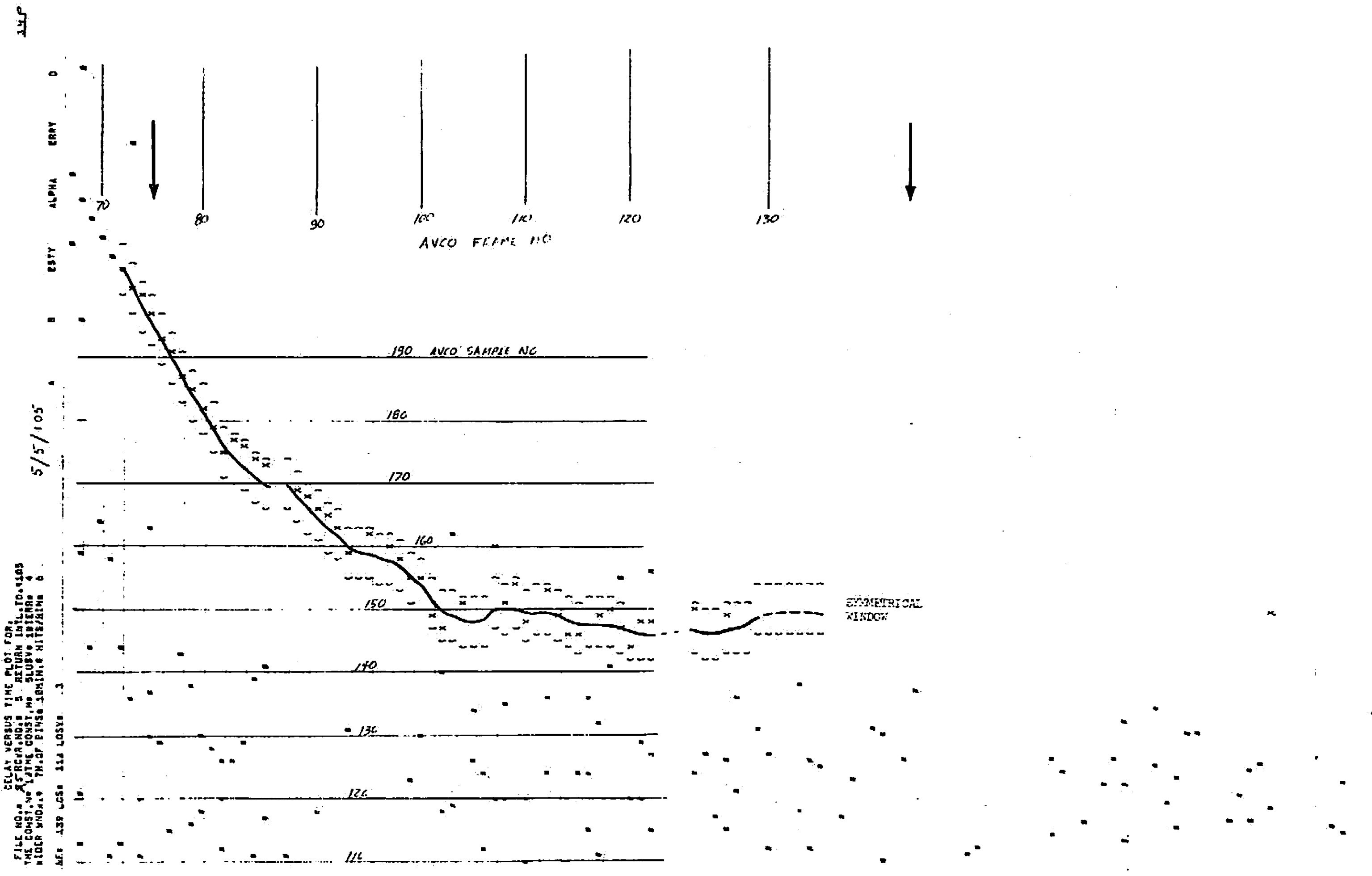


Figure 5a



Improved Track from Figure 18a (~100 samples into frame)

Figure 18b

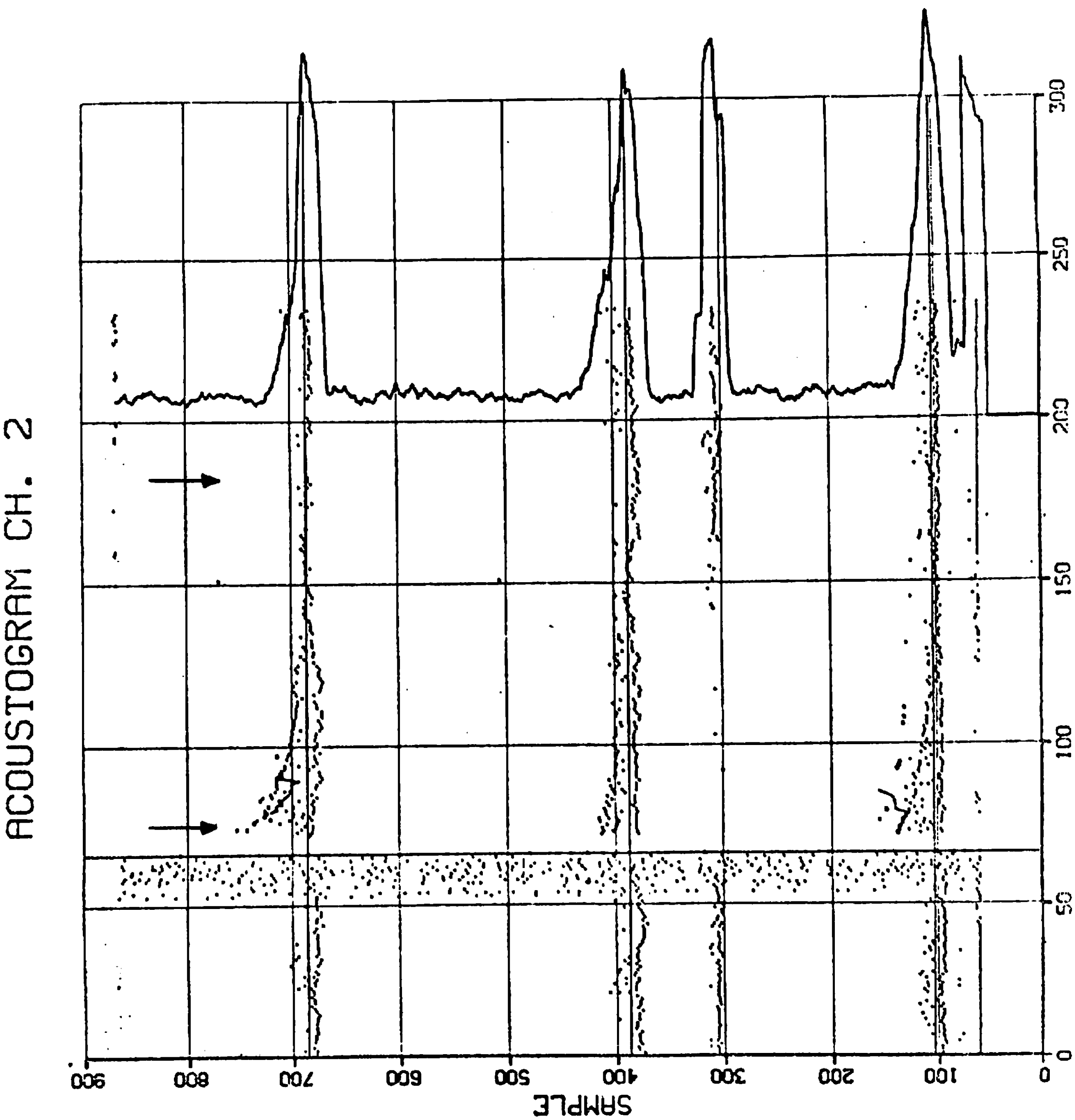
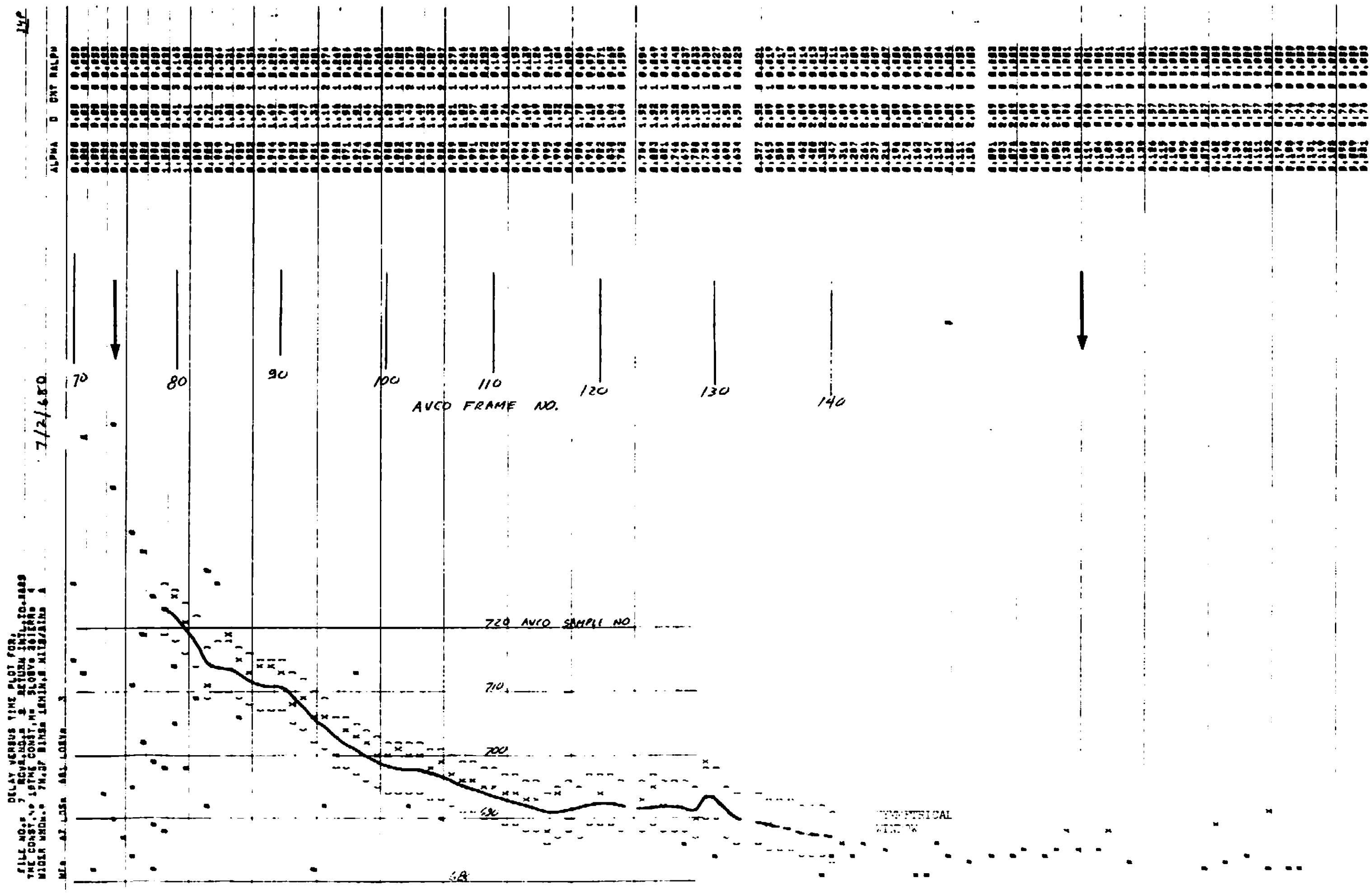

ACOUSTOGRAM CH. 2
SAMPLE
0
100
200
300
400
500
600
700
800
900
0
50
100
150
200
250
300

Figure 19a



Improved Track from Figure 19a (~700 samples into frame)

Figure 19b

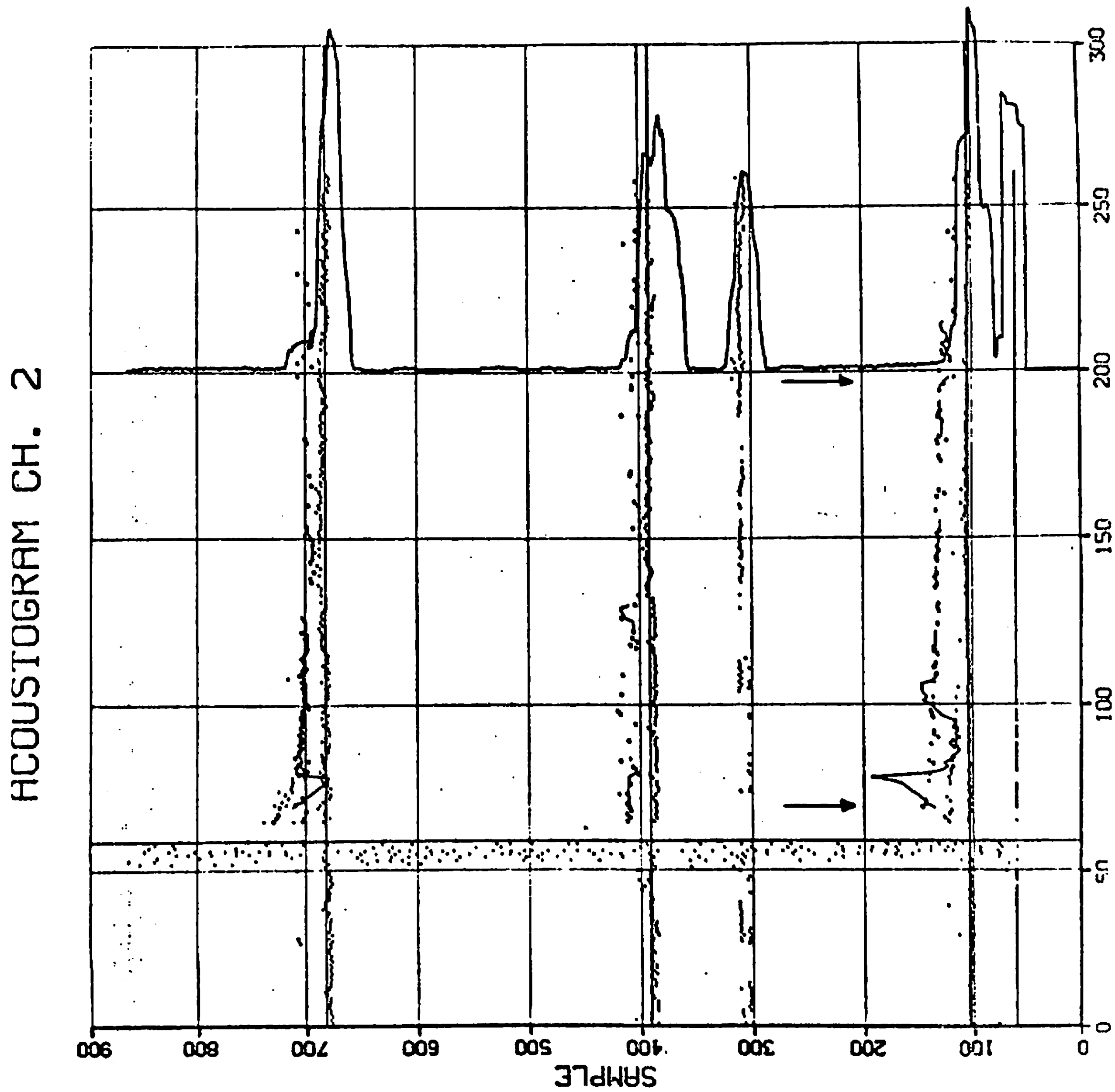
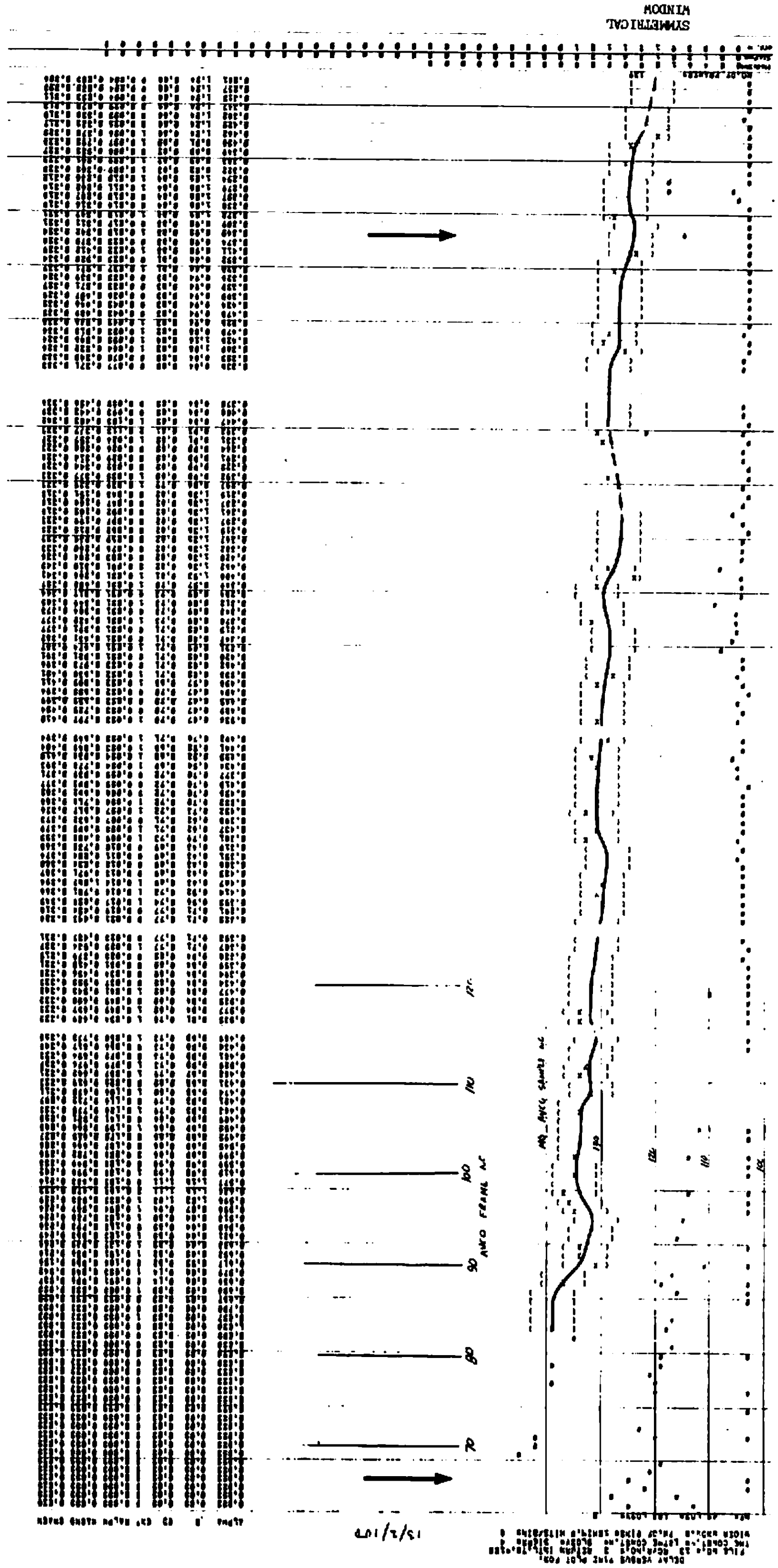
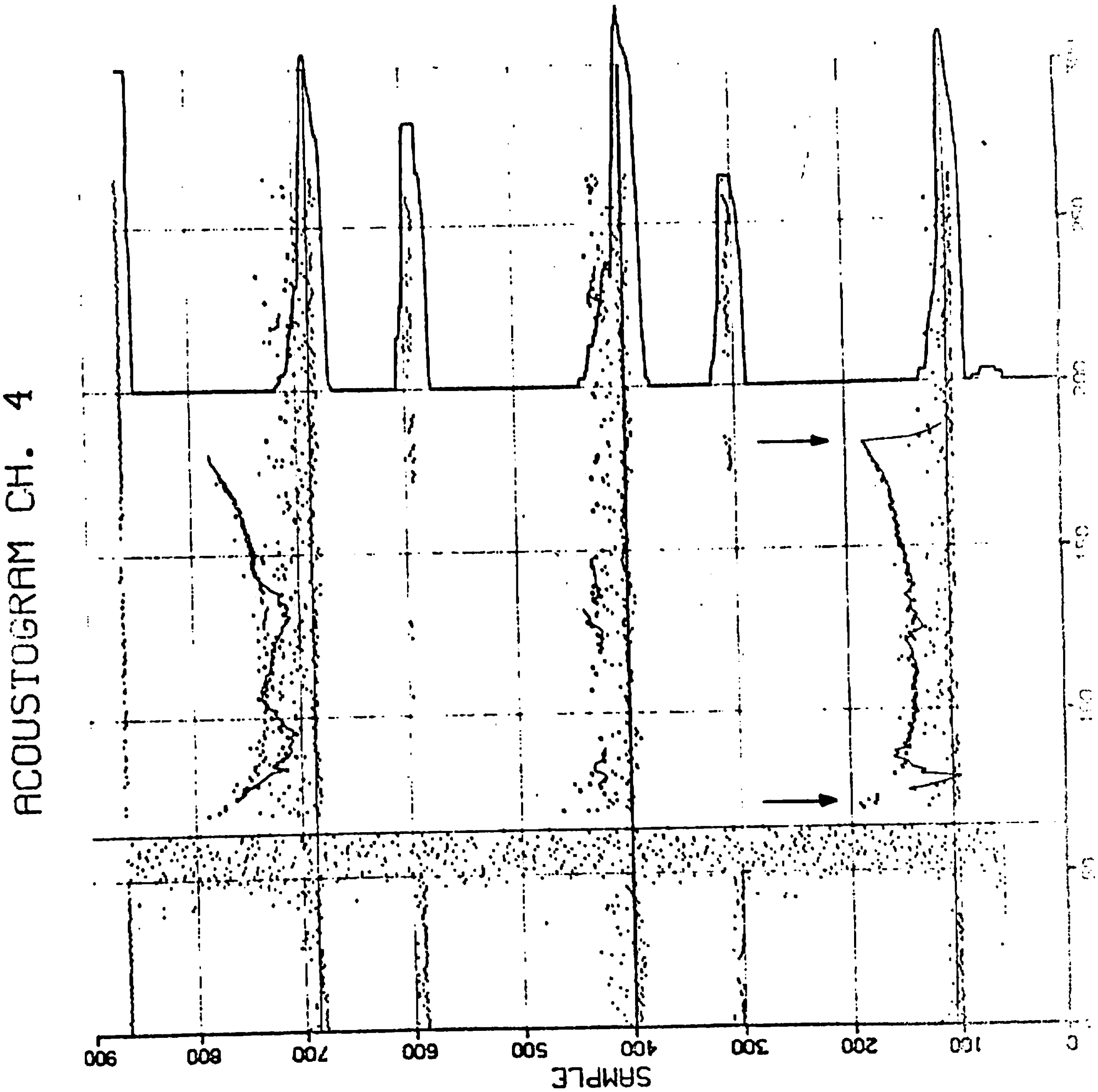

ACOUSTOGRAM CH. 2
SAMPLE
0
100
200
300
400
500
600
700
800
900
0
50
100
150
200
250
300

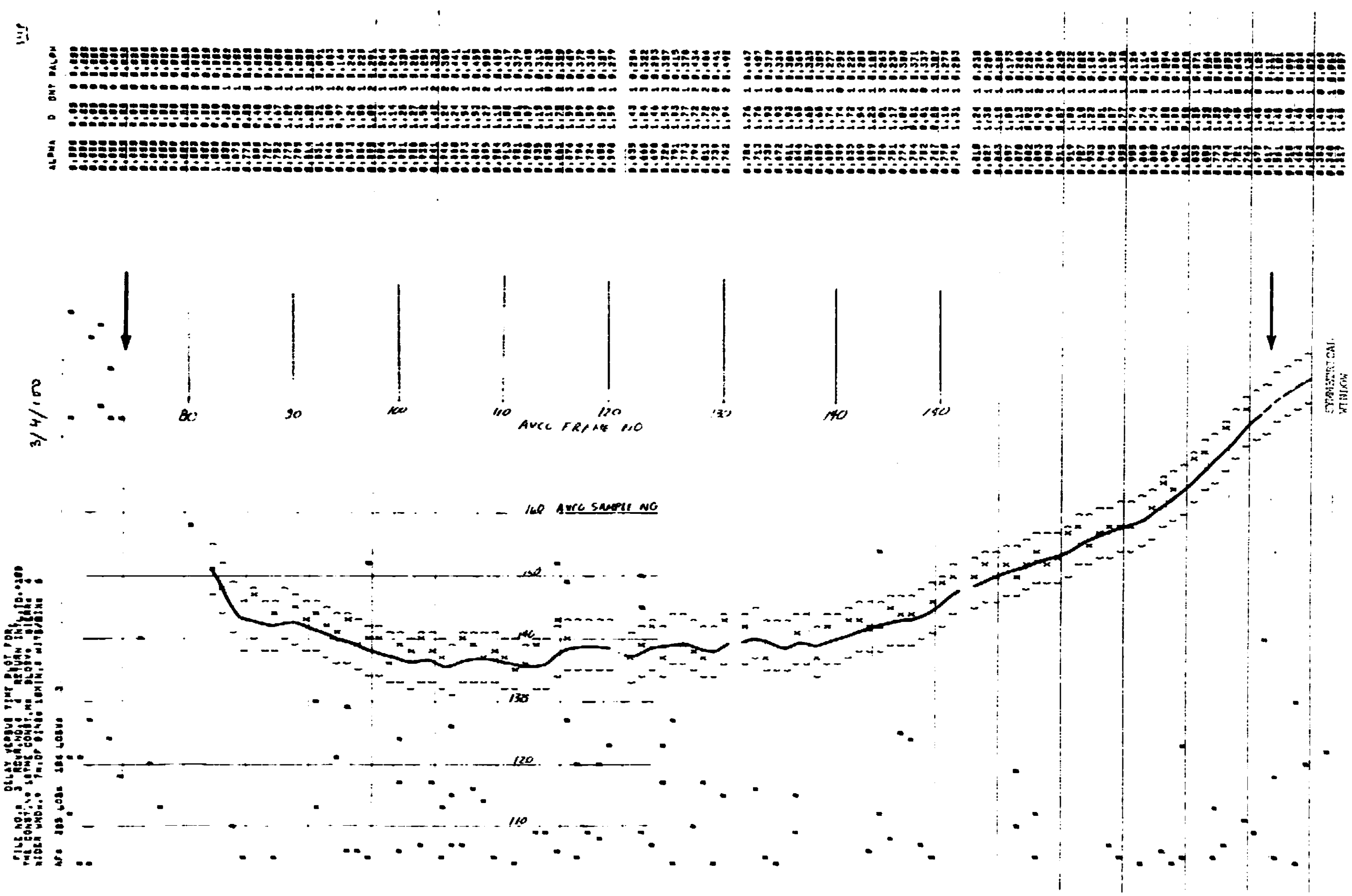
Figure 2a



Improved Track from Figure 20a (~100 samples into frame)

