M-62-66-8

ANUARY 1967

FINAL REPORT

PCM TELEMETRY DATA COMPRESSION STUDY, PHASE II

(25 AUGUST 1965 TO 25 DECEMBER 1966)

Contract No NAS 5-9729

Prepared by

LOCKHEED MISSILES & SPACE COMPANY
Sunnyvale, California

for

GODDARD SPACE FLIGHT CENTER
Greenbelt, Maryland

FACILITY FORM 602

N67-31840 (ACCESSION NUMBER)

(THRU)

410 (PAGES)

1 (CODE)

CR-85894 (NASA CR OR TMX OR AD NUMBER)

07 (CATEGORY)

RQ 7-44454

M-62-66-8

JANUARY 1967

FINAL REPORT

PCM TELEMETRY DATA COMPRESSION STUDY, PHASE II

(25 AUGUST 1965 TO 25 DECEMBER 1966)

by

T. BJORN, JR.

W. J. FRICCERO

J. E. MEDLIN

Contract No. NAS 5-9729

Prepared by

LOCKHEED MISSILES & SPACE COMPANY

Sunnyvale, California

for

GODDARD SPACE FLIGHT CENTER

Greenbelt, Maryland

FOREWORD

This report describes Phase II of a Lockheed Missiles & Space Company study of PCM Telemetry Data Compression. The work was performed for the National Aeronautics and Space Administration under Contract No. NAS 5-9729 during the period between 25 August 1965 and 25 December 1966. In addition to this report, Refs. 1, 2, and 3 describe the first three quarters of Phase II. Phase I of the study, completed on 15 August 1965 is reported in Ref. 4

The authors wish to thank GSFC scientist Gerald J. Hogg for his significant contributions throughout the investigation, and the GSFC OGO satellite data processing personnel and experimenters for their explanations of the system, the data, and the tolerance accuracy requirements. We also wish to thank LMSC programmers James L. Hunts, Christopher S. Raymond, and Gerald W. O'Shaughnessy for their work in developing and modifying the computer programs.

PRECEDING PAGE BLANK NOT FILMED.

SUMMARY

The objective of this study was to investigate ways to improve the utilization efficiency of NASA/GSFC PCM telemetry links through data compression. The study was divided into two phases. Phase I included a literature search and subsequent evaluation of known data compression techniques and a study of time and channel identification.

Phase II extended the data compression technique evaluation of Phase I, investigated transmission link noise effects, analyzed the impact of compressed data on computer operations, and performed a conceptual design of a ground-based data compression and reconstruction system. This report describes the work performed during Phase II of the study.

DATA COMPRESSION TECHNIQUE EVALUATION

The data compression technique evaluation was divided into two major parts: an evaluation of redundancy removal techniques, and a study of buffer queuing control methods.

Redundancy Removal Evaluation

The redundancy removal technique evaluation examined, by digital computer-simulation, the compression effectiveness of nine sample selectors, using selected sensors from Explorer XVII (S-6) and Orbiting Geophysical Observatory-A (S-49) satellite telemetry data. The results of these tests showed, as did the Phase I tests, that a particular type of first-order polynomial interpolator achieved the highest overall performance in reducing the bandwidth of the S-6 and the S-49 telemetry data. This interpolator is

capable of fitting straight lines to the sampled data waveform by adjusting their end-points with 4 degrees of freedom.*

However, when factors other than redundancy removal effectiveness were weighed, it was concluded that the zero-order predictor would be the most desirable selector for both spaceborne and ground-based applications, mainly because of its relative simplicity a the ease of reconstructing ZOP-c presse data.

Queuing Control Evaluation

Early in Phase II, a theoretical study was mad of buffer queuing contr methods. One of the major conclusions of this study that an adaptive tolerance system which monitors a combination of the queue length and its integral held promise of being an effective queuing control system. This method was therefore evaluated later in the study by computer simulation. In addition, adaptive precompression filtering and sample deletion techniques were empirically evaluated.

The empirical study found, using S-6 multiplexed data, that the addition of integral feedback to an adaptive tolerance control system definitely enhances overall system performance. Ranges of tolerance control parameter values yielding the highest performance, according to the adopted criteria, were determined.

Adaptive sample deletion was found be relatively effective with S-6 data, but reconstructed data fidelity was considered on some sensors unacceptable The adaptive filtering techniques tested were completely ineffective with S-6 data.

*It was found that in general the S-49 data were much more redundant than were the S-6. Combined bandwidth compression ratios were found to be on the order of 10 for the S-49 data, compared with 3 for the S-6, using comparable tolerances.

To maintain fidelity, all the digital channels in the S-49 multiplexed data were excluded from the queuing control loop. Because of the large proportion of digital channels, it was found that none of the control systems tested were effective with these data. It was therefore recommended that a priority-sensor elimination system be evaluated for S-49 data.

COMPRESSED DATA TRANSMISSION ERROR ANALYSIS

The transmission link noise study was divided into a theoretical analysis and an experimental evaluation. The theoretical study derived expressions for, and found values of, rms error in data reconstructed from compressed PCM bit streams contaminated with ground transmission link noise. Waveform aberrations caused by errors in nonredundant sample amplitude, sensor identification, and minor frame flag were analyzed.

The experimental study simulated a noisy compressed PCM ground transmission link using a digital computer. After deliberately contaminating the compressed bit stream with Pareto noise, the data were decoded, reconstructed, and analyzed on a numerical and visual basis to determine the effect of link noise on reconstructed data.

Both studies indicated that at normal operating signal-to-noise ratios, ground transmission link noise will not degrade the compressed data to any appreciable extent, if compressed PCM formats and bit stream decoding procedures are designed to minimize the effect of link noise.

REAL-TIME COMPUTER ANALYSIS

The real-time computer analysis investigated the capability of the SDS-920 and 930 digital computers to handle real-time compressed OGO data for spacecraft status checks and experiment monitoring. Based on the compression ratios obtained on the OGO experimenters' data during Phase II of the study, it was found that the

use of compressed data would (1) double the average available service and special-purpose (SSP) calculation time by the SDS-920 computer in the accelerated subcom mode, and (2) have a negligible average effect on other computer operating times examined.

Because the results of the study were based on the compressibility of experimenters' data, rather than spacecraft status data in the accelerated subcom mode, it was recommended that computer simulation tests be performed on OGO spacecraft status data in their accelerated mode.

DATA PROCESSING COMPUTER ANALYSIS

The data processing analysis investigated the effect of compressed data on the OGO data processing system. The results showed that if the data were delivered to the experimenters in compressed form, data compression would (1) reduce the decommutation time from the present requirements by at least a factor of 10, and (2) reduce the experimenters' tape storage requirements by about a factor of 13, and those of the decommutation input tape by more than a factor of 2. The advantages of maintaining the data in compressed form throughout the processing operation were found to be substantial.

SYSTEM CONFIGURATION STUDY

The configuration of a data compression and reconstruction system for the GSFC OGO ground data transmission network, and ground facility modifications required for its incorporation, were determined. To incorporate ground-based data compression, it was found that a compressor should be installed only at the Rosman, North Carolina, and the Fairbanks, Alaska, data acquisition stations, unless transmission links with approximate capacities of 6.4 kilobits/second could be installed between the other stations and GSFC. A data reconstructor consisting of a decoder an organizer/buffer, and an interpolator, would be needed at the central location. The zero-order predictor was recommended for incorporation into this system, along

with a method of channel and time coding which provides a minor frame channel identification with each transmitted sample, and forces a readout during each empty minor frame. The consideration of queuing control by low-priority sensor elimination was also recommended for this system. Finally, the following steps toward the incorporation of data compression into the OGO ground telemetry network were recommended:

- Demonstrate the potential of OGO data compression, using existing equipment.
- Perform a detailed hardware design, based on the data compression system configuration presented in this report.
- Construct a breadboard, and test this equipment in the OGO ground telemetry system.
- Install the final data compression and reconstruction system into the OGO telemetry network.

CONTENTS

Section		Page
	FOREWORD	iii
	SUMMARY	v
	ILLUSTRATIONS	
	TABLES	
1	INTRODUCTION	1-1
	1.1 Purpose	1-1
	1.2 Scope	1-2
	1.3 Project	1-2
	1.4 Background	1-5
2	COMPRESSION MODEL EVALUATION	2-1
	2.1 New Data Compression Techniques	2-6
	2.2 Selector Performance	2-12
	2.3 Broadside Data Compression Evaluation	2-34
	2.4 Secondary Data Compression	2-38
3	QUEUING CONTROL EVALUATION	3-1
	3.1 Theoretical Analysis	3-2
	3.2 Empirical Analysis	3-4
4	COMPRESSED DATA TRANSMISSION ERRORS	4-1
	4.1 Theoretical Bit Error Study	4-1
	4.2 Experimental Bit Error Study	4-24
	4.3 Theoretical-Experimental Comparison	4-36
	4.4 Conclusions and Recommendations	4-38
5	COMPRESSION MODEL TRADEOFF ANALYSIS	5-1
	5.1 Performance Criteria	5-2
	5.2 Weightings of Performance Criteria	5-5
	5.3 Relative Compression Model Performance	5-7
	5.4 Conclusions and Recommendations	5-9

Section		Page
6	REAL-TIME COMPUTER ANALYSIS	6-1
	6.1 Introduction	6-1
	6.2 Real-Time Computer Operations at GSFC	6-3
	6.3 Real-Time Computer System Analysis	6-11
	6.4 Conclusions and Recommendations	6-37
7	DATA PROCESSING COMPUTER ANALYSIS	7-1
	7.1 Introduction	7-1
	7.2 Data Processing Operations at GSFC	7-2
	7.3 The Impact of Compression on Data Processing	7-15
	7.4 Conclusions	7-38
8	DATA COMPRESSION SYSTEM CONFIGURATION	8-1
	8.1 Existing System	8-1
	8.2 Data Compression System Requirements	8-4
	8.3 Data Compression and the Goddard System	8-7
	8.4 Data Compression Model	8-10
	8.5 Conclusions and Recommendations	8-13
9	CONCLUSIONS AND RECOMMENDATIONS	9-1
	9.1 Conclusions	9-1
	9.2 Recommendations for Future Work	9-5
	BIBLIOGRAPHY	10-1
Appendix		
I	COMPARISON OF COMPRESSION ERROR WITH TOTAL RMS ERROR	I-1

ILLUSTRATIONS

Figure		Page
2-1	S-6 Data Specimens	2-13
2-2	S-49 Data Spe imens	2-15
2-3	Selector Performance S-4ᴝ Sensor 14	2-18
2-4	Selector Performance, S-4ᴝ Sensor 37	2-19
2-5	Selector Perfcman e S-4ᴝ Eensc 50	2-20
2-6	Selector Performance S-4ᴝ Sensc 60	2-21
2-7	Selector Perfcmance, S-4ᴝ Sensor 63	2-22
2-8	Selector Performance, S-6 Eenec 2	2-23
2-9	Selector Performanoe E-6 Sen c 3	2-24
2-10	Selector Performance, S-6 Sensor 7	2-25
2-11	Selector Performance, S-6 Sensor ᴝ	2-26
2-12	S ctc Performance, S-6 Senec 11	2-27
2-13	S-6 Sen c 7 Original Data	2-36
2-14	S-6 Sensor 7 Pseudo Sensc Data	2-37
2-15	Typical Sampled Data Waveform	2-40
2-16	Original and Reconstructed Data, FOIDIS Selector	2-41
2-17	Original and Reconstructed Data, ZOI Selector, Tolerance = 2 Counts	2-42
2-18	Original and Reconstructed Data, Secondary Compression, Tolerance = 2 Counts	2-43
2-19	Original and Reconstructed Data, ZOI Selector, Tolerance = 1 Count	2-44
2-20	Original and Reconstructed Data, Secondary Compression, Tolerance = 1 Count	2-45
3-1	Queue Length Behavior With Adaptive Tolerance, S-49 Data	3-40
3-2	Sensor 37 Reconstructed Data, Run S49-B: No Queuing Contr	3-43

Figure		Page
3-3a	Sensor 37 Reconstructed Data, Run S49-E: Adaptive Tolerance, Compression Ratio Monitoring	3-44
3-3b	Sensor 37 Tolerance Variations, Run S49-E: Adaptive Tolerance, Compression Ratio Monitoring	3-45
3-4a	Sensor 37 Reconstructed Data, Run S49-F: Adaptive Tolerance, Queue Length Monitoring	3-46
3-4b	Sensor 37 Tolerance Variations, Run S49-F: Adaptive Tolerance, Compression Ratio Monitoring	3-47
3-5	Control Function for Queue/Integral-Queue Control System	3-48
3-6	Characteristic Tolerance and Queue Length Behavior, With Integral Feedback Added	3-49
3-7	Typical Reconstructed Data, With Integral Feedback Added	3-50
3-8	Maximum Tolerance Variations, With Integral Feedback Added	3-51
3-9	Average Tolerance Variations, With Integral Feedback Added	3-52
3-10	Control System Behavior Without Integral Feedback	3-53
3-11	Maximum Queue Length Variations, With Integral Feedback Added	3-54
3-12	Figure-of-Merit Variations, With Integral Feedback Added	3-55
3-13	Tolerance and Queue Length Behavior, Run S49-H	3-56
3-14	Tolerance and Queue Length Behavior, Run S49-I	3-59
3-15	Tolerance and Queue Length Behavior, Run S49-J	3-62
3-16	Sensor 50 Reconstructed Data, Run S49-I: Queue/Integral-Queue Control	3-65
3-17	Sensor 50 Reconstructed Data, R S43-J Queue/Integral-Queue Control	3-66
3-18	Exponential Filtering Control Function	3-67
3-19	Running-Average Filtering Control Function	3-68
3-20	Queue Length Time Histories, Runs S6-A, S6-G, and S6-H	3-69
3-21	Sensor 7 Reconstructed Data, Run S6-A: No Filtering or Queuing Control	3-70
3-22	Sensor 7 Reconstructed Data, Run S6-B: Exponential Filtering, Fixed-Parameter	3-71
3-23	Sensor 7 Reconstructed Data, Run S6-G Running-Average Filtering, Fixed-Parameter	3-72

Figure		Page
3-24	Sensor 7 Reconstructed Data, Run S6-E: Sample Deletion Only, Fixed-Parameter	3-73
3-25	Averaging Period Variations, Run S6-F Adaptive Running-Average Filtering	3-74
3-26	Sensor 7 Reconstructed Data, Run S6-G: Adaptive Sample Deletion	3-75
3-27	Sensor 7 Reconstructed Data, Run S6-H: Adaptive Tolerance	3-76
3-28	Sensor 37 Reconstructed Data, Run S49-K: Exponential Filtering, Fixed-Parameter	3-77
3-29	Sensor 37 Reconstructed Data, Run S49-L: Running-Average Filtering, Fixed-Parameter	3-78
3-30	Sensor 37 Reconstructed Data, Run S49-M: Sample Deletion Only, Fixed-Parameter	3-79
3-31	Queue Length Behavior With Adaptive Filtering and Sample Deletion, S-49 Data	3-80
3-32	Sensor 37 Reconstructed Data, Run S49-N: Adaptive Exponential Filtering	3-82
3-33b	Sensor 37 Exponential Factor Variations, Run S49-N: Adaptive Exponential Filtering	3-83
3-33a	Sensor 37 Reconstructed Data, Run S49-P: Adaptive Running-Average Filtering	3-84
3-33b	Sensor 37 Averaging Period Variations, Run S49-P: Adaptive Running-Average Filtering	3-85
3-34	Queue Length Behavior With Finite Buffer, Run S49-R	3-86
3-35	Sensor 37 Reconstructed Data, Run S49-Q: With Buffer Overflow	3-88
3-36	Sensor 37 Reconstructed Data, Run S49-R: With Buffer Overflow	3-89
4-1	CPCM Bit Stream Record Formats	4-4
4-2	Expected Error Versus Original Data Magnitude	4-8
4-3	Error in Reconstruction Caused by Faulty Bit Stream Magnitude Word, First-Order Selectors	4-10
4-4	Errors in Reconstruction Caused by Faulty Sensor ID, First-Order Selectors	4-11
4-5	Sensor ID Errors Caused by Missing Line Segments First-Order Selectors	4-13
4-6	Error Due to Faulty Line Segments in the ZOP/ZOI Format	4-17
4-7	Rms Error/$\sqrt{R_E}$ Versus Sample Compression Ratio	4-21
4-8	CPCM Transmission Link Simulation	4-25

Figure		Page
4-9	PCM Transmission Link Simulation	4-26
4-10	Data Reconstruction and Error Analysis	4-27
4-11	Error Caused by Magnitude Bit Error, First-Order Interpolator	4-29
4-12	Error Caused by Line Segment Additions With Wrong ID Assignments, First-Order Interpolator	4-30
4-13	Error Caused by Line Segment Eliminations With Wrong ID Assignments, First-Order Interpolator	4-31
4-14	Error Caused by Magnitude Bit Error, Zero-Order Selector	4-33
4-15	Error Caused by Line Segment Additions With Wrong ID Assignments, Zero-Order Selector	4-34
4-16	Error Caused by Line Segment Eliminations With Wrong ID Assignments, Zero-Order Selector	4-35
4-17	Rms Error Versus Bit Error Rate	4-37
5-1	Performance Criteria Weighting	5-6
6-1	Layout of OGO Operations Control Center	6-4
6-2	Input/Output Device for OGO Real-Time Computer System	6-5
6-3	Block Diagram of OGO Operations Control Center Equipment Layout	6-6
6-4	36-Bit Word From PCM Station	6-8
6-5	Minor Frame Computer Word Format	6-9
6-6	Subcommutator Computer Word Format	6-9
6-7	Block Diagram of Real-Time Computation System	6-11
6-8	Functional Block Diagram of Control Center Real-Time Processing System	6-12
6-9	Timing for Critical Operations of Computation System	6-14
6-10	Limit Check Boundary Intercept Determination for FOIDIS-Compressed Data	6-9
6-11	Operating Times With Compressed Data, SDS-320	6-26
6-12	Operating Times With Compressed Data, SDS-330	6-29
6-13	Simplified Block Diagram of Real-Time Calculation Operation on Compressed Test Data	6-32
6-14	Pictorial Presentation of Compressed Data With and Without Forced Output	6-33
6-15	Interlace Register	6-36

Figure		Page
7-1	GSFC Data Processing Operations	7-5
7-2	Buffer Tape Data Record Format	7-8
7-3	Edit Tape Data Record Format	7-11
7-4	Decommutation Process	7-13
7-5	Decommutation Data Matrix	7-14
7-6	Normal Production Decommutation Process	7-16
7-7	Compressed Tape Frame Size	7-18
7-8	Pictorial Presentation of Compressed Data With & Without Forced Output	7-19
7-9	Compressed Data Reconstruction	7-21
7-10	Transfer of Data From Bulk to Operating Storage for Data Reconstruction	7-23
7-11	Predetermined Forced Outputs	7-25
7-12	Estimated Timing for Decommutation Program	7-27
7-13	Compressed Data Arrays	7-28
7-14	Record Size as a Function of Compression Ratio	7-32
7-15	Tape Readin Time Differences	7-34
7-16	Tradeoff Between Tape Read Time and Reconstruction Time	7-35
8-1	Simplified Diagram of OGO Control Center and Communications Center	8-2
8-2	Data Acquisition Station	8-5
8-3	Data Compression System	8-6
8-4	Remote Station With Vehicle Compression	8-8
8-5	GSFC Compressed Data Handling System	8-9
8-6	Remote Station With Ground Compression	8-11

TABLES

Table		Page
2-1	S-6 Sensor Designations	2-3
2-2	S-49 Sensor Designations, Equipment Group 1	2-7
2-3	Selector Performance for S-6 Data	2-28
2-4	Selector Performance for S-49 Data	2-29
2-5	Bit Plane Encoding Performance for S-6 and S-49 Data	2-30
2-6	Performance of Exponential Interpolator on S-6 Data	2-31
2-7	Performance of Short-Term Frequency Component Selector	2-32
2-8	Combined Bit Compression Ratios, $\overline{N}_b$, for Rms Error of 0.5% Full Scale	2-33
2-9	Broadside Performance Results	2-35
3-1	Compression Run Summary, Queue Length and Compression Ratio Monitoring, S-49 Data	3-7
3-2	S-49 Sensor Tolerances	3-8
3-3	Results by Individual Sensor, Queue Length and Compression Ratio Monitoring, S-49 Data	3-19
3-4	Rms Errors in Reconstructed Data, Queue/Integral-Queue Control System	3-26
3-5	Compressor Run Summary, Queue/Integral-Queue Control System	3-30
3-6	Results by Individual Sensor, Queue/Integral-Queue Control System, S-49 Data	3-31
3-7	Compressor Run Summary, Adaptive Filtering and Sample Deletion, S-6 Data	3-27
3-8	Assigned Tolerances for S-6 Data	3-28
3-9	Results by Individual Sensor Adaptive Filtering and Sample Deletion, S-6 Data	3-30
3-10	Compressor Run Summary, Adaptive Filtering and Sample Deletion, S-49 Data	3-33
3-11	Results by Individual Sensor, Adaptive Filtering and Sample Deletion, S-49 Data	3-35

Table		Page
4-1	Bit Error Probabilities	4-5
4-2	Theoretical Rms Errors	4-23
4-3	Rms Errors Found in Reconstructed Data	4-28
4-4	Rms Errors Found in Uncompressed Data	4-36
5-1	Compression Model Performance Factors	5-8
6-1	Minor Frame Format Modes	6-2
6-2	Pe[illegible] OGO Operations Control Center Equipment	6-7
6-3	Service and Special Purpose Operations Performed at the Control Center	6-7
6-4	Approximate Real Time Data Processing Times	6-8
6-5	Summary of Real-Time System Computation Capacity	6-13
6-6	Tabulation of Program INTERP Instructions	6-20
6-7	Estimated Computation Time for Linear Interpolation	6-20
6-8	Continuous [illegible] Check Parameters for OGO-C	6-22
6-9	Operating Times With Compressed Data, SDS-920	6-25
6-10	Operating Times With Compressed Data, SDS-930	6-28
6-11	Time Required for Reconstruction of Parameters for SSP Calculations	6-35
7-1	Minor Frame Format Modes	7-2
7-2	Central Station Computation Systems Equipment Listing	7-3
7-3	Estimated Timing for Major OOO Computer Programs	7-6
7-4	Buffer Tape Format	7-7
7-5	Edit Tape Format	7-10
7-6	Estimated Computation Times for Data Reconstruction	7-22
7-7	Time for Reconstruction and Restoration – UNIVAC 1107	7-29
7-8	Time for Reconstruction and Restoration – UNIVAC 1103	7-30
8-1	NASA/GSFC DSN Acquisition and Tracking Stations	8-3

Section 1
INTRODUCTION

Data compression provides an answer to a major problem that currently faces the field of information technology: the increasing volume of scientific and engineering data being telemetered and processed.

Telemetry, an absolute necessity to the development of missiles and spacecraft, is a wasteful user of the radio-frequency spectrum. There are vehicles operating now that must use several r-f links to transmit all the data they obtain. As space experiments and vehicles become more complex, the dilemma becomes more serious and more widespread: masses of data virtually saturate ground networks and tax their ability for vital real-time data handling; preflight checkout time and costs increase; demands on data-processing time and facilities exceed practical limits; and data-storage archives overflow.

1.1 PURPOSE

The objective of this study was to investigate ways to improve the utilization efficiency of NASA/GSFC PCM telemetry links through data compression. The work was in three general areas:

- Study and evaluation of data compression system techniques
- Study of the effect of transmission link noise on compressed data
- Analysis of the effect of compressed data on GSFC computer operations
- Determination of the requirements of a ground-based data compression and reconstruction system

Activity during Phase I was confined primarily to the first area. This included a literature search and subsequent evaluation of known data compression techniques

and a study of time-and-channel identification (Ref. 4). The technique evaluation was done with the aid of digital computer simulation using, as a model, PCM telemetry a from the S-6, Explorer XVII satellite.

The second phase of the study extended the first-phase evaluation by using additional telemetry data from the S-49, Orbiting Geophysical Observatory-A (OGO-A) satellite, by investigating transmission link noise effects, and by performing computer operations analyses. Near the end of the second phase, a study was made of the configuration of a data compression and reconstruction system for the OGO ground data transmission network, and of required ground facility modifications for the incorporation of data compression.