Figure 20b


ACOUSTOGRAM CH. 4
SAMPLE
900
800
700
600
500
400
300
200
100
0



Improved Track from Figure 21a (~100 samples into frame)

Figure 21b

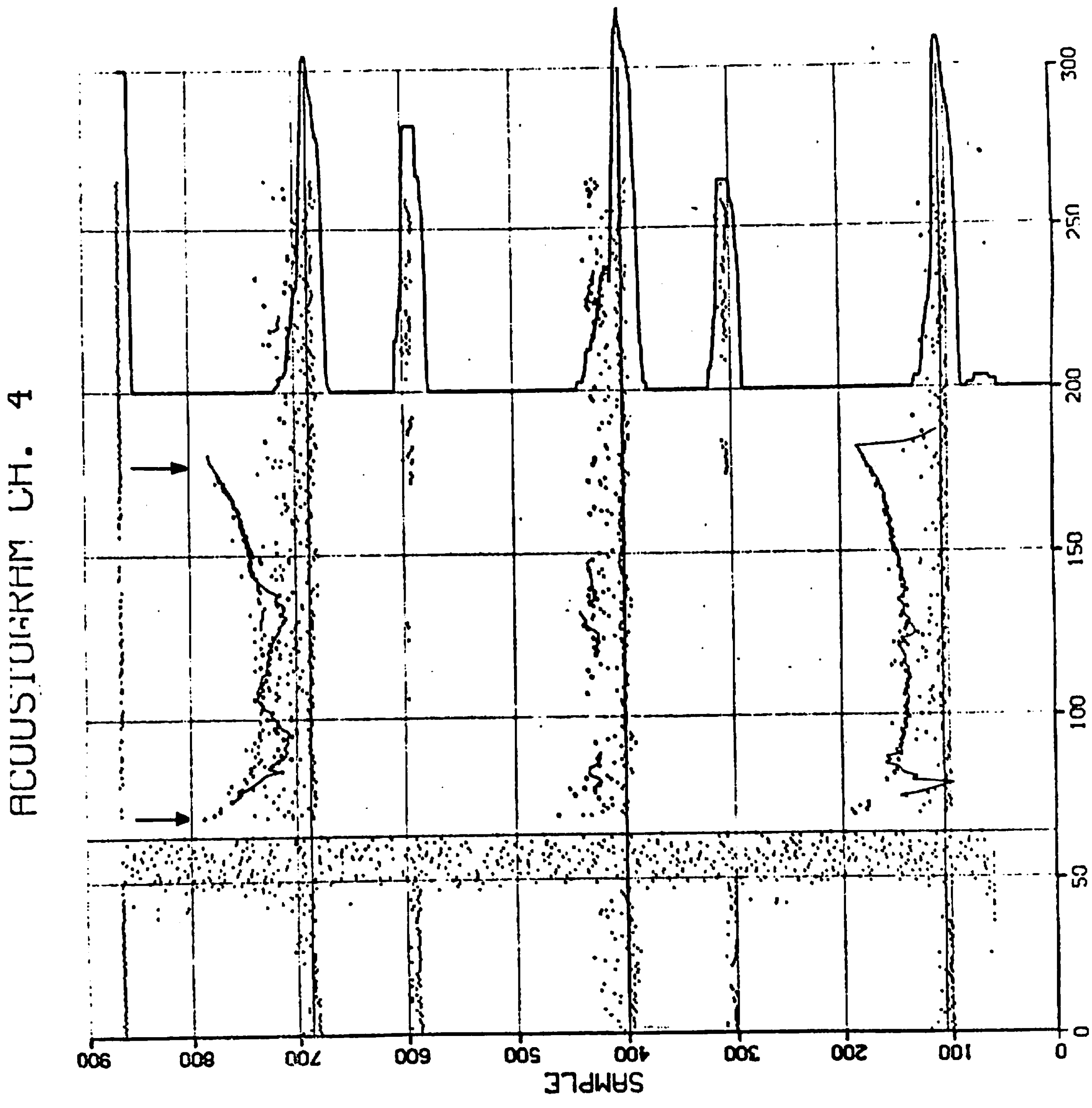
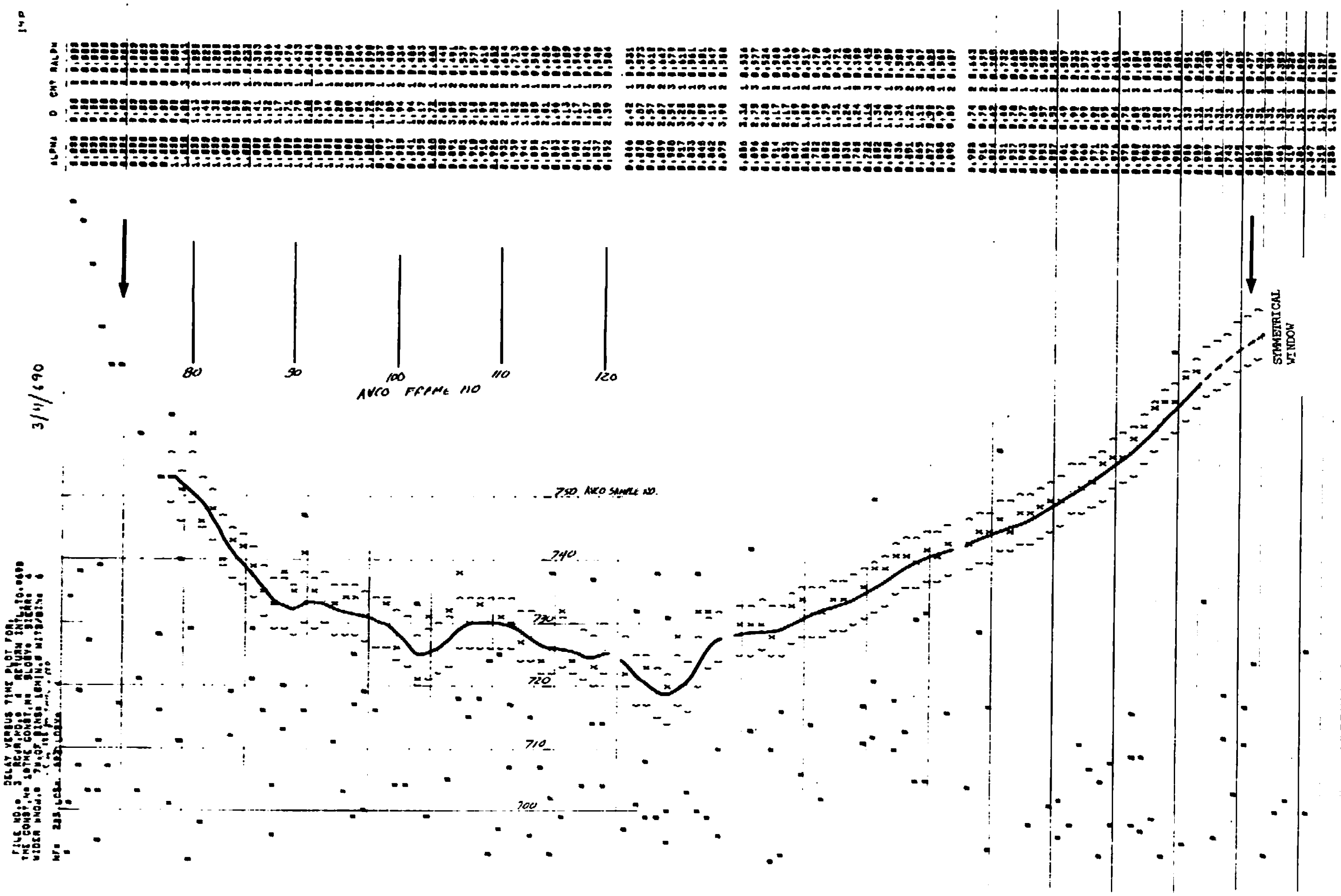
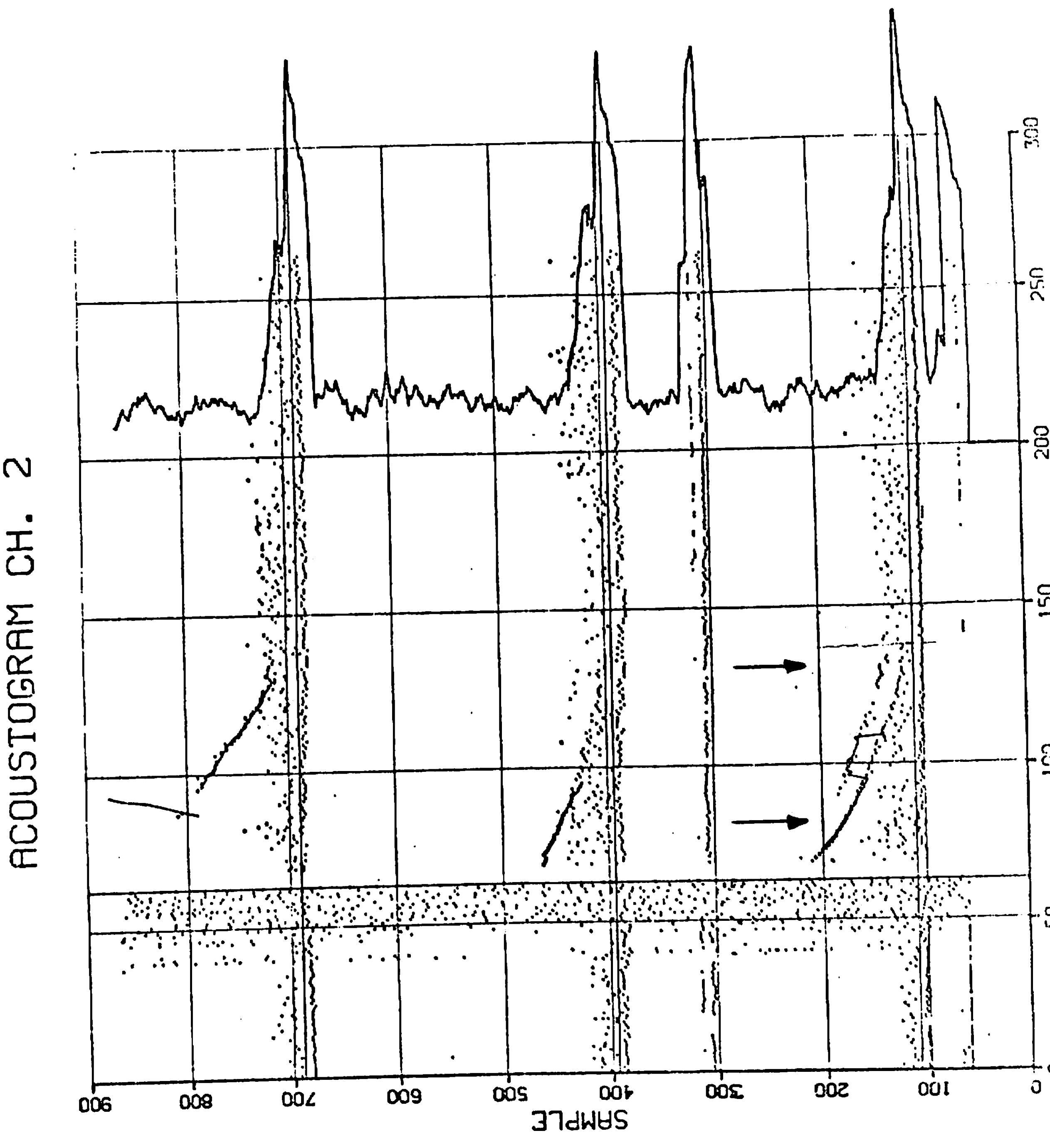

ACOUSTOGRAM CH. 4
SAMPLE
0
100
200
300
400
500
600
700
800
900
0
50
100
150
200
250
300

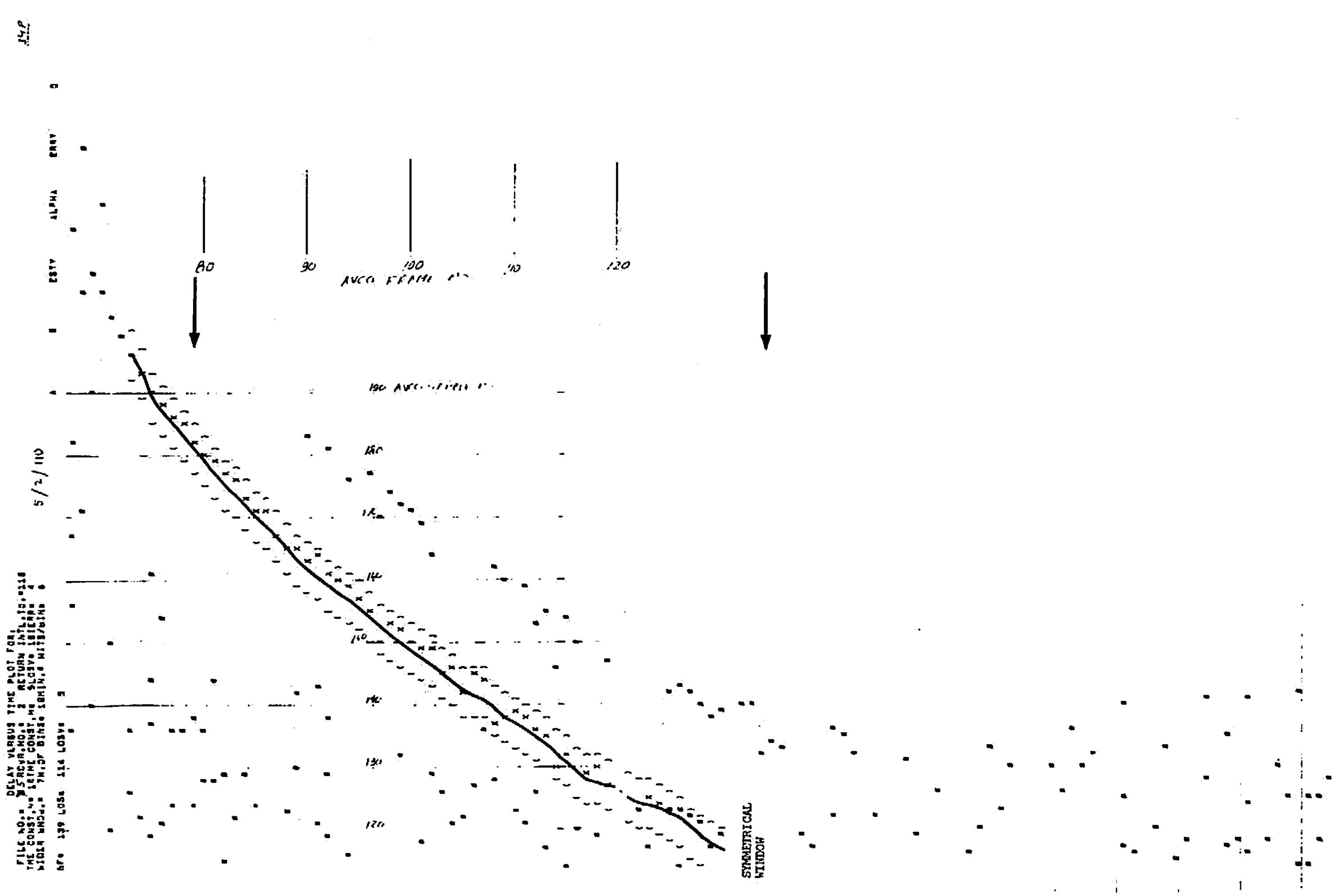
Figure 22a



Improved Track from Figure 22a (~700 samples into frame)

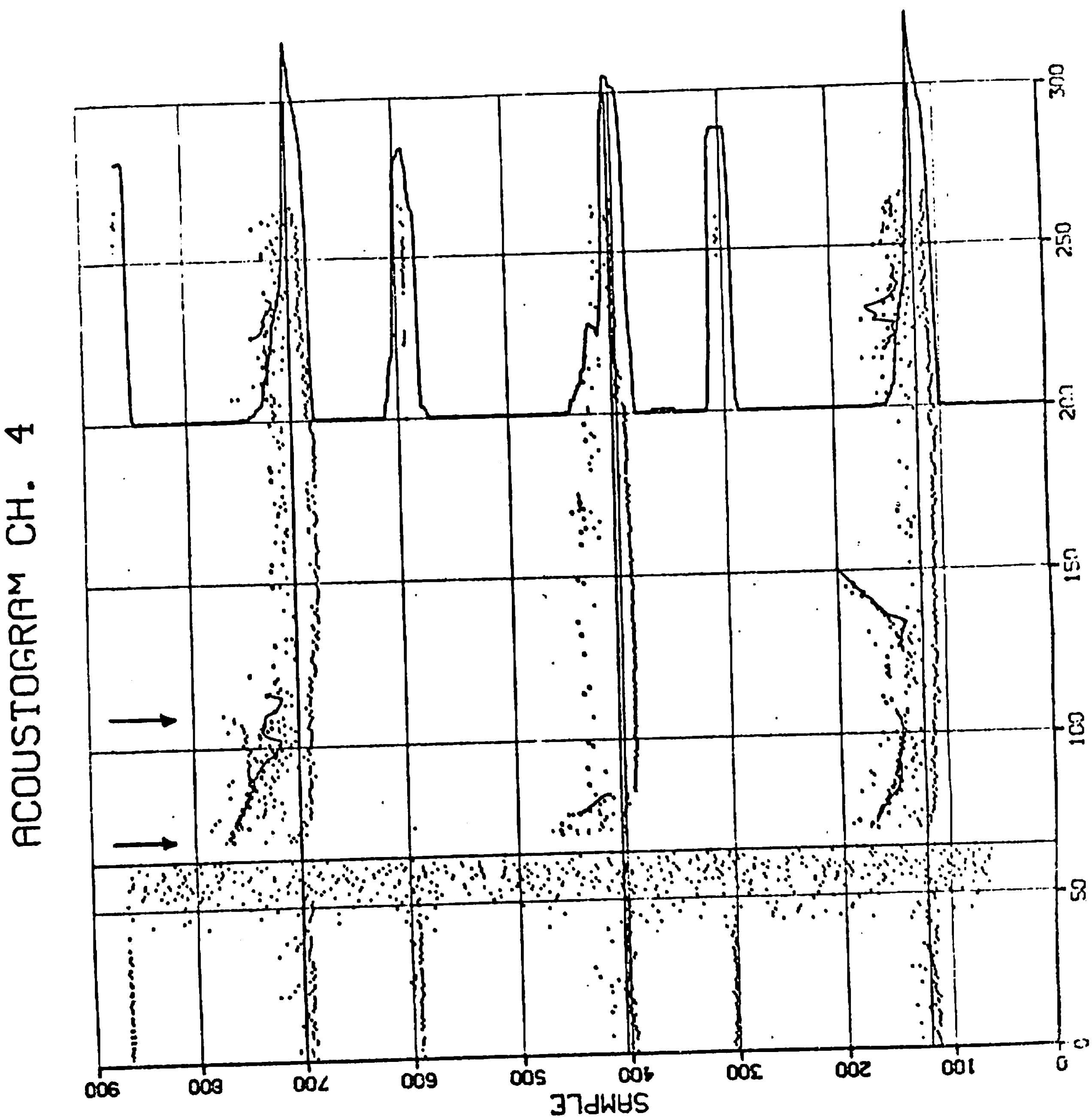
Figure 22b

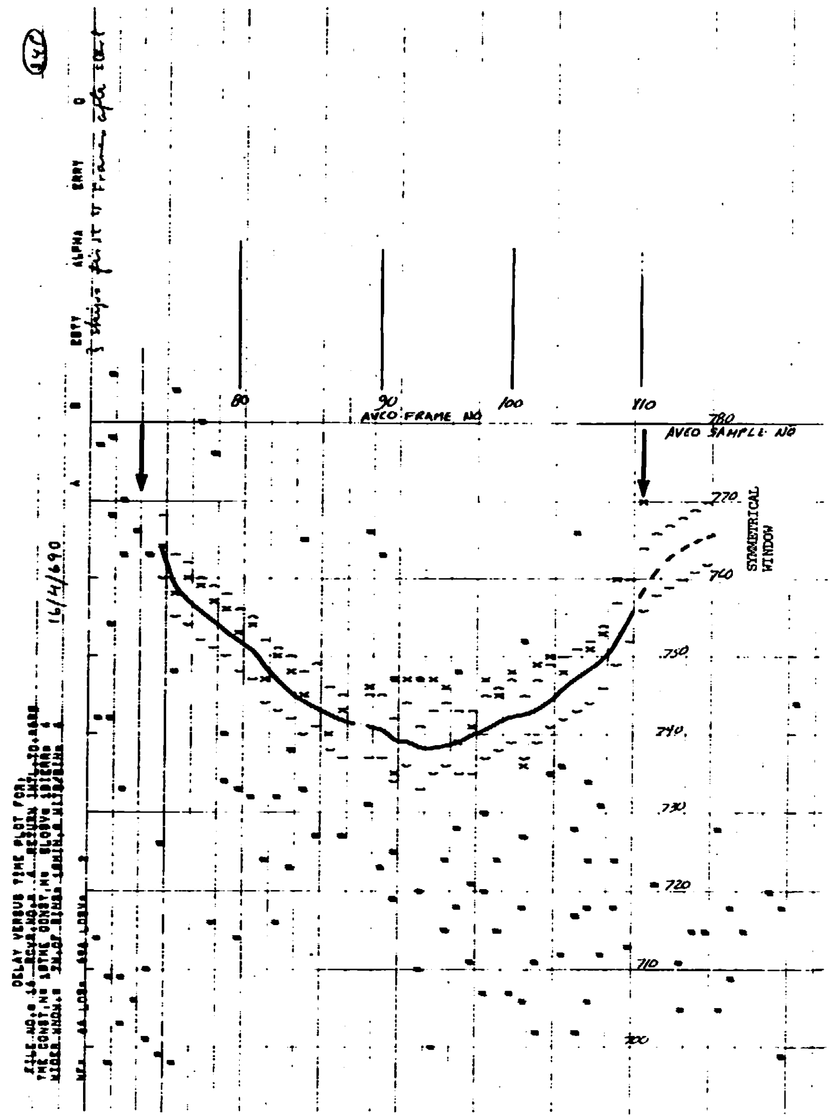

ACOUSTOGRAM CH. 2
SAMPLE
0
100
200
300
400
500
600
700
800
900
0
50
100
150
200
250
300



Improved Track from Figure 23a (~150 samples into frame)

Figure 23b


ACOUSTOGRAM CH. 4
SAMPLE
900
800
700
600
500
400
300
200
100
0
50
100
150
200
250
300



Improved Track from Figure 24a (~700 samples into frame)

Figure 24b

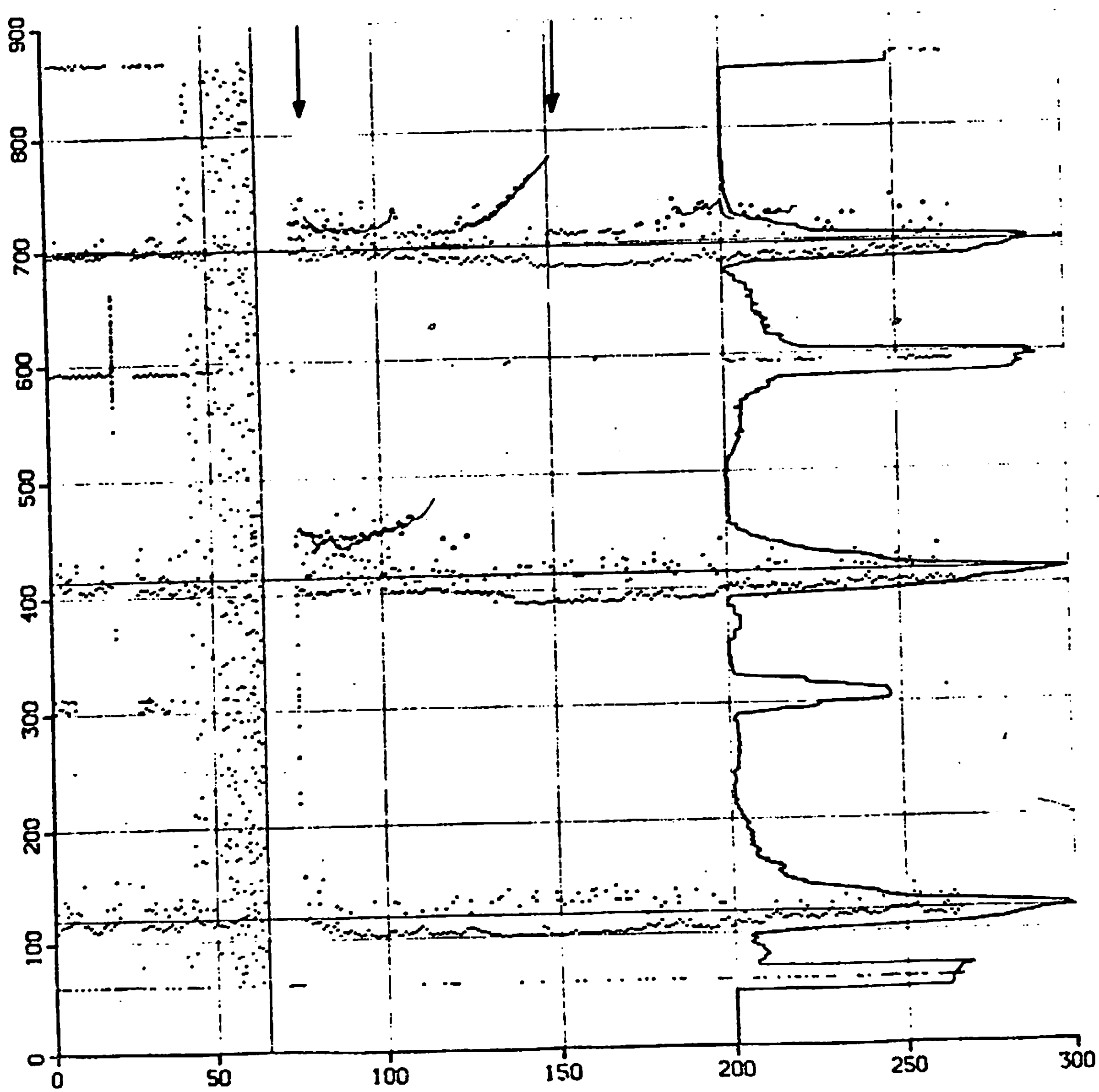
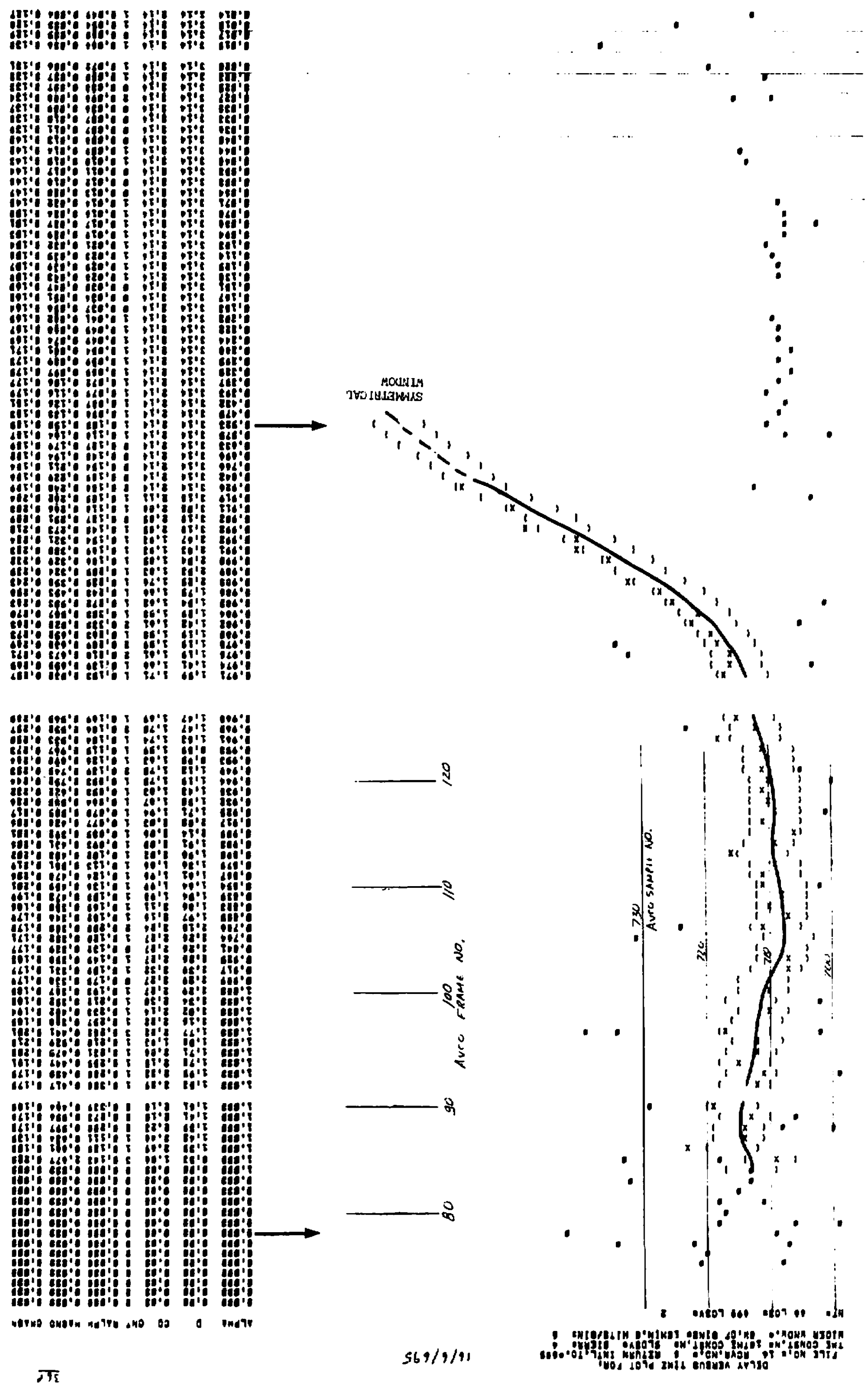


Figure 25a



Improved Track from Figure 25a (~700 samples into frame)

Figure 25b

ACOUSTOGRAM CH. 2

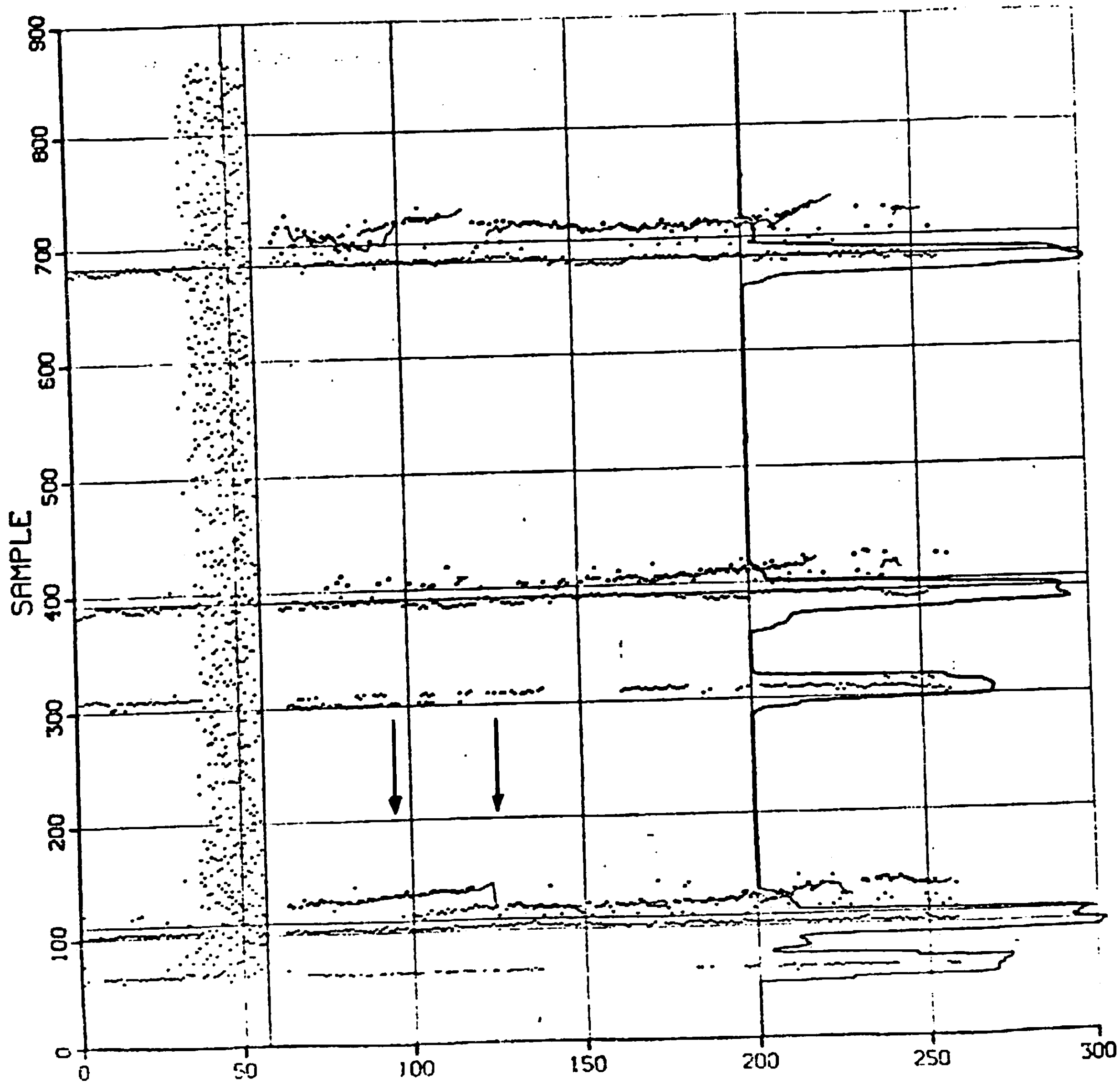
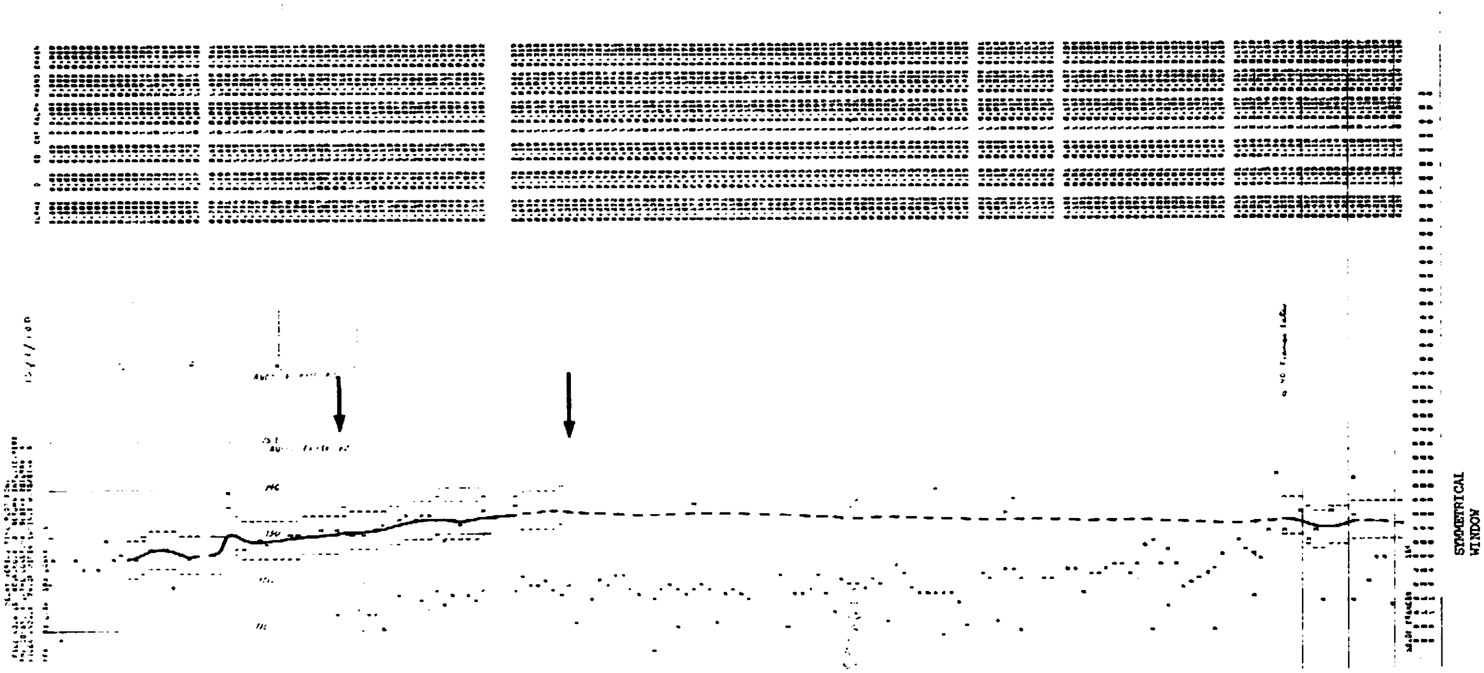


Figure 20a



Improved Track from Figure 26a (~100 samples into frame)

Figure 26b

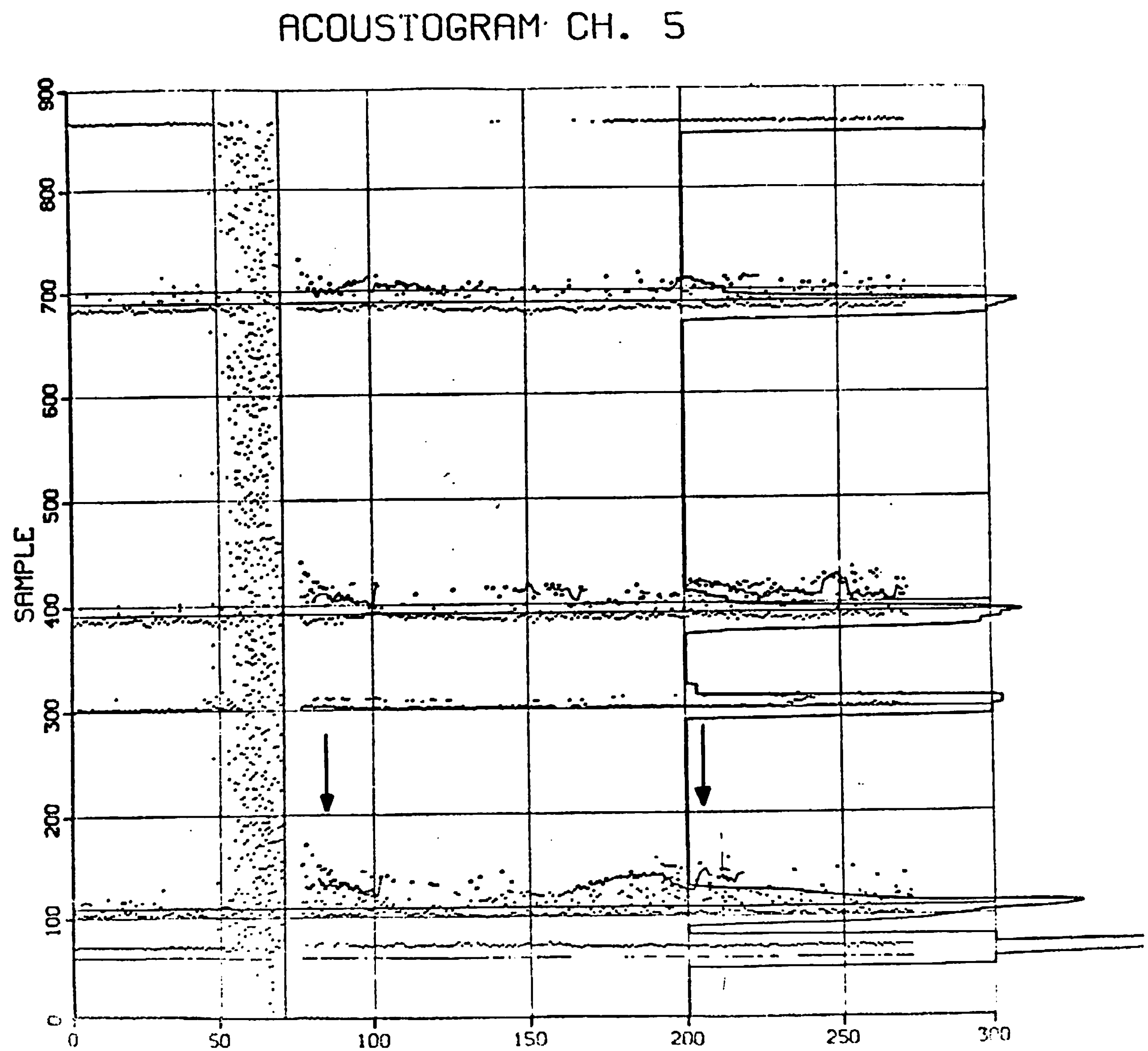
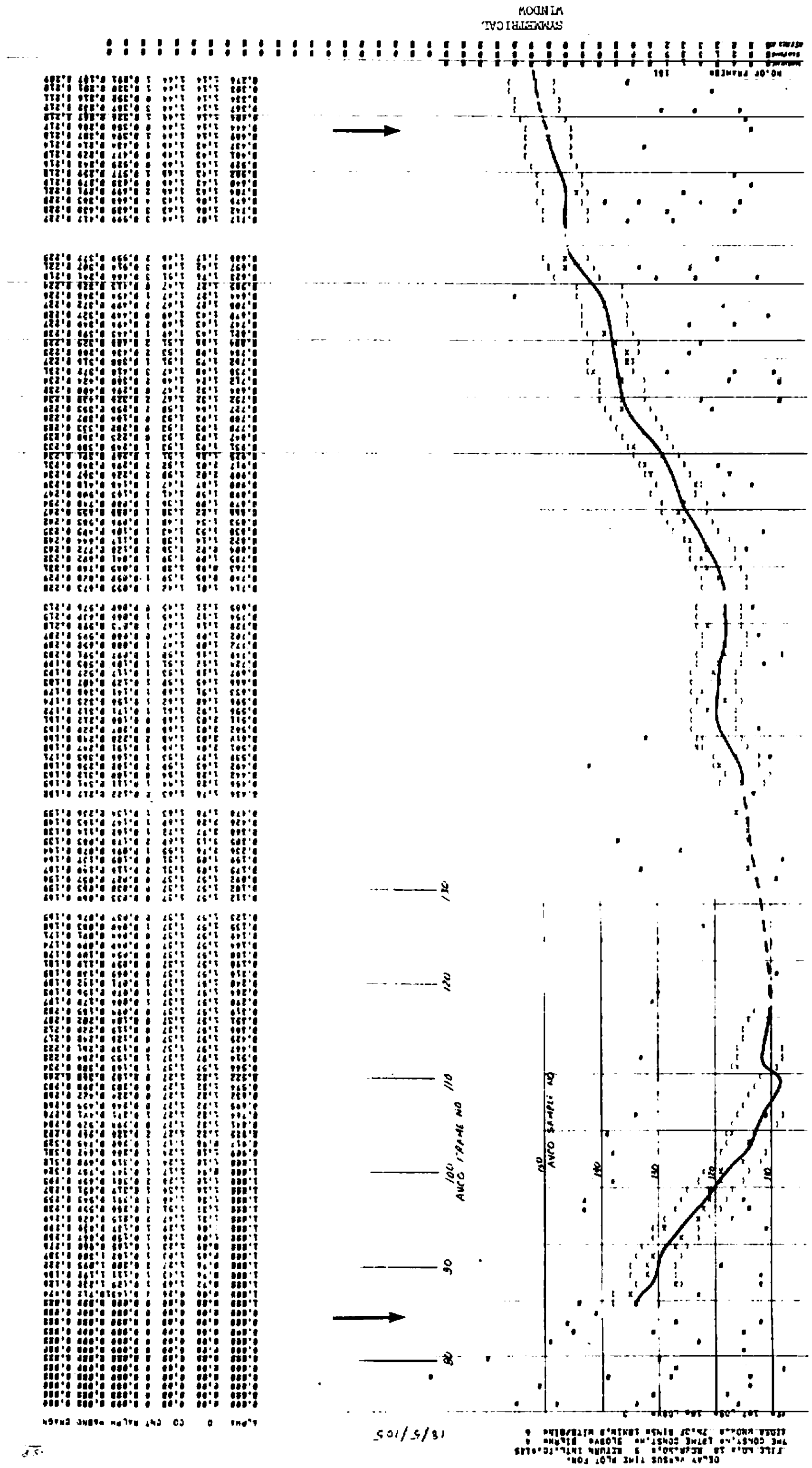

ACOUSTOGRAM CH. 5
SAMPLE
900
800
700
600
500
400
300
200
100
0
0
50
100
150
200
250
300

Figure 27a



Improved Track from Figure 27a (~100 samples into frame)

Figure 27b

ACOUSTOGRAM CH. 4

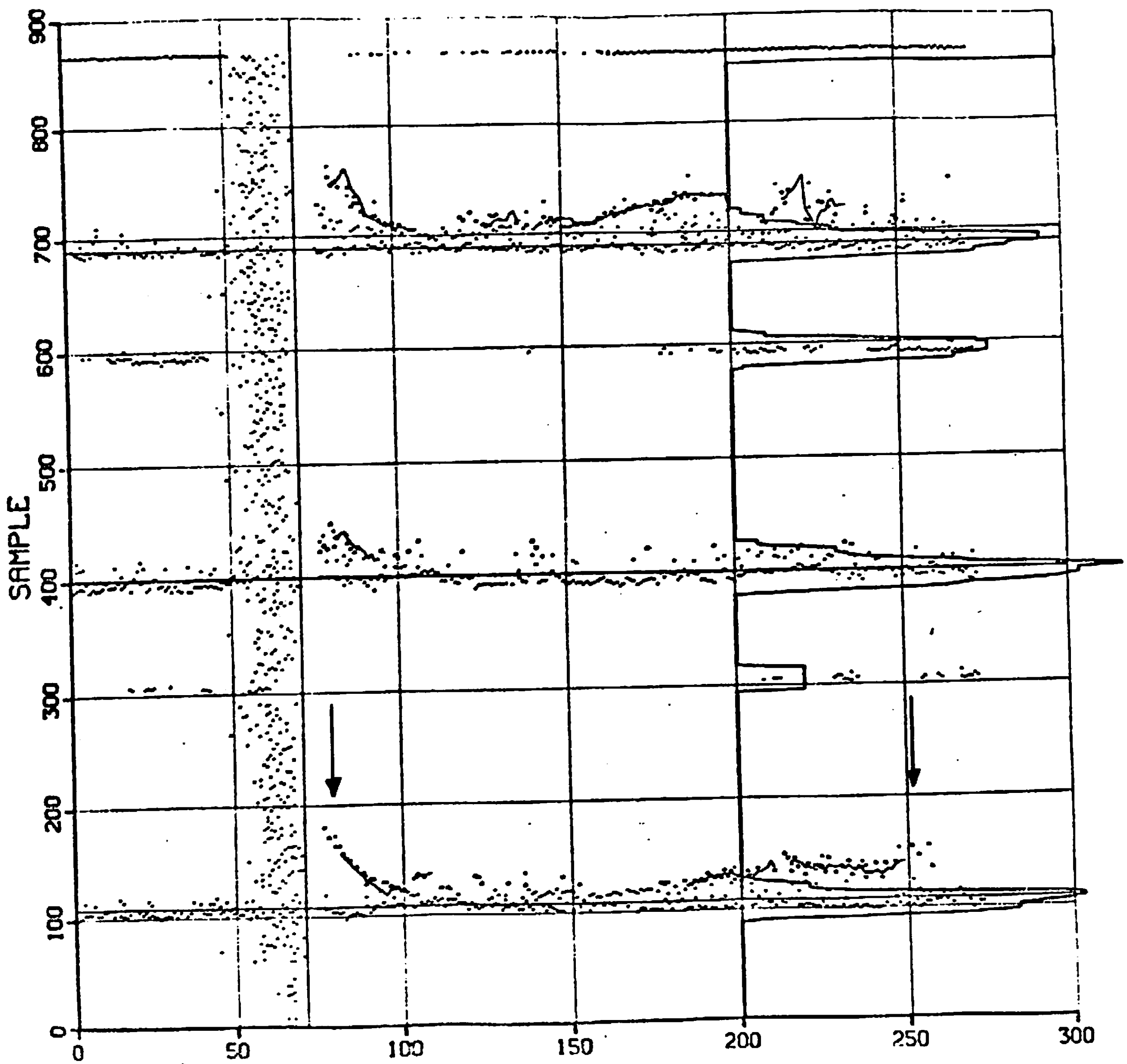
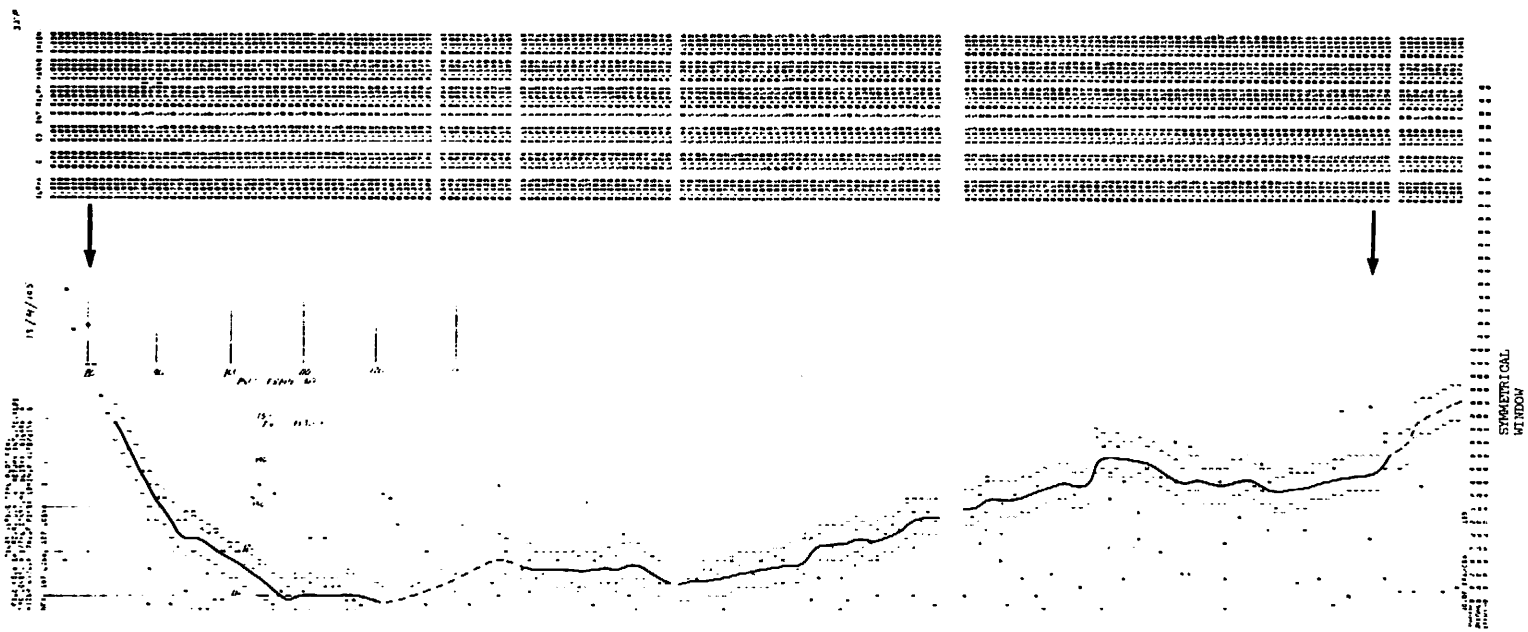
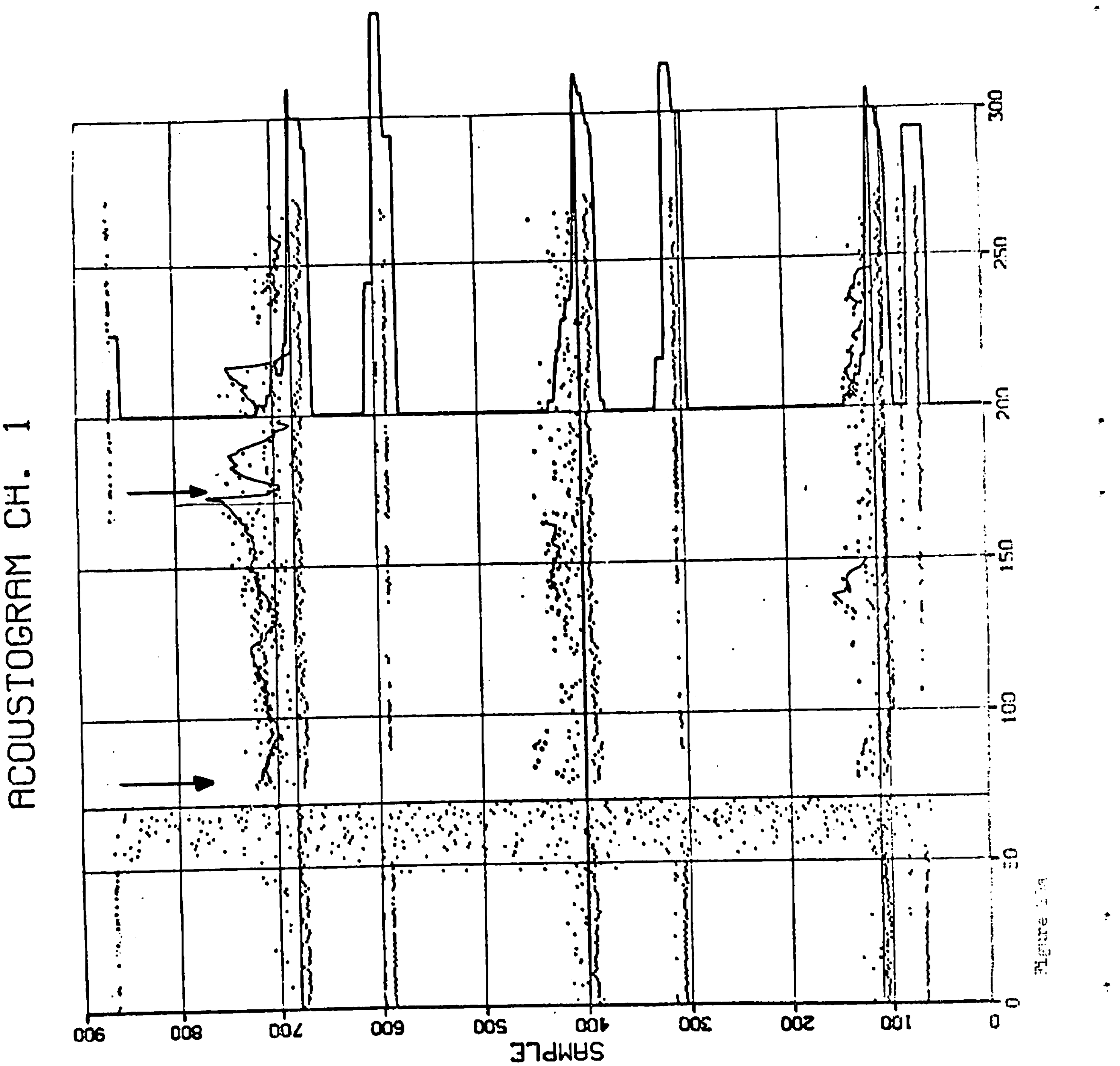