1.2 SCOPE

This report has 10 sections and 1 appendix. Section 1 introduces the problem, the project, and the report. An evaluation of compression models based on redundancy reduction effectiveness is presented in Section 2. Section 3 evaluates buffer queuing control and Section 4 describes the study of the effect of transmission link noise on compressed data. A tradeoff analysis of the compression models is presented in Section 5. Analyses of digital computer operations with compressed data are found in Sections 6 and 7. Section 8 presents the results of the data compression system configuration study. Conclusions and recommendations are given in Section 9, and the bibliography is in Section 10. Appendix I compares the effect of compression error with the total rms errors encountered in the link noise study.

1.3 PROJECT

Phase II, which includes an evaluation of techniques, a noise study, a computer utilization study, and a system conceptual design study, was divided into seven tasks, beginning with Task VII.

1.3.1 Task VII, S-49 Compression Model Analysis

Task VII involved an extension of the compression technique evaluation by computer simulation under Phase I, using telemetry data from a more recently launched satellite than the S-6. The data used were from the S-49, OGO-A satellite.

1.3.2 Task VIII, New Data Compression Model Evaluation

Task VIII evaluated, by computer simulation, new data compression techniques which were developed late in the first phase of the study. These techniques included sample selectors as well as [illegible] of buffer feedback for the purpose of queuing control.

1.3.3 Task IX, Theoretical Buffer Analysis

Task IX involved a study of adaptive buffer fullness control on a purely theoretical basis. The theoretical evaluation of new feedback methods before computer simulation materially increased the efficiency of the search for new techniques. The evaluation of existing methods lent credence to the results of tests previously performed.

1.3.4 Task X, Data Compression Mechanization Study

This task was concerned with the problems of incorporating a data compression and reconstruction system into the NASA/GSFC ground information transmission network. It was divided into three major subtasks.

1.3.4.1 <u>System Configuration Analysis</u>. This subtask investigated OGO data acquisition stations, the control center station, and data transmission requirements, to determine (1) the configuration of a data compression and reconstruction system for the ground data transmission network, and (2) the required ground facility modifications for the incorporation of this system.

1.3.4.2 Secondary Data Compression. This subtask studied the effects of compressing data which have been previously compressed, assuming that secondary compression is implemented by a higher-order compression algorithm.

1.3.4.3 Transmission Link Simulation. This subtask investigated, by computer simulation, the effect of ground transmission link noise on compressed data. The effect on the reconstructed data of noise added to the compressed data transmission link bit stream was compared with the effect of the same noise added to the uncompressed data bit stream.

1.3.5 Task XI, Theoretical Transmission Error Analysis

Task XI analyzed, on a theoretical basis, the effect of errors which may occur in a compressed data bit stream. An analysis of the effect of transmission errors investigated in Task X on various data compression systems was also done as a part of Task XI.

1.3.6 Task XII Real-Time Computer Analysis

This task investigated the capability of a digital computer to handle real-time compressed data for spacecraft status checks and experiment monitoring. A comparison was made between the times required to handle telemetry data in a compressed format and those necessary to handle the same data in the present standard format.

1.3.7 Task XIII, Data Processing Computer Analysis

Task XIII compared computer operating times and other requirements necessary to process telemetry data in a compressed format with requirements for processing the same data in the present standard format.

1.4 BACKGROUND

During the first phase of this study, a preliminary investigation was made of the relative effectiveness of known data compression techniques as applied to GSFC satellite telemetry data.

The following steps were taken:

- Literature search
- Compression model selection
- Compression model and buffer simulation
- Time and channel identification study
- Proposal of optimum compression system

1.4.1 Literature Search

The initial step was a search of international literature on data compression. A list of over 30 different methods was compiled; each method was critically examined to determine its probable merits.

1.4.2 Compression Model Selection

The second step was a elimination process to reduce the number of models found during the literature search to a small group containing those showing the greatest promise of high performance. Six data compression models were placed in this group for subsequent testing. These included five polynomial sample selectors and one bit plane encoder technique.

1.4.3 Compression Model and Buffer Simulation

The next step, constituting the bulk of the work, was divided into two parts: (1) a compression model simulation analysis and (2) an adaptive buffer queuing control study.

1.4.3.1 Compression Model Simulation Analysis. Using an IBM 7094 digital computer, simulation analyses were made of the compression models chosen in the second step. S-6 data waveforms, used in this simulation, were selected to match particular data classes specified by GSFC. The results of these tests showed that a particular type of first-order polynomial interpolator achieved the highest performance in reducing the bandwidth of the S-6 telemetry data. This interpolator is capable of fitting straight lines to the sampled data waveform by adjusting their end points with 4 degrees of freedom.

Another part of the simulation analysis investigated the effects of precompressed filtering and adaptive techniques on compression efficiency and on reconstructed data fidelity. The results of this investigation indicated that both adaptive and non-adaptive precompression filtering may effectively enhance data compressor performance. This portion of the study, however, was too brief to be conclusive.

1.4.3.2 Adaptive Buffer Queuing Control Study. In this study, simulation tests were performed to determine the type of queuing control, if any, that would be needed to curtail or eliminate data loss from buffer overflow in a telemetry data compressor for an S-6 satellite. The most important result of this study was that, because of the data stationarity, adaptive queuing control would not be required for the S-6 data as long as the buffer readout rate exceeded the average readin rate by a factor of at least 1/0.98. However, to guard against the possibility of an abnormally low readout rate setting, a control system which monitored queue length was found to be necessary.

1.4.4 Time and Channel Identification Study

Relative efficiencies of several methods of time and channel coding were investigated. It was found that the transmission of a minor frame channel identification word with each data sample was the most efficient coding technique for S-6 data among those studied. With this method, at least one data/channel word in each minor frame must be transmitted in order to maintain a time reference and to identify subcommutated sensors.

1.4 5 Proposal of Optimum Compression Models

The final step consisted of a tradeoff analysis of the six compression models tested. The models which demonstrated the highest overall performance were tentatively proposed for compressing S-6 and similar data.

The spread in compression model performance, resulting from the tradeoff analysis, was surprisingly low. The first-order 4-degree-of-freedom interpolator mentioned above was proposed for the ground-based data compression applications where use this model is relatively complex, a simpler – though less effective – first-order interpolator was proposed for space applications. Queuing control, time and channel coding, and reconstruction methods were proposed for both applications.

As the study progressed, it became apparent that the effectiveness of existing compression techniques on S-6 data was not as high as has been experienced previously at LMSC on other data. A typical example of the overall bandwidth compression ratios obtained on the S-6 data is 3.3:1. A number of methods for improving the compression were proposed including partial data processing before transmission (signal reduction) and methods to take advantage of the periodicity which exists in many of the S-6 data waveforms. It was estimated that, with the proper application of these techniques, bandwidth compression ratios exceeding 10:1 can be achieved on S-6 data. Therefore recommendations for future study include investigations of these data compression techniques.

Section 2
COMPRESSION MODEL EVALUATION

One of the major objectives of Phase II was to extend the investigation of selector effectiveness begun during Phase I. This extension followed several paths. The compression techniques evaluated during Phase I (Ref. 4) were reexamined, using telemetry data from the S-49, OGO-A, satellite, and additional data from the S-6, Explorer XVII, satellite. In addition, a number of compression techniques, which were developed but not tested during Phase I, were simulated and empirically evaluated during Phase II. These new compression techniques are exponential interpolation, short-term frequency component selection, and broadside data compression.

A complete list of the data compression techniques evaluated in this study is as follows:

- Zero-order predictor (floating-aperture), ZOP
- Zero-order interpolator (computed sample transmitted), ZOI
- First-order interpolator (4-degree-of-freedom), FOI4DG
- First-order interpolator (disjoined line segment), FOIDIS
- First-order interpolator (offset-out-of-tolerance), FOIOOT
- Bit plane encoding (BPE)
  - Binary coded
  - Gray-coded
- Exponential interpolator (EXPINT)
- Short-term frequency component selector (STFC)
- Broadside data compression

Descriptions of the first six models on the list are found in Ref. 4. The new compression techniques (the last three on the list) are described in Par. 2.1. A performance evaluation of all the techniques except broadside data compression is given in Par. 2.2. Broadside compression is evaluated in Par. 2.3.

The investigation of data compression techniques included a study of secondary data compression. This subtask examined the effects of compressing data which have been previously compressed The study is described in Par. 2 4

The S-6 satellite data chosen for this study was received on 10 April 1963. The nominal overall sample rate of the S-6 data was 900 samples per second, and, since one frame consisted of 45 channels, the frame rate was 20 frames per second. Three of the channels (17, 31, and 33) were submultiplexed with 16 positions e The data words consisted of 9 bits. Most of the sensors were cross-s applied onto more than one channel; the data amplitude range of 0 to 511 decimal counts was left undisturbed. Assigned sensor identification numbers, and their corresponding channel numbers, are shown in Table 2-1. Explorerer XVII was spin-stabilized at a rate of approximately 1.5 revolutions per second. This caused many of the sensor waveforms to be cyclic in nature.

T E-49 PCM telemetry data used in this study consisted of 9-bit words, taken and transmitted in real time at a rate of 64 kilobits per second One minor frame contained 128 data channels The frame rate was thus 55 6 frames per second, with an overall word rate of 7111 words per second The multiplexed format was that of Equipment Group 1 A 128-word minor frame contained the following:

- 78 words representing analog data (only 8 bits of the 9-bit words were used)
- 27 digital words depicting digital information
- 3 sync words
- 3 accumulated spacecraft time words
- 3 position and status words
- 3 subcom words
- 1 spare channel

There were 68 data sensors, excluding the subcom channels, 67 of which were grouped into 18 experiments. The 68th sensor denoted the shaft position of Orbital Plane

Table 2-1

S-6 SENSOR DESIGNATIONS

Sensor No.	Channel and Position No.	Avg. Sample Rate (sps)	Sensor Name
1	1	20	Position and status
2	2, 7, 12, 18, 23, 28, 34, 39, 44	180	Electron temp #1 output
3	16, 40	40	Electron temp #2 output
4	3, 19, 35	60	Bayard alpert guage #1 DC
5	4, 20, 36	60	Bayard alpert guage #1 AC
6	5, 21, 37	60	Bayard alpert guage #2 DC
7	6, 22, 38	60	Bayard alpert guage #2 AC
8	8, 32	40	Aspect
9	9, 25, 41	60	Mass spect #1 log amp output
10	24	20	Mass spect #1 elect biased output
11	10, 26, 42	60	Redhead guage #1
12	11, 27, 43	60	Redhead guage #2
13	13, 29, 45	60	Mass spect #2 log amp output
14	30	20	Mass spect #2 elect biased output
15	14	20	Zero-volt reference
16	15	20	Five-volt reference
17	17, pp. 15	1.25	Ionization current/voltage
18	17, pp. 0	1.25	High-voltage monitor
19	17, pp. 1	1.25	Repeller monitor
20	17, pp. 2	1.25	Amplifier B+ monitor
21	17, pp. 3	1.25	Mass marker
22	17, pp. 4	1.25	Log amp temp
23	17, pp. 5	1.25	Electrometer temp
24	17, pp. 13	1.25	Zero-volt refere e
25	17, pp. 14	1.25	Five-volt reference
26	31, pp. 15	1.25	BA emission current #1
27	31, pp. 0	1.25	BA emission current #2

Table 2-1 (cont'd)

Sensor No.	Channel and Position No.	Avg. Sample Rate (sps)	Sensor Name
28	≥1, pp 1	1.25	BA guage temp #1
29	≥1, pp. 2	1.25	BA guage temp #2
30	≥1, pp. 3	1.25	BA electrometer temp #1
31	o1, pp. 4	1.25	BA electrometer temp #2
32	≥1, pp. 5	1.25	BA voltage bias #1
33	≥1, pp. 6	1.25	BA voltage bias #2
34	≥1, pp. 7	1.25	Sphere pressure
35	≥1, pp. 8	1.25	Nutation damper squib monitor
36	≥1, pp. 9	1.25	Experiment squib monitor
37	≥1, pp. 10	1.25	Turn-on counter
38	≥1, pp. 11	1.25	T/M voltage (+20.15v)
39	≥1, pp. 12	1.25	9.3v battery monitor
40	≥1, pp. 13	1.25	Five-volt reference
41	≥1, pp. 14	1.25	Zero-volt refe>e e
42	≥3, pp. 15	1.25	Zero-volt reference
43	≥3, pp. 0	1.25	Five-volt reference
44	≥3, pp. 1	1.25	BA card temp #1
45	≥3, pp. 2	1.25	BA card temp #2
46	≥3, pp. 3	1.25	BA ion trap filament monitor #1
47	≥3, pp. 4	1.25	BA ion trap filament monitor #2
48	≥3, pp. 5	1.25	6.2v battery monitor
49	≥3, pp. 6	1.25	-27.9v battery monitor
50	≥3, pp. 7	1.25	Thermistor #1 (lower MS boss)
51	≥3, pp. 8	1.25	Thermistor #2 (skin, lower 45°)
52	≥3, pp. 9	1.25	Thermistor #3 (ambient, top of RH)
53	≥3, pp. 10	1.25	Thermistor #4 (skin, equator)
54	≥3, pp. 11	1.25	+13.95v monitor
55	o3, pp. 12	1.25	Thermistor #6 (upper MS boss)

Table 2-1 (concl'd)

Sensor No.	Channel and Position No.	Avg. Sample Rate (sps)	Sensor Name
56	33,pp. 13	1.25	Thermistor #5 (skin, upper 45°)
57	33,pp. 14	1.25	Spare
58	17,pp. 6	1.25	Ionization current/voltage
59	17,pp. 7	1.25	High-voltage monitor
60	17,pp. 8	1.25	Repeller monitor
61	17,pp. 9	1.25	Amplifier B+ monitor
62	17,pp. 10	1.25	Mass marker
63	17,pp. 11	1.25	Log amp temp
64	17,pp. 12	1.25	Electrometer temp

Experimental Package (OPEP)2. Sensor identification numbers of the S-49 data, together with corresponding experiment and channel numbers and sensor sample rates, are shown in Table 2-2.

2.1 NEW DATA COMPRESSION TECHNIQUES

2.1.1 Exponential Interpolator

The principle of exponential interpolation is similar to that of polynomial interpolation, differing only in the use of exponential reference patterns instead of polynomial reference patterns. The exponential interpolation process consists of fitting the longest possible exponential waveform to the data, such that a given peak error criterion between the data and this exponential waveform is not exceeded.

Although this process is very similar to that of polynomial interpolation, at least one significant complication must be considered. Not just one, but two, different classes of exponential reference patterns exist, and should be considered simultaneously. The first class of reference patterns consists of growing exponential waveforms, while the second class of reference patterns consists of asymptotic exponential waveforms. A general expression which describes the members of the first class of reference patterns is

$$y_n = a_1 \pm b_1 u_1^n; (u_1 > 1), n = 0, 1, 2, \qquad (2\text{-}1a)$$

A general expression which describes the members of the second class of reference patterns is

$$y_n = a_2 \pm b_2 (1 - u_2^n); (u_2 < 1), n = 0, 1, 2, \qquad (2\text{-}1b)$$

where

y_n = n^{th} sample magnitude

a_1, b_1, c_1, a_2, b_2, and u_2 = constants which establish the exact pattern shape

Table 2-2

S-43 SENSOR DESIGNATIONS

EQUIPMENT GROUP 1

Experiment No.	Sensor No.	Channel No.	Avg. Samp e Rate (sps)	Analog or Digital	Sensor Name
1	1	81, 82, 83, 84	222.4	D	Pulse Height Data
2	2	4	55.6	A	SOEP 2 Probe, Output 1
	3	5	55.6	A	SOEP 2 Probe, Output 2
	4	6	55.6	A	SOEP 2 Probe, Output 3
	5	7	55.6	A	SOEP 2 Probe, Output 4
	6	21	55.6	A	OPEP 2 Probe, Output 1
	7	22	55.6	A	OPEP 2 Probe, Output 2
	8	23	55.6	A	High Voltage Readout
3	9	10, 42, 74, 106	222.4	A	Proton Cup Data & Monitors
	10	36, 107	111.2	A	Electron Cup Data & Monitors
4	11	55	55.6	A	A Pulse Height Data
	12	56	55.6	A	B Pulse Height Data
	13	57	55.6	D	Pulse Count
5	14	13, 77	111.2	A	Electrometer Output
	15	25, 89	111.2	D	Particle Count
	16	104	55.6	A	Absorber Wheel Position
6	17	58	55.6	D	A Pulse Height Data
	18	59	55.6	D	B Pulse Height Data
	19	60	55.6	D	Events Count, Both Telescopes

Table 2-2 (Continued)

Experiment No.	Sensor No.	Channel No.	Avg. Sample Rate (sps)	Analog or Digital	Sensor Nam
7	20	19	55.6	D	D'1 Pulse Height Data
	21	20	55.6	A	D4 Rate Data
	22	28,52	111.2	D	Combination Rate Data
	23	29	55.6	A	D5 Rate Data
	24	49	55.6	D	D1 Pulse Height Data
	25	50	55.6	D	D3 Pulse Height Data
	26	51	55.6	D	D4 Pulse Height Data
	27	53	55.6	A	D'4 Rate Data
8	28	114	55.6	D	Particle Count, 2^8 to 2^{14}
	29	115	55.6	D	Particle Count, 2^0 to 2^7
9	30	43,90,113	166.8	D	Ion Chamber Data
	31	100	55.6	A	Ion Chamber Electrometer Output
	32	116,121,122, 123,124	278.0	D	Electron Spectrometer Data
10	33	14,37,61,86, 117	278.0	A	X Axis Waveform
	34	15,38,62,87, 118	278.0	A	Y Axis Waveform
	35	16,39,63,88, 119	278.0	A	Z Axis Waveform
	36	80	55.6	A	Z Axis Waveform 10 cps

Table 2-2 (Continued)

Experiment No.	Sensor No.	Channel No.	Avg. Sample Rate (sps)	Analog or Digital	Sensor Name
11	37	30,94	111.2	A	Fluxgate Coil A, -30 to +30 Gamma
	38	31,95	111.2	A	Fluxgate Coil B, -30 to +30 Gamma
	39	32,96	111.2	A	Fluxgate Coil C, -30 to +30 Gamma
	40	91	55.6	A	Fluxgate Coil A, -500 to +500 Gamma
	41	92	55.6	A	Fluxgate Coil B, -500 to +500 Gamma
	42	93	55.6	A	Fluxgate Coil C, -500 to +500 Gamma
	43	102	55.6	A	Rb Vapor Magnetometer Data
13	44	125	55.6	A	Electron Sphere Output
	45	126	55.6	A	Sphere Voltage Status
	46	127	55.6	A	Ion-Proton Sphere Output
	47	128	55.6	A	Sweep, Bias, & Energy Modes
14	48	9,12,41,44, 73,76,105,108	444.8	D	Sensor Controls & Data
15	49	24	55.6	A	1-6 amu 10^{-10} Current Data
	50	27	55.6	A	1-6 amu 10^{-11} Current Data
	51	40	55.6	A	1-6 amu 10^{-12} Current Data
	52	54	55.6	A	1-6 amu 10^{-13} Current Data
	53	68	55.6	A	1-6 amu 10^{-14} Current Data
	54	69	55.6	A	7-45 amu 10^{-10} Current Data
	55	70	55.6	A	7-45 amu 10^{-11} Current Data
	56	71	55.6	A	7-45 amu 10^{-12} Current Data
	57	72	55.6	A	7-45 amu 10^{-13} Current Data
	58	109	55.6	A	7-45 amu 10^{-14} Current Data

Table 2-2 (Continued)

Experiment No.	Sensor No.	Channel No.	Avg. Sample Rate (sps)	Analog or Digital	Sensor Name
19	59	46,110	111.2	A	0.2–1.6 kc Receiver Output
	60	47,111	111.2	A	1.6–12.5 kc Receiver Output
	61	48,112	111.2	A	12.5–100 kc Receiver Output
18	62	8,26,45,64, 85,103,120	389.2	A	Receiver IF Voltage
19	63	78	55.6	A	Ion Chambers A, B, C, Combined Output
	64	79	55.6	A	Ion Chamber D Output
20	65	11	55.6	D	Filter Number
	66	17	55.6	D	Light Intensity
	67	18	55.6	D	Intensity & Scan Line No
—	68	75	55.6	A	OPEP 2 Shaft Position
—	69	101	55.6	—	Spare
—	70	1	55.6	D	Sync Word 1
	71	2	55.6	D	Sync Word 2
	72	3	55.6	D	Sync Word 3
—	73	33	55.6	D	Accumulated Time, 2^{24} to 2^{18}
	74	34	55.6	D	Accumulated Time, 2^{17} to 2^{9}
	75	35	55.6	D	Accumulated Time, 2^{8} to 2^{0}
—	76	65	55.6	D	Position & Status 1
	77	66	55.6	D	Position & Status 2
	78	67	55.6	D	Position & Status 3
—	—	97	—	—	Subcom 1 Omitted
	—	98	—	—	Subcom 2 Omitted
	—	99	—	—	Subcom 3 Omitted

In each case, the sign of the second term determines whether the waveform increases or decreases with increasing n

In order to specify a particular member of either class of exponential waveforms, three quantities must in turn be specified; namely, a, b, and u. If it is desired to fit an exponential waveform from a given class through two given data values, there exist, in general, an infinite number of possible combinations of a, b, and u which achieve the desired fit. To allow only one possible curve, b is specified a priori; the values of the remaining two quantities a and u which achieve the desired fit thus specify a single curve.

2.1.2 Short-Term Frequency Component Selector

With the short-term frequency component selector, a Fourier analysis is performed on a prescribed number of most recent data samples, the analysis being updated as new data samples are received. This short-term frequency analysis is then used to decide whether the most recent data sample is to be retained. If a significant fraction of the total short-term data energy is present at relatively high frequencies, the most recent data sample is likely to be retained. If a significant fraction of the total short-term data energy is present at relatively low frequencies, the most recent data sample is likely to be discarded. This sample selection technique tends to maintain the short-term sampling rate of the data in proportion to the short-term information bandwidth of the data. This selector provides no peak error guarantee.

2.1.3 Broadside Data Compression

Because of the modulation due to satellite spin, many Goddard satellite data waveforms are periodic. This periodicity is added to normal slow data variations caused by the orbital movement of the vehicle. Such data are very difficult to compress, although they are inherently highly redundant. One way to cope with periodic waveforms, such as those caused by spin of the satellite vehicle, is to separate data samples that occur at corresponding portions of succeeding waveform cycles into

groups, or "pseudo-sensors," and then to compress, concurrently and independently, each of the groups. Reconstruction can be accomplished by first restoring the missing samples in each group, and then reassembling the waveform by interlacing the groups. In this way the cyclic nature of the waveform, as well as its phase, would be preserved.

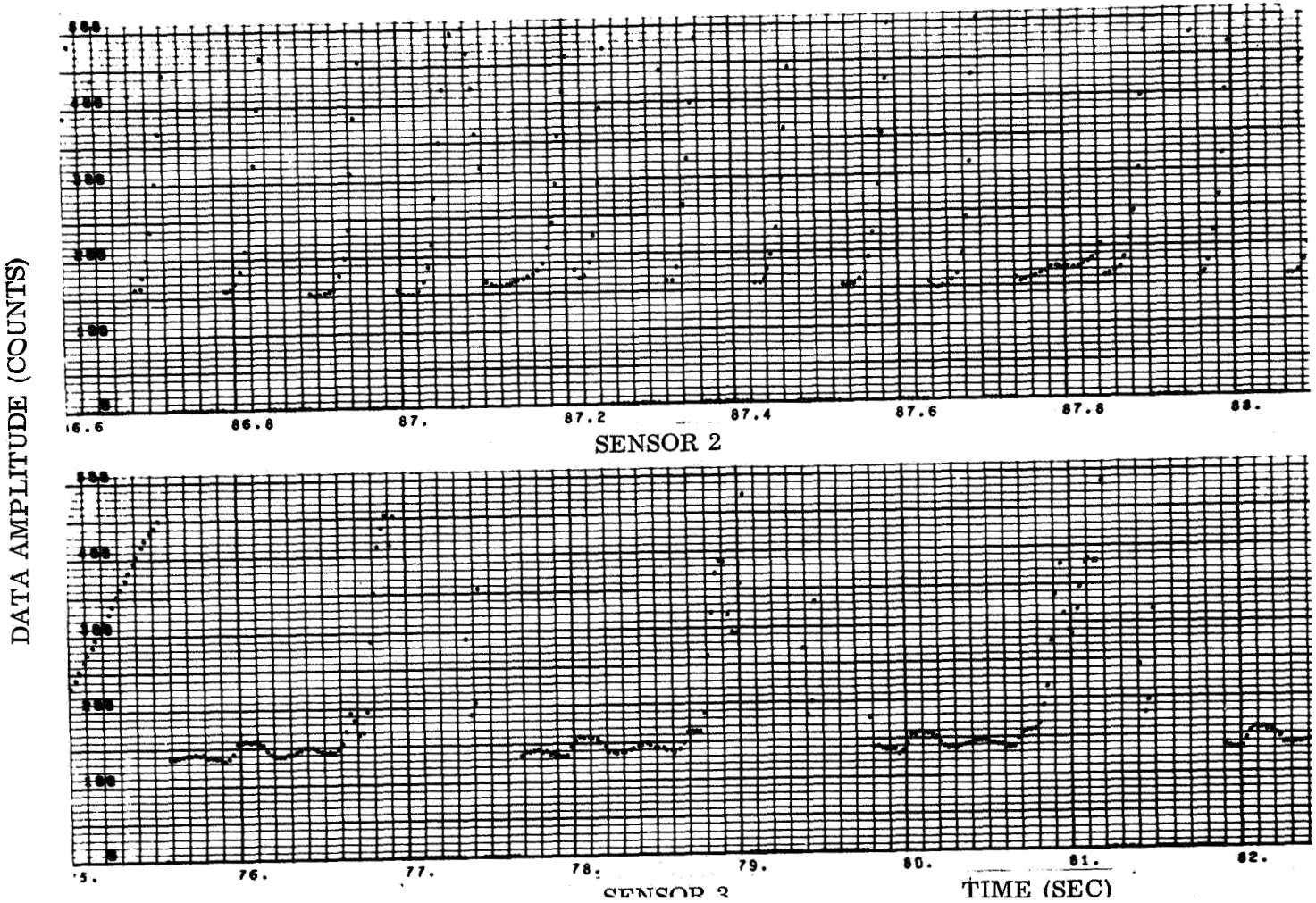
Concurrent action on all portions of the waveform cycle is a "broadside" approach to the problem of redundancy removal from cyclic waveforms. It may be thought of as a method of putting cyclic data into a form which can be more efficiently compressed using conventional redundancy removal techniques. Thus the broadside technique is itself not a selector, although it embodies sample selection.