Figure 28a

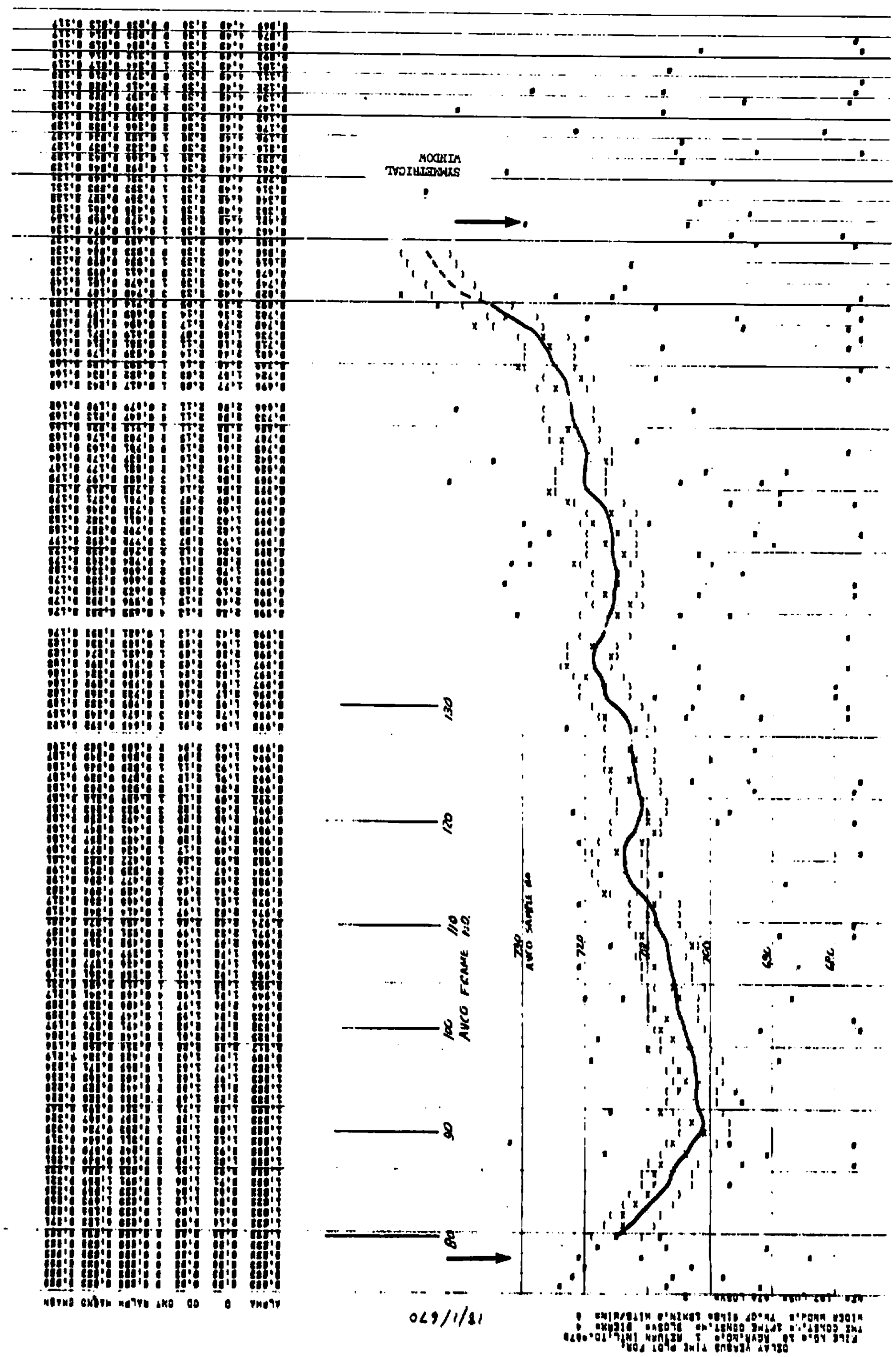


Improved Track from Figure 28a (~100 samples into frame)

Figure 28b


ACOUSTOGRAM CH. 1
SAMPLE
0
100
200
300
400
500
600
700
800
900
50
100
150
200
250
300

A-72



Improved Track from Figure 29a (~700 samples into frame)

Figure 29b

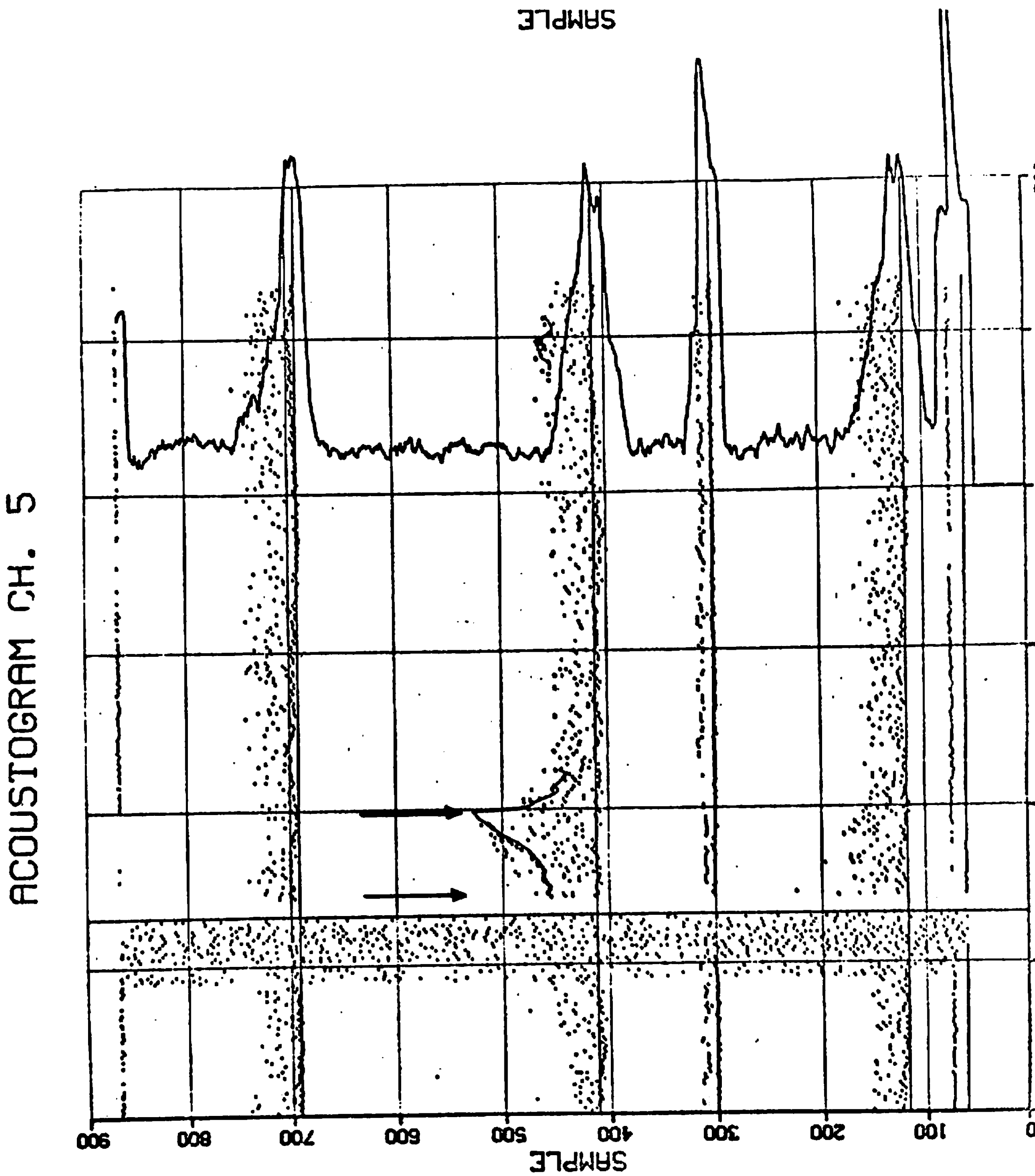
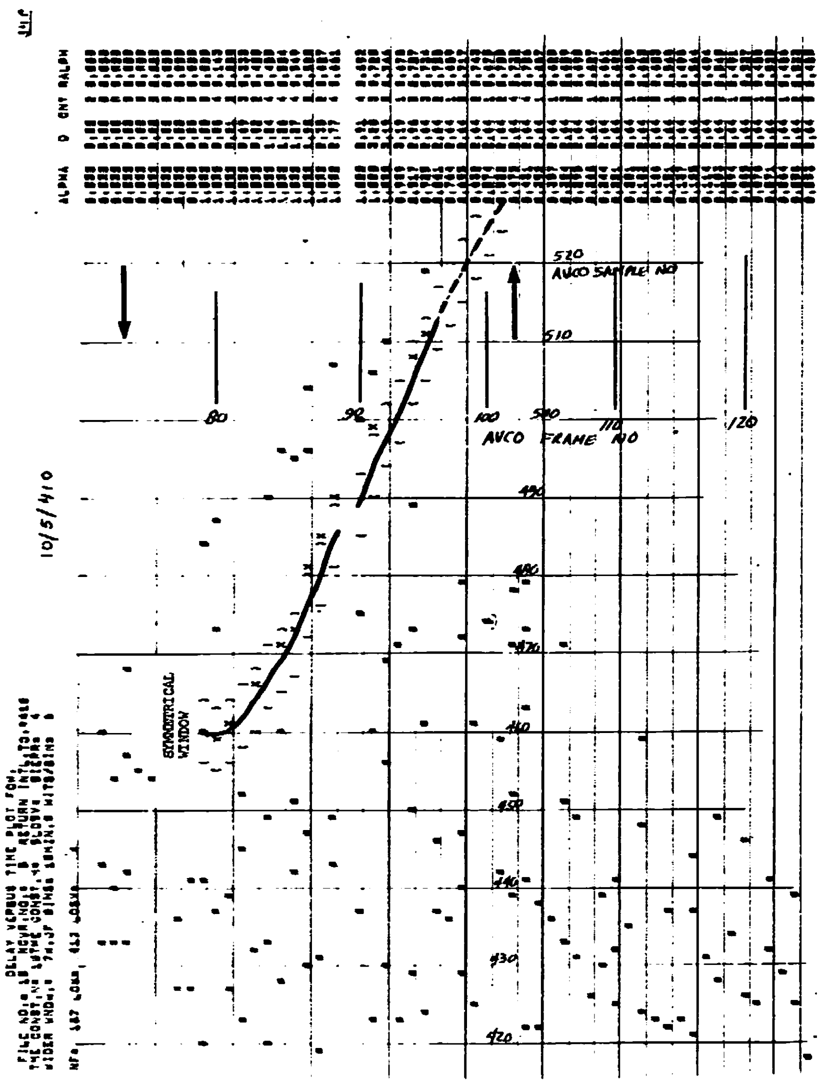
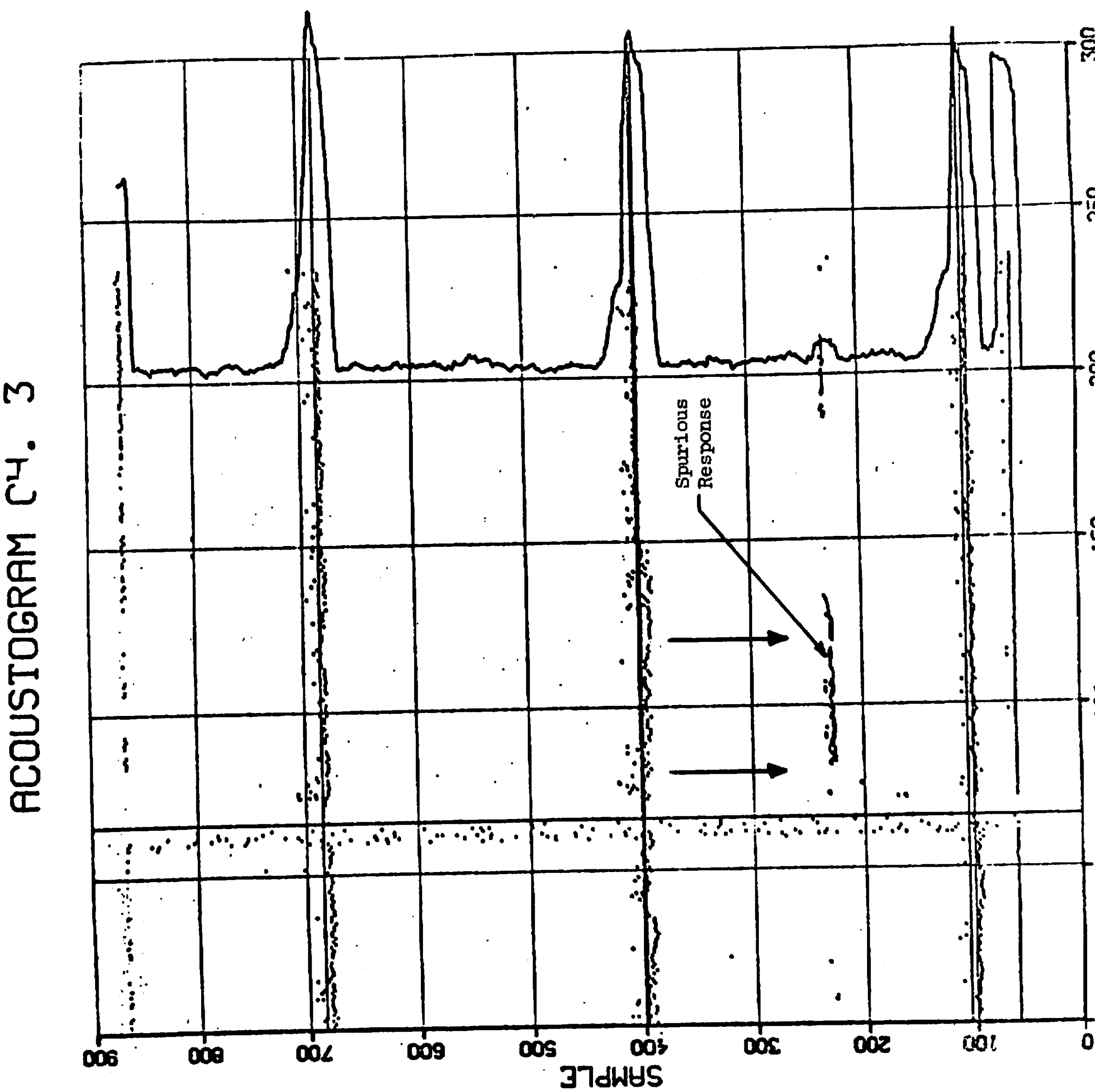

ACOUSTOGRAM CH. 5
SAMPLE
0
100
200
300
400
500
600
700
800
900
SAMPLE
0
50
100
150
200
250
300

Figure 30a



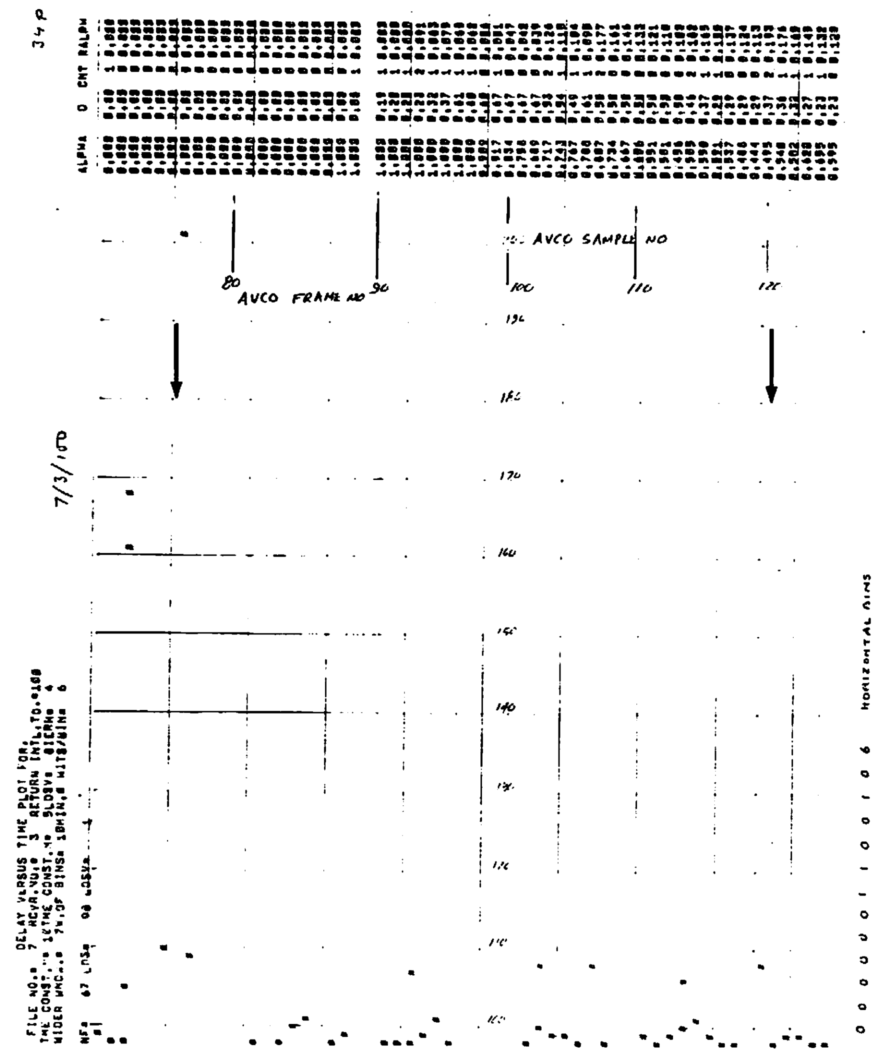
Improved Track from Figure 30a (~400 samples into frame)

Figure 30b



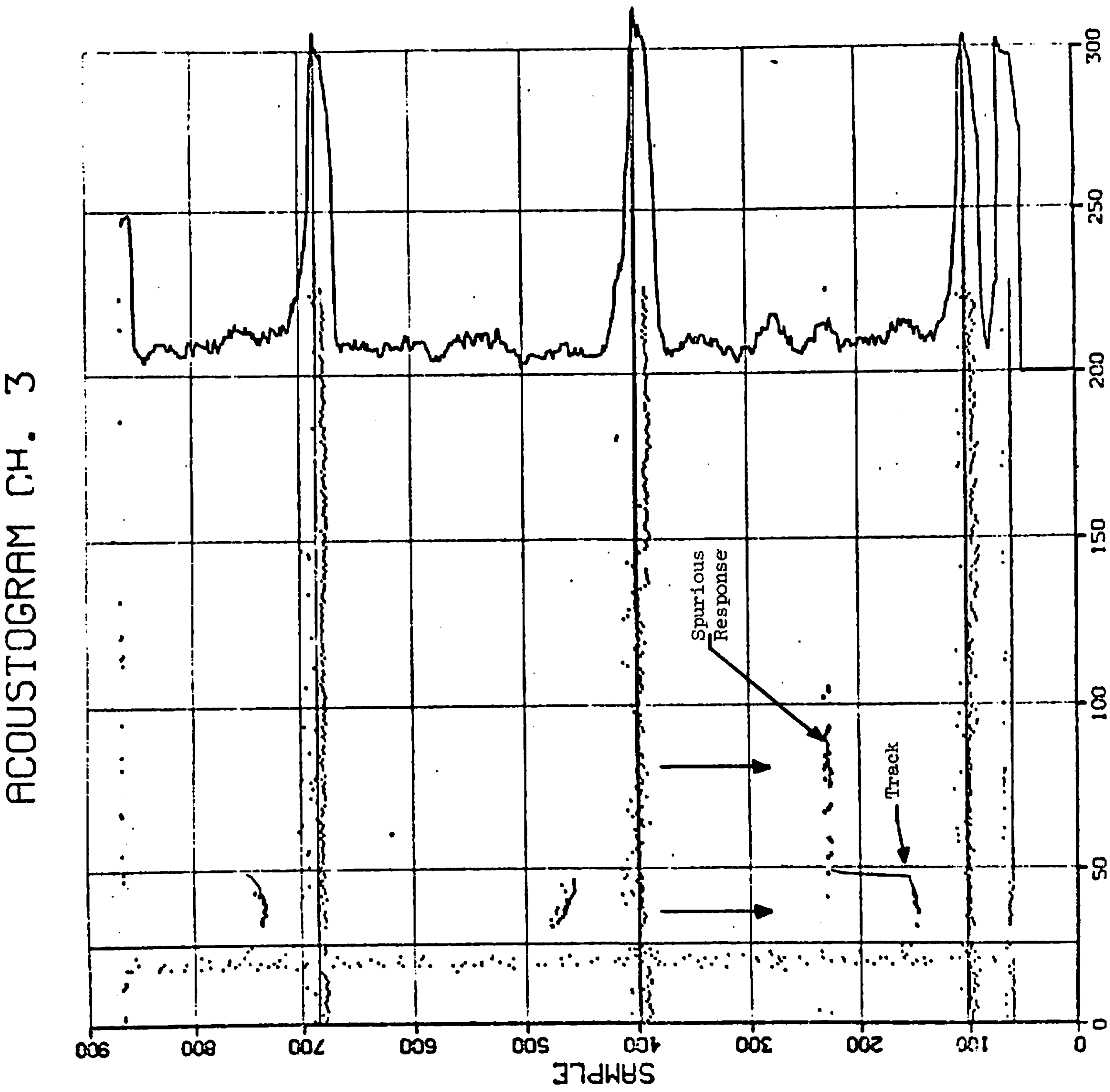
ACOUSTOGRAM CH. 3
Spurious Response
SAMPLE
0
100
200
300
400
500
600
700
800
900
0
50
100
150
200
250
300

Figure 31a



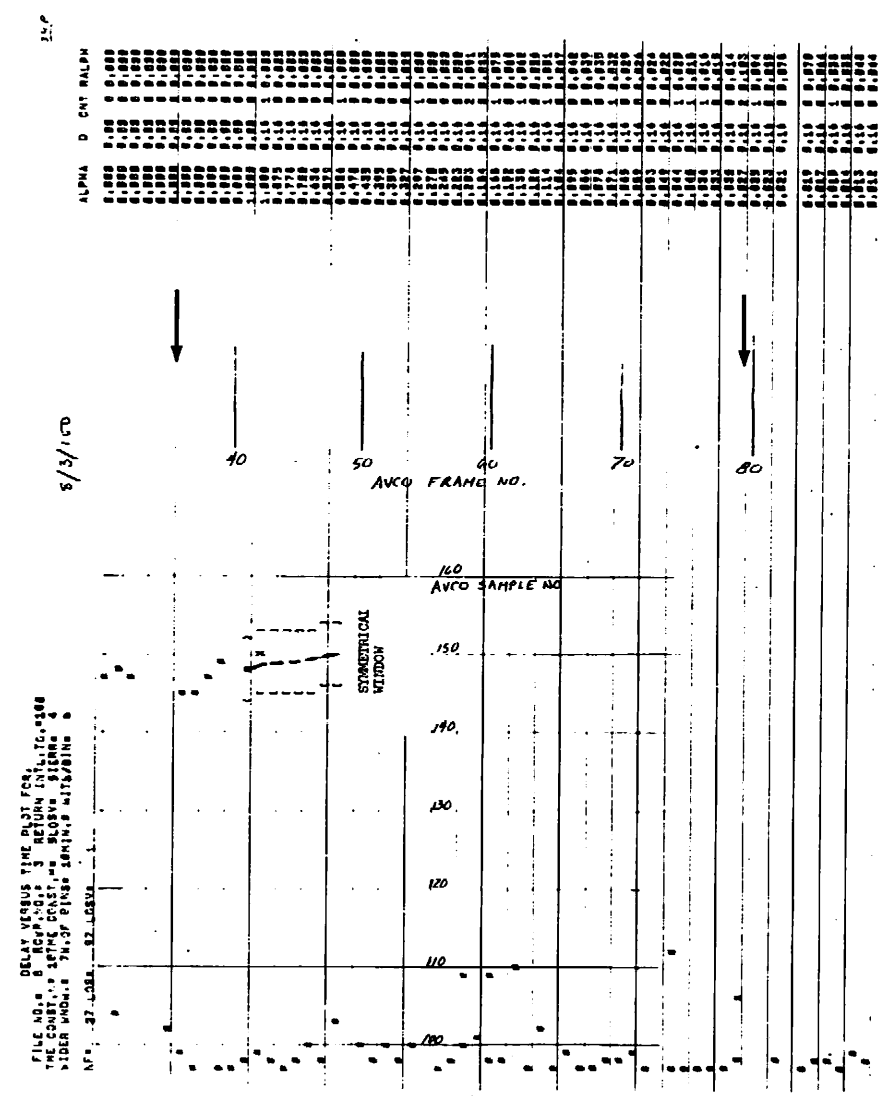
Rejection of Spurious Response, Figure 31a (~230 samples into frame)

Figure 31b



ACOUSTOGRAM CH. 3
SAMPLE
0
100
200
300
400
500
600
700
800
900
0
50
100
150
200
250
300
Spurious Response
Track

Figure 32a



Improved Track from Figure 32a (~100 samples into frame)

Figure 32b

NOT USED

Section 3

CONCLUSIONS AND RECOMMENDATIONS

The objectives of this effort were to come up with an algorithm for the tracking of LOS and vortex delay which did not have any of the problems associated with the basic minimum mean square error (MMSE) tracker, so that it can eventually be implemented in real-time on a minicomputer such as PDP-11. In spite of the fact that the processed and digitized data did not contain the amplitude and bandwidth (or pulse width) information, which were considered to be serious handicaps, all the goals of this effort have been accomplished and suggestions for improvement and further work are also made.

Starting with the basic MMSE tracker, all the problems associated with it were individually analyzed and various solutions tried. Successful solutions consisting of the Start Routine, tighter tracking with non-symmetric window, LOS masking and temporary stop were implemented and tried on a wide cross-section of track data from 15 landings. Furthermore the solutions consisted of simple procedures done in fixed point arithmetic which can easily be added to the basic tracker with realistic storage and time requirements.

For further improvements in the rejection rate for the noise only situations it is recommended that the modified noise density criterion mentioned in Section 2.3.1.5 be implemented. It would also be interesting to further pursue the fixed point arithmetic suggested in Section 2.3.3 for the basic tracker by implementing the total modified tracker in fixed point arithmetic and trying it on the same 15 landings or any other later improved data. Real-time implementation of the final algorithm coded in machine language on a minicomputer such as PDP-11 would also be of great value.

Appendix A

RUNNING MMSE DELAY TRACKER CALCULATIONS

N = Time constant of running average of track density

M = Time constant of running MMSE tracker

i = Frame number

Y_i = Value of the delay point in frame i

$$\beta_i \begin{cases} =1 & \text{if a hit occurs in frame } i \\ =0 & \text{otherwise} \end{cases}$$

TRACK DENSITY, $$\alpha_i = \frac{N \cdot \alpha_{i-1} + \beta_i}{N + 1}$$

TC MODIFICATION, $$M_s = \alpha_i \cdot M$$

FRAME NUMBER, $$XS_i = \frac{M_s \cdot XS_{i-1} + i}{M_s + 1}$$

DELAY, $$YS_i = \frac{M_s \cdot YS_{i-1} + Y_i}{M_s + 1}$$

(FRAME)·(FRAME) $$XSS_i = \frac{M_s \cdot XSS_{i-1} + i^2}{M_s + 1}$$

(FRAME)·(DELAY), $$XYS_i = \frac{M_s \cdot XYS_{i-1} + i \cdot Y_i}{M_s + 1}$$

SLOPE, $$A = \frac{XYS_i - XS_i \cdot YS_i}{XSS_i - XS_i \cdot XS_i}$$

INTERCEPT, $$B = YS_i - XS_i \cdot A$$

DELAY ESTIMATE, $$Y_i = B + A \cdot i$$

DELAY ERROR, $$D_i = \frac{M_s \cdot D_{i-1} + |\hat{Y}_i - Y_i|}{M_s + 1}, \text{ if } B_i = 1$$

$$D_i = D_{i-1} \qquad , \text{ if } B_i = 0$$

Appendix B

INITIAL VALUES OF THE TRACKER PARAMETERS

Assume the Start Routine gives,

1) hit threshold = MAXH
2) winning delay = MAXI
3) winning bin slope = SLOPE

Track Density, $\alpha_{IN} = 1$

Frame Number, $XS_{IN} = (MAXH + 1)/2$

Delay, $YS_{IN} = (MAXI - SLOPE \cdot (MAXH - 1)/2$

(Frame) · (Frame), $XSS_{IN} = (MAXH+1)(2 \cdot MAXH+1)/6$

(Frame) · (Delay), $XYS_{IN} = \frac{MAXI\,(MAXH+1)}{2} - \frac{SLOPE}{MAXH} \cdot \left\{ \sum_{i=1}^{MAXH-1} i \cdot (MAXH-i) \right\}$

Delay Error, $D_{IN} = 0$

NOT USED

Appendix C

WAKE VORTEX PROGRAM

```
      DIMENSION IBETA(500),ALPHA(500),XS(500),YS(500)
      DIMENSION XSS(500),XYS(500),D(500),D1(500)
      DIMENSION YSS(500)
      DIMENSION IDATA(498,2),IX(500)
      DIMENSION JX(500),JS(500),JSU(500)
      DIMENSION Y(500),Y1(500),YS1(500),XYS1(500)
      DIMENSION RALPH(500),IRBET(500)
      DIMENSION CALPH(500),CD(500),CRALP(500)
      DATA I3/1H /,IS/1H*/,IST/1HX/,IS1/1H(/,IS2/1H)/
  340 FORMAT (1H1,T14,'DELAY VERSUS TIME PLOT FOR,')
  330 FORMAT ('  FILE NO.=',I3,'  RCVR.NO.=',I3,'  RETURN INTL.TO.=',I3)
  350 FORMAT(' TME CONST.N=',I3,'TIME CONST.M=',I3,'LOSV=',I3,'IERR=',I3)
  360 FORMAT(' WIDER WNDW.=',I3,'W.OF BINS=',I3,'MIN.# HITS/BIN=',I3)
  370 FORMAT(1X,T79,'A'T93,'B'T95,'ESTY'T126,'ALPHA'T116,'ERRY'T127,'D')
  380 FORMAT(1X,T95,'ALPHA'T103,'D    CD  CNT'T116,'RALPH MAGND CMAGN')
  310 FORMAT (I12)
  230 FORMAT (1H*,92A1,F6.3,F6.2,F6.2,I3,F6.3,F6.3,F6.3)
  320 FORMAT (' NO.OF FRAMES=',I10)
  450 FORMAT(1X,40I3)
  460 FORMAT(1X,20I3)
  321 CONTINUE
      TYPE 250
  250 FORMAT(' IFLE ?ICVR ?INITL ? = I3')
      ACCEPT 252,IFLE,ICVR,INITL
      IF(IFLE-10) 402,402,404
  402 CONTINUE
  251 FORMAT (' NCONST, ?MCONST ?LOSV ?IERR ? = I3')
      N=10
      M=5
      IERR=4
  253 FORMAT(' WIDER WNDW.? WIDTH OF BINS? MIN.# OF HITS/BIN? = I3')
      IWDE=7
      IBW=10
      MNH=5
      IWW=130
      MIF1=5
      ISLP=3
      IWS=20
      NS=40
      NIB=IWW/IBW
      IBW1=IBW-1
      IBW2=IBW/2
  252 FORMAT (I3)
      PRINT 340
      PRINT 330,IFLE,ICVR,INITL
      PRINT 350, N,M,LOSV,IERR
      PRINT 360, IWDE,IBW,MNH
C     START,END,CHANGE OF FRAME,& RECEIVER NO. INFO. IN THE ADDRESS
      IADDR=2048
      ISTRT=IADDR+15
      IEND=IADDR+14
      IFRAM=IADDR
      ICVR=IADDR+ICVR
C     INITIALIZE ALL STATISTICAL VARIABLES (FRAME,DELAY,ERROR,ETC.)
      ALPHA(0)=0.
      XS(0)=0.
      YS(0)=INITL
      YS(1)=INITL
      YSS(0)=0.
      YSS(1)=0.
```

```
      XSS(0)=0.
      XYS(0)=0.
      D(0)=0.
      CD(0)=0.
C     INITIALIZE ALL BINS TO ZERO HITS
      DO 412 I=1,22
  412 JX(I)=0
      DO 413 I=1,320
      JSO(I)=0
  413 JS(I)=0
C     INITIALIZE THE RETURN FOR FIRST FRAME TO BLANKS
      DO 226 I=1,IUW
  226 IX(I)=IB
C     INITIALIZE START,RECORD NO.,FRAME NO.,THRESHOLD-X-ING,&-DELAY HIT
      JSTRT=0
      MSTRT=0
      JSTOP=0
      IRST=0
      NDEL1=133
      NDEL2=133
      MAXJ=0
      IREC=1
      IBSF=0
      NF=1
      IBETA(1)=0
      IBB=0
      IBS=0
      CALL DEFINE FILE (1,996,1R,'DATA.DAT',0,0)
  212 READ(1#IREC)IDATA
  100 FORMAT(' IREC =',I2,T11,'|'T21,'|'T31,'|'T41,'|'T51,'|'T61,'|')
C     GO THROUGH 498 ADDRESSES AND 498 DATA POINTS PER RECORD
      DO 202 I=1,498
      J=IDATA(I,1)
      IF(JSTRT-1) 246,257,248
  246 IF(J-ISTRT) 241,244,241
  241 IF(J.NE.IFRAM) GO TO 247
      NF=NF+1
      IBS=0
      YS(NF)=YS(NF-1)
      YSS(NF)=YSS(NF-1)
  247 IF(J.NE.ICVR) GO TO 202
      IDEL=IDATA(I,2)
      JDEL=IDEL-INITL
      JDA=IABS(JDEL)
      IF(JDA.GE.INS) GO TO 202
      IF(NF.GT.NS) GO TO 618
      N1=NF-1
      GO TO 622
  618 N1=NS
  622 CONTINUE
      YS(NF)=(N1*YS(NF-1)+IDEL)/(N1+1)
      EST1=YS(NF)
      LOSE1=ABS(EST1-IDEL)
      IF(IBS.EQ.0) GO TO 243
      IF(LOSE1.GE.LOSE) GO TO 245
  243 LOSE=LOSE1
      EST=EST1
      IBS=1
      YSS(NF)=(N1*YSS(NF-1)+LOSE)/(N1+1)
  245 LOS=EST
```

A-89

```
      LOSV=YSS(NF)
      GO TO 222
  200 FORMAT(' NF=',I5,' LOS=',I5,' LOSV=',I5)
  244 JSTRT=1
      PRINT 302
      PRINT 200,NF,LOS,LOSV
      NF=0
      LOSV=2*LOSV
      LOSV=MAX0(LOSV,5)
      LOSV1=LOSV
      LOSV=20
      GO TO 242
  257 IF(J.EQ.IFRAM) NF=NF+1
      IF(NF.LT.4) GO TO 202
      JSTRT=2
      NF=1
      GO TO 222
  240 IF (J-IFRAM) 232,213,238
  213 IF(MSTRT-1) 442,444,444
  442 NDEL1=133
      NDEL2=133
      IF(NF.GE.26) GO TO 232
      IF(MAXJ.LT.MNH) GO TO 446
      MSTRT=1
      ESTY=MAXI
      NDEL1=ESTY-IERR
      NDEL2=ESTY+IERR
      LOSV=LOSV1
      NF=MNH
      NFM1=NF-1
      ALPHA(NF)=1.
      XS(NF)=(NF+1.)/2.
      YS(NF)=MAXI-A*NFM1/2.
      XSS(NF)=(NF+1.)*(NF*2.+1.)/6.
      ISUM=0
      DO 472 KIN=1,NFM1
      MIN=NF-KIN
      ISUM=ISUM+KIN*MIN
  472 CONTINUE
      XYS(NF)=MAXI*(NF+1.)/2.-A*ISUM/NF
      D(NF)=0.
      CD(NF)=0.
      B=YS(NF)-XS(NF)*A
      GO TO 446
  444 IBSF=IBSF+1
      IF(IBSF-8) 446,464,464
  464 JSTOP=1
      NFST=0
  446 CONTINUE
      IF((JSTOP.EQ.1).AND.( IBST.GE.MNH)) GO TO 232
      IF(JSTOP.LT.1) GO TO 466
      NDEL1=133
      NDEL2=133
  466 CONTINUE
      IX(NDEL1)=IS1
      IX(NDEL2)=IS2
      BM=0.
      CBM=0.
      IF(D(NF).GT.0.01) BM=ALPHA(NF)/((RALPH(NF)+1.)*D(NF))
      IF(CD(NF).GT.0.01) CBM=CALPH(NF)/((CRALP(NF)+1.)*CD(NF))
```

```
C      PRINT VALUES OF THE DATA POINTS EVERY CHANGE OF FRAME
       PRINT 232,((X(L),L=1,92),ALPHA(NF),D(NF),
      &CD(NF),ICNT,RALPH(NF),BM,CBM
       IF(JSTOP.EQ.1) NFST=NFST+1
       IF(NFST.GE.50) GO TO 232
       NF=NF+1
C      INITIALIZE THE RETURN FOR EACH FRAME TO BLANKS
       DO 312 K=1,IWW
  312  IX(K)=IB
C      INITIALIZE DELAY HIT,THRESHOLD X-ING,AND DELAY ERROR FOR EACH FRAM
       IBB=0
       IBETA(NF)=0
       ICNT=0
       IRBET(NF)=0
C      N1=NF-1, M1=NF-1, FOR NF < OR = THE TWO TIME CONSTANTS N AND M
       IF(NF-N) 716,716,718
  716   N1=NF-1
       GO TO 722
  718  N1=N
  722  CONTINUE
       IF(NF-M) 724,724,726
  724  M1=NF-1
       GO TO 728
  726  M1=M
  728  CONTINUE
C      UPDATING OF THE STATISTICAL VARIABLES AT EACH THRESHOLD CROSSING
       RALPH(NF)=(N1*RALPH(NF-1)+IRBET(NF))/(N1+1)
       CRALP(NF)=((NF-1)*CRALP(NF-1)+IRBET(NF))/NF
       CALPH(NF)=((NF-1)*CALPH(NF-1)+IBETA(NF))/NF
       ALPHA(NF)=(N1*ALPHA(NF-1)+IBETA(NF))/(N1+1)
       AMS=ALPHA(NF)*M1
       XS(NF)=XS(NF-1)
       YS(NF)=YS(NF-1)
       XSS(NF)=XSS(NF-1)
       XYS(NF)=XYS(NF-1)
       ESTY=B+NF*A
       NDEL1=ESTY-IERR
       NDEL2=ESTY+IERR
       D(NF)=D(NF-1)
       CD(NF)=CD(NF-1)
       GO TO 202
  203  IF(J-IEND) 234,232,202
  234  IF(J-ICVR) 202,204,202
  204  IDEL=IDATA(I,2)
       JDEL=IDEL-LOS
C      ACCEPT DATA POINT FOR GIVEN RCVR,IF WITHIN GIVEN DELAY INTERVAL
       IF(JDEL-IWW) 206,206,202
  206  IF(JDEL) 202,202,207
  207  IX(JDEL)=IS
C      EXCLUDE THE DATA POINT IF TOO CLOSE TO GROUND (LOSV)
       IF(JDEL-LOSV) 202,202,209
  209  CONTINUE
       IF(MSTRT-1) 434,436,436
  434  CONTINUE
  422  J1=JDEL+IBW1
       J2=J1/IBW
       J3=JDEL+ISLP*NF+IBW1
       J4=J3/IBW
       J5=JDEL+IBW2+IBW1+ISLP*NF
       J6=J5/IBW
```

```
      JX(J2)=JX(J2)+1
      JS(J4)=JS(J4)+1
      JSO(J6)=JSO(J6)+1
      IF(JS(J4)-JX(J2)) 452,454,454
  452 IF(JX(J2)-JSO(J6)) 451,451,453
  453 IF(JX(J2)-MAXJ) 426,426,432
  432 MAXJ=JX(J2)
      MAXI=JDEL
      A=?.
  426 CONTINUE
      GO TO 457
  454 IF(JS(J4)-JSO(J6)) 451,451,455
  455 IF(JS(J4)-MAXJ) 456,456,458
  458 MAXJ=JS(J4)
      MAXI=JDEL
      A=0.-ISLP
  456 CONTINUE
      GO TO 457
  451 IF(JSO(J6)-MAXJ) 457,457,459
  459 MAXJ=JSO(J6)
      MAXI=JDEL
      A=0.-ISLP
  457 CONTINUE
      GO TO 202
  436 CONTINUE
      ICNT=ICNT+1
      IF(ICNT,GE,2) IRBET(NF)=1
      RALPH(NF)=(N1*RALPH(NF-1)+IRBET(NF))/(N1+1)
      CRALP(NF)=((NF-1)*CRALP(NF-1)+IRBET(NF))/NF
C     IS THE HIT WITHIN WIDER WINDOW AROUND ESTY?
      ILOW=ESTY-IWDE
      IUPP=ESTY+IWDE+7
      IF(JDEL,LT,ILOW) GO TO 202
      IF(JDEL,GT,IUPP) GO TO 202
C     THRESHOLD CROSSING CHANGED TO 1 FOR ACCEPTED POINT
      IBETA(NF)=1
C     N1=NF-1, M1=NF-1, FOR NF < OR = THE TWO TIME CONSTANTS N AND M
      IF(NF-N) 516,516,518
  516  N1=NF-1
      GO TO 522
  518 N1=N
  522 CONTINUE
      IF(NF-M) 524,524,526
  524 M1=NF-1
      GO TO 528
  526 M1=M
  528 CONTINUE
C     UPDATING OF THE STATISTICAL VARIABLES AT EACH THRESHOLD CROSSING
      CALPH(NF)=((NF-1)*CALPH(NF)+IBETA(NF))/NF
      CAMS=CALPH(NF)*(NF-1)
      ALPHA(NF)=(N1*ALPHA(NF-1)+IBETA(NF))/(N1+1)
      AMS=ALPHA(NF)*M1
      XS(NF)=(AMS*XS(NF-1)+NF)/(AMS+1.)
      Y1(NF)=JDEL
      YS1(NF)=(AMS*YS(NF-1)+Y1(NF))/(AMS+1)
      XSS(NF)=(AMS*XSS(NF-1)+NF*NF)/(AMS+1)
      XYS1(NF)=(AMS*XYS(NF-1)+NF*Y1(NF))/(AMS+1)
      IF(NF-1) 564,564,562
  564 A1=A
      B1=B
```

```
      GO TO 563
  562 A1=(XYS1(NF)-XS(NF)*YS1(NF))/(XSS(NF)-XS(NF)*XS(NF))
      B1=YS1(NF)-XS(NF)*A1
  563 CONTINUE
      ESTY1=B1+NF*A1
      ERRY1=ABS(ESTY1-Y1(NF))
      D1(NF)=(AMS*D(NF-1)+ERRY1)/(AMS+1)
      IF(IBB-1) 552,554,554
  554 IF(ERRY1-ERRY) 553,553,544
  553 IX(JDG2)=IS
  552 Y(NF)=Y1(NF)
      YS(NF)=YS1(NF)
      XYS(NF)=XYS1(NF)
      A=A1
      B=B1
      ESTY=ESTY1
      ERRY=ERRY1
      D(NF)=D1(NF)
        CD(NF)=(CAMS*CD(NF-1)+ERRY)/(CAMS+1)
      NDEL1=ESTY-IERR
      NDEL2=ESTY+IERR
      IBB=1
      JDG2=JDEL
      IX(JDEL)=IST
      IF(ERRY-IERR) 502,502,505
  502 IBSE=0
      IF(JSTOP.LT.1) GO TO 505
      IRST=IRST+1
      IF(IRST.GE.MNH) JSTOP=0
  505 CONTINUE
  544 CONTINUE
  202 CONTINUE
      IREC=IREC+1
      GO TO 212
  232 CONTINUE
      PRINT 320,NF
      PRINT 460, (JX(IK),IK=1,20)
      PRINT 450, (JS(IL),IL=1,40)
        PRINT 450, (JSC(IM),IM=1,40)
      GO TO 321
  404 CONTINUE
      STOP
      END
```

NOT USED

REFERENCES

1. *Pulsed Acoustic Vortex Sensing System Design Evaluation Report,* AVCO Government Products Group, AVCO Systems Division; for Department of Transportation, Transportation Systems Center, Cambridge, Mass. under Contract No. DOT-TSC-620.

2. *Aircraft Wake Vortex Sensing Systems,* D. Burnham, M. Gorstein, J. N. Hallock, R. Kodis, T. Sullivan and I. G. McWilliams, Transportation Systems Center, Cambridge, Mass., June 1971, Technical Report for DOT/FAA.

3. *Vortex Sensing Tests at NAFEC,* D. Burnham, J. Hallock, R. Kodis, T. Sullivan, Transportation Systems Center, Cambridge, Mass., January 1972, Technical Report for DOT/FAA.

4. *Vortex Sensing Tests at Logan and Kennedy Airports,* T. Sullivan, D. Burnham, R. Kodis, Transportation Systems Center, Cambridge, Mass., December 1972, Final Report for DOT/FAA.

5. *Extracts from a Proposal for Pulsed Acoustic Vortex Sensing System,* AVCO Government Products Group, AVCO Systems Division; for Department of Transportation, Transportation Systems Center, Cambridge, Mass. under Request for Proposal TSC/TEC-0062-GF, 7 December 1972.

APPENDIX B

Vortex Tracker Report

Arcon Corporation, R74-2W, dated 18 March 1974

R74-2W

VORTEX TRACKER REPORT

18 MARCH 1974

TABLE OF CONTENTS

VORTEX TRACKER REPORT

General Status

The following brief report is an account of a new design for the delay tracker in the AVCO pulsed vortex tracking system. The philosophy of this design is to produce an integrated and efficient algorithm which remedies some of the observed performance deficiencies of the current system. Among these difficulties are:

a. Excessive running time due to use of floating point arithmetic.

b. Erratic start-up of tracks.

c. Failure to correlate data with track in ambiguous situations.

The new system is a more consistent and powerful practical method for delay tracking.

Additionally, a simplified version of this general design has been programmed and checked out for use with the AVCO delay tracking simulator. The simplification has been to omit the secondary search procedure and the generation of split tracks in ambiguous data situations. The program has undergone preliminary tests (only as a means of checkout) and results are encouraging. No formal test results are available at this time.

Further development of this program would require parameter tuning to optimally match data characteristics. The full algorithm should also be implemented to assess the effectiveness of its additional features. Extensive testing should be performed to validate the algorithm performance with real data.

Design Outline

The following design description is organized around Figures 1 through 13. The remaining figures provide some background information.

Figure 1. Delay Tracker Features

All fixed point processing is used to minimize computer running time. The smoothing algorithm is such that special scaling to preserve accuracy is unnecessary and it may be programmed in fixed point directly as written. (Delays from zero to maximum are accommodated with sufficient accuracy by the fixed point fractions 0 to 1.)

The smoothing algorithm is computationally very efficient and estimates track position and velocity recursively. It is a Kalman-Bucy filter based on an idealized model of track dynamics but should be empirically useful in the present context.

Each receiver/transmitter combination is handled separately for tracking. For each combination several tracks may be held simultaneously. This permits tracking of ghosts or baseline in addition to the main track; but, more importantly, if one track inadvertently follows false data, a second track can simultaneously acquire and follow the true data. The choice between them is made on the higher level when their data are selected for vortex calculations.

In order to select the best from among several tracks, running measures of the status and quality of a track must be kept. These measures are:

a. Type status – The track is being initiated upon (still tentative), it is undergoing normal tracking (normal), or it is a split (parent or trial).

b. Firmness – A measure of the past consistency of data-track correlations which controls smoothing constant and bin size selection.

c. Hit/Miss Count – A count of consecutive hits for initiation decisions or of consecutive misses for drop decisions.

Multiple tracking and a variety of data situations are handled by an extended correlation logic. This logic implements a continuing auto-initiation of new tracks on any data frame. It also selects the best data for each track and performs all track status decisions. In particular, when data of a normal track is missing in the primary bin, it constructs a larger secondary bin. A unique datum in the secondary bin leads to a split trial track, while the original track is also extrapolated (the parent). This situation is resolved on subsequent frames.

Figure 2. Input Processing

Input processing is executed once per frame.

Input data is screened to eliminate known fixed targets and the baseline returns (if desired). This is accomplished as in the current system by assigning gating times which are used to suppress all delay reports in the desired time intervals.