The implementation of broadside data compression could be accomplished by synchronizing the multiplexer to the roll period of the satellite. If synchronous multiplexer operation is difficult or impossible it is feasible to operate the multiplexer asynchronously, but to group together the corresponding data samples by means of a post-multiplexer sorting process. This can be done with reference to appropriate aspect information.

2.8 SELECTOR PERFORMANCE

Figures 8-1 and 8-8 show portions of the waveforms of the data specimens used in the selector performance evaluation. The total number of samples processed in each specimen was as follows:

S-6 Sensor 2	: 1053 samples
S-6 Sensor 3	: 1170 samples
S-6 Sensor 7	: 1170 samples
S-6 Sensor 9	: 1170 samples
S-6 Sensor 11	: 1170 samples
S-49 Sensor 14	: 1170 samples
S-49 Sensor 37	: 1403 samples
S-49 Sensor 50	: 702 samples
S-49 Sensor 60	: 1000 samples
S-49 Sensor 63	: 701 samples



Fig, **2-1** ***S-6*** Data Specimens

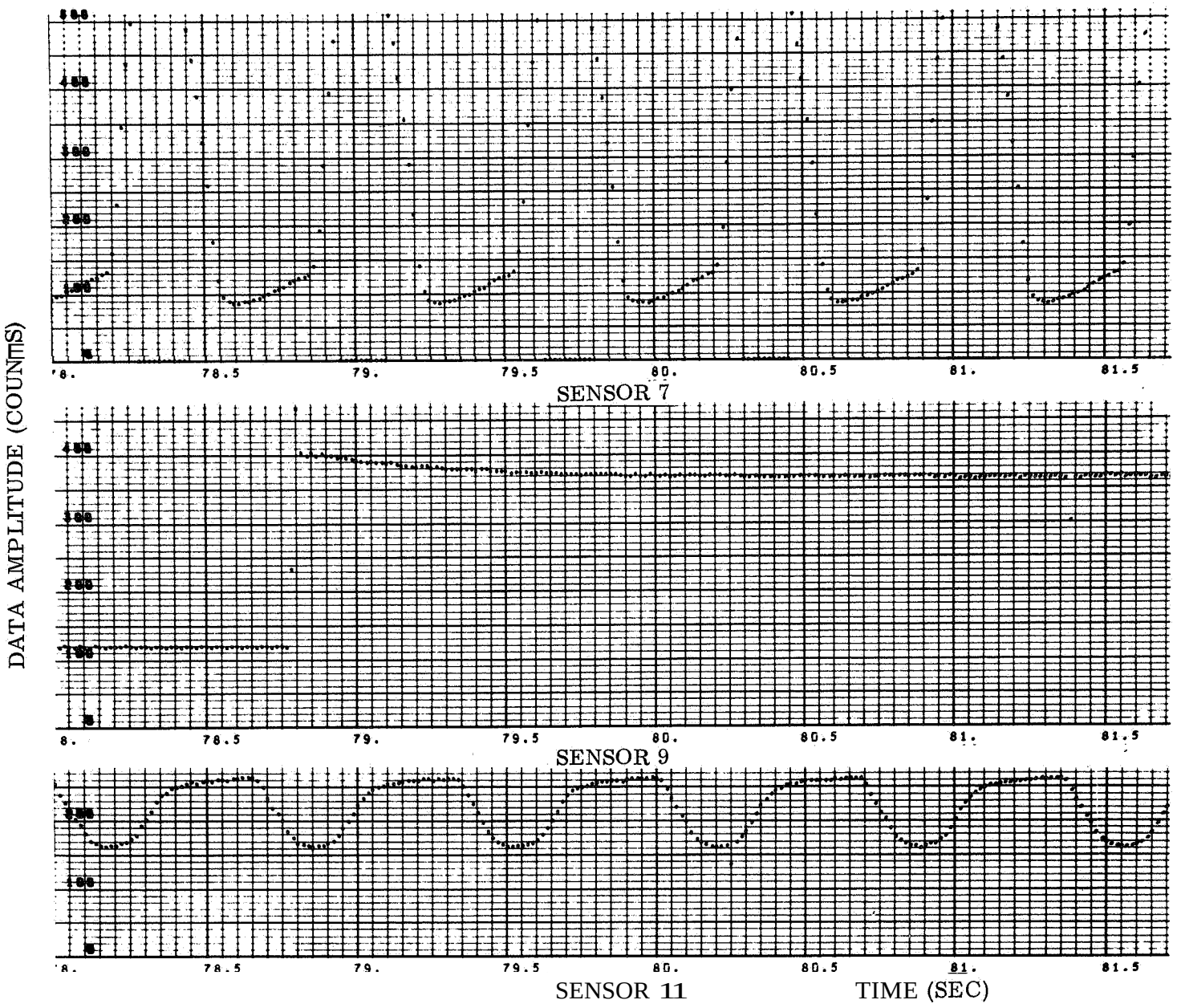


Fig. 2-1 S-6 Data Specimens (Cont'd)

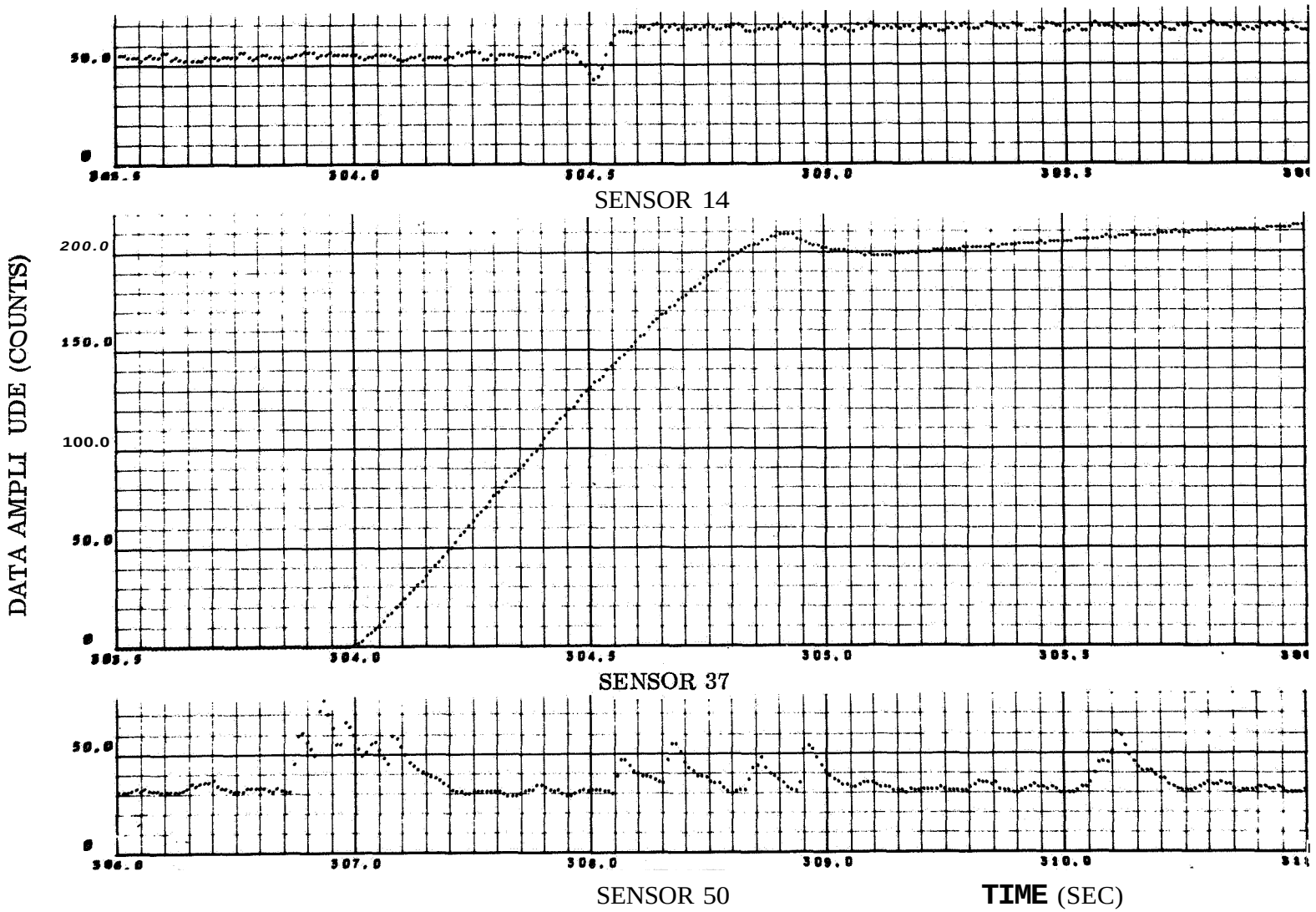


Fig. 2-2 5-49 Data Specimens

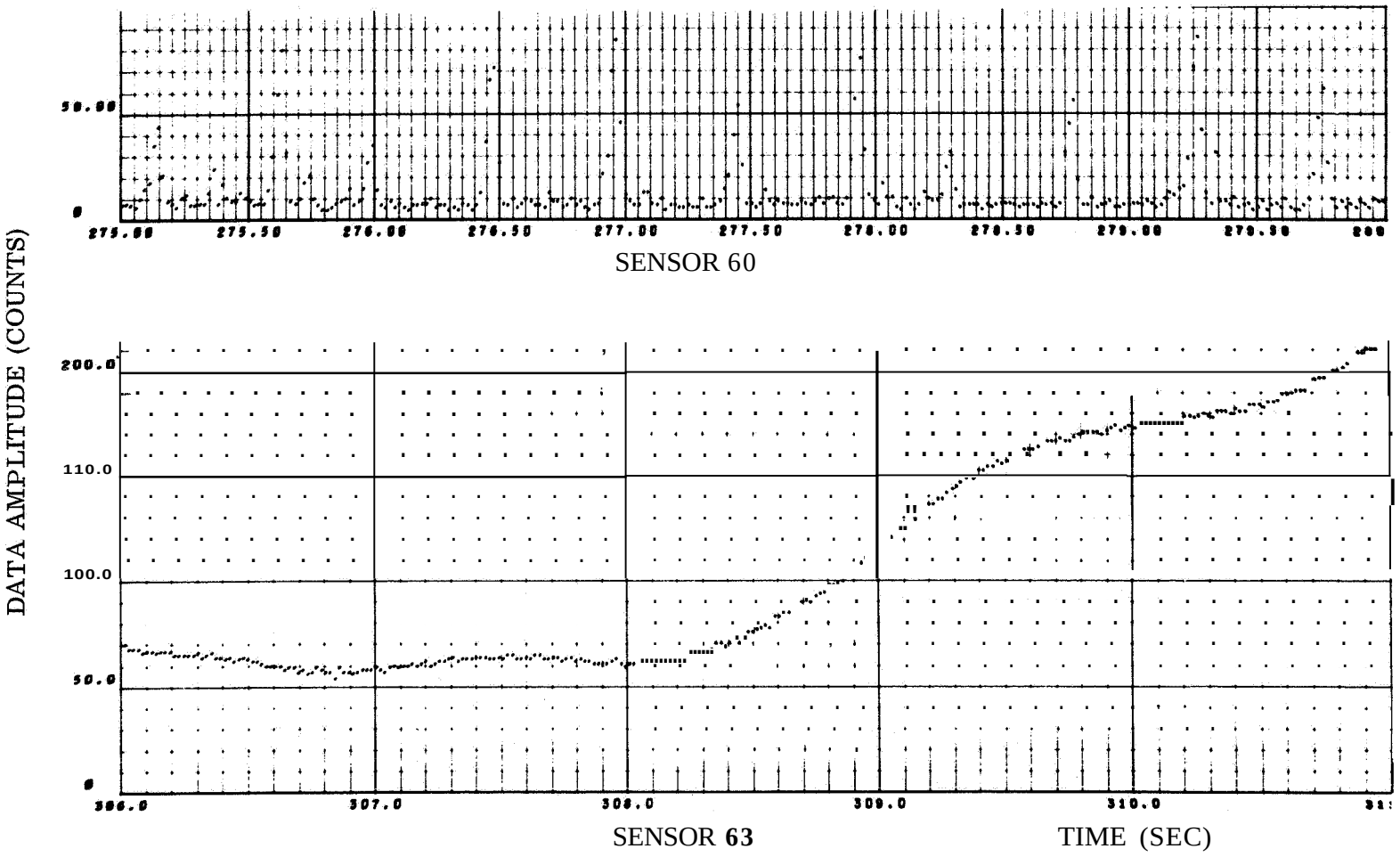


Fig. 2-2 S-49 Data Specimens (Cont'd)

The curves in Figs. 2-3 through 2-12 allow a comparison to be made, among the various selectors, of compression ratio and reconstructed data fidelity. Note that the ordinate scales on these curves vary considerably; this was done to facilitate detailed inspection of all the curves. The data used to plot these curves are given in Tables 2-3 through 2-7.

Combined bit compression ratios over all 10 sensors are shown for 7 of the selectors in Table 2-8. These values were calculated using the compression ratios shown on the curves for an rms error of 0.5%. The number of nonredundant samples corresponding to these compression ratios was summed over the 10 data specimens, and this sum was divided into the sum of the samples processed. (The EXPINT and STFC selectors are not included in Table 2-8 because of insufficient data.)

Table 2-8 indicates that the FOI4DG selector outperforms the remaining selectors on the average, for the chosen value of rms error. The FOI4DG selector is closely followed by FOIDIS and FOIOOT. It is interesting to note that the difference in combined bit compression ratio between the poorest and the best performing model is less than a factor of 2

On an individual-sensor basis, it is evident from Figs. 2-3 through 2-12 that the relative ranking of the FOI4DG, FOIDIS, ZOI, and ZOP selectors, in that order, prevails for almost every sensor, at an rms error equal to 0.5%. [The one notable exception is S-49 Sensor 50 (Fig. 2-5), where the compression ratio achieved by the ZOI selector is abnormally high.]

However, neither the FOIOOT selector nor bit plane encoding behaved in this manner. In general, the FOIOOT selector performed better, compared with the other four polynomial selectors, on quieter data (as shown in Figs. 2-3, 2-4, and 2-7), and relatively poorer on more active data. It therefore appears that the FOIOOT selector (with the chosen values of offset*) is highly sensitive to data activity.

---

*Throughout this study, the offset of the FOIOOT selector was made equal to the tolerance.