The data is next sorted into a set of input tables, one for each of the 36 receiver/transmitter combinations. Each data word consists of a delay time and a separate bit (normally zero) which can be set to indicate that the data has been utilized for track processing.

Figure 3. Track Files

Track information is kept in a set of track files, one for each receiver/transmitter combination. Each file has room for a maximum of several tracks (four tracks have been alloted in the initial program version). Each track slot consists of three words.

The first track word (0) contains information on track status. Bit 0 is an occupancy indicator showing whether or not the track slot is being used. If the slot is not occupied, the additional information in it is old and of no consequence. Bits 1-2 indicate the status type of the track (see Figure 8). Bits 3-5 are used if this is a parent track (for a track split). They indicate which of the tracks in the file is the trial (split) track associated with this parent. Bits 6-10 hold the firmness of the track (see Figure 5). Bits 11-15 contain the hit/miss count which is utilized in track initiation and drop decisions (see Figure 8).

The second track word (1) contains the smoothed "position" (actually time delay) estimate of the track. It is the quantity used directly in vortex position calculations.

The third track word (2) contains the smoothed velocity estimate which is necessary for the internal operation of the smoothing algorithm.

These estimates are fixed point quantities, fractions whose scaling can be fixed once and for all by considering the maximum delay as the unit of position. The velocity can be held as the per frame change of delay.

Figure 4. Smoothing Equations

The smoothing equations recursively update position and velocity estimates of each track using the track's data for each frame. The equations are a practical implementation of a two-state Kalman-Bucy filter.

The first step is to extrapolate the previous smoothed position, velocity to the present frame time (a one-frame advance). Then the deviation between the datum delay time d_n and the predicted position (delay time) $\hat{\hat{D}}_n$ is calculated, and fractions (α, β) of it are added to the predictions to produce the updated, new smoothed position, velocity.

In the case where no data is available on a frame, the extrapolated values are accepted as the new smoothed values. In such cases, when a new datum is finally found on a later frame, the β constant is modified by dividing by the number of frames since the last data was received (information available from the miss counter). Normally, for steady, no-miss data conditions, this divisor will be one.

Figure 5. Tracking Parameter Lookup

The selection of α, β smoothing constants is implemented by a table lookup which is keyed upon the track firmness F_n. Firmness ranges from 0 to 15 in steps of 1. When the first datum is received for a track, a firmness of 0 is assigned. Subsequent hits increment F_n by one, while misses decrement it by two. The subsequent F_n excursions are limited to the range 1-15.

As the firmness increases, the appropriate α_n, β_n decrease to provide increased smoothing. The maximum value of F_n which is permitted is a compromise between the maximum degree of smoothing desired vs. the need to follow quick variations in the delay dynamics. The value 15 is only an initial guess of the appropriate parameter.

Search bins sizes are also made functions of firmness. The values listed are only illustrative and may be calculated from the known standard deviation (σ) of delay noise. The bin size for $F_n = 1$ is additionally based on the expected maximum change of delay position per frame. Bin sizes, Δ_n, may require experimental adjustment, since the theory on which they are based oversimplifies the actual track dynamics.

Figure 6. α, β Values Used in Delay Tracker

The α, β values listed in the firmness table are taken from a standard curve which can be derived by Kalman theory. (Except for the $F_n = 0, 1$ values which are special.) We have,

$$\alpha_n = \frac{2(F_n + 1)}{(F_n + 1)(F_n + 2)} \qquad (F_n > 1)$$

$$\beta_n = \frac{2\alpha_n^2}{2 - \alpha_n + 2\sqrt{1 - \alpha_n}}$$

$$\Delta_n = \frac{4\sigma}{\sqrt{1 - \alpha_n}}$$

Figure 7. Equivalent Data Weights

The α, β smoothing is linear for constant α, β. Thus the impulse response of the filter, the weighting of the past data values to achieve a position (or velocity) estimate, can be exhibited. This figure shows the position estimate weights for the α, β at a steady firmness of 15. It reveals that the weights are approximately exponentially tapered over about 10 samples.

Figure 8. Track Status Transitions

The possible track statuses are tentative, normal, parent and trial.

A tentative track is created when data is initially received. It requires N consecutive hits to be verified as a real track (rather than noise). The status then changes to normal. If any miss occurs during the tentative period, the track is dropped and must be reacquired if new data arrives.

A normal track is maintained until too many consecutive misses are counted. It is then dropped and must be reacquired if data continues.

Under special circumstances (no data in primary search bin, one datum in larger secondary bin), the normal track splits into: a parent track which extrapolates without data, and a trial track which follows the data. A special α, β selection may be appropriate for this trial track smoothing, since a fast dynamic variation is indicated, and α, β should not be too small to follow it.

On the next frame, the trial track survives (becomes normal, and the parent is dropped) only if the parent receives no data in its primary bin and the trial receives one datum in its primary bin. Otherwise, the trial track is dropped and the parent reverts again to normal.

Neither parent, trial or tentative, tracks can be split in this manner, and secondary search is unnecessary for them.

Figure 9. Primary/Secondary Correlation

As noted above, primary search is applied to all tracks, and the additional secondary search is used only with normal tracks.

Bins are constructed by centering at the predicted position, $\hat{\hat{D}}_n$, for this frame. The primary tolerance is $\pm \Delta_n$, and the secondary, if needed, is $\pm 1.5\ \Delta_n$. The enlargement factor can be experimentally adjusted.

Figures 10-11-12-13. Correlation/Smoothing for Delay Tracks

These figures exhibit the proposed correlation/smoothing program flow logic. This logic is entered three times for each frame processing of a track file. Pass 1 processes the normal and parent tracks in a file. Pass 2 processes the trial tracks, and pass 3 processes the tentative tracks. This order is necessary because the various track categories have different priorities of access to the available data. Data which are used in any pass are so marked in the input file, and cannot be used by a subsequent track or in a subsequent pass.

The logic contains the features previously described. Some additional points are as follows:

a. Before changing a tentative track to normal, its final estimated velocity is checked against prescribed limits to reject all tracks with impossible velocities. (Bin size also limits excessive velocity excursions implied by the first two data points.)

b. If a normal track has more than one datum in the primary bin, that nearest to the prediction is selected. However, such a condition is counted as a partial miss (miss increment of 1) to prevent continued tracking through heavy multiple noise reports. (Ordinary misses increment the miss counter by 2.)

c. If a track file is full, splits cannot be accommodated and a normal track is continued as a missed data case (not shown).

d. Following the three passes through primary/secondary correlation, some unused data may remain in the input file. This data is utilized to set up new tentative tracks (initiation processing). The data is selected in order of the largest delay time (as is done in the current AVCO system) and set in the position estimate word of an unoccupied track slot. The velocity is set to zero. The occupied bit and other status information are initialized (status = tentative, $I_n = 1$, Hit count = 0). Note that the insertion of position data is equivalent to a smoothing step in which $\alpha = 1$, $\beta = 0$.

Background

Figures 14-21 provide some background material relating to the firmness control of α, β and bin sizes.

Figure 14. Variance Factor of Position Prediction Errors Due to Data Noise.

This figure gives contours of equal position prediction smoothing in the α-β plane. Of special note is the stable triangle of α, β values.

Outside this triangle the recursive filter is unstable. The standard deviation of position prediction (due to noise effects) is $K\sigma$ where σ is the standard deviation of data noise, and K^2 contours are exhibited in the figure. Small α, β, in general, imply heavy smoothing, but the smoothing effects are rather modest for the range of values considered in present design. The exact formula is

$$K^2 = \frac{2\beta + \alpha(2\alpha + \beta)}{\alpha(4-2\alpha-\beta)} .$$

Figure 15. Variance Factor Contours and Optimal α, β Curves

This figure restricts the α-β plane to the practical region (the unit square) and redraws the K^2 noise contours of the previous figures (now called K_1^2). In addition, contours for errors from another source are also plotted. These are errors produced by a simplified model of the track maneuver dynamics. The simple model assumes that the frame to frame accelerations are independently selected from a zero mean distribution with prescribed variance (a^2).

Optimal selection of α, β is based on keeping one source of error constant while minimizing the other (this is equivalent to minimizing a weighted sum of the two errors). A locus of appropriate α, β is thus generated by the points of mutual tangency of these two sets of contours. The "standard" α, β curve is thus produced. A specific α, β selection from this curve then depends on the relative weight given to the two error effects.

Figure 16. Optimal α, β Curves and Damping Contours

An alternate approach to α, β selection is to examine the transient response of the filter to abrupt changes in data. Pole-zero positions can be calculated and damping factors can be determined (for the envelope of samples). This analysis reveals that the desirable damping factor of $\zeta = .7$ produces an α, β curve close to the previously determined standard.

Figure 17. Optimal α, β Start-Up Sequence

The foregoing figures and analyses were based on steady-state assumptions. By utilizing the same models, the problems of start-up and missing data can also be analyzed. This figure shows the optimal start-up sequence of α, β (small circles) which begins at $\alpha = 1$, $\beta = 1$ and descends asymptotically to a particular steady-state point. By repeating this calculation with various relative weights on noise/maneuver error effects, other sequences can be generated. These α, β points all lie between the standard curve and the "start-up envelope" curve. Successive points fall somewhere on the heavy traces shown in the figure.

Since the start-up envelope is perturbed only very modestly from the standard curve, we adopt values of α, β on the latter also for start-up. The first point ($F_n = 1$) is an exception where the true start-up value $\alpha = 1$, $\beta = 1$ is utilized.

Figures 18-19

These figures illustrate a similar calculation for missed data in which the frames-since-last-data factor (T_n) used in velocity smoothing (see Figure 4) has been utilized. Starting from the steady-state condition (SS), it successively assumes that one miss, two misses, etc. occur. During the misses, the track is, of course, extrapolated. On the first datum after the misses, the optimal α, β are plotted (labeled 1M after one miss, 2M after two misses, etc.). On the second datum, the optimal α, β drop to near the steady-state value.

The calculations are made for two different steady-state starting conditions in the respective figures.

These results indicate that near optimal α, β for missed data conditions can be selected from the standard curve with a suitable firmness control and a T_n factor correction.

Figures 20-21

The optimal adjustment of bin sizes is based on a comparison of false report density with the probability density of the predicted position of the report. The width of the latter distribution is produced by the noise/maneuver errors described previously. If data is missed, the width increases but the height of the distribution decreases. At first this increases the appropriate bin, but finally, the bin narrows and vanishes.

These effects are built into practical design only very roughly. Thus, the enlarged secondary bin is utilized only in a very restricted way. Also the vanishing of a bin is equivalent to the decision to drop a track, which is implemented directly by counting consecutive misses.

The remaining figure illustrates results of a calculation of deviation (difference between data and predicted position) variance during track start-up. The different curves are for different noise/maneuver weightings and lead to different steady-state values. Since the suggested maximum firmness is 15, the appropriate steady-state parameter is $\lambda \approx .005$ (see Figure 15). By following this curve and multiplying the standard deviation by a safety factor of 3.5 to 4, bin sizes listed in Figure 5 are obtained. (Cf. the simpler formula given in the description of this figure. This formula assumes a $\lambda = 0$ curve, which leads to nearly the same result.)

DELAY TRACKER FEATURES

- ALL FIXED POINT PROCESSING (FOR SPEED)
- EFFICIENT SMOOTHING ALOGRITHM
- MULTIPLE TRACKS PER RECEIVER/TRANSMITTER
- FINER DISTINCTIONS ON TRACK QUALITY/STATUS
  - TYPE STATUS
  - FIRMNESS
  - HIT/MISS COUNT
- EXTENDED CORRELATION LOGIC
  - CONTINUING AUTO-INITIATION
  - PRIMARY/SECONDARY SEARCH
  - TRIAL (BRANCHING) TRACK INITIATION AND RESOLUTION

FIGURE 1.

FIGURE 2.

INPUT PROCESSING

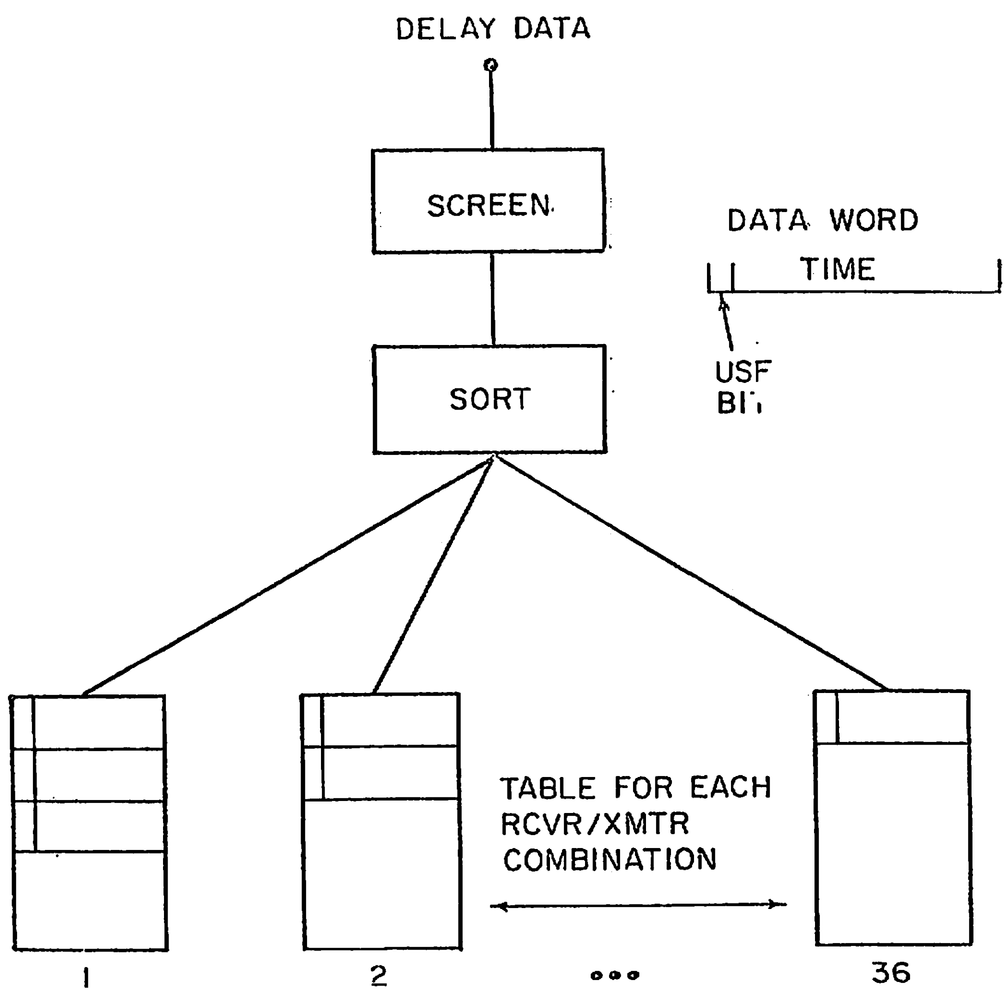
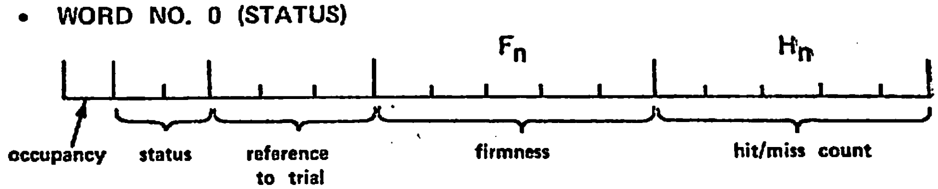


FIGURE 3.

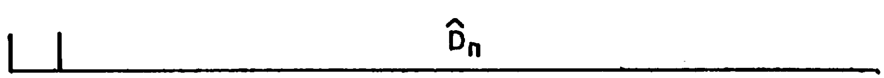
TRACK FILES

- ONE FILE FOR EACH RECEIVER/TRANSMITTER
- SEVERAL TRACKS PER FILE
- THREE WORDS PER TRACK

- WORD NO. 0 (STATUS)



- WORD NO. 1 (POSITION ESTIMATE)



- WORD NO. 2 (VELOCITY ESTIMATE)

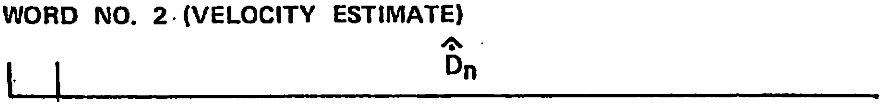


FIGURE 4.

SMOOTHING EQUATIONS

- RECURSIVE
- PRACTICAL IMPLEMENTATION OF AN OPTIMAL KALMAN FILTER
- EXTRAPOLATE

$$\hat{\hat{D}}_n = \hat{D}_{n-1} + \hat{\dot{D}}_{n-1}$$

$$\hat{\hat{\dot{D}}}_n = \hat{\dot{D}}_{n-1}$$

- SMOOTH

$$\hat{D}_n = \hat{\hat{D}}_n + \alpha_n(d_n - \hat{\hat{D}}_n)$$

$$\hat{\dot{D}}_n = \hat{\hat{\dot{D}}}_n + \frac{\beta_n}{T_n}(d_n - \hat{\hat{D}}_n)$$

- USE D IN VORTEX CALCULATION
- $\hat{D}_n, \hat{\dot{D}}$ SMOOTHED POSITION, VELOCITY

$\hat{D}_n, \hat{\dot{D}}$	SMOOTHED POSITION, VELOCITY
$\hat{\hat{D}}_n, \hat{\hat{\dot{D}}}_n$	PREDICTED POSITION, VELOCITY
d_n	DELAY DATA
α_n, β_n	SMOOTHING PARAMETERS
T_n	FRAMES SINCE LAST DATA

FIGURE 5.

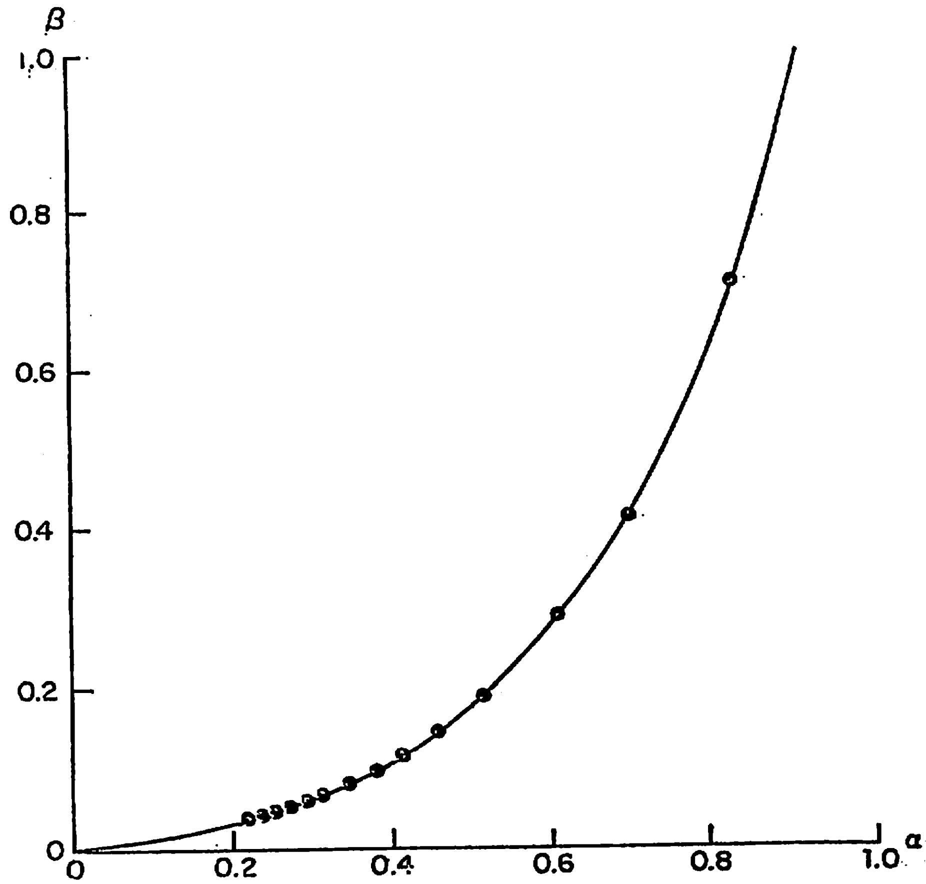
TRACKING PARAMETER LOOKUP

Firmness F_n	Position Smooth α_n	Velocity Smooth β_n	Search Bin Δ_n
0	1.000	.000	—
1	1.000	1.000	$7.3\sigma + \Delta t \cdot v_{max}$
2	.833	.700	9.8σ
3	.700	.409	7.3σ
4	.600	.270	6.3σ
.	:	:	:
:	.	.	.
15	.228	.030	4.6σ

IF HIT n: $F_{n+1} = F_n + 1$ $(F_{n\ max} = 15)$

IF MISS: $F_{n+1} = F_n - 2$ $(F_{n\ min} = 1)$

INITIAL HIT: $F_o = 0 \rightarrow F_1 = 1$



α, β VALUES USED IN DELAY TRACKER (COMPARED WITH STANDARD CURVE)

FIGURE 6.

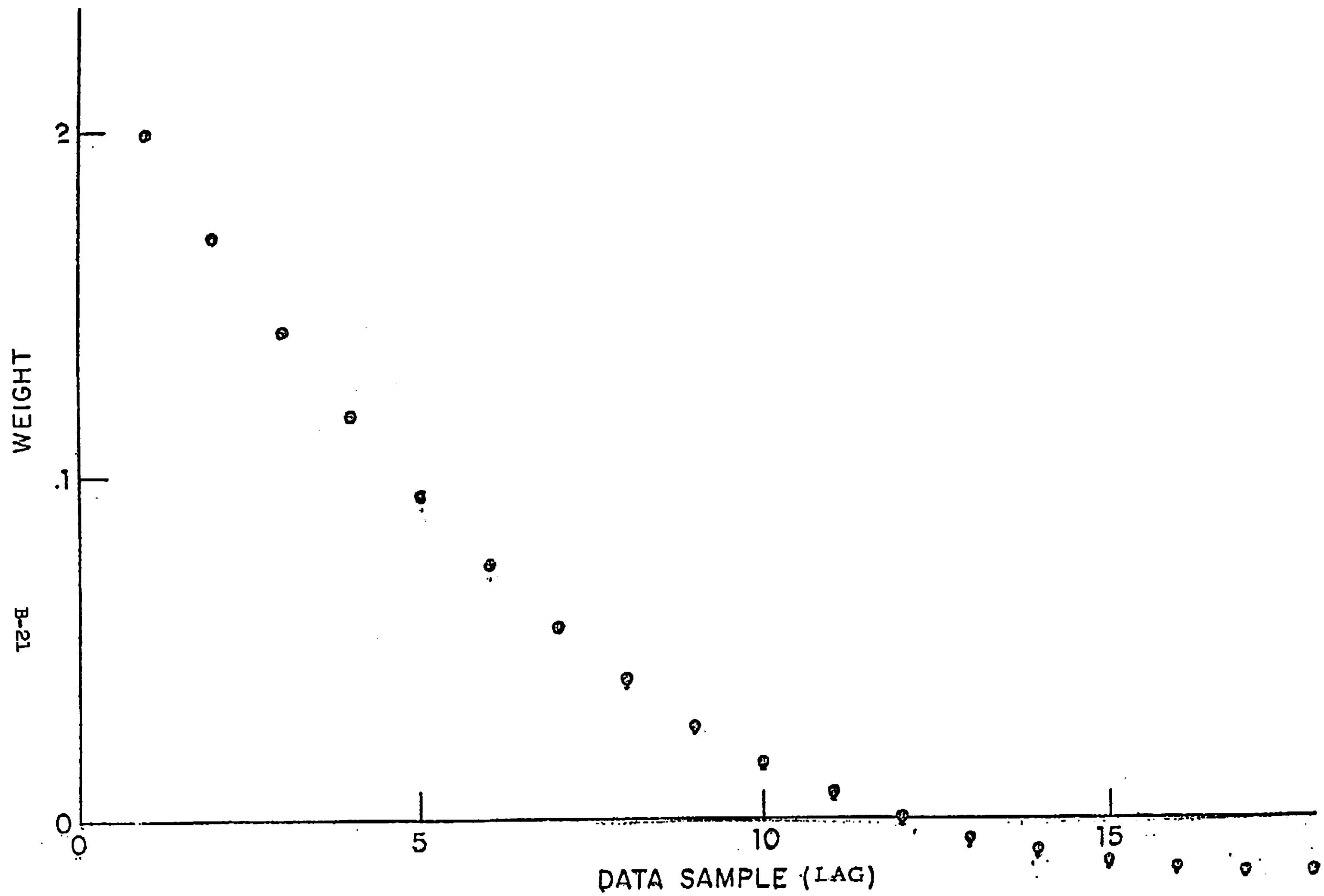


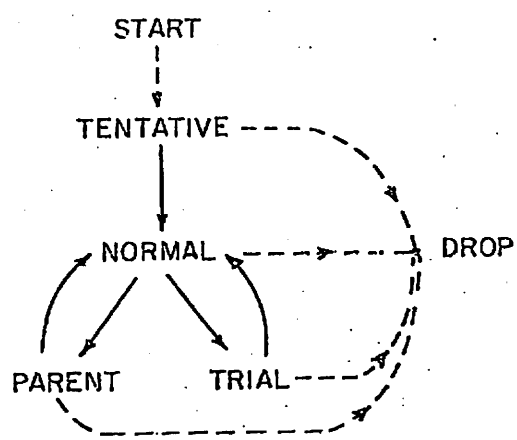
FIGURE 7.

EQUIVALENT DATA WEIGHTS FOR SMOOTHING AT MAXIMUM FIRMNESS

FIGURE 8.

TRACK STATUS TRANSITIONS

STATUS BITS	STATUS
00	TENTATIVE
01	TRIAL
10	NORMAL
11	PARENT

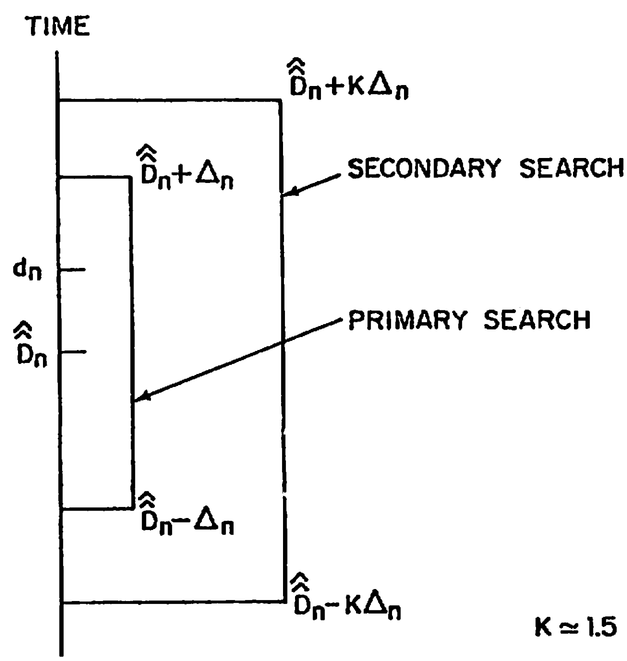


- INITIATE NORMAL : N CONSECUTIVE HITS
- NORMAL DROP : M CONSECUTIVE MISSES

FIGURE 9.

PRIMARY/SECONDARY CORRELATION

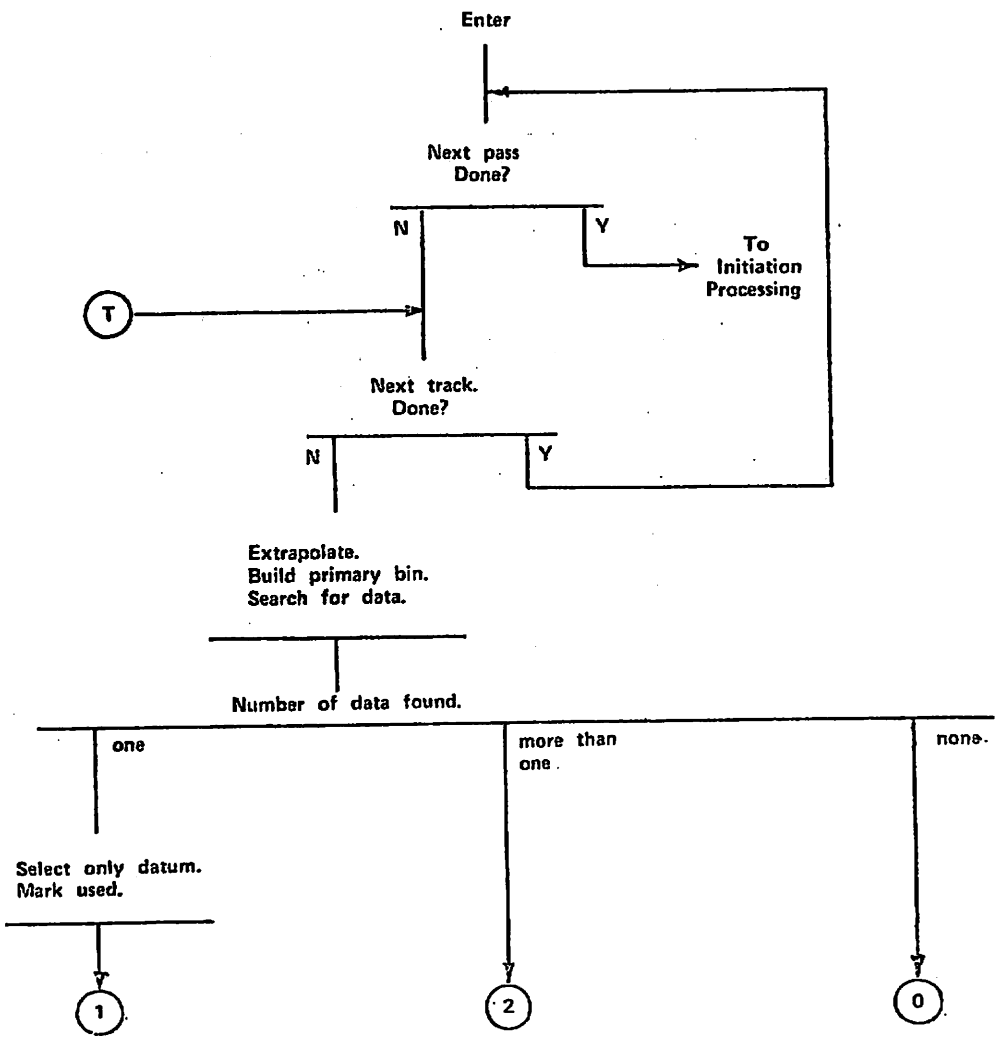
- PRIMARY SEARCH – ALL TRACKS
- SECONDARY SEARCH – NORMAL TRACKS



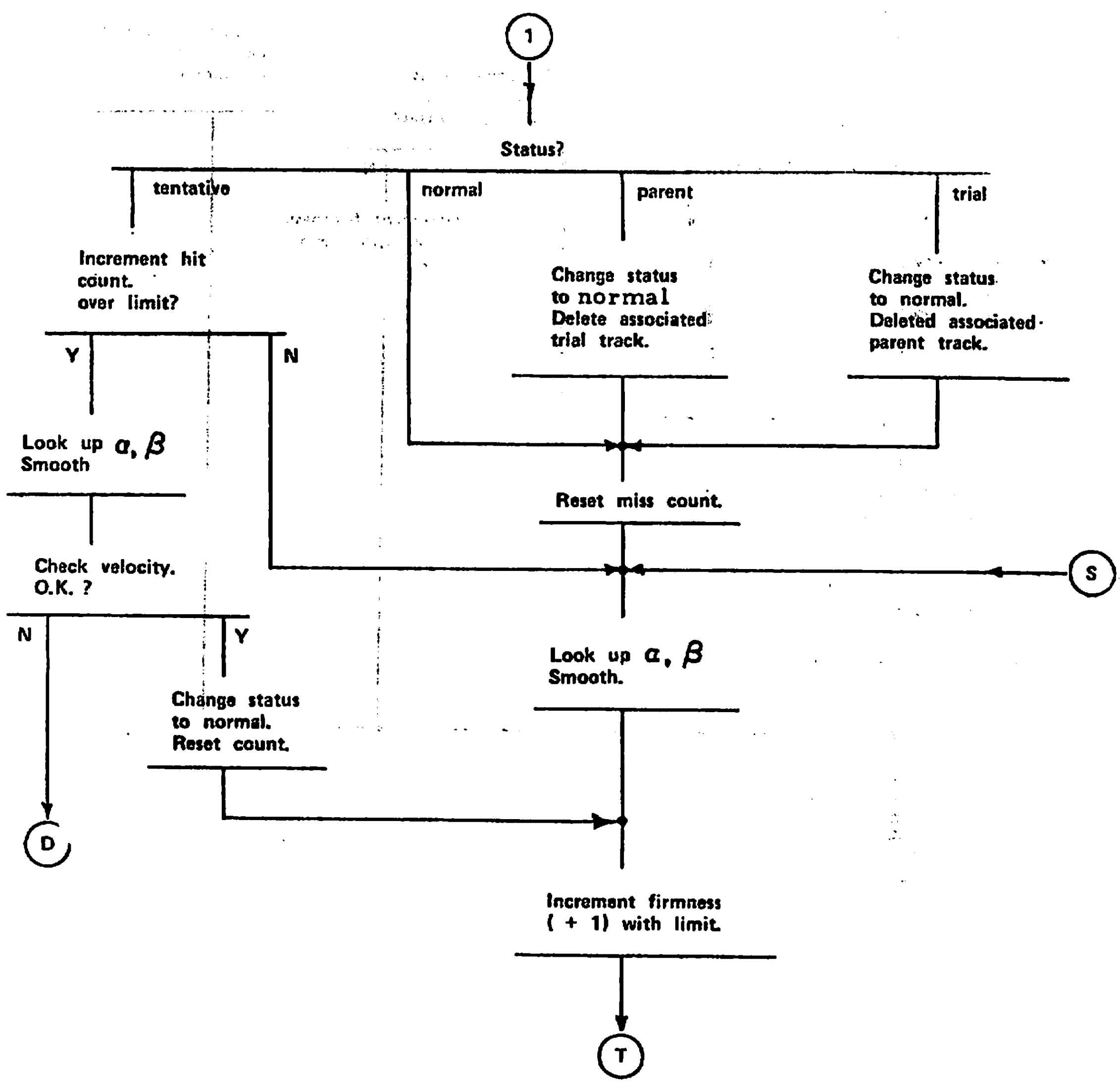
FIGURES 10-11-12-13.

CORRELATION/SMOOTHING FOR DELAY TRACKS

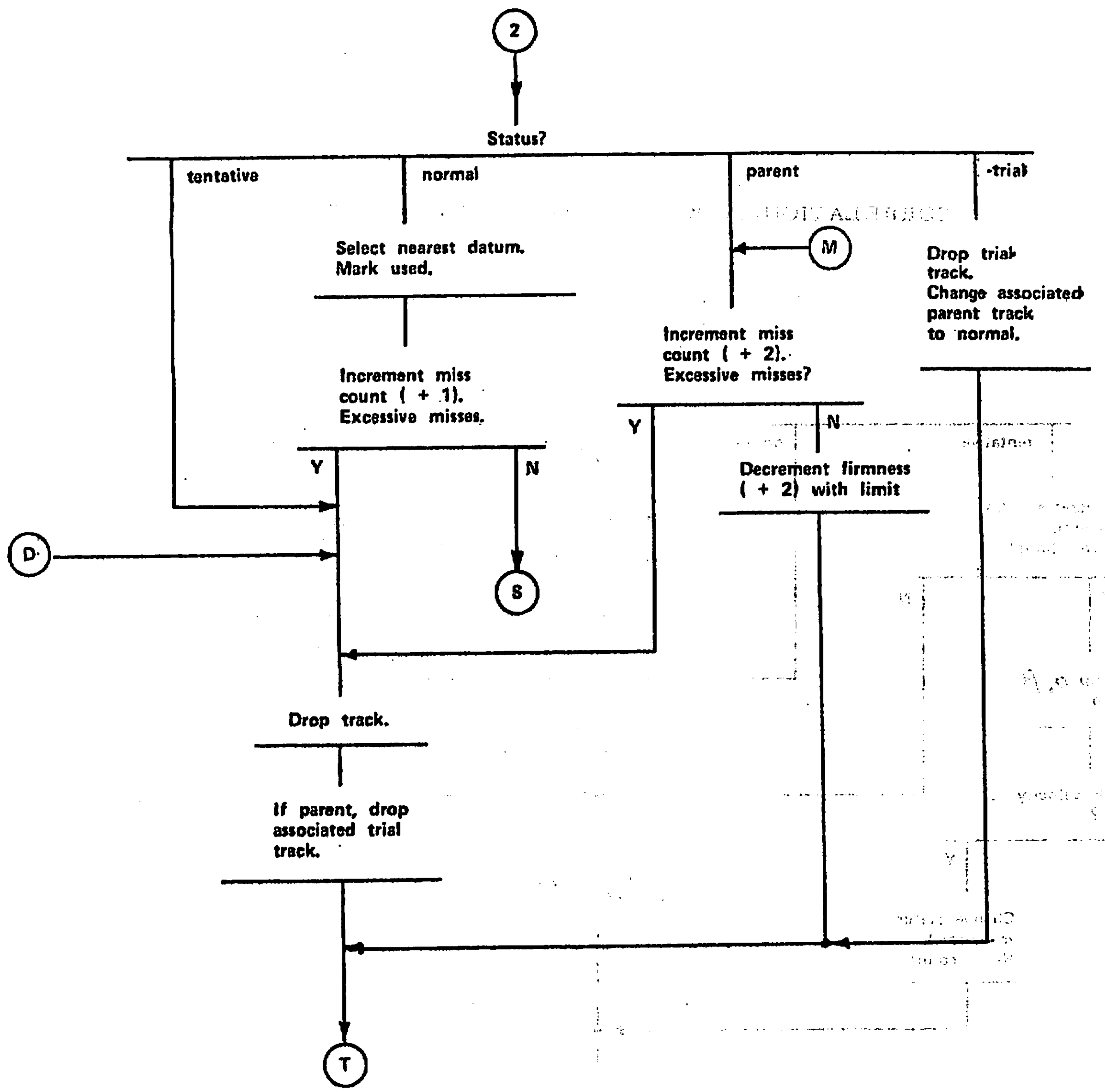
PASS	TRACK CLASSES
1	normal & parent
2	trial
3	tentative



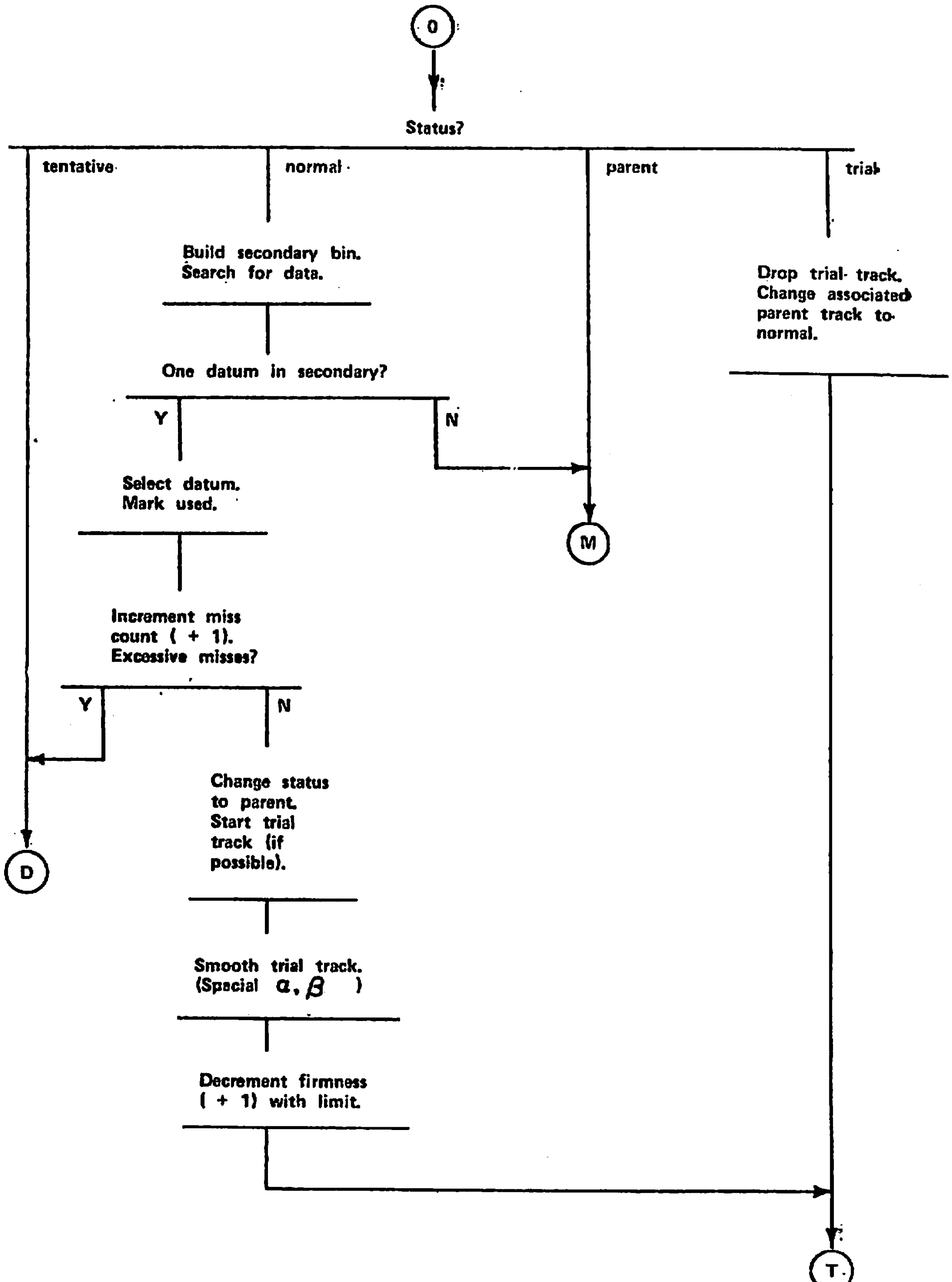
FIGURES 10-11-12-13.
CORRELATION/SMOOTHING FOR DELAY TRACKS



FIGURES 10-11-12-13.
CORRELATION/SMOOTHING FOR DELAY TRACKS



FIGURES 10-11-12-13.
CORRELATION/SMOOTHING FOR DELAY TRACKS



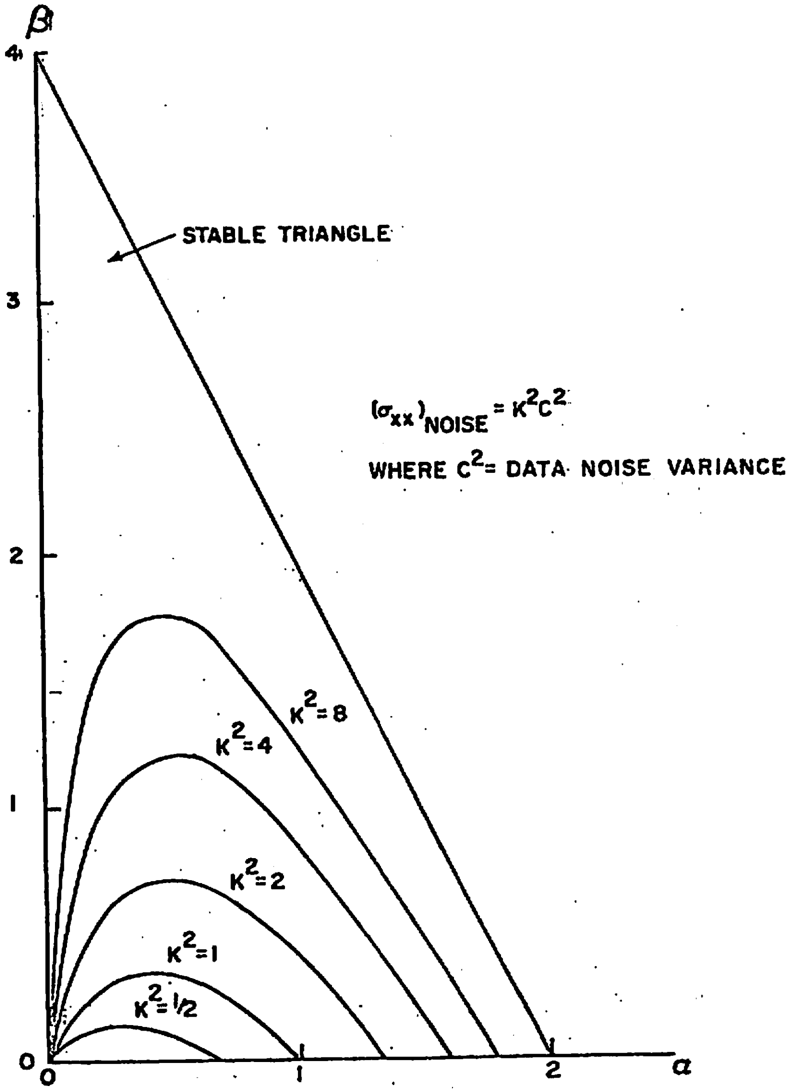
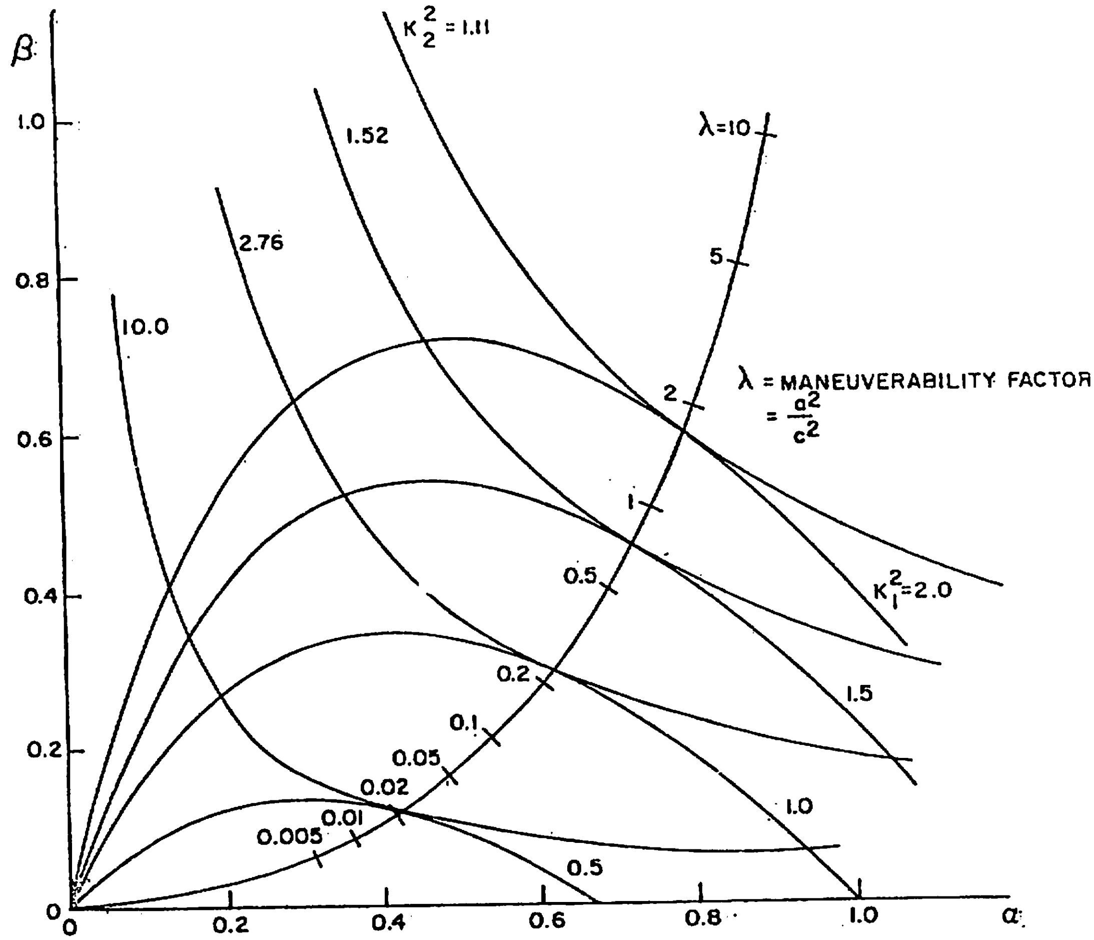


FIGURE 14.

VARIANCE FACTOR OF POSITION PREDICTION ERRORS DUE TO DATA NOISE



$K_1^2 \cdot c^2$ = PREDICTION VARIANCE (COMPONENT) DUE TO OBSERVATION ERRORS

$\lambda K_2^2 \cdot c^2$ = PREDICTION VARIANCE (COMPONENT) DUE TO RANDOM MANEUVERS

FIGURE 15.

VARIANCE FACTOR CONTOURS AND OPTIMAL α, β CURVE

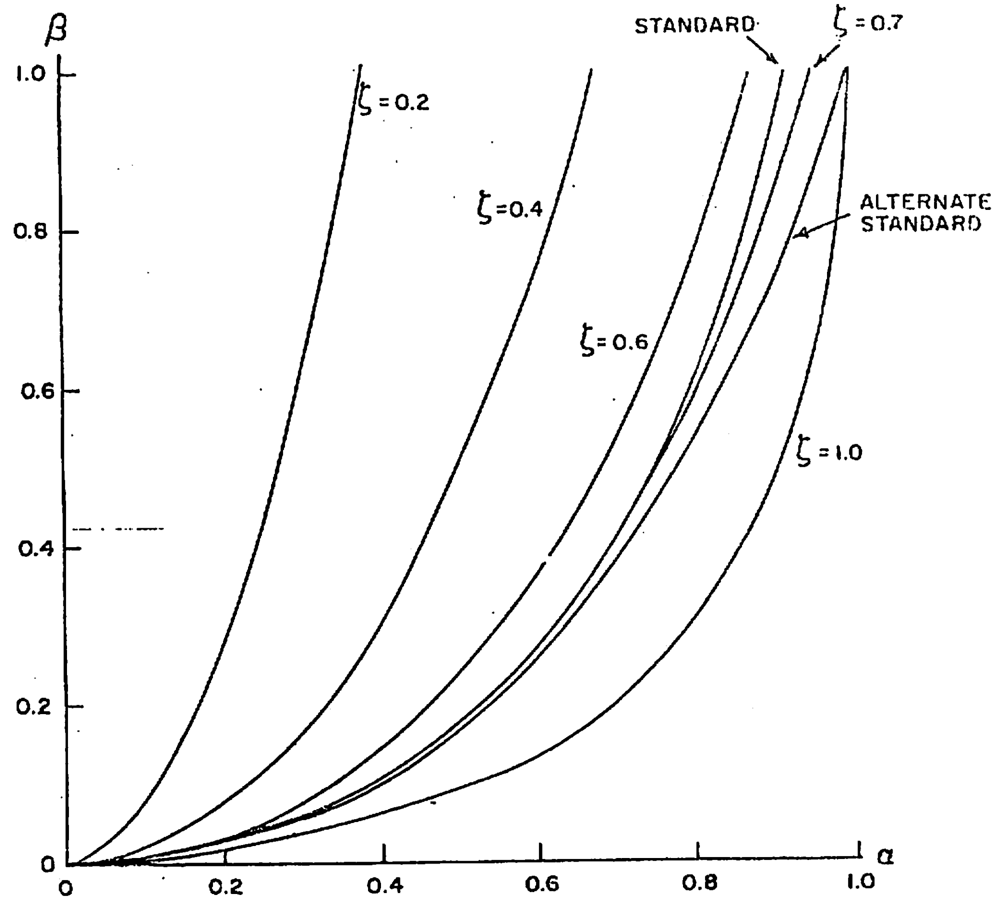


FIGURE 16.
OPTIMAL α, β CURVES AND DAMPING CONTOURS

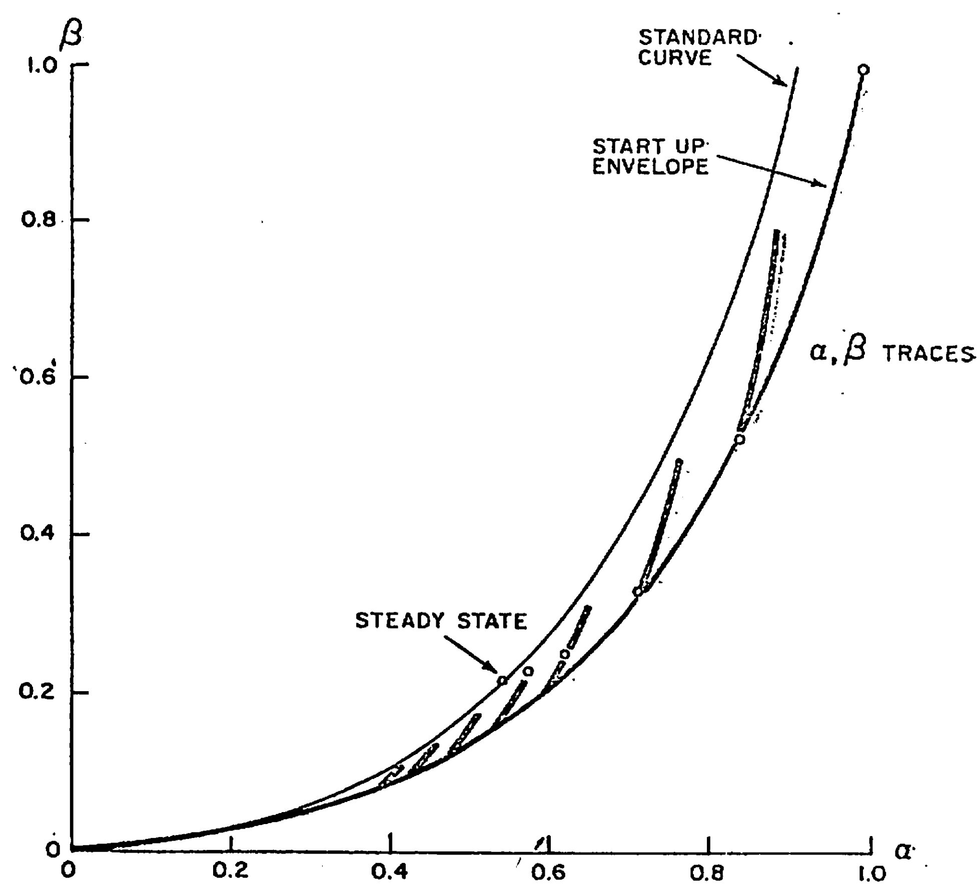


FIGURE 17.