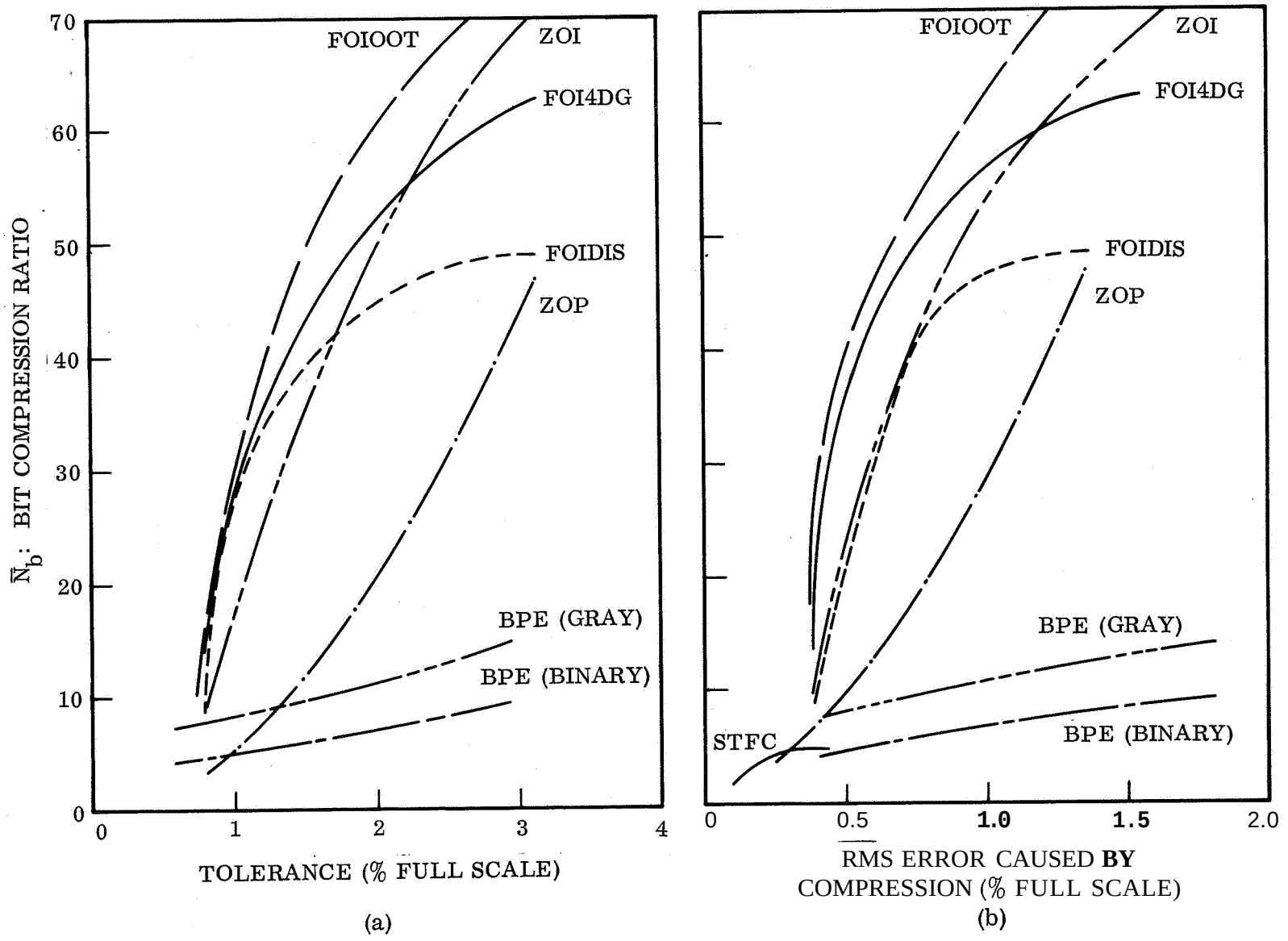


Fig. 2-3 Selector Performance, 5-49 Sensor 14

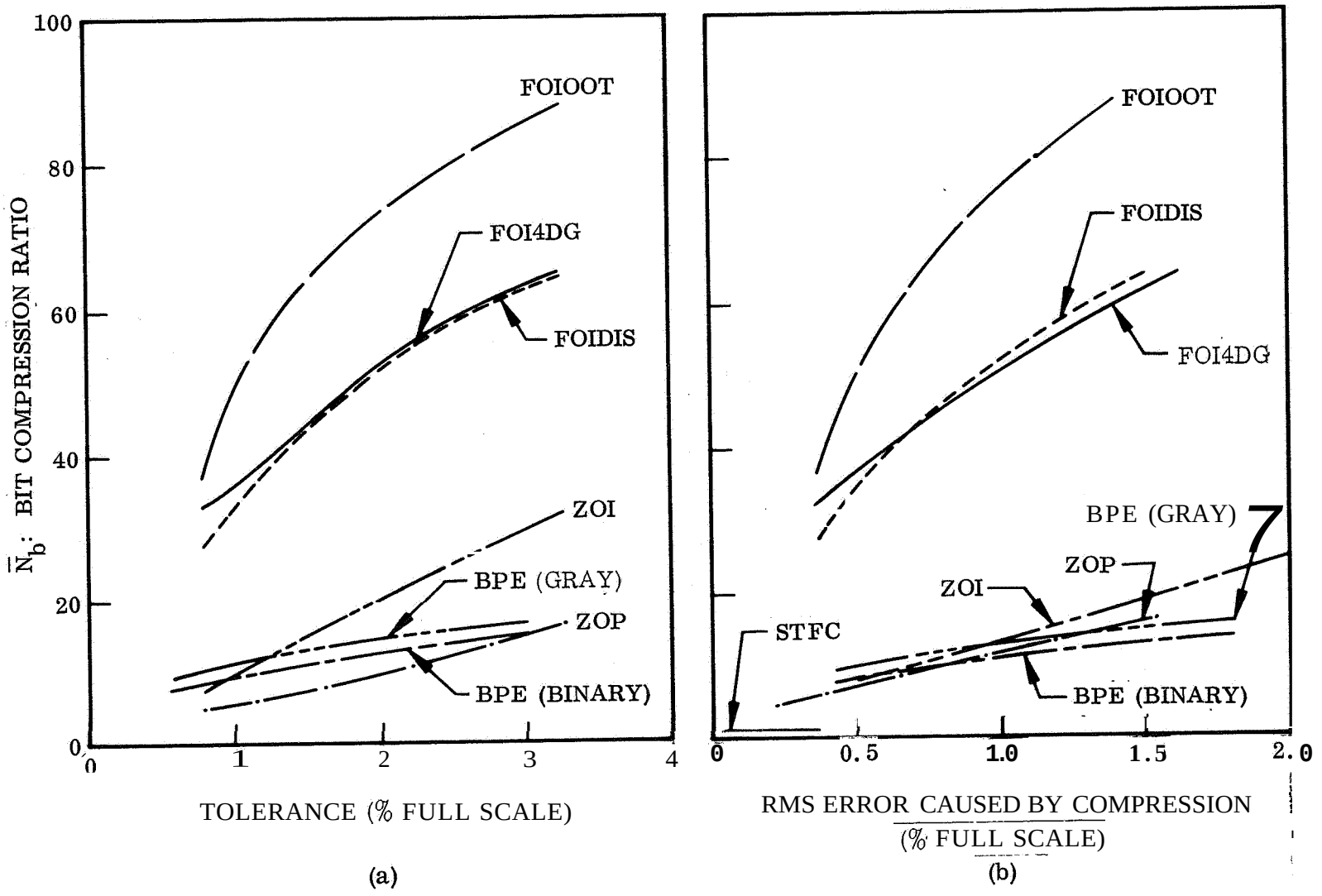
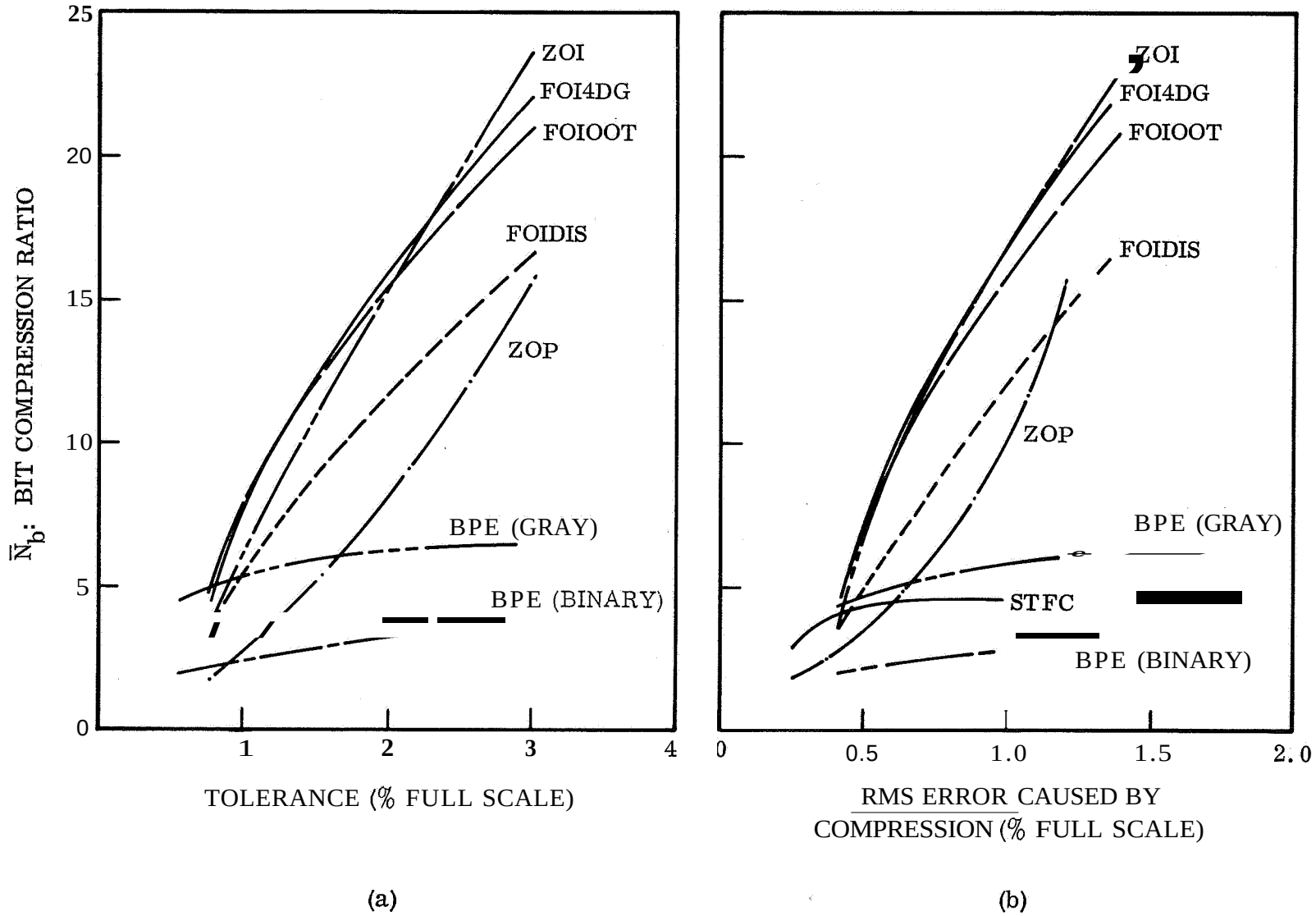
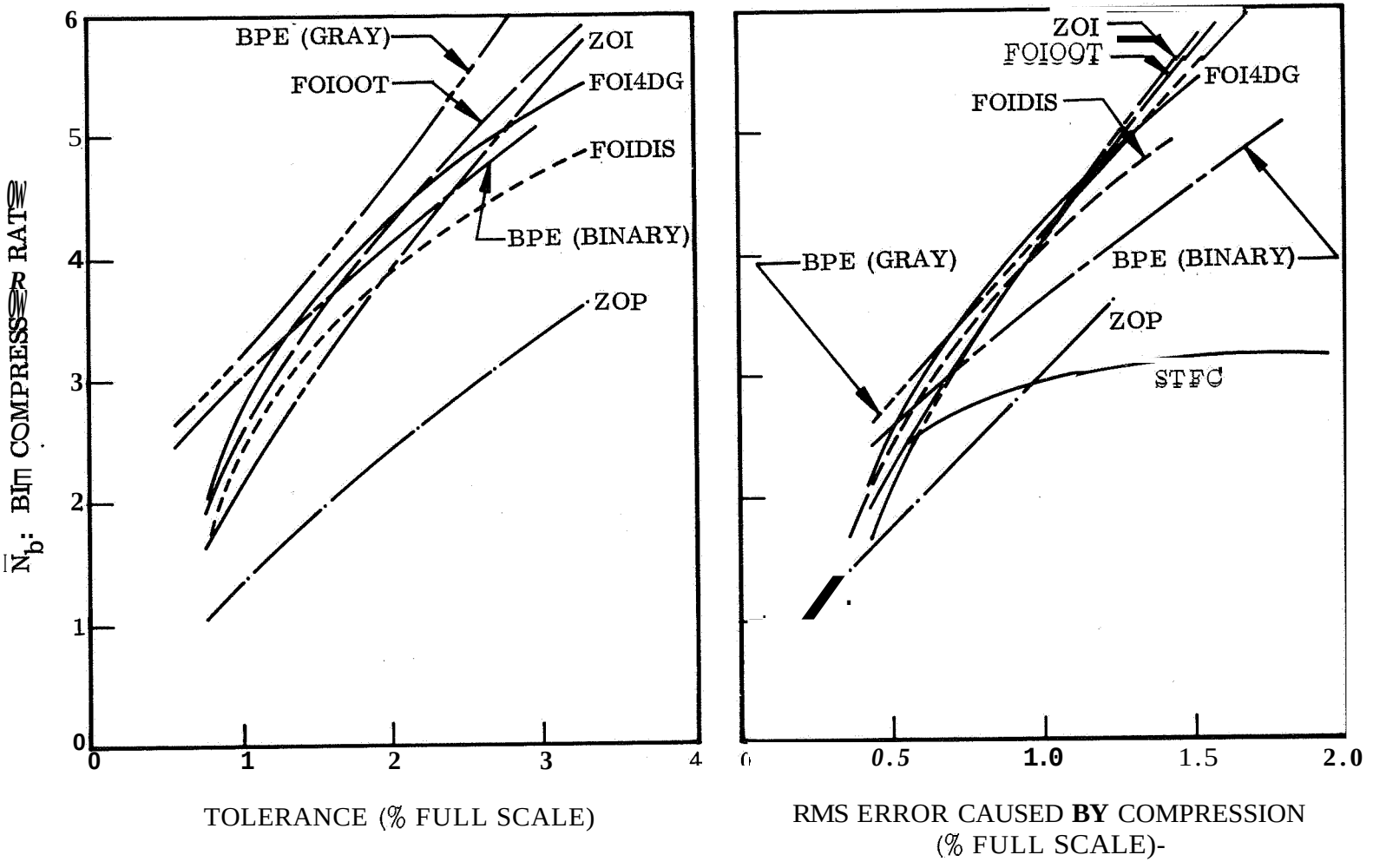


Fig. 2-4 Selector Performance, S-49 Sensor 37

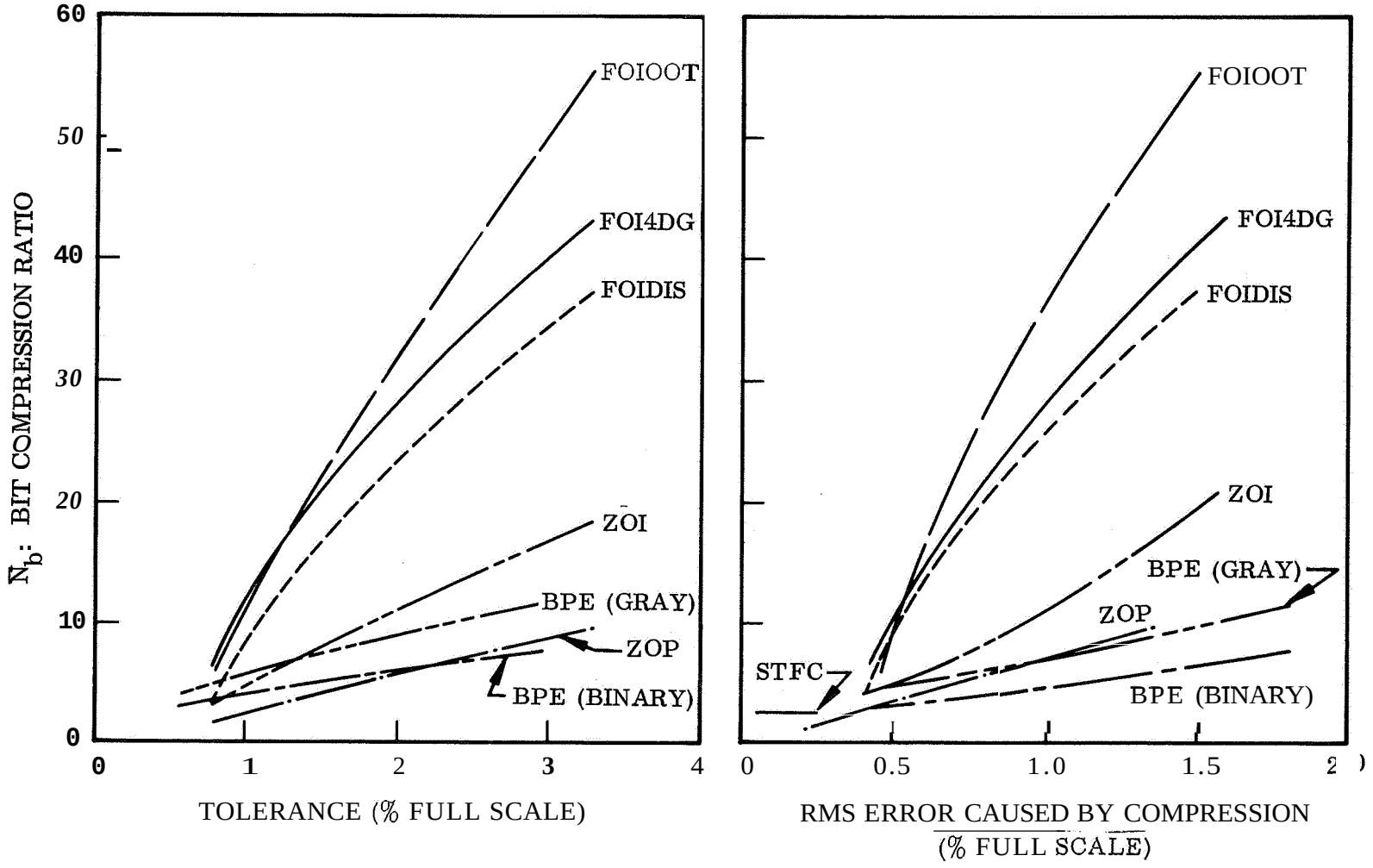


Fig, 2-5 Selector Performance, S-49 Sensor 50



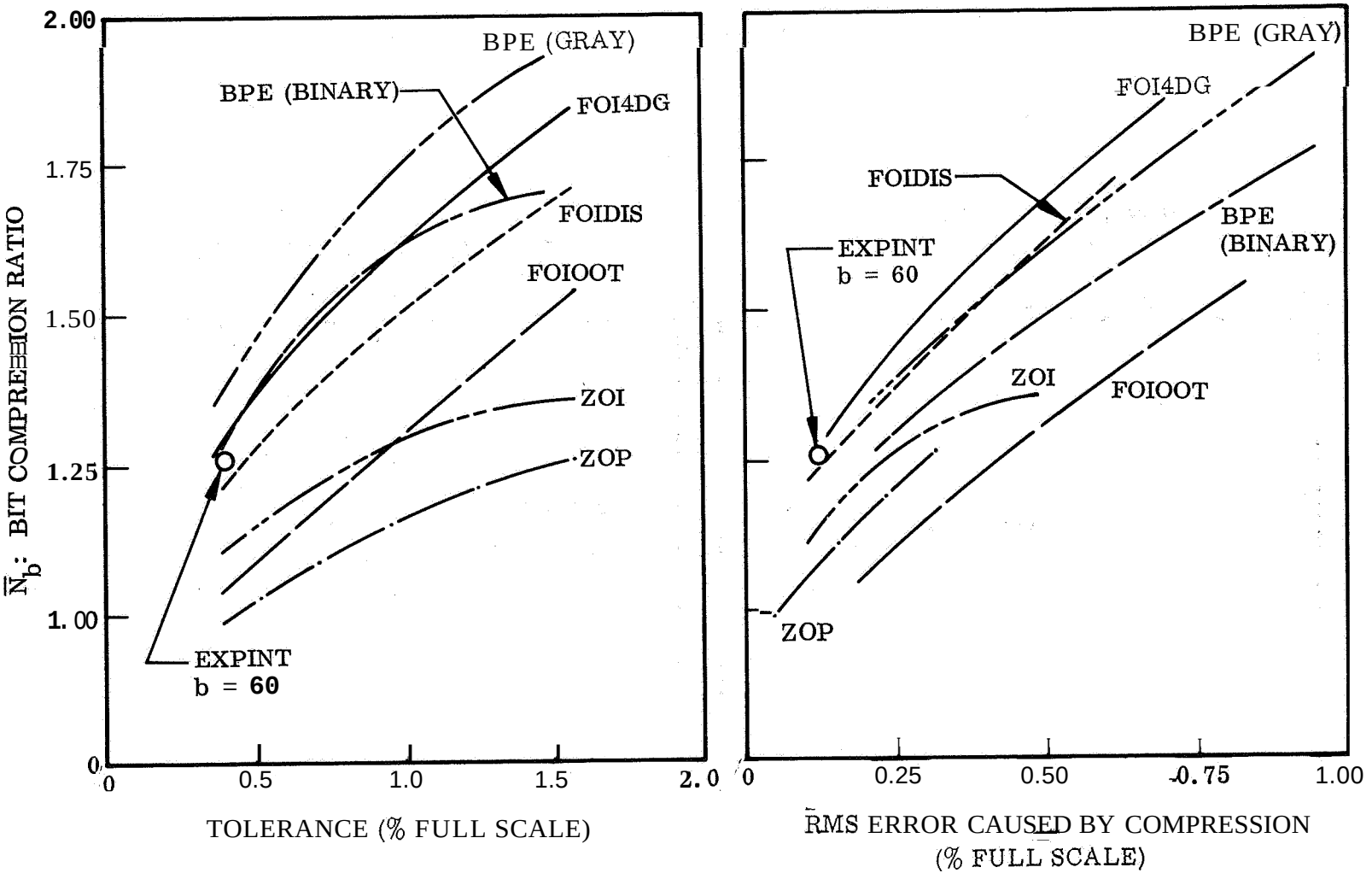
(a) @)

Fig. 2-6 Selector Performance, **S-49** Sensor **60**



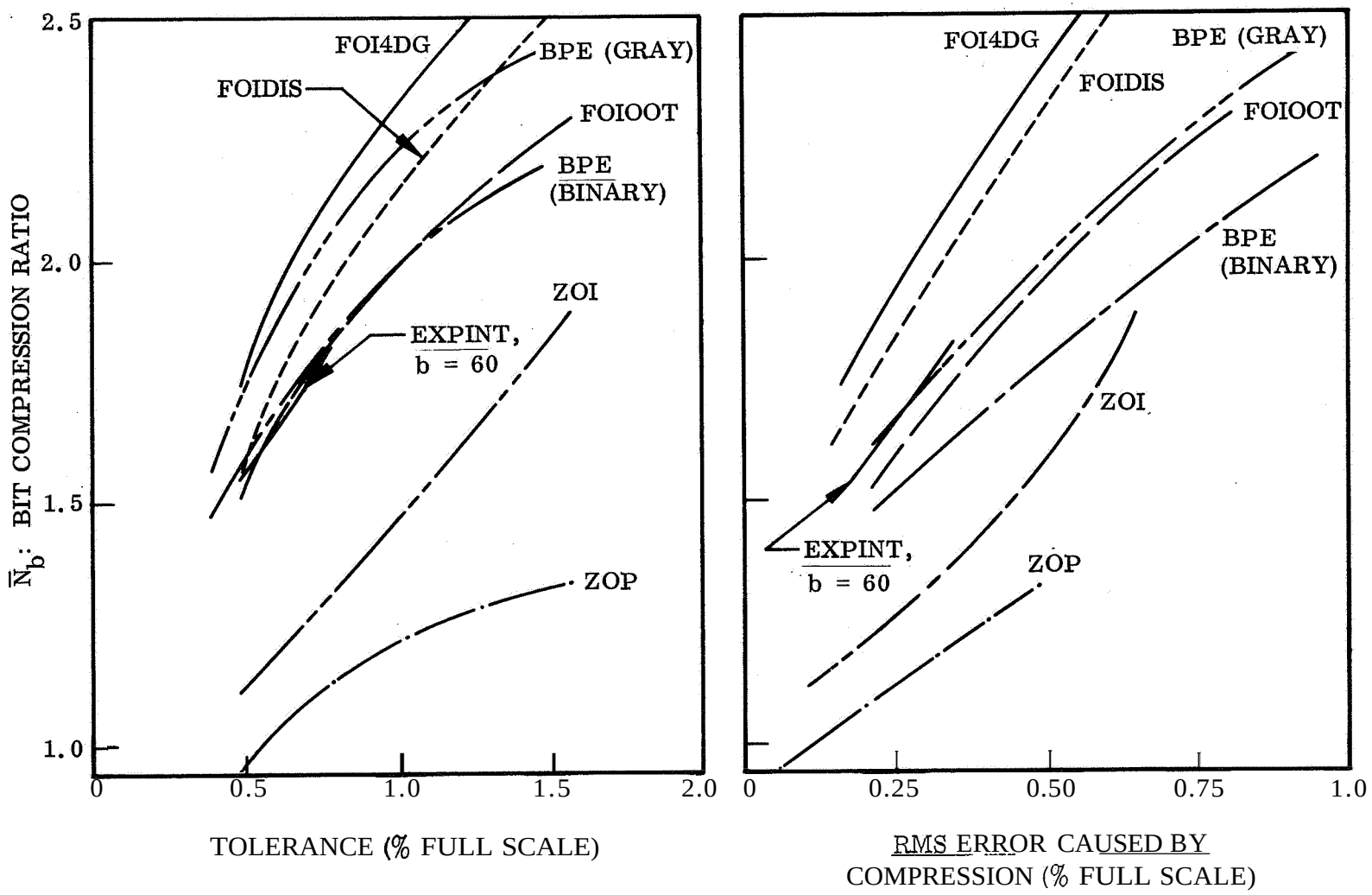
(a) (b)

Fig. 2-7 Selector Performance, **S-49** Sensor *63*



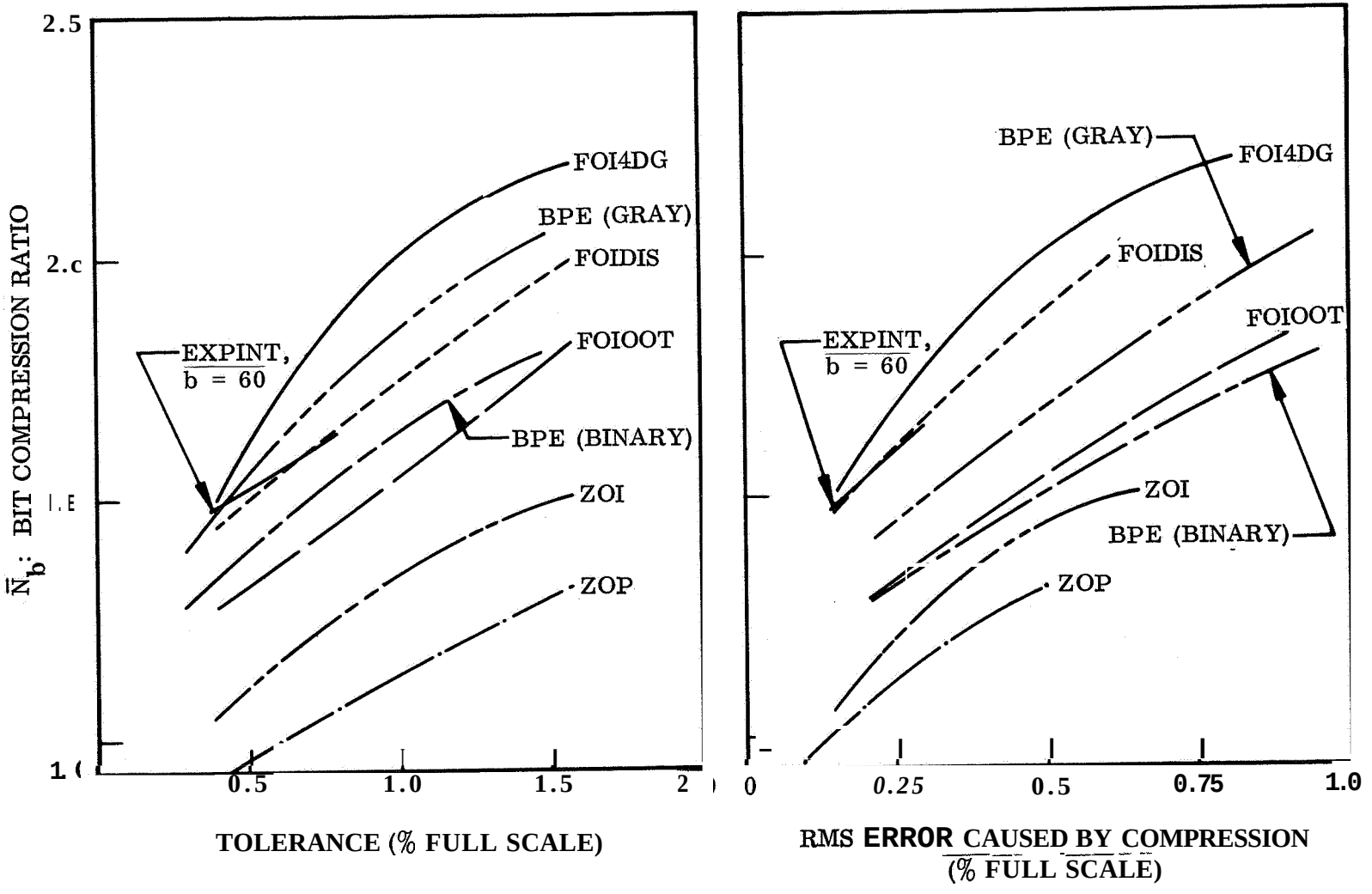
(a) (b)

Fig. 2-8 Selector Performance, *S-6* Sensor 2



(a) (b)

Fig. 2-9 Selector Performance, S-6 Sensor 3



(a) (b)