OPTIMAL α, β START-UP SEQUENCE

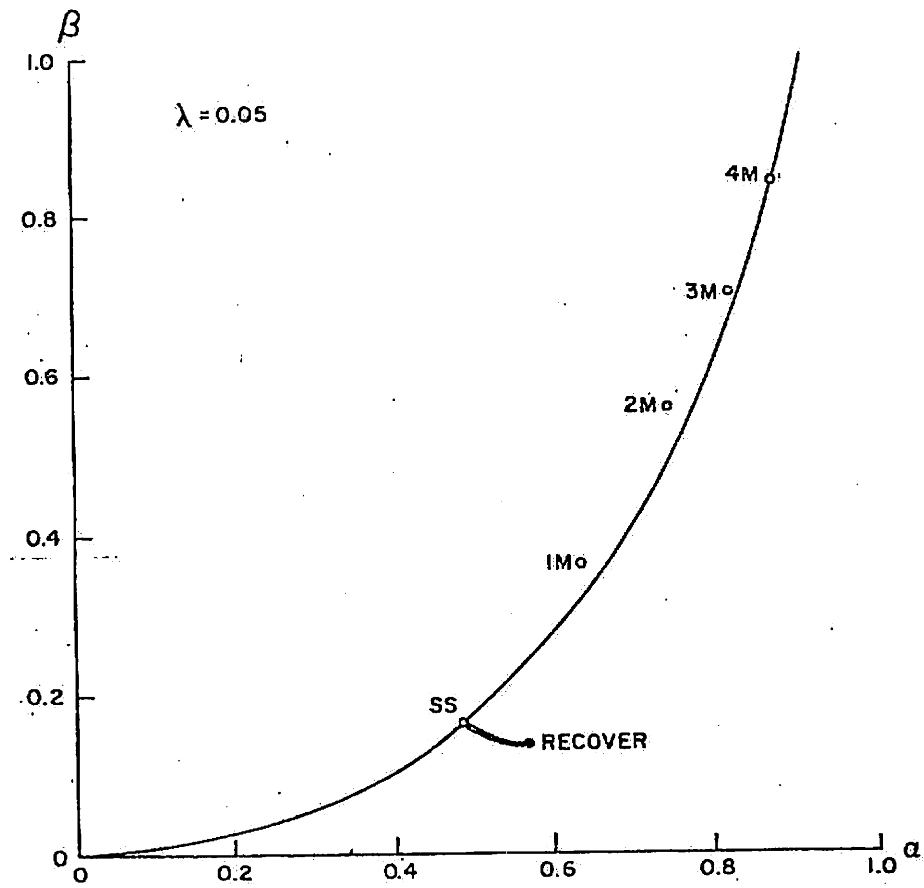


FIGURE 18

OPTIMAL α, β FOR MISSING DATA SEQUENCE
(λ = 0.05).

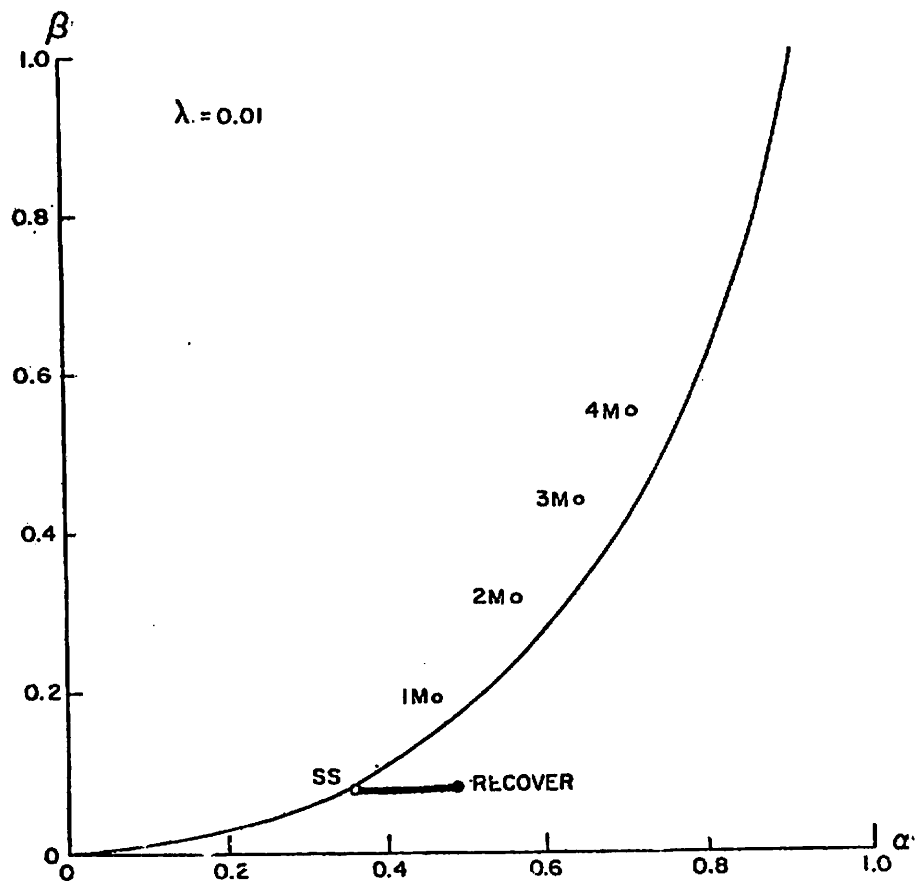


FIGURE 19.

OPTIMAL α, β FOR MISSING DATA SEQUENCE
(λ = 0.01).

DETERMINATION OF BIN SIZES

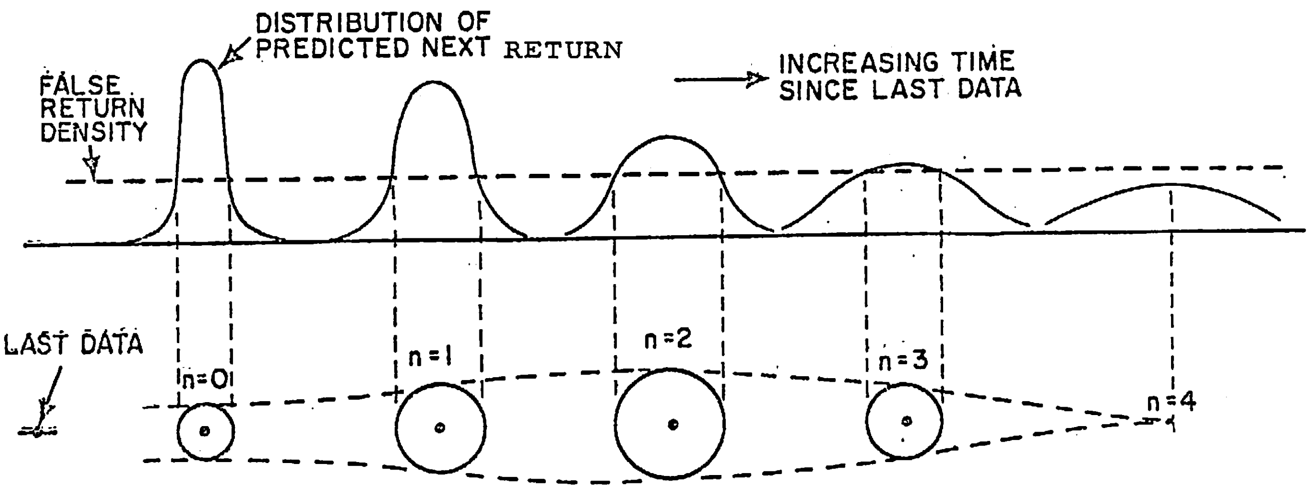


FIGURE 20.

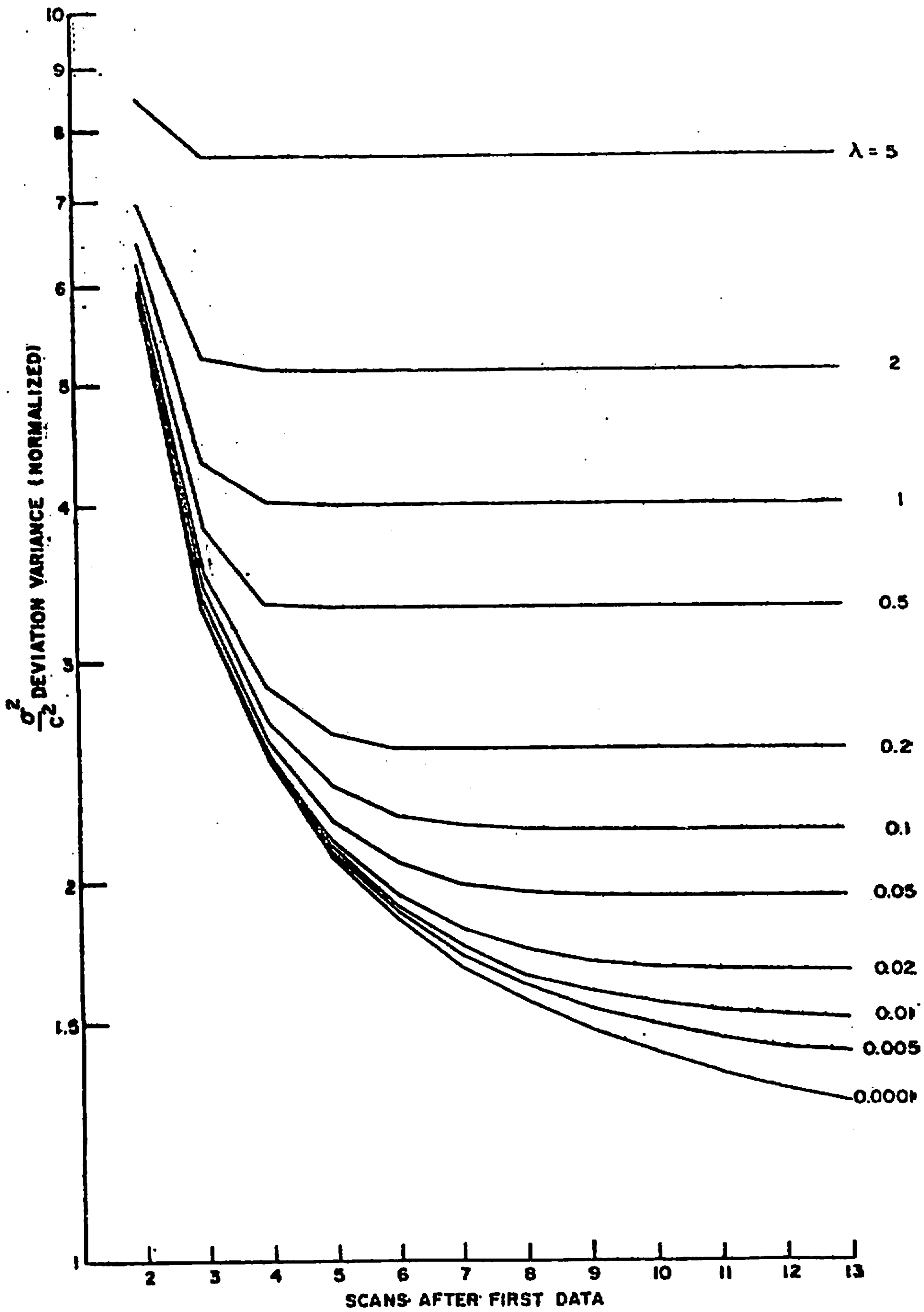


FIGURE 21.

DEVIATION VARIANCE BEHAVIOR IN START-UP SEQUENCE

APPENDIX A

Delay Tracking Program

This program was designed to improve system performance by enhancing the reliability and accuracy of delay tracking. The improvement results in an overall increased immunity to noise and a reduced susceptibility to data discontinuities. This program is a simplified version of the complete system described in the body of the report.

The program consists of six routines:

1. Main routine ARCON provides the bulk of logical and control operations of all other routines.
2. Routine INITIA performs the process of new track initiation.
3. Routine INSERT inserts new tracks which are initiated by INITIA-routine into track file.
4. Routines LOOKTR and QUEST look for a track of the proper status which should be continued and the data for the job.
5. Routine DROP eliminates tracks which cannot be continued properly (see flow chart of the algorithm).

```
COMPILER OPTICNS - NAME=  MAIN,OPT=01,LINECNT=50,SIZE=0000K,
                  SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
      SUBROUTINE ARCON(KDATA,NTRACK,NMB,NFRM,NMBR,KODTR,PARAM,DT)
      DIMENSION TRC(18,4,2),NST(18,4,5)
      DIMENSION DT(18,11),NTRACK(2),KODTR(2),NMBR(18,4,2),KDATA(18)
      DIMENSION PARAM(15,3)
      NDATA=10
      NT=4
      NHITLM=3
      VMAX=10.
      IF(NFRM)1,2,1
    2 NFRM=NFRM+1
      DO 7000 NN=1,18
      DO 6700 KK=1,4
      DO 6600 LM=1,5
 6600 NST(NN,KK,LM)=0
 6700 CONTINUE
 7000 CONTINUE
      CALL INITIA(DT,TRC,NST,NTRACK,NDATA,NT,NMBR,NMB,NFRM)
      DO 11 I=1,18
      KDATA(I)=1
   11 DT(I,1)=7777.
      RETURN
    1 NFRM=NFRM+1
      DO 3 NPASS=1,2
      IBEGIN=1
      JBEGIN=0
      I=NTRACK(NPASS)
      IF(I)3,3,4
    4 NPT=KODTR(NPASS)
      DO 5 KP=1,I
      WRITE(6,500)NTRACK
  500 FORMAT(2I10)
      WRITE(6,55)IBEGIN,JBEGIN,I,NPT
   55 FORMAT(1X,4I10)
      CALL LOOKTR(IBEGIN,JBEGIN,NPT,NST,NT)
      PRED=TRC(IBEGIN,JBEGIN,1)+TRC(IBEGIN,JBEGIN,2)
      UPBIN=PRED+PARAM(NST(IBEGIN,JBEGIN,3),3)
      DOWBN=PRED-PARAM(NST(IBEGIN,JBEGIN,3),3)
      CALL QUEST(IBEGIN,DT,NDATA,DOWBN,UPBIN,PRED,ID,ND)
      IF(ND-1)6,7,8
    6 IF(NPASS-1)3,12,13
    8 IF(NPASS-1)3,9,13
    9 NST(IBEGIN,JBEGIN,4)=NST(IBEGIN,JBEGIN,4)+1
      IF(NST(IBEGIN,JBEGIN,4)-MSLIM)18,18,13
   12 NST(IBEGIN,JBEGIN,4)=NST(IBEGIN,JBEGIN,4)+2
      IF(NST(IBEGIN,JBEGIN,4)-MSLIM)10,10,13
   10 IF(NST(IBEGIN,JBEGIN,3)-2)14,14,15
```

```
 14   NST(IBEGIN,JBEGIN,3)=1
      GC TO 5
 15   NST(IBEGIN,JBEGIN,3)=NST(IBEGIN,JBEGIN,3)-2
      GC TC 5
 13   NST(IBEGIN,JBEGIN,1)=0
      NTRACK(NPASS)=NTRACK(NPASS)-1
      CALL DRCP(NST,NMBR,IBEGIN,JBEGIN)
      GC TO 5
  7   IF(NPASS-1)3,16,17
 16   NST(IBEGIN,JBEGIN,4)=0
 18   TRC(IBEGIN,JBEGIN,1)=PRED+PARAM(NST(IBEGIN,JBEGIN,3),1)*
     1(DT(IBEGIN,ID)-PRED)
      TN=NST(IBEGIN,JBEGIN,4)+1
      TRC(IBEGIN,JBEGIN,2)=TRC(IBEGIN,JBEGIN,2)+
     1PARAM(NST(IBEGIN,JBEGIN,3),2)*(DT(IBEGIN,ID)-PRED)/TN
      IF(NPASS-1)3,24,19
 17   NST(IBEGIN,JBEGIN,5)=NST(IBEGIN,JBEGIN,5)+1
      GC TO 18
 19   IF(NST(IBEGIN,JBEGIN,5)-NHITLM)24,27,27
 27   IF(TRC(IBEGIN,JBEGIN,2))20,21,21
 20   V=(-TRC(IBEGIN,JBEGIN,2))
      GC TO 22
 21   V=TRC(IBEGIN,JBEGIN,2)
 22   IF(V-VMAX)23,23,13
 23   NST(IBEGIN,JBEGIN,2)=2
      NST(IBEGIN,JBEGIN,4)=0
      NTRACK(1)=NTRACK(1)+1
      NTRACK(2)=NTRACK(2)-1
 24   IF(NST(IBEGIN,JBEGIN,3)-15)25,26,25
 25   NST(IBEGIN,JBEGIN,3)=NST(IBEGIN,JBEGIN,3)+1
      CALL INSERT(TRC,NST,NMBR,NFRM,IBEGIN,JBEGIN)
 26   DC 28 NA=ID,NDATA
      DT(IBEGIN,NA)=DT(IBEGIN,NA+1)
      IF(DT(IBEGIN,NA)-7777.)28,5,28
 28   CCNTINUE
  5   CCNTINUE
  3   CCNTINUE
      CALL INITIA(DT,TRC,NST,NTRACK,NDATA,NT,NMBR,NMB,NFRM)
      DO 30 I=1,18
      KDATA(I)=1
 30   DT(I,1)=7777.
      RETURN
      END
```

```
 COMPILER OPTIONS - NAME=  MAIN,OPT=01,LINECNT=50,SIZE=0000K,
                    SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
                    SUBROUTINE INITIA(DT,TRC,NST,NTRACK,NDATA,NT,NMBR,NMB,NFRM)
                    DIMENSION DT(18,11),TRC(18,4,2),NST(18,4,5),NTRACK(2),NMBR(18,4,2)
                    DIMENSION DIN(4)
                    VAVER=-1.
                    DO 15 I=1,18
                    K=0
                    DO 5 J=1,NT
                    IF(NST(I,J,1))1,1,5
           1        K=K+1
           5        CONTINUE
                    IF(K)1,15,7
           7        NH=0
                    DO 8 J=1,K
                    D=0
                    DO 9 L=1,NDATA
                    IF(DT(I,L)-7777.)10,14,10
          10        IF(DT(I,L)-D)9,9,11
          11        D=DT(I,L)
                    LL=L
           9        CONTINUE
          14        IF(D)1,18,17
          17        NH=NH+1
                    DIN(J)=D
                    DO 13 L=LL,NDATA
                    DT(I,L)=DT(I,L+1)
                    IF(DT(I,L)-7777.)13,8,13
          13        CONTINUE
           8        CONTINUE
          18        IF(NH)1,15,19
          19        J=1
                    DO 12 L=1,NT
                    IF(NST(I,L,1))1,16,12
          16        NST(I,L,1)=1
                    NST(I,L,2)=0
                    NST(I,L,3)=1
                    NST(I,L,4)=0
                    NST(I,L,5)=0
                    NTRACK(2)=NTRACK(2)+1
                    TRC(I,L,1)=DIN(J)
                    TRC(I,L,2)=VAVER
                    J=J+1
                    NMB=NMB+1
                    NMBR(I,L,1)=NMB
                    CALL INSERT(TRC,NST,NMBR,NFRM,I,L)
                    IF(J-NH)12,12,15
          12        CONTINUE
```

```
   15 CONTINUE
   21 CONTINUE
   20 CONTINUE
      RETURN
      END
```

```
COMPILER OPTIONS - NAME=  MAIN,OPT=01,LINECNT=50,SIZE=0000K,
                   SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
      SUBROUTINE LOOKTR(IBEGIN,JBEGIN,NPT,NST,NT)
      DIMENSION NST(18,4,5)
      IF(JBEGIN-NT)5,6,1
    5 JBEGIN=JBEGIN+1
      GO TO 7
    6 JBEGIN=1
      IBEGIN=IBEGIN+1
    7 DO 1 I=IBEGIN,18
      DO 2 J=JBEGIN,NT
      IF(NST(I,J,1))1,2,3
    3 IF(NST(I,J,2)-NPT)2,4,2
    2 CONTINUE
      JBEGIN=1
    1 CONTINUE
    4 IBEGIN=I
      JBEGIN=J
      RETURN
      END
```

```
  COMPILER OPTIONS - NAME=  MAIN,OPT=01,LINECNT=50,SIZE=0000K,
                       SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
                SUBROUTINE QUEST(IBEGIN,DT,NDATA,DB,UB,PRED,ID,K)
                DIMENSION DT(18,11)
                IC=0
                K=0
                CLST=10000.
                DO 1 I=1,NDATA
                IF(DT(IBEGIN,I)-7777.)9,3,9
        9       IF(DT(IBEGIN,I).GE.DB.AND.DT(IBEGIN,I).LE.UB) GO TO 4
                GO TO 1
        4       K=K+1
                DI=DT(IBEGIN,I)-PRED
                IF(DI)5,6,6
        5       DI=(-DI)
        6       IF(DI.GT.CLST) GO TO 1
                ID=I
                CLST=DI
        1       CONTINUE
        3       RETURN
                END
```

```
COMPILER OPTIONS - NAME=  MAIN,OPT=01,LINECNT=50,SIZE=0000K,
                   SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
          SUBROUTINE INSERT(TRC,NST,NMBR,NFRM,I,J)
          DIMENSION TRC(18,4,2),NST(18,4,5),NMBR(18,4,2)
          COMMON/AAA/IFRM(18,4,20),RESULT(18,4,20)
          MFIRM=5
          NMBR(I,J,2)=NMBR(I,J,2)+1
          IFRM(I,J,NMBR(I,J,2))=NFRM
          RESULT(I,J,NMBR(I,J,2))=TRC(I,J,1)
          IF(NMBR(I,J,2)-20)1,2,1
2         IF(NST(I,J,3)-MFIRM)3,4,4
4         WRITE(6,10)NMBR(I,J,1),(IFRM(I,J,K),K=1,10,1)
10        FORMAT(I10,10I10)
          WRITE(6,11)RESULT(I,J,1),(RESULT(I,J,K),K=2,10,1)
11        FORMAT(F20.2,9F10.2)
          WRITE(6,12)IFRM(I,J,11),(IFRM(I,J,K),K=12,20,1)
12        FORMAT(I20,9I10)
          WRITE(6,13)RESULT(I,J,11),(RESULT(I,J,K),K=12,20,1)
13        FORMAT(F20.2,9F10.2)
3         NMBR(I,J,2)=0
1         RETURN
          END
```

```
 COMPILER OPTIONS - NAME=  MAIN,OPT=01,LINECNT=50,SIZE=0000K,
                    SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
              SUBROUTINE DROP(NST,NMBR,I,J)
              DIMENSION NST(18,4,5),NMBR(18,4,2)
              COMMON/AAA/IFRM(18,4,20),RESULT(18,4,20)
              MFIRM=5
              IF(NST(I,J,3)-MFIRM)2,1,1
        1     IF(NMBR(I,J,2)-10)3,3,4
        3     M=NMBR(I,J,2)
              WRITE(6,10)NMBR(I,J,1),(IFRM(I,J,K),K=1,M,1)
       10     FORMAT(I10,10I10)
              WRITE(6,11)RESULT(I,J,1),(RESULT(I,J,K),K=2,M,1)
       11     FORMAT(F20.2,9F10.2)
              GO TO 2
        4     WRITE(6,10)NMBR(I,J,1),(IFRM(I,J,K),K=1,10,1)
              M=NMBR(I,J,2)
              WRITE(6,11)RESULT(I,J,1),(RESULT(I,J,K),K=2,10,1)
              WRITE(6,12)IFRM(I,J,11),(IFRM(I,J,K),K=12,M,1)
       12     FORMAT(I20,9I10)
              WRITE(6,11)RESULT(I,J,11),(RESULT(I,J,K),K=12,M,1)
        2     NMBR(I,J,2)=0
              RETURN
              END
```


180 copies

☆ U. S. GOVERNMENT PRINTING OFFICE: 1977--701-346--90