Fig. 2-10 Selector Performance, S-6 Sensor 7

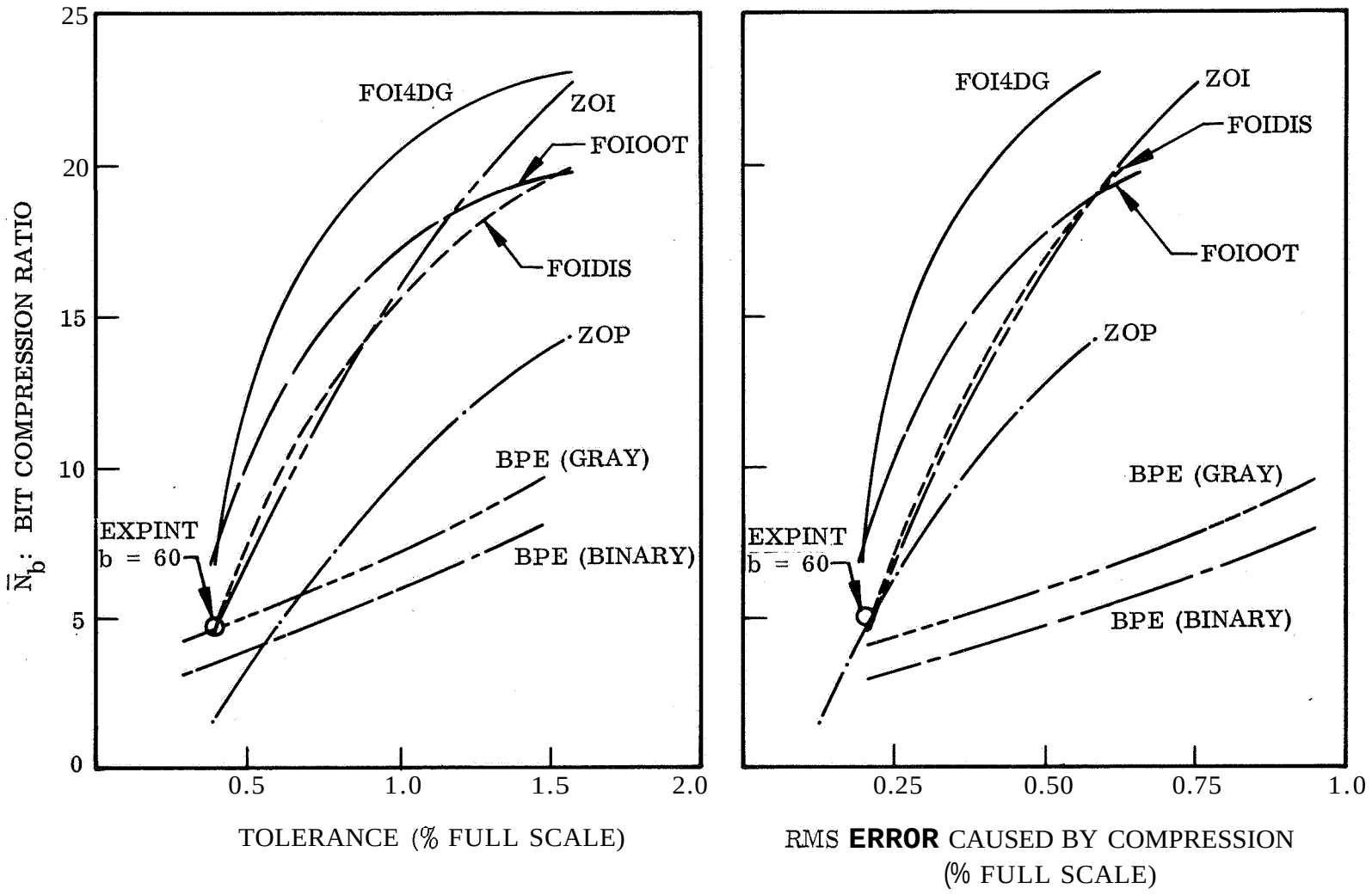


Fig. 2-11 Selector Performance, S-6 Sensor 9

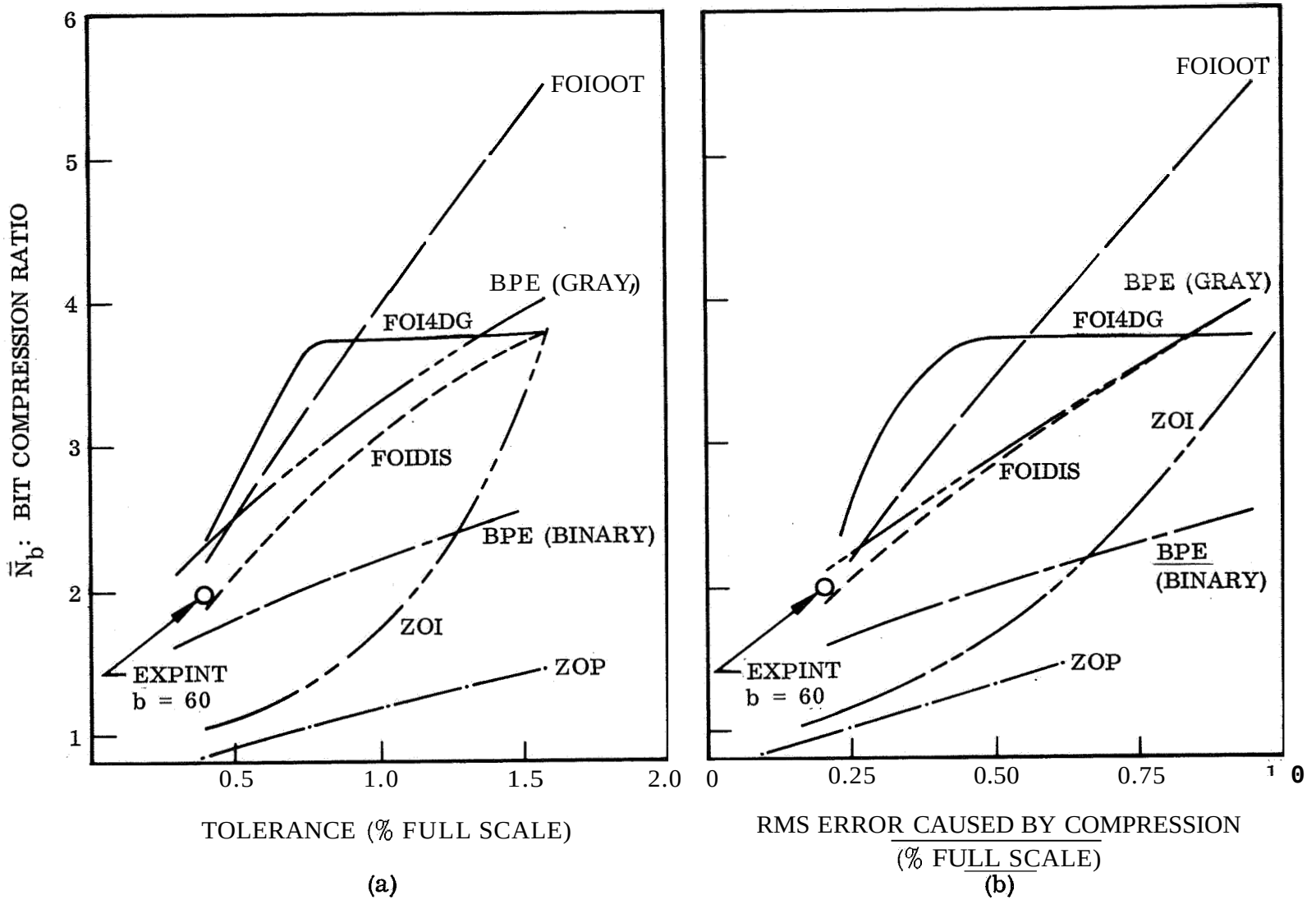


Fig. 2-12 Selector Performance, S-6 Sensor 11

Table 2-3

SELECTOR PERFORMANCE FOR S-6 DATA

Selector	Tol., counts	Sensor 2			Sensor 3			Sensor 7			Sensor 9			Sensor 11			$\bar{N}_b/\bar{N}$
		$\bar{N}$	$\bar{N}_b$	ϵ	$\bar{N}$	$\bar{N}_b$	ϵ	$\bar{N}$	$\bar{N}_b$	ϵ	$\bar{N}$	$\bar{N}_b$	ϵ	$\bar{N}$	$\bar{N}_b$	ϵ	
ZOP	2	1.66	0.99	0.038	1.61	0.96	0.056	1.46	0.88	0.055	2.66	1.60	0.126	1.40	0.84	0.083	
	4	1.86	1.11	0.137	1.89	1.13	0.182	1.76	1.06	0.188	11.94	7.15	0.280	1.79	1.07	0.247	9/15
	8	2.10	1.26	0.315	2.23	1.34	0.442	2.18	1.31	0.480	23.88	14.30	0.570	2.38	1.43	0.622	
ZOI	2	1.85	1.11	0.097	1.84	1.10	0.119	1.73	1.04	0.131	7.80	4.67	0.190	1.75	1.05	0.168	
	4	2.06	1.24	0.221	2.20	1.32	0.290	2.12	1.27	0.300	20.53	12.30	0.327	2.29	1.38	0.178	9/15
	8	2.27	1.36	0.450	3.13	1.89	0.648	2.52	1.51	0.654	37.75	22.70	0.740	4.50	3.70	0.983	
FOIDIS	2	1.62	1.22	0.100	2.14	1.61	0.144	1.94	1.46	0.145	6.04	4.55	0.194	2.50	1.88	0.198	
	4	1.90	1.42	0.278	2.60	1.95	0.310	2.21	1.66	0.310	17.21	12.90	0.372	3.64	2.72	0.418	18/24
	8	2.29	1.72	0.610	3.36	2.54	0.630	2.65	1.99	0.620	26.60	19.95	0.616	5.00	3.75	0.860	
FOI4DG	2	1.71	1.28	0.137	2.30	1.73	0.164	2.00	1.50	0.160							
	4	2.02	1.52	0.327	2.84	2.14	0.350	2.49	1.88	0.380							
	8	2.46	1.85	0.692	3.62	2.72	0.709	2.92	2.20	0.820							
FOIOOT	2	1.85	1.04	0.176	2.70	1.52	0.206	2.27	1.28	0.202	12.32	6.91	0.204	3.92	2.20	0.241	
	4	2.15	1.21	0.376	3.28	2.29	0.821	2.25	1.83	0.920	26.60	15.00	0.370	5.94	3.34	0.464	9/16
	8	2.73	1.54	0.828	4.07	2.29	0.821	3.25	1.83	0.920	35.46	19.80	0.640	9.76	5.48	0.960	

Explanation of symbols:

$\bar{N}$: Sample compression ratio

$\bar{N}_b$: Bit compression ratio

ϵ: Rms error in % full scale (100% = 511 counts)

Table 2-4

SELECTOR PERFORMANCE FOR S-49 DATA

Selector	Tol., counts	Sensor 14			Sensor 37			Sensor 50			Sensor 60			Sensor 63			$\bar{N}_b/\bar{N}$
		$\bar{N}$	$\bar{N}_b$	ϵ	$\bar{N}$	$\bar{N}_b$	ϵ	$\bar{N}$	$\bar{N}_b$	ϵ	$\bar{N}$	$\bar{N}_b$	ϵ	$\bar{N}$	$\bar{N}_b$	ϵ	
ZOP	2	5.45	3.27	0.275	7.84	4.71	0.235	3.12	1.87	0.255	1.77	1.06	0.204	2.72	1.63	0.240	
	4	22.08	13.20	0.629	13.50	8.09	0.660	8.67	5.19	0.668	3.32	1.99	0.576	7.31	4.38	0.604	9/15
	8	78.00	46.80	1.360	26.99	16.20	1.520	26.00	15.60	1.210	6.10	3.66	1.255	15.94	9.54	1.340	
ZOI	2	15.20	9.21	0.376	13.24	7.98	0.520	6.11	3.66	0.413	2.78	1.67	0.380	5.40	3.24	0.395	
	4	61.58	36.90	0.680	25.51	15.30	1.190	18.48	11.08	0.730	5.32	3.19	0.750	13.75	8.25	0.795	9/15
	8	117.00	70.20	1.670	51.97	31.20	2.530	39.00	23.40	1.435	9.71	5.82	1.560	30.48	18.30	1.590	
FOIDIS	2	11.70	8.78	0.396	36.92	27.65	0.356	4.81	3.60	0.410	2.24	1.64	0.372	5.01	3.76	0.414	
	4	53.18	40.00	0.733	58.46	43.80	0.740	12.10	9.04	0.780	4.50	3.38	0.745	23.37	17.55	0.728	18/24
	8	65.00	48.75	1.380	87.69	65.50	1.550	21.94	16.45	1.370	6.50	4.88	1.430	50.08	37.56	1.505	
FOI4DG	2	17.73	13.30	0.378	43.84	32.80	0.372	5.95	4.47	0.453	2.75	2.06	0.445	8.16	6.11	0.445	
	4	58.50	43.90	0.610	58.46	43.80	0.760	16.72	12.52	0.756	5.16	3.76	0.835	29.21	21.90	0.805	18/24
	8	83.58	62.50	1.520	87.69	65.50	1.630	29.25	21.90	1.360	7.25	5.42	1.540	58.42	43.80	1.620	
FOIOOT	2	28.54	16.05	0.386	66.81	37.60	0.342	8.26	4.63	0.459	3.44	1.93	0.463	11.50	6.46	0.448	
	4	90.00	50.60	0.700	116.92	65.60	0.750	21.94	12.32	0.752	6.46	3.63	0.836	41.24	23.20	0.710	9/16
	8	130.00	73.10	1.340	155.89	87.60	1.400	36.95	20.80	1.370	10.53	5.91	1.585	100.15	56.20	1.500	

Explanation of symbols:

$\bar{N}$: Sample compression ratio

$\bar{N}_b$: Bit compression ratio

ϵ: Rms error in % full scale (100% = 255 counts)

Table 2-5

BIT PLANE ENCODING PERFORMANCE FOR S-6 AND S-49 DATA

S-6					S-49				
Sensor	K	K_f	$\bar{N}_b$: Bin	$\bar{N}_b$: Gray	Sensor	K	K_f	$\bar{N}_b$: Bin	$\bar{N}_b$: Gray
2	1/2	1	1.114		14	1/2	1	2.894	
	3/2	2	1.271	1.354		3/2	2	4.197	7.413
	7/2	3	1.480	1.594		7/2	3	6.217	9.921
	15/2	4	1.772	1.93		15/2	4	9.243	14.337
3	1/2	1	1.276		37	1/2	1	5.609	
	3/2	2	1.483	1.617		3/2	2	7.49	9.399
	7/2	3	1.776	1.971		7/2	3	9.956	12.483
	15/2	4	2.213	2.459		15/2	4	13.576	15.937
7	1/2	1	1.124		50	1/2	1	1.68	
	3/2	2	1.284	1.41		3/2	2	2.093	4.528
	7/2	3	1.498	1.672		7/2	3	2.714	5.83
	15/2	4	1.797	2.04		15/2	4	3.669	6.372
9	1/2	1	2.291		60	1/2	1	1.939	
	3/2	2	3.072	4.248		3/2	2	2.472	2.693
	7/2	3	4.665	5.825		7/2	3	3.408	3.71
	15/2	4	8.073	9.685		15/2	4	5.189	6.312
11	1/2	1	1.373		63	1/2	1	2.257	
	3/2	2	1.62	2.141		3/2	2	3.013	4.344
	7/2	3	1.976	2.81		7/2	3	4.528	6.792
	15/2	4	2.532	4.085		15/2	4	7.805	11.566

Explanation of symbols:

K : Tolerance in counts (full-scale = 511 counts for S-6 and 255 counts for S-49)

K_f: Number of Class-C planes

$\bar{N}_b$: Bandwidth compression ratio

Table 2-6

PERFORMANCE OF EXPONENTIAL INTERPOLATOR ON S-6 DATA

			Sensor 2			Sensor 3			Sensor 7			Sensor 9			Sensor 11		
	b_1	b_2	$\bar{N}$	$\bar{N}_b$	ϵ	$\bar{N}$	$\bar{N}_b$	ϵ	$\bar{N}$	$\bar{N}_b$	ϵ	$\bar{N}$	$\bar{N}_b$	ϵ	$\bar{N}$	$\bar{N}_b$	ϵ
T	20	20	1.70	1.27	0.109	1.88	1.41	0.144	1.86	1.40	0.141	6.96	5.22	0.194	2.46	1.84	0.198
O	40	40	1.70	1.28	0.122	1.97	1.48	0.150	1.90	1.42	0.152	7.05	5.28	0.197	2.54	1.91	0.217
L	60	60	1.66	1.24	0.117	2.04	1.54	0.159	1.98	1.48	0.156	6.96	5.22	0.200	2.62	1.96	0.216
0.392%	80	80	1.70	1.27	0.124	2.10	1.58	0.165	1.95	1.46	0.154	6.58	4.94	0.204	2.54	1.90	0.219
(2 counts)	100	100	1.70	1.28	1.250	2.14	1.60	0.163	1.95	1.46	0.157	6.58	4.94	0.200	2.62	1.96	0.216
T	20	20				2.22	1.62	0.309	2.05	1.54	0.294						
O	40	40				2.40	1.80	0.341	2.18	1.64	0.268						
L	60	60				2.44	1.83	0.331	2.20	1.64	0.298						
0.784%	80	80				2.47	1.86	0.326	2.32	1.74	0.305						
(4 counts)	100	100				2.54	1.90	0.326	2.48	1.86	0.323						

Explanation of symbols:

b_1, b_2: Constants [cf. **Eq.** *(2.1)*]

$\bar{N}$: Sample compression ratio

$\bar{N}_b$: Bit compression ratio ($\bar{N}_b/\bar{N}$ = 18/24)

ϵ: Rms error, % full scale (100% = 511 counts)

Table 2-7

PERFORMANCE OF SHORT-TERM FREQUENCY COMPONENT SELECTOR

		Sensor 2		Sensor 3		Sensor 7		Sensor 9		Sensor 11	
$N^{(a)}$	Thresh	$\bar{N}_b$	$\epsilon^{(b)}$	$\bar{N}_b$	ϵ	$\bar{N}_b/N$	ϵ	$\bar{N}_b$	ϵ	$\bar{N}_b/N$	ϵ
7	0.50	3.39	22.8	4.02	12.4	3.84/7 4.46/10	21.0 22.6	3.95	2.2	4.2 /7 3.33/5	0.953 0.800
7	0.667	2.56	21.0	3.64	8.2	2.96/7 3.53/10	20.7 10.8	2.26	1.79	2.37/7 3.33/5	0.509 0.562
7	0.75	2.27	13.5	3.36	5.9	2.67/7	14.0	2.52	1.22	2.53/7 3.33/5	0.475 0.460
7	0.85	1.78	7.2	2.40	3.2	2.94/7	4.6	2.20	.785	2.3 /7	0.081

S-49 Data:		Sensor 14		Sensor 37		Sensor 50		Sensor 60		Senso. 63	
$N^{(a)}$	Thresh	$\bar{N}_b$	ϵ	$\bar{N}_b$	ϵ	$\bar{N}_b$	ϵ	$\bar{N}_b$	ϵ	$\bar{N}_b$	ϵ
7	0.50	4.70	0.456	1.08	0.360	4.70	0.988	3.17	1.94	2.58	0.173 0.173
7	0.667	4.65	0.296	1.08	0.160	4.65	0.722	3.18	1.54	2.68	0.160
7	0.75	4.64	0.238	1.08	0.132	4.58	0.554	2.98	1.0	2.77	0.142
7	0.85	1.43	0.106	1.01	0.0512	2.88	0.257	2.47	0.561	2.57	0.103

(a) 2N+1 = Number of data samples upon which short-term power spectrum is based.
(b) ϵ = rms error, % of full scale.

Table 2-8

COMBINED BIT COMPRESSION RATIOS, $\bar{N}_b$, FOR RMS ERROR OF 0.5% FULL SCALE

Selector	Combined $\bar{N}_b$
FOI4DG	3.97
FOIDIS	3.60
FOIOOT	3.30
BPE-GRAY	3.07
ZOI	2.66
BPE-BINARY	2.46
ZOP	2.24

Bit plane encoding, on the other hand, appears to be relatively insensitive to data activity. Both binary and Gray-coded BPE performed comparatively better on the sensors that exhibited a generally low compression ratio, and vice versa.

Figures 2-3 through 2-12 also show that the slopes of the $\bar{N}_b$-versus-tolerance curves for BPE are generally lower than those for the polynomial selectors. As the tolerance decreases, the curves tend to cross; thus, for very small tolerances, BPE (at least, Gray-coded BPE) exceeds the polynomial selectors in compression performance. This result (which is also shown in Figs. 6-2 and 6-3 of Ref. 4) suggests that BPE may outperform the polynomial selectors when operating on digital sensors. To reproduce digital sensors exactly, it is necessary to set the tolerance so low that all bit changes between words are noted. Thus, very low compression ratios are often obtained with the other redundancy removal techniques. The above observations show that BPE may be expected to perform better in these instances.

The use of Gray coding increased the compression ratio over that observed with binary coding. This was expected, since Gray coded data words change in only one bit position at a time as the word changes from one count to the next, while binary-coded data words may have many bit changes.

The compression performance of the EXPINT selector proved to comparable to that of the FOIDIS selector. For the run parameters chosen, it never exceeded the performance of the FOI4DG selector, even when operating on Class 3 (exponential) data (Ref. 4, Fig. 3-1). The STFC selector performed poorly for practically all the data specimens used in the tests.

It should be noted that only compression rate and rms error are considered in this section. Other factors such as complexity, ease of reconstruction, and sensitivity to noise in the compressed transmission link, are considered in Section 5

2.3 BROADSIDE DATA COMPRESSION EVALUATION

As noted in Par. 2.1.3, broadside compression is not a sample selector, although it includes s ection. It is a two-step process which transforms the data into a form that can more efficiently compressed using sample selectors. Because of its special nature, broadside compression is evaluated separately from the other compression techniques.

To test the performance of broadside data compression under conditions of nonsynchronous multiplexer operation, a computer program was written to process data from any sensor (in the aspect sensor) and to sort these data samples into groups, each containing samples from corresponding portions of consecutive waveform periods. Using this program, the samples from each of these groups were written onto a separate file on magnetic tape. This tape was then processed as if each group (pseudo sensor) constituted data from individual sensor, using the conventional redundancy removal techniques.

The telemetry data sensor that was used for the study of broadside data compression was S-6 Sensor 7, since it has highly periodic data, and also because compression performance results using this sensor with other models are available for comparison purposes. Table 2-9 presents the compression ratios obtained using the broadside technique in conjunction with various selectors along with the results obtained using those selectors alone.

Table 2-9

BROADSIDE PERFORMANCE RESULTS

Selector	Compression Ratio	
	Broadside	Non-broadside
ZOI	2.38	2.12
FOIDIS	3.65	2.29
FOI4DG	4.35	2.49

Table 2-9 shows improvement obtained by using the broadside technique. Further study and research could probably develop much better broadside techniques. One basic problem with the present system is that only slight improvement in compression ratio is obtained, because most periodic waveforms do not repeat in an integral number of samples. (The multiplexer is not synchronized to the waveform variations.) This causes each of the pseudo-sensors to form the same waveform as the original data, but with a large amount of aliasing error introduced. If the original waveform contained an integral number of samples, then each pseudo-sensor would have 100-percent aliasing, and much higher compression ratios would be obtained using the present method of broadside data compression. The waveform used for broadside compression is seen in Fig. 2-1. Figure 2-13 shows the original data, and Fig. 2-14 shows the data represented by a pseudo-sensor. Both figures have a compressed time scale so that comparable data points on the sensor and pseudo-sensor could be observed. As the period of the pseudo-sensor approaches ∞, it is seen that the broadside methods presented here would become most effective.

Broadside data compression holds high promise for two reasons: (1) it provides a method of extracting redundancy from periodic waveforms, and (2) it produces a reconstructed waveform that is more natural to the data analyst than the "choppy" waveform obtained by normal non-broadside compression techniques. For these two reasons, it is believed that effort should be extended to develop broadside techniques which either produce more aliasing error in each of the pseudo-sensors or effectively cope with the problem of nonintegral periods.

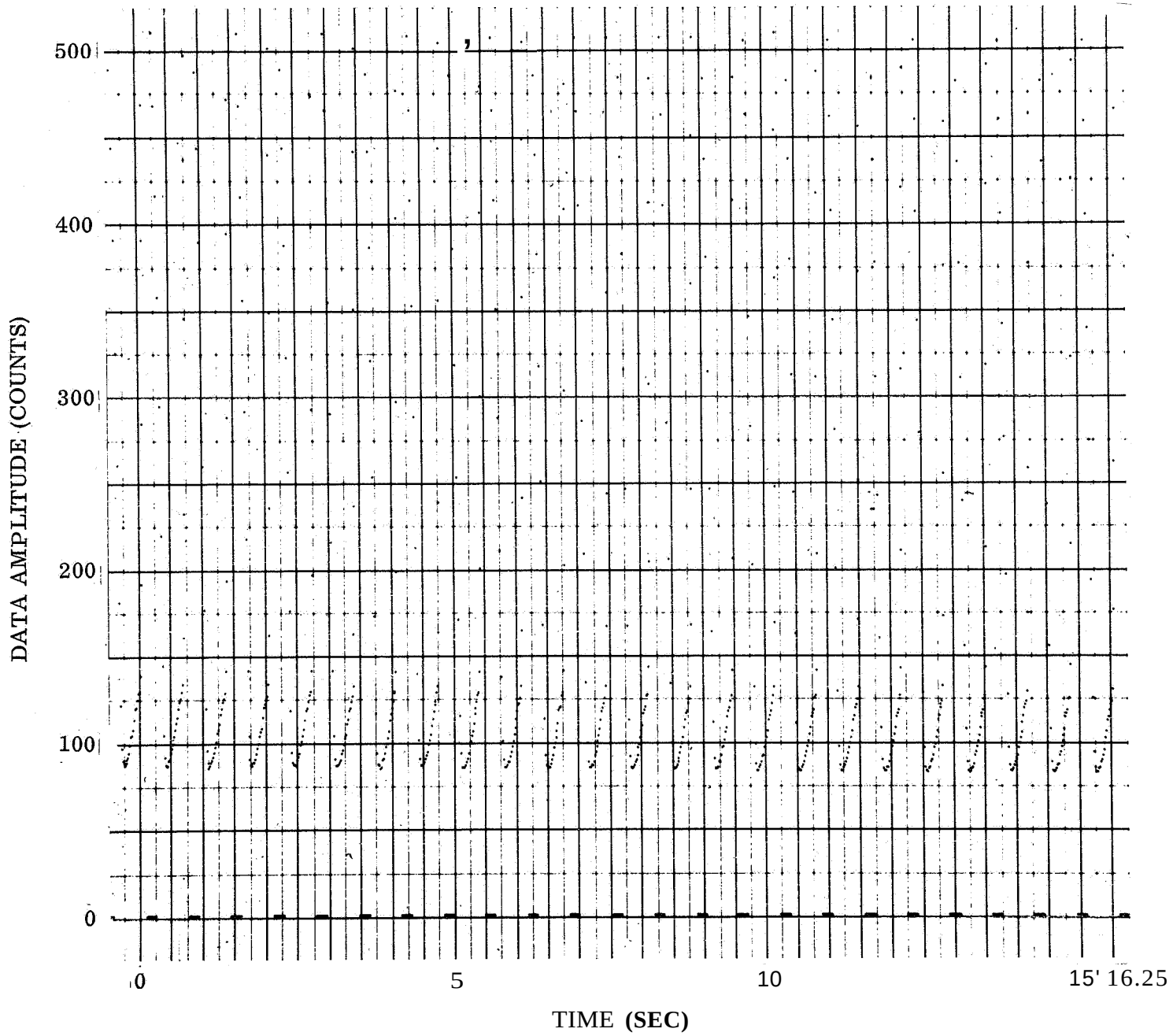


Fig. 2-13 S-6 Sensor 7 Original Data

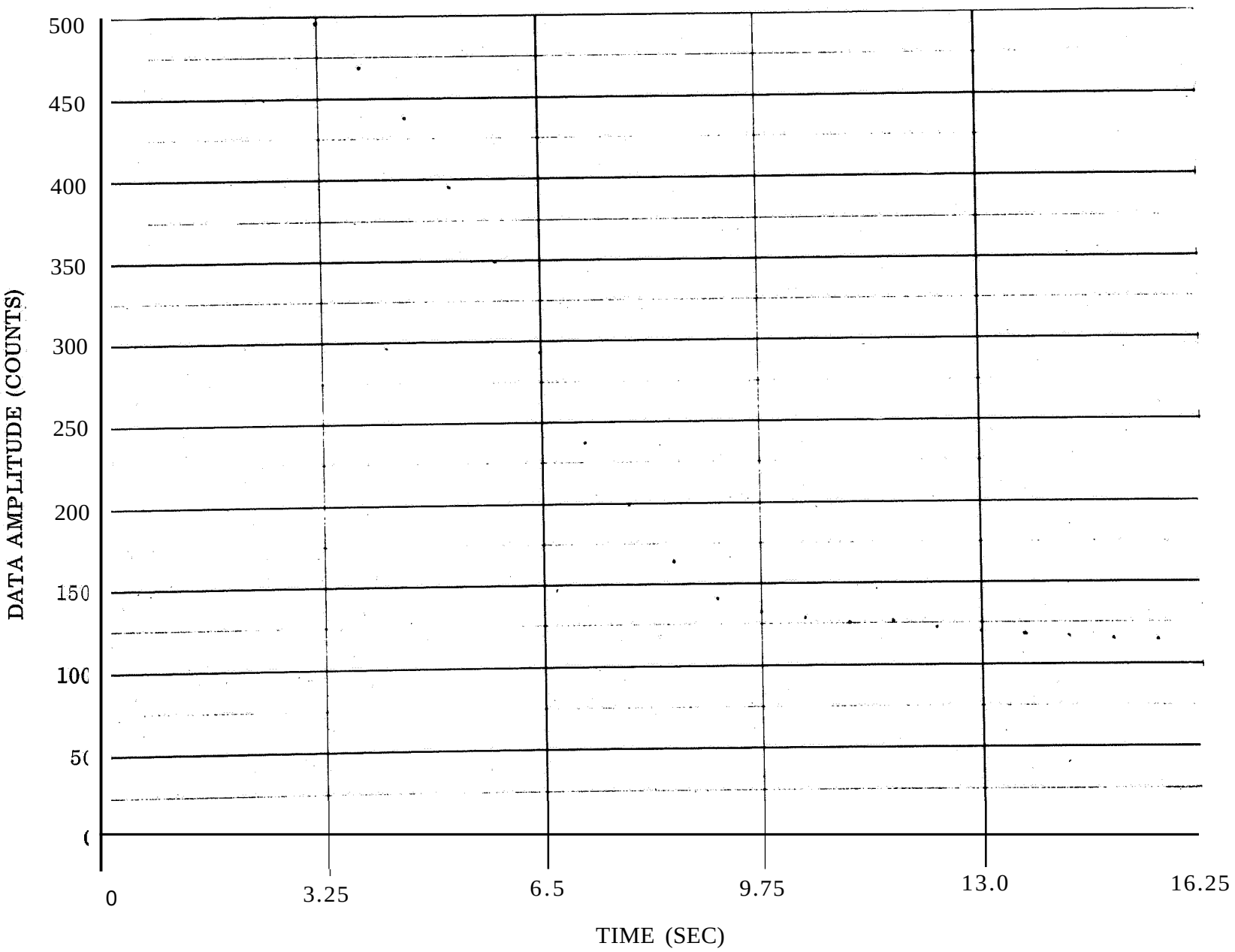


Fig. 2-14 **S-6** Sensor 7 Pseudo-Sensor Data

One method for making the period of a pseudo-sensor equal ∞ would be to filter out the periodic waveform variations. However, if this were done it would be impossible to reconstruct the original data waveform.

2.4 SECONDARY DATA COMPRESSION

The investigation of secondary data compression centered mainly on the use of higher order polynomial compression algorithms on data which were previously compressed with lower order selectors. Some attention was also given to the use of bit plane encoding as a secondary compression technique. Both m[illegible] are [illegible]c se[illegible] in the following paragraphs.

It is interesting to note that any redundancy reduction algorithm can be regarded as a secondary data compressor if some form of parameter extraction is employed ahead of the compressor. (That is, specific information about the waveform is extracted from it, and this information is sent on to the compressor.) Regarded in this light, a study of secondary data compression involves an examination of methods for combining t se [illegible] techniques efficiently. While it is believed that a substantial payoff in data compression lies in this area, this study was not attempted because of its extended scope

2.4.1 Bit Plane Encoding

Bit plane encoding is one possible method of secondary data compression. To employ this method, there are two ways that the compressed data samples could be arranged [illegible]r encoding:

- The combined data magnitude [illegible] channel ID word[illegible] from all the sensors c[illegible] be mixed together in a single BPE word block as they are received from the primary compressor. This method does not appear promising, because the statistical independence of the sample/ID words in the block is likely to preclude efficient encoding. Thus, data "expansion," rather than compression, could result from the sec[illegible] compression attempt.

- There could be a separate BPE word block for each sensor As each block filled, it would be encoded and the resulting bit stream sent to a buffer, along with the sensor ID for that block. Each data amplitude word in the block must be accompanied by a time word, since without it the original time base would be lost. The time word would be encoded along with the amplitude word This method would require a large random-access storage capability and would probably increase the time delay between data measurement and reconstruction. In addition, the necessary inclusion of time information with each data amplitude word, even though it too can be encoded, may offset whatever additional compression bit plane encoding could accomplish. It is estimated that this method of BPE could gain no more than an additional factor of 2 in average compression ratio; the advisability of pursuing it any further as a study item is thus questionable.

2 2 Use of Higher-Order Compression on Previously Compressed Data That Have Been Reconstructed

In studying this method of secondary data compression, one important fact must be kept in mind: any first-order disjoined selector has the same capability as a comparable zero-order selector. Therefore, the use of two compression algorithms in tandem in an effort to increase compression efficiency is not feasible. This is demonstrated by the following graphical analysis. Figure 2-15 shows a portion of a typical sampled data waveform, and Fig. 2-16 shows the data compressed and reconstructed, using the FOIDIS selector with a tolerance of two decimal counts. The compression ratio is 8-1/3 and the rms error is 1.145 counts. Figures 2-17 and 2-18 show successive compression operations, first with ZOI and a tolerance of 2 counts and then with FOIDIS and a tolerance of 2. The overall compression ratio is still 8-1/3, but the rms error is now 1.5 counts. It was thought that this large rms error was due to the fact that a tolerance of 2 was used for each compression, thus causing the total peak error to equal 4 counts. Therefore, the data were again compressed using successive selectors but with a tolerance of 1 count. This makes the overall peak error 2 counts, which equals that obtained when the data are compressed with FOIDIS only. Figures 2-19 and 2-20 show these compression operations. In this case the

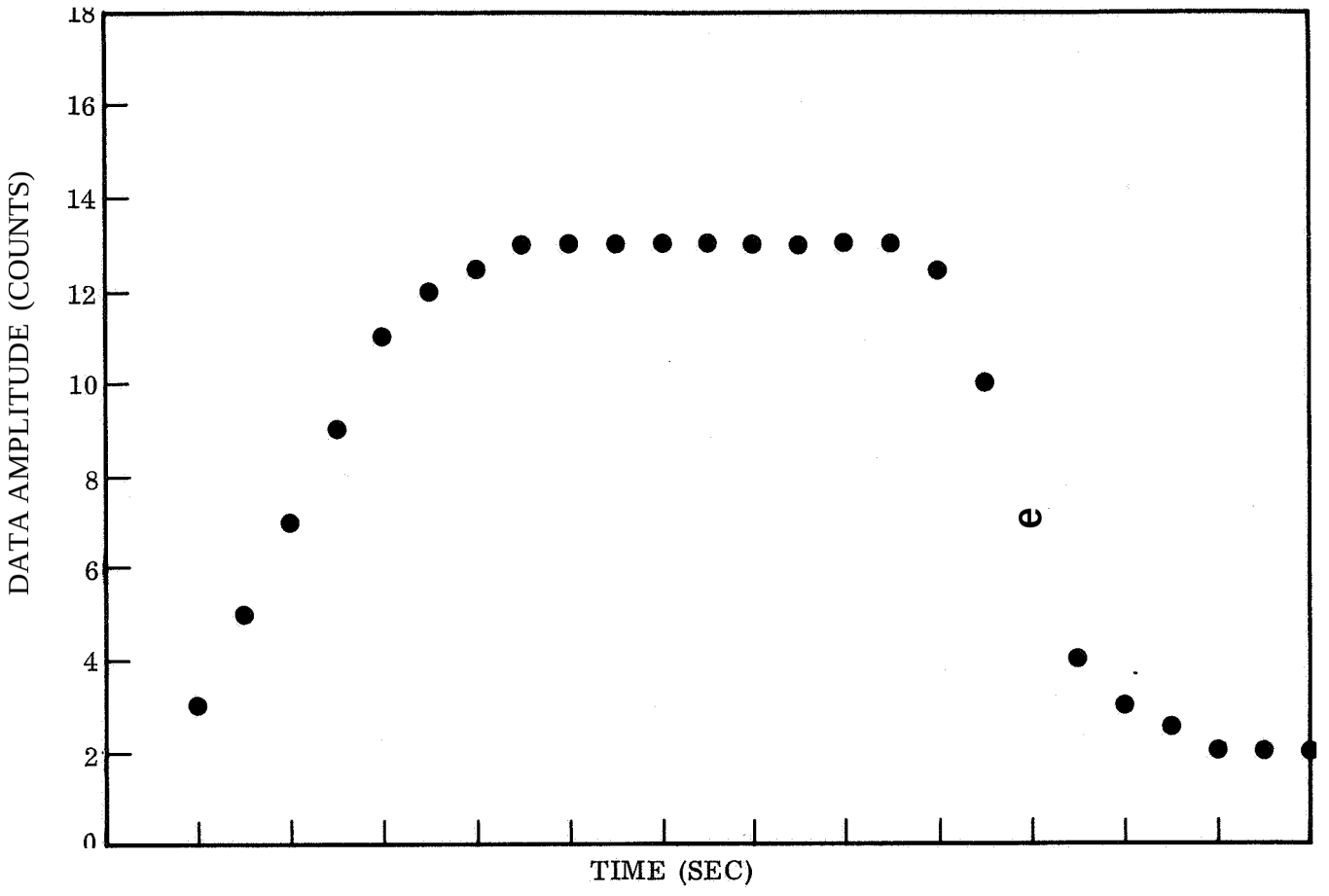


Fig. 2-15 Typical Sampled Data Waveform

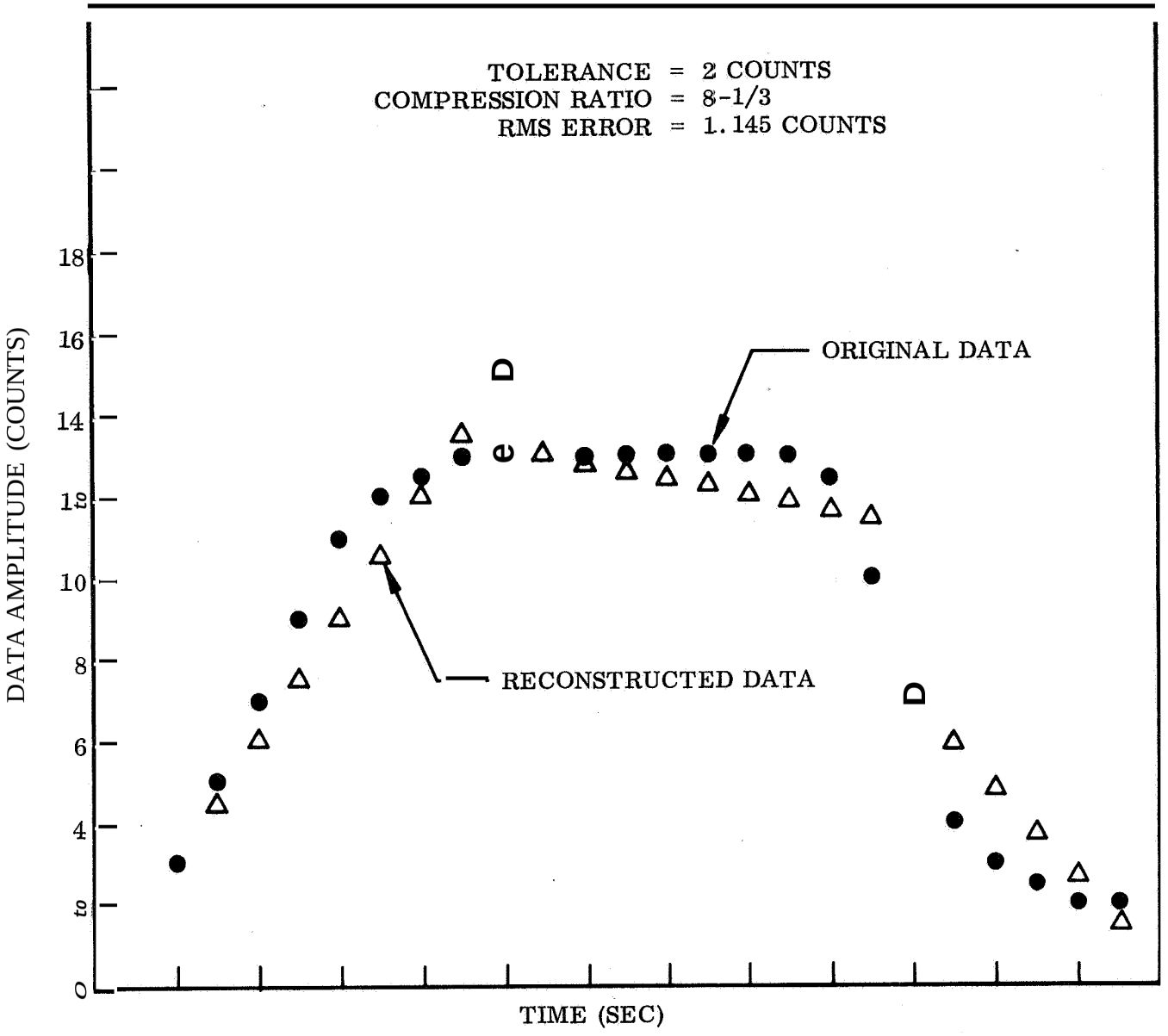


Fig. 2-16 Original and Reconstructed Data, FOIDIS Selector

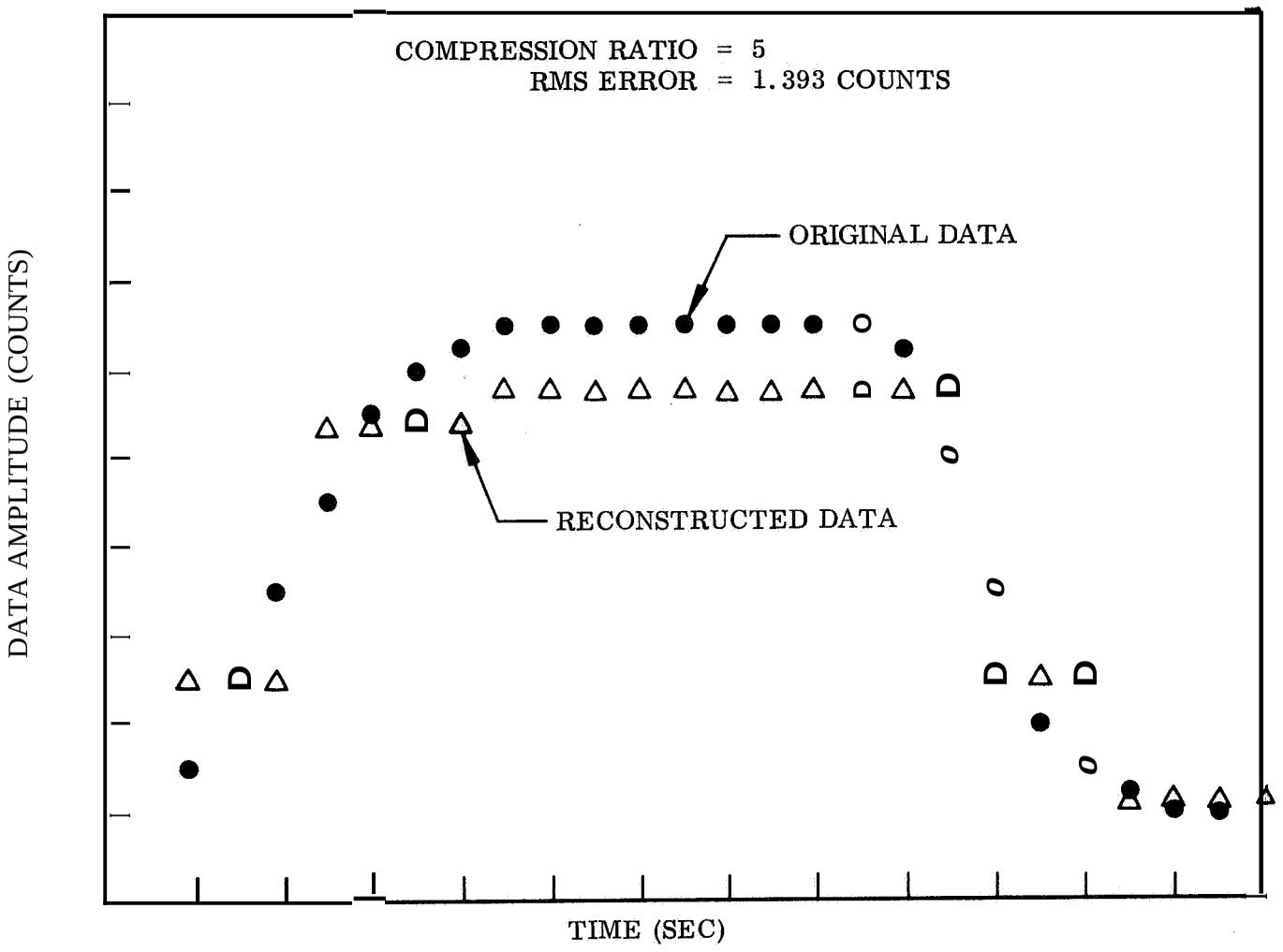


Fig. 2-17 Original and Reconstructed Data, ZOI Selector, Tolerance = 2 Counts

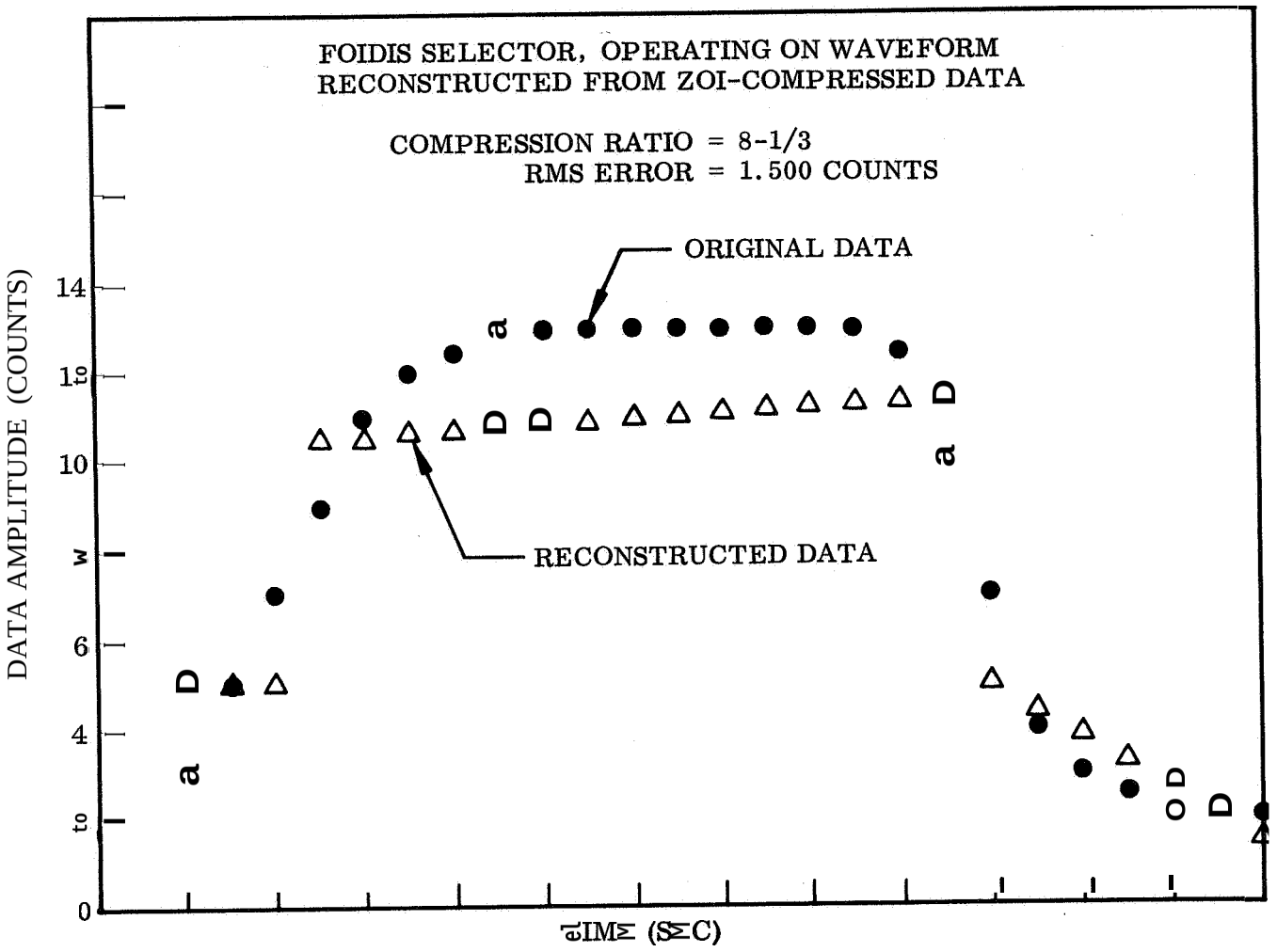


Fig. 2-18 Original and Reconstructed Data, Secondary Compression, Tolerance = 2 Counts

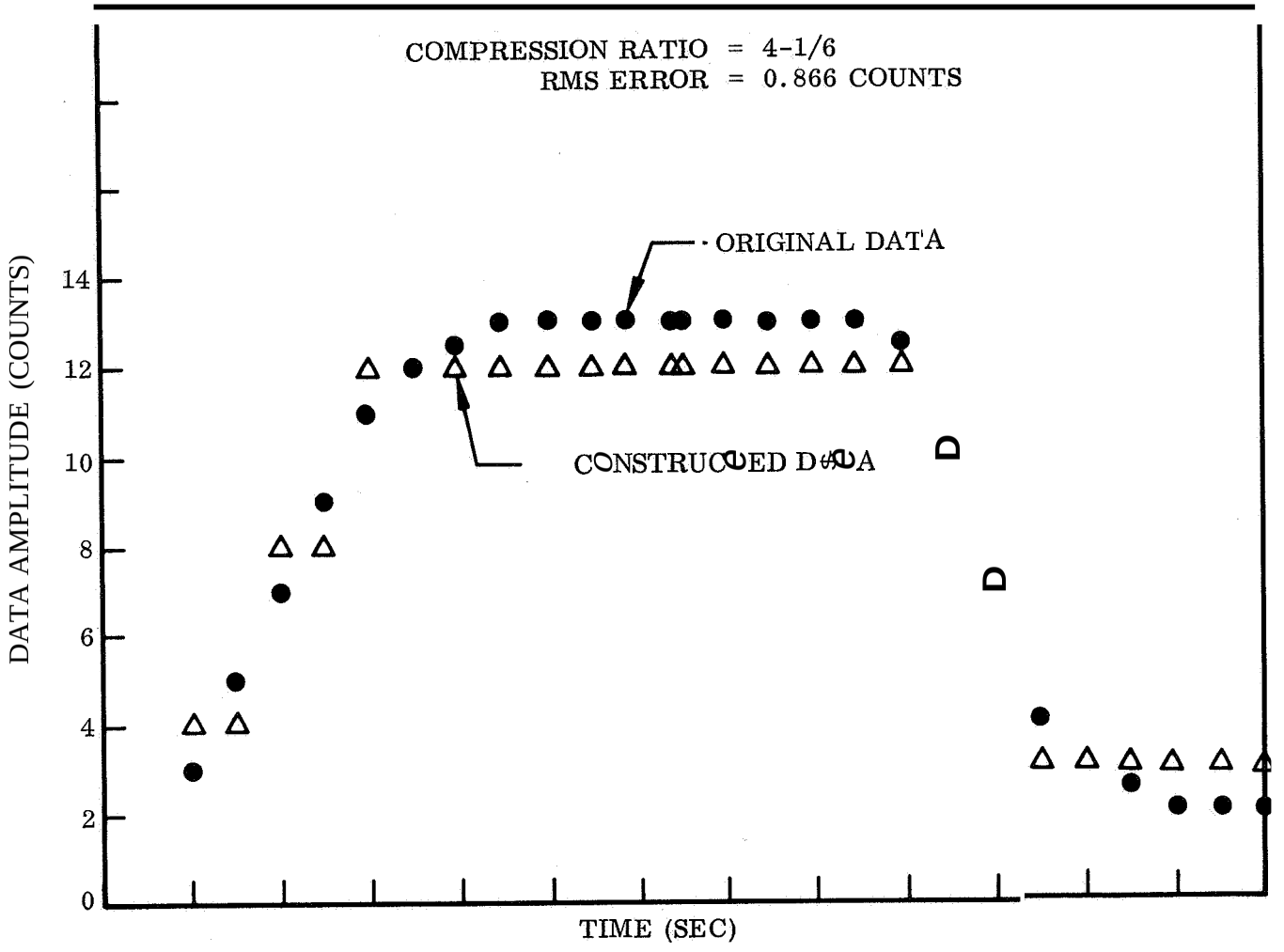


Fig. 2-19 Original and Reconstructed Data, ZOI Selector, Tolerance = 1 Count

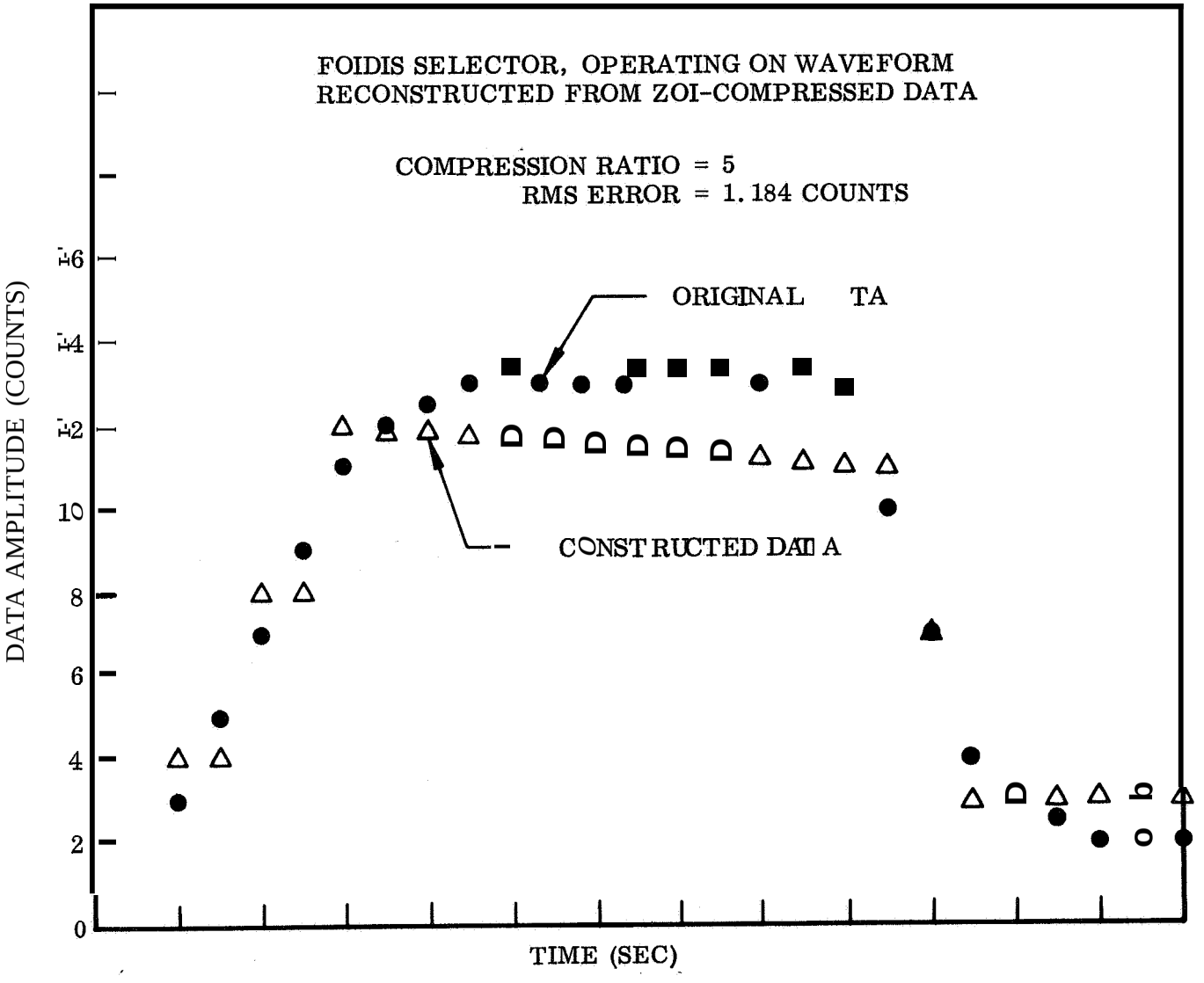


Fig. 2-20 Original and Reconstructed Data, Secondary Compression, Tolerance = 1 Count

overall compression ratio is 5 and the rms error is 1.184. The compression ratio is lower than when only FOIDIS was used while the rms error is still larger. While this is only one special example, it demonstrates what was previously stated: higher order disjoined selectors accomplish everything that comparable lower order selectors do.

This presentation would seem to disqualify the use of higher order data compression selectors from secondary compression, but this is not completely true. There is an application where compression of this type could be more than acceptable. If a vehicle did not have sufficient capability for any compression selector except a simple zero-order system it might be wise to compress the data a second time as they were received at the ground station in order to more efficiently distribute the data over the ground links. The desirability of such a scheme is demonstrated by the fact that, for compression using a ZOI selector alone (Fig. 2-17), the compression ratio is 5 and the rms error is 1 [illegible] counts, while secondary compression (Fig. 2-20) yielded the same compression ratio but [illegible] rms error of only 1.184 counts.

2.4.3 Use of Higher Order Compression on Previously Compressed Data That Have Not Been Reconstructed

When zero-order compressed data are further compressed using a first-order selector before reconstruction, data magnitudes may be eliminated. However, the time of occurrence of these redundant samples must be saved. There could be as few as zero of these "secondarily" redundant samples per line segment or there could be many.

To analyze this secondary compression technique, a zero-order compressed data bit stream format, modified from that shown in Fig. 4-1b, was chosen as a model. This format has two types of odd compressed data words (in addition to the normal time/position and sync words). One type of compressed PCM data word is identical to word 1 shown in Fig. 4-1b, except there is one additional bit to indicate that this is a data magnitude word. (Thus each word is 16 bits long.) The other type of

word is used to transmit time and ID information for the secondarily redundant samples. This is accomplished by transmitting channel ID and flag (Bits 10 through 14 of Word 1, Fig. 4-1b) just as for the normal magnitude word. Since identification of this nature takes 5 bits, three of these time identification words could be placed in one coded compressed bit stream word if each word were made 17 bits long. This is where t saving of bits occurs

Neglecting the time/position and sync words at the end of each bit stream record, the zero-order formatted data has a one-to-one ratio of compressed samples to coded compressed words. Thus, after the first (zero-order) compression there are S_C coded compressed words. Now, the secondary bit compression ratio may be defined as $N_{bs} = S_C/S_R$, where S_R is the number of coded compressed words after secondary data compression. Since every three secondarily redundant samples form one coded compressed word, $S_R = S_C - S_E + (S_E/3)$ where S_E is the number of coded compressed words that are eliminated. Therefore $\bar{N}_S = (S_C)/[S_C - (2S_E/3)]$. The maximum number of samples that could ever be eliminated is S_C; thus, even if all the remaining samples were eliminated the second time, so that $S_E = S_C$, the maximum secondary bit compression ratio would be 3. (Actually it would be less than 3 if time/position and sync were considered.) The question is, "Is a secondary bit compression ratio of 3 good enough?"

To find the answer to that question, assume that data have been compressed with a tolerance of 2K, using some first-order interpolator. If data are to be compressed using a zero-order and a first-order selector in tandem, then a tolerance of K would be needed in each case to allow the same overall peak error as was produced by the first-order interpolator alone with a tolerance of 2K. If the overall bandwidth compression obtained by compressing data with the zero-order selector and a tolerance of K and then with a first-order interpolator with a tolerance of K is greater than the compression ratio obtained by using the first-order interpolator and a tolerance of 2K, then secondary data compression is worthwhile. Taking an average of the bit compression ratios observed for the 10 sensors used in the S-6 and S-49 selector performance study, it was found that $\bar{\bar{N}}_b = 3.07$ for both zero-order selectors with a

tolerance of K = 2 counts, and $\overline{\overline{N}}_b$ = 25.13 for the three first-order interpolators with a tolerance of 4. Thus, on the average, a compression ratio of 25.13/3.07 = 8.18 is necessary when the zero-order compressed data is secondarily compressed using a first-order interpolator. Since the maximum attainable compression ratio is 3, it is seen that secondary data compression of this type is not useful in attaining higher compression ratios than were obtained using a first-order interpolator alone. Also, because of the limiting factor of 3, this type of secondary compression is considered to be less promising than secondary compression of compressed/reconstructed data, for use in conjunction with vehicle compression which was limited to a simple zero-order selector.

2.4.4 Conclusions and Recommendations

Secondary data compression according to this study, appears to have little value except for special applications. The secondary data compression technique not discussed here, parameter extraction, may be promising. However, to make a detailed, comprehensive evaluation of its possibilities would involve an extensive study. This would have to include an evaluation and study of all the parameter extraction used on the S-6 and S-49 vehicles, as well as other techniques that could be used in the future, and is beyond the scope of the present effort.

Section 3
QUEUING CONTROL EVALUATION

Buffer queuing control prevents data sample loss caused by buffer overflow. This buffer, placed at the output of a redundancy reduction data compressor to store temporarily aperiodic data, overflows when its capacity is exceeded. Queuing control, however, transforms this overflow into controlled data degradation, thus minimizing its effect on OAO data quality.

Reference 4 describes an evaluation of queuing control systems conducted during Phase I. That initial study examined the following types of adaptive tolerance control systems.

- Systems which monitor the average run length of a given fixed number of most recent line segments* for each individual sensor (called compression ratio monitoring systems in Ref. 4)
- Systems which monitor the buffer queue length

Each of these systems controls the tolerance associated with each sensor individually. The Phase I study was made with the aid of digital computer-simulation, using selected portions of telemetry data from the S-6, Explorer XVII, satellite. The major conclusion derived from this study was that the relative stationarity of the buffer input sample arrival statistics with the S-6 data precluded the need for queuing control to guard against short-term queue buildup, as long as the compressed data transmission sample rate was sufficiently high. However, adaptive aperture queuing control would be required to ensure against buffer overflow during high long-term activity, when the average buffer input rate would have, in the absence of control, crept above the fixed transmission rate. Under these circumstances, a run length monitoring system was found to be completely ineffective in controlling the queue length.

*Line segment run length is defined as the amount of time (measured in suitable units, e.g., seconds, number of data samples) between two significant data samples (or sample pairs, in the case of the FOIDIS or FOI4DG selectors) from a given sensor.

During Phase I the queuing control study was continued. A theoretical analysis was performed to aid the search for new control techniques. The empirical study was extended to systems employing adaptive filtering and sample deletion. Multiplexed telemetry data from the S-49, OGO-A, satellite was used in addition to the S-6 satellite data specimen employed during Phase I. A major output of the theoretical analysis was a promising queuing control system which monitors a combination of the queue length and its time integral. This technique was also examined empirically.

To preserve the continuity of the text, all the illustrations in Section 3 are presented after the text.

3.1 THEORETICAL ANALYSIS

The theoretical study was divided into three major parts: a deterministic feedback system analysis, the investigation of a predictive queuing control technique, and a study of optimum queuing control. The analysis, which is reported in detail in Ref. 1, is summarized below.

3.1.1 Deterministic Feedback System Analysis

In the analysis of deterministic queuing control systems, equations were derived describing the time behavior of a data compressor model with (1) several different degrees of polynomial queue feedback,* (2) first-degree polynomial derivative- and integral-queue feedback, and (3) exponential queue feedback.** These equations were then programmed on a digital computer, and SC4020 plots obtained, showing queue length as function of time for various values of feedback parameters and data

*With polynomial queue feedback, the tolerance is a polynomial function of the queue length.

**With exponential queue feedback, the tolerance is an exponential function of the queue length.

activity. It was found that (1) dynamic buffer queue behavior is relatively independent of the shape or functional form of the feedback relation used, and (2) a system employing first-degree polynomial derivative queue feedback is basically unstable. It was also found that by combining the queue length and its time integral to control the data tolerance, the control system can be adjusted to reach and maintain any equilibrium value of buffer queue selected by the user, independent of data activity. It was recommended that this feedback technique be further investigated, by means of computer-simulation.

3.1.2 Predictive Queuing Control

The queuing control technique analyzed during this portion of the study predicts future data activity from past compressor behavior, and adjusts the tolerance in accordance with this prediction in order to prevent buffer overflow. This technique appears to offer several advantages, one being good-quality reconstructed data, even during those periods following a surge of data activity, when conventional queuing control techniques generally excessively degrade data. It was recommended that this feedback technique be further investigated by means of computer-simulation.

3.1.3 Optimum Queuing Control

The third part of the study investigated the synthesis of an optimum adaptive tolerance feedback system with the objective of minimizing the rms tolerance used by the compressor. This analysis showed that the task will be formidable because of severe mathematical difficulties. However, these difficulties do not appear insurmountable, and additional work was recommended. This study provided a foundation for the work and suggested those areas on which it should concentrate.

3 2 EMPIRICAL ANALYSIS

The empirical queuing control study was divided into the following parts:

- A series of tests was performed on the queue length and compression ratio monitoring adaptive tolerance systems studied during Phase I using S-49 telemetry data, instead of the S-6 data used during Phase I.
- An investigation was made of the control system which monitors a combination of the queue length and its time integral,* using both S-6 and S-49 data.
- A study was made of adaptive precompression filtering and sample selection control systems, using both S-6 and S-49 data.

This paragraph describes each of these studies and presents their results. It is divided into four major parts. The first three parts discuss the three portions of the queuing control study; the last part presents a comparative evaluation of all the control systems, both for S-6 and S-49 telemetry data.

The basic simulated data compression system used in this study is described in Appendix IV of Ref. 4. The first-order interpolation disjoined (FOIDIS) selector was used in this system throughout the study. Descriptions of the S-6 and S-49 data are presented in Section 2. As in Phase I, the evaluation was based on two criteria of queuing control performance. The first was the ability of the system to keep the queue length within specified bounds, and the second was the fidelity of the compressed (and subsequently reconstructed) data waveforms. To facilitate the observation of queuing behavior, all simulated data compressor runs except the last two discussed in Par 3 2 3.3, were made with the buffer capacity set so high that overflow would have been impossible under any conditions.

Because of the relative stationarity of the buffer input sample arrival statistics with the S-6 telemetry data the performances of the several queuing control systems using

*This system will hereafter be referred to as the "queue/integral-queue control system."

these data were investigated under the circumstance of an excessively low buffer readout rate.

Throughout the queuing control study using S-49 data, it was assumed that any degradation of the OGO digital data would render them essentially useless to the data analyst. Therefore, the tolerance was set on all digital sensors to equal 0.01 decimal counts* so that even a word change in the least significant bit position would be noted. Likewise, to preserve this fidelity, filtering, sample deletion, and queuing control were never applied to any of the digital sensors.

The 384 S-49 subcommutated channels (128 per minor frame channel) were omitted from the study at the customer's request.

To evaluate the various queuing control systems, a number of computer runs were made simulating the operations of the data compressor, both with and without queuing control. These runs, which are described in the following paragraphs, are identified as follows. Computer runs using S-6 data will be designated S6-A, S6-B, and so on, in the order that they appear in the text. Likewise, runs using S-49 data will be designated S49-A, S49-B, etc.

3 2.1 Queue Length and Compression Ratio Monitoring

This paragraph describes the tests performed on the adaptive tolerance queuing control systems examined during Phase I, using S-49 telemetry data. These are the compression ratio and the queue length monitoring systems. These systems are described in Appendix IV of Ref. 4.

A total of six simulated data compressor runs were made in the investigation of queue length and compression ratio monitoring. The first four were made without queuing control, to examine queue behavior, compression ratios, and reconstructed data.

---

*Full scale = 511 decimal counts for digital sensors.

The last two were queuing control runs. The data specimen was taken from GSFC S-49 data tape No. 1125, between 338.4 and 342.4 seconds from the beginning of the tape. Approximately 28,400 samples were processed. The specimen was selected for its varying activity; so that different queuing control systems could be more fully evaluated.

The six runs are summarized in Table 3-1. On the second run (S49-B) the buffer readout rate was set at 544 samples per second, making the ratio of average buffer sample readin rate to readout rate, $\bar{\rho}$, equal to 0.95. The maximum queue length achieved in the simulated infinite-capacity buffer on this run was 200 samples.

The queuing control system used on Run S49-E monitored the short-term average line segment run length. On both queuing control runs, control was applied only to the analog sensors of the experiments which were in operation, as shown in Table 3-2. The control curve is shown as the straight line in Fig. IV-2, Ref. 4. Its slope and intercept were chosen for each analog sensor according to the following criteria. The tolerance was set equal to the value shown in Table 3-2, when the short-term average run length equaled the value averaged over the entire run. The maximum tolerance (achieved at an average run length equal to one sample period) was set equal to three times the value shown in Table 3-2. The quantity J_R * was set to 7, and the buffer readout rate was made equal to that of Run S49-B, viz., 544 samples per second.

The queuing control system used on Run S49-F monitored the queue length. The control curve for this run is shown as the straight line in Fig. IV-3, Ref. 4. Its slope and intercept were chosen for each analog sensor according to the following criteria. Below a queue length of 20 samples, the tolerance was set equal to the value shown in Table 3-2. Above 20, the tolerance was increased linearly and was made to triple at a queue length equal to 200 samples. The buffer readout rate was set equal to 544, as in Run S49-E.

---

*The quantity J_R is the number of most recent line segments, including the current run, over which the average run length is computed.

Table 3-1

COMPRESSOR RUN SUMMARY, QUEUE LENGTH AND COMPRESSION RATIO MONITORING, S-49 DATA

Run	Queuing Control System	M	$\bar{\rho}$	Combined $\bar{N}$	Maximum Queue Length (samples)
S49-A	None	574	0.91	13.32	163
S49-B	None	544	0.96	13.32	200
S49-C	None	528	0.99	13.32	220
S49-D	None	344	1.51	13.32	850
S49-E	ATCR	544	0.96	13.23	215
S49-F	ATQL	544	0.93	13.79	185

Explanation of symbols:

M: Buffer readout rate in samples per second

$\bar{\rho}$: Ratio of average buffer sample readin rate to readout rate

$\bar{N}$: Sample compression ratio

ATCR: Adaptive tolerance, compression ratio monitoring

ATQL: Adaptive tolerance, queue length monitoring

Table 3-2

S-49 SENSOR TOLERANCES

Experiment Number	Sensor Number	Minor Frame Channel No.	Analog or Digital	In[illegible] er-ati[illegible] ?	Tolerance (counts)(a)
1	1	81, 82, 83, 84	D	Yes	0.01
2	2	4	A	Yes	2
	3	5	A	Yes	2
	4	6	A	Yes	2
	5	7	A	Yes	2
	6	21	A	Yes	2
	7	22	A	Yes	2
	8	23	A	Yes	4
3	9	10, 42, 74, 106	A	No	1000
	10	36, 107	A	No	1000
4	11	55	A	Yes	4
	12	56	A	Yes	4
	13	57	D	Yes	0.01
5	14	13, 77	A	Yes	2
	15	25, 89	D	Yes	0.01
	16	104	A	Yes	8
6	17	58	D	No	1000
	18	59	D	No	1000
	19	60	D	No	1000
7	20	19	D	Yes	0.01
	21	20	A	Yes	2
	22	28, 52	D	Yes	0.01
	23	29	A	Yes	2
	24	49	D	Yes	0.01
	25	50	D	Yes	0.01
	26	51	D	Yes	0.01
	27	53	A	Yes	2

Table 3-2 (cont.)

Experiment Number	Sensor Number	Minor Frame Channel No.	Analog or Digital	In Operation ?	Tolerance (counts)[a]
8	28	114	D	Yes	0 01
	29	115	D	Yes	0 01
3	30	43, 90 113	D	No	1000
	31	100	A	No	1000
	32	116, 121, 122, 123, 124	D	No	1000
10	33	14, 37, 81, 86, 117	A	Yes	2
	34	15, 38, 62, 87, 118	A	Yes	2
	35	16, 39 63 88, 113	A	Yes	2
	36	80	A	Yes	2
11	37	30, 94	A	Yes	2
	38	31, 95	A	Yes	2
	39	32, 96	A	Yes	2
	40	31	A	Yes	2
	41	32	A	Yes	2
	42	33	A	Yes	2
	43	102	A	Yes	2
12	44	125	A	No	1000
	45	126	A	No	1000
	46	127	A	No	1000
	47	128	A	No	1000
13	48	9, 12, 41, 44, 73, 76, 105, 108	D	Yes	0 01
15	49	24	A	Yes	4
	50	27	A	Yes	4
	51	40	A	Yes	4

Table 3-2 (cont)

Experiment Number	Sensor Number	Minor Frame Channel No.	Analog or Digital	In Operation ?	Tolerance (counts)[a]
	52	54	A	Yes	4
	53	68	A	Yes	4
	54	69	A	Yes	4
	55	70	A	Yes	4
	56	71	A	Yes	4
	57	72	A	Yes	4
	58	109	A	Yes	4
	59	46, 110	A	No	1000
	60	47, 111	A	No	1000
	61	48, 112	A	No	1000
	62	8, 26, 45, 64, 85, 103, 120	A	No	1000
	63	78	A	Yes	2
	64	79	A	Yes	2
	65	11	D	Yes	0 01
	66	17	D	Yes	0.01
	67	18	D	Yes	0 01
	68	75	A	Yes	4
	69	101	–	No	1000
	70	1	D	Yes	0.01
	71	2	D	Yes	0.01
	72	3	D	Yes	0.01
	73	33	D	Yes	0.01
	74	34	D	Yes	0.01
	75	35	D	Yes	0.01
	76	65	D	Yes	0.01
	77	66	D	Yes	0.01
	78	67	D	Yes	0.01

(a) Full scale = 511 counts for digital sensors and 255 counts for analog sensors

The buffer queue lengths for Runs S49-B, S49-E, and S49-F are plotted as functions of time in Fig. 3-1, and the maximum queue lengths are shown for all the runs in Table 3-1. It is evident that, with the parameter values chosen neither control system was very effective in reducing the maximum queue length, although the queue length monitoring (ATQL) system was slightly more effective than did the compression ratio monitoring (ATCR) system. There are two reasons for this apparent lack of effectiveness. First, the "stiffness" of control (i.e., the increase in tolerance for a given increase in queue length or decrease in average run length) was relatively low. Second, and probably most important, control was applied only to the 54 analog minor-frame channels, which constituted only 61 percent of the 89 channels in operation. Any contribution to the queue buildup from the digital sensors could not be directly controlled; hence, the overall control effectiveness was materially lowered.

Three analog sensors were reconstructed from Runs S49-B, S49-E, and S49-F, and their rms errors obtained. These values are presented, along with average and maximum tolerances and sample compression rates, in Table 3-3. The table shows higher rms errors with queuing control, and, for two of the three sensors examined, higher errors with the ATQL system than with the ATCR system. Time histories of the Sensor 37 original and reconstructed data are shown along with the reconstructed data error for Run S49-B (no control) in Fig. 3-2. Corresponding plots for the same segment of data are shown for Runs S49-E and S49-F in Figs. 3-3a and 3-4a, respectively. Figures 3-3b and 3-4b present sensor 37 tolerances as functions of time for the two queuing control runs over the same time periods.

Considering both control effectiveness and reconstructed data fidelity, it is judged that the queue length monitoring system performed better on the S-49 data than did the compression ratio monitoring system. However, with the parameters chosen, neither system was very effective in controlling the queue length.

Table 3-3

RESULTS BY INDIVIDUAL SENSOR, QUEUE LENGTH AND COMPRESSION RATIO MONITORING, S-49 DATA

Sensor	Run	Queuing Control System	Average Tolerance (counts)(a)	Maximum Tolerance (counts)	Rms Error (counts)	Sample Compression Ratio
14	S49-B	None	2.00	2.00	1.01	12.40
	S49-E	ATCR(b)	2.46	6.16	1.03	17.12
	S49-F	ATQL	2.99	5.66	1.11	17.12
37	S43-B	None	2.00	2.00	1.01	35.60
	S43-E	ATCR	3.90	6.05	1.42	55.50
	S43-F	ATQL	3.04	5.64	1.97	55.50
50	S43-B	None	4.00	4.00	1.99	24.27
	S43-E	ATCR	5.58	12.29	2.58	18.50
	S43-F	ATQL	6.03	11.16	2.10	27.75

(a) Full scale = 2.55 counts for analog sensors
(b) See explanation of symbols in Table 3-1

3.2.2 Queue/Integral-Queue Control System

This paragraph describes the investigation of the queue/integral-queue control system. Answers to the following questions were sought:

- Does the addition of integral feedback enhance of degrade the performance of the adaptive tolerance control system?
- What set of control parameter values will yield the highest performance?

3.2.2.1 Control System Description. A weighted sum of the queue length and its time-integral, recomputed with each input data sample, was used to determine the tolerance for each sensor. The control curves for the queue/integral-queue system are shown in Fig. 3-5. The control equations are

$$K_i = K_i' = K_{1i} + \beta_{Qi} n_j + \beta_{Ii} \sum_{k=0}^{j} (n_k - n_S) \quad \text{if} \quad K_i' \geq 0 \qquad (3.1)$$

$$K_i = 0 \quad \text{if} \quad K_i' < 0$$

where

K_i = tolerance for Sensor i
K_{1i} = initial tolerance for Sensor i
β_{Qi} = queue length feedback constant for Sensor i
β_{Ii} = integral feedback constant for Sensor i
n_j = present queue length (after the jth input sample to the data compressor)
n_k = queue length after the kth input sample to the data compressor; $k \leq j$
n_S = desired quiescent queue length

The variables controlled by data cards were n_S, K_{1i}, β_{Qi}, a β_{Ii}, $1 \leq i \leq S_T$, where S_T = total number of sensors.

3.2.2.2 Tests Using S-6 Telemetry Data. This paragraph presents the results of the tests performed on the queue/integral-queue control system, using S-6 data. These tests involved a total of 25 simulated data compressor runs. Each run employed a different combination of values of the queue length feedback constant β_{Qi}, the integral feedback constant β_{Ii}, and the initial tolerance K_{1i}, shown in Eq. (3.1). None of the remaining parameter values were varied. The desired quiescent queue length n_S was set equal to 50 samples throughout the experiment.

The parameters β_{Qi}, β_{Ii}, and K_{1i} were made identical for all sensors, so that $\beta_{Q1} = \beta_{Q2} = \ldots = \beta_{Q64} = \beta_Q$, $\beta_{I1} = \beta_{I2} = \ldots = \beta_{I64} = \beta_I$, and $K_{11} = K_{12} = \ldots = K_{164} = K_1$. In all runs the quantities K_1 and β_Q were chosen so that if the summation term in Eq. (3.1) were zero, the tolerance K would equal 4 counts, when the queue length n equaled 50 samples, i.e.,

$$K = 4 = K_1 + 50\,\beta_Q \tag{3.2}$$

Thus, K_1 and β_Q always formed a pair of values. Five pairs of values of K_1 and β_Q were used, along with five values of β_I, for a total of 25 combinations. The parameter β_Q was varied between 0.025 and 0.4 in multiples of 2, and K_1 was varied in accordance with Eq. (3.2). The quantity β_I was varied between 0.00002 and 0.0025 in multiples of 5, for four values. The fifth value of β_I was zero; i.e., no integral feedback was used on five of the computer runs.

The readout rate from the buffer was set to 148 samples per sec so that the average readin rate would exceed the readout rate if the tolerance was fixed, without control, at 4 counts on all sensors. Thus, under these circumstances, the buffer would continually fill. The telemetry data specimen was taken from GSFC S-6 data tape No. 59 between 74 and 85 seconds from the beginning of the tape. Approximately 18,800 samples were processed.

The characteristic behavior of the adaptive tolerance system with integral feedback added is shown in Fig. 3-6. The queue length, shown in the lower part of the figure,

tends to oscillate about the desired quiescent value (in this case, 50 samples). The tolerance also tends to oscillate, as shown in the upper portion of the figure. The amplitudes of these oscillations generally decrease as β_Q increases, and their frequency decreases as β_I decreases. (For the largest value of β_Q used in the experiment, namely, 0.4, the queue length oscillations could not be distinguished above the variations caused by statistical changes in data activity.)

The cyclic variation in tolerance shown in Fig. 3-6 characteristically produces errors in reconstructed data waveforms similar to those shown in Fig. 3-7. This figure indicates that the fidelity of the reconstructed data is closely related to the tolerance, because when the tolerance increases the fidelity becomes poorer. The relationship between rms error, caused by compression and reconstruction, and maximum and average tolerance is shown in Table 3-4. This table presents rms errors observed in reconstructed data waveforms from three sensors, for four compressor simulation runs with queuing control, and one without. Because the tolerance is so closely related to data fidelity, it was used in this experiment as the measure of fidelity.

The maximum and average values of tolerance observed on all runs except those for $\beta_I = 0$ are shown in Figs. 3-8 and 3-9, respectively. Tolerance values for $\beta_I = 0$ are shown in Fig. 3-10. The following system behavior is evident from these curves:

- In general, the maximum and average values of tolerance decrease as β_I is decreased, and are at their lowest values when $\beta_I = 0$
- For higher values of β_Q, the rate of this decrease diminishes, and becomes nearly zero as β_I is decreased. (The curves in Figs. 3-8 and 3-9 become flatter.)
- For most values of β_I other than zero, as β_Q is increased, both the average and maximum tolerances decrease to minimum values, and then begin to increase again. (This is shown in Figs. 3-8 and 3-9.)
- Figure 3-10 shows that the maximum and average values of tolerance increase as β_Q increases, when $\beta_I = 0$

Table 3-4

RMS ERRORS IN RECONSTRUCTED DATA, QUEUE/INTEGRAL-QUEUE CONTROL SYSTEM

Sensor	β_Q	β_I	Average Tolerance (counts)(a)	Maximum Tolerance (counts)	Rms Error (counts)
7	(b)	(b)	4.0	4.0	1.57
	0.05	0	10.5	13.6	4.65
	0.05	0.0005	14.8	30.3	8.18
	0.05	0.0025	21.3	58.6	10.09
	0.10	0.0005	12.5	24.3	6.41
9	(b)	(b)	4.0	4.0	1.82
	0.05	0	10.5	13.6	3.75
	0.05	0.0005	14.8	30.3	6.11
	0.05	0.0025	21.3	58.6	11.79
	0.10	0.0005	12.5	24.3	5.88
11	(b)	(b)	4.0	4.0	2.26
	0.05	0	10.5	13.6	6.20
	0.05	0.0005	14.8	30.3	10.21
	0.05	0.0025	21.3	58.6	15.48
	0.10	0.0005	12.5	24.3	7.85

Explanation of symbols:

β_Q: Queue length feedback constant

β_I: Integral feedback constant

(a) Full scale = 511 counts

(b) This data compressor simulation run was made without queuing control in order to obtain a reference.

The quantity chosen to measure the effectiveness of queuing control was the maximum queue length attained during each run. Figure 3-11 presents these values for all runs except those in which β_I was set at zero. The maximum queue lengths obtained when β_I equaled zero are shown in Fig. 3-10. These curves indicate the following behavior:

- For all values of β_I, including zero, the maximum queue length decreases as β_Q increases.
- For a given value of β_Q, the maximum queue length is relatively insensitive to changes in β_I, for values of β_I above approximately 0.0001. As β_I is decreased below 0.0001, the influence of the summation term in Eq. (3.1) becomes too weak, and the maximum queue length begins to rise toward its highest value, which is attained when $\beta_I = 0$.

Using these results, it should now be possible to find a combination of values of β_Q and β_I that yields the highest system performance, at least under the operating conditions and with the telemetry data used in the experiment. The results indicate that the choice of control parameters based only on the criterion of reconstructed data fidelity does not agree with what the choice would be if only the criterion of control effectiveness were considered. (The lowest maximum queue length is obtained when $\beta_Q = 0.4$ and $\beta_I = 0.0001$, but the lowest values of maximum and average tolerance are observed when $\beta_Q = 0.025$ and $\beta_I = 0$.) Therefore, a pair of compromise values should be derived by combining the two critieria.

It is evident that there are many ways of combining the performance criteria. Each method may, of course, produce a different result. The following method was used in this study because of its simplicity, and because it was desired not to give special weight to either criterion. A figure of merit was defined such that

$$f_m = \frac{10,000}{K_{max}\, n_{max}} \tag{3.3}$$

where

f_m = figure of merit

$K_{m\sqcup\sqcup}$ = maximum tolerance (counts)

$n_{m\sqcup\sqcup}$ = maximum queue length (samples)

The maximum value of tolerance was used in Eq. (3.3) because it is believed that this is a fairer measure of reconstructed data fidelity than average tolerance.

Variations of the figure of merit f_m with feedback constants β_Q and β_I are shown in Figs. 3-12 and 3-10. These curves show a definite peak in the surface formed in the three-dimensional f_m, β_Q, β_I coordinate system. The peak, however, appears to be relatively broad, in both the β_Q and β_I dimensions. Using the methods of evaluation discussed above, maximum queuing control system performance is thus obtained for values of the queue length feedback constant β_Q in the range of 0.1 to 0.2, combined with values of the integral feedback constant β_I in the range of 0.00005 to 0.0001. It should be noted, however, that other values of quiescent queue length, buffer readout rate, initial tolerance, and other system parameters, as well as other telemetry data, could yield different specific results from those obtained in this experiment.

3.2.2.3 Tests Using S-49 Telemetry Data. To investigate the queue/integral-queue control system using S-49 telemetry data, a total of four simulated data compressor runs were made. All but the first run employed queuing control. The data specimen was taken from GSFC S-49 data tape No. 1125, between 338.0 and 342.8 seconds from the beginning of the tape. Approximately 34,100 samples were processed. This specimen, which was used in all of the S-49 queuing control investigations, is characterized by three successive time periods having distinctly different average buffer input sample rates: a low rate, followed by a higher rate, followed again by the original low rate. The effect of this behavior is shown in the queue length time history for Run S49-B in Fig. 3-1.

A summary of the runs is shown in Table 3-[illegible]. The first run of the group, S4[illegible]-G, was made without queuing control in order to find the average buffer readin rate, for the following sensor tolerances. On all the digital sensors, the tolerance was set to 0.01 decimal counts, and on the sensors not in operation, 1000 counts. On all the operating analog sensors the tolerance was made equal to 2 counts.

On the three computer runs employing queuing control (S49-H, S4[illegible]-I and S4[illegible]-J), the buffer readout rate M was set equal to the average readin rate observed on Run S49-G, 536 samples per second. The desired quiescent queue length n_S was set to 50 samples. For the sensors on which control was exercised (i.e., the analog sensors in operation), the initial tolerance K_1 and the queue length feedback constant β_Q were always chosen so that, if there were no integral-queue feedback, the tolerance would equal 2 counts at a queue length n of 50. On Runs S49-H and S49-I, K_1 was set equal to 0.75 counts, and β_Q equaled 0.025. On Run S49-J, β_Q was increased to 0.100, thus requiring K_1 to be set to -3.0. The integral feedback constant β_I was varied among the runs as shown in Table 3-5

The average and maximum values of tolerance, and the maximum queue lengths observed on these runs are shown in Table 3-5. The rms errors measured on three reconstructed sensor waveforms are shown in Table 3-6. Tolerances and queue lengths are shown as functions of time in Figs. 3-13, 3-14, and 3-15.

It is evident from Fig. 3-1 that the increase in data activity, which tends to fill the buffer, begins at about 339.8 seconds. Before this time, the control system on all three runs more or less successfully maintained the queue length at an average value of 50 samples. However, after 339.8 seconds, the system was unable, even by increasing the tolerances to values greater than full scale, to check the steady rise in queue length. It therefore must be assumed that the m[illegible], if not the entire, cause of the rise is the increased activity of the digital sensors, which were omitted from the control loop.

It is possible to compare the control parameter settings on the three runs by observing the behavior of the tolerance and queue length prior to 339.8 seconds. During this

Table 3-5

COMPRESSOR RUN SUMMARY, QUEUE/INTEGRAL-QUEUE CONTROL SYSTEM, S-49 DATA

Run	M	β_Q	β_I	Average Tolerance (counts)(a)	Maximum Tolerance (counts)	Combined $\overline{N}$	Maximum Queue Length (samples)
S49-G(b)	544	—	—	—	—	12.92	234
S43-H	536	0.025	0.0025	1646	33 14	13.46	239
S49-I	536	0.025	0.0005	268	8 44	13.46	217
S43-J	536	0.100	0.0005	231	53 5	13.35	215

Explanation of symbols:
M: Buffer readout rate in samples per sec
β_Q: Queue length feedback constant
β_I: Integral feedback constant
$\overline{N}$: Sample compression ratio
(a) Run S49-G was a reference run, made without queuing control
(b) Full scale = 255 counts for analog sensors

Table 3-6

RESULTS BY INDIVIDUAL SENSOR, QUEUE/INTEGRAL-QUEUE CONTROL SYSTEM, S-49 DATA

Sensor	Run	β_Q (a)	β_I	Tolerance (counts)(b)	Rms Error (counts)	Sample Compression Ratio
14	S49-G(c)	–	–	2	1.01	12.11
	S49-H	0.025	0.0025	–	10.42	53.30
	S49-I	0.025	0.0005	–	114.96	53.30
	S49-J	0.100	0.0005	–	21.32	33.31
37	S49-G	–	–	2	1.01	38.14
	S49-H	0.025	0.0025	–	139.96	53.40
	S49-I	0.025	0.0005	–	119.22	53.40
	S49-J	0.100	0.0005	–	139.96	133.50
50	S49-G	–	–	2	1.08	5.56
	S49-H	0.025	0.0025	–	90.71	44.50
	S49-I	0.025	0.0005	–	62.70	66.75
	S49-J	0.100	0.0005	–	52.14	16.69

(a) See explanation of symbols in Table 3-5.
(b) Full scale = 255 counts for analog sensors.
(c) Run S49-G was a reference run, made without queuing control.

time period it is seen that the maximum tolerance achieved on Run S49-H was about 525 counts, while the maximum tolerance on Run S49-I was about 75 counts. However, the maximum queue length was slightly higher on Run S49-H. Hence, the lower value of β_I used on Run S49-I is more desirable. When β_Q was increased by a factor of four on Run S49-J, while maintaining β_I at the same value as on Run S49-I the maximum tolerance during the initial time period is reduced to about 25 counts. The maximum queue length during this period is also reduced and the queue length is controlled more effectively. Therefore, the higher value of β_Q is more desirable. It must be remembered, however, that during the time period prior to 339.8 seconds, no control would be needed at all to keep the queue length under 50, even with constant tolerances of 2 counts. The applicability of this type of control system to S-49 data is therefore highly questionable. A portion of the reconstructed data waveform from Sensor 50 is shown for Runs S49-I and S49-J in Figs. 3-16 and 3-17 respectively.

3.2.3 Adaptive Filtering and Sample Deletion Queuing Control

As pointed out in Ref. 4, another possible way to reduce the input [illegible] rate to the buffer, besides increasing the tolerance, is either to filter the input data to the compressor or to delete [illegible] from the input data, or both. It was the object of this portion of the investigation to evaluate these methods of control [illegible] to compare them with adaptive tolerance techniques. Data from two different satellites were used in this study, so that a broader analysis could be made.

3.2.3.1 <u>Description of Control Systems</u>. Three queuing control systems were examined in this portion of the study. These are adaptive exponential filtering (AEF), adaptive running-average filtering (ARAF), and adaptive sample deletion (ASD). All three systems employed only the queue length as the monitored parameter.

<u>Adaptive Exponential Filtering</u>. The general expression for the digital exponential filter on a given sensor's sampled waveform is

$$\hat{y}_j = ay_j + b\hat{y}_{j-1}$$

where

$\hat{y}_j$ = filtered j^{th} data sample

y_j = unfiltered j^{th} data sample

$\hat{y}_{j-1}$ = filtered $j - 1^{st}$ data sample

b = $1 - a$ = exponential filtering factor

Note that

$$a + b = 1$$

The variable controlled by the queue length was the filtering factor b_i, $1 \leq i \leq S_T$, where S_T is the total number of sensors. The factor b_i was made to vary in such a manner that the equivalent low-pass filter cutoff (half-power) frequency varied linearly with the queue length.

The control curve for this system is shown in Fig. 3-18. The control equations are

$$b_i = \alpha_i = e^{-\pi} e^{2\pi \beta_{Ei} [n - (n_{min})_i]} \qquad \text{if} \quad e^{-\pi} \leq b_i \leq 1$$

$$b_i = 0 \qquad \text{if} \quad \alpha_i < e^{-\pi}$$

$$b_i = 1 \qquad \text{if} \quad \alpha_i > 1$$

where

b_i = exponential filtering factor for Sensor i

α_i = assigned feedback constant for Sensor i

n = buffer queue length

$(n_{min})_i$ = assigned threshold value of queue length for Sensor i

The variables controlled by data cards were α_{Ei} and $(n_{min})_i$, $1 \leq i \leq S_T$, where S_T is the total number of sensors.

Adaptive Running-Average Filtering. The general expression for the runnin average filter on a given sensor's sampled waveform is

$$\hat{y}_j = \frac{1}{T} \sum_{k=0}^{T-1} y_{j-k}$$

where

$\hat{y}_j$ = filtered j^{th} data sample

y_{j-k} = unfiltered $j - k^{th}$ data sample

T = number of samples averaged together

The variable controlled by the monitored parameter was T_i , $1 \leq i \leq S_T$. The quantity T_i was made to vary in such a manner that the equivalent low-pass filter cutoff frequency varied stepwise linearly with the queue length. The program was written so that T_i never exceeded 20.

The control curve for this system is shown in Fig. 3-19. The control equations are

$$\frac{1}{T_i} = \frac{1}{T_i'} = 1 - \beta_{Ri} [n - (n_{min})_i] \qquad \text{if} \qquad 0.05 \leq \frac{1}{T_i'} \leq 1$$

The variable T_i is rounded off to the nearest integer.

$$T_i = 1 \qquad \text{if} \qquad \frac{1}{T_i'} > 1$$

$$T_i = 20 \qquad \text{if} \qquad \frac{1}{T_i'} < 0\ 05$$

where

T_i = number of samples averaged together for Sensor i

β_{Ri} = assigned feedback constant for Sensor i

n = buffer queue length

$(n_{min})_i$ = assigned threshold value of queue length for Sensor i

The variables controlled by data o s were β_{Ri} and $(n_{min})_i$, $1 \leq i \leq S_T$.

T running-average filtering system was capable of operating in two different modes:

- Filtering only – filter output sample rate equal to filter input sample rate
- Post-filtering sample deletion before compression – compressor input sample rate reduced by a factor of $1/T_i$ below the output rate of the filter; i.e., every T_i^{th} sample would be retained

Adaptive Sample Deletion. The sample deletion system retained, for a given sensor, one out of every T samples. The queue length controlled T_i , $1 \leq i \leq S_T$. The quantity T_i was made to vary in such a manner that the resulting sample rate, f_{si}/T_i varied stepwise linearly with the monitored parameter, where f_{si} = original sample rate for Sensor i.

The control curves and equations are identical to those for the running-average filtering system, except that for sample deletion there was no upper limit on T_i . Also for sample deletion, T_i is redefined as the number of samples in the group from which one is retained.

3.2.3.2 Tests Using S-6 Telemetry Data. The adaptive filtering and sample deletion queuing control system investigation using S-6 data required a total of eight simulated data compressor runs. The first run used no filtering or control, and was made for comparative purposes. The next four were "static" runs, i.e., filtering and/or sample deletion was used, but their parameter values were fixed throughout the run. There was no queuing control on these runs. The next two runs employed queuing control. The last run used adaptive tolerance queuing control without integral feedback, and employed no filtering or sample deletion. The telemetry data specimen was taken from GSFC S-6 data tape No. 59, between 74 and 90 seconds from the

beginning of the tape. Approximately 14,400 samples were processed. Filtering, deletion, and control were applied only to the sensors which generated sampled analog waveforms; Sensors 1, 8, and 37 were thus excluded.

A summary of the runs is presented in Table 3-7 In all the runs except Run S6-H, the fixed tolerance was set for each sensor to the value shown in Table 3-8. On all the static runs the buffer readout rate was set to 2266 samples per second, which, with no filtering or sample deletion, would cause the ratio of average buffer sample readin rate to readout rate $\bar{\rho}$, to equal 0 97 (Run S6-A in Table 3-7) The first of the static filtering runs shown in Table 3-7 Run S6-B, employed exponential filtering The exponential filtering factor b_i was set to 0.368 for all the filtering sensors, so that the time constant equal one sample period. T second, Run -C, used running average filtering. The averaging period T_i equal three sample per s for each analog sensor Run S6-D also employed running-average filtering with T equal to 3, but only every third sample of each filtered waveform entered the data compressor Run S6-E retained every third sample, as in Run S6-D, except that no prior filtering had been done.

A time history of the queue length is shown for a portion of Run S6-A in Fig 3-20. The combined sample compression ratios and maximum queue lengths observed on the four runs are shown in Table 3-7. It is evident from the table that the exponential filtering used on Run S6-B lowered the combined sample compression ratio but raised the maximum queue length in comparison with Run S6-A. Running-average filtering on the other hand raised the compression ratio slightly and lowered the maximum queue length a little. However, sample deletion, either with or without filtering raised the compression ratio and lowered the maximum queue length quite appreciably.

Reconstructed data from Sensor 7 are shown for Runs S6-A, S6-B, S6-C, and S6-E in Figs. 3-21 through 3-24 (In reconstructing deleted samples, first-order interpolation was used) It is evident from these figures that digital filtering and/or sample deletion can appreciably alter the shape of a waveform of this nature, especially where abrupt changes occur in the waveform.

Table 3-7

COMPRESSOR RUN SUMMARY, ADAPTIVE FILTERING AND SAMPLE DELETION, S-6 DATA

Run	Filter	Queuing Control System	M	$\bar{\rho}$	Combined $\bar{N}$	Maximum Queue Length (samples)
S6-A	None	None	226	0.97	4.10	55
S6-B	EFFP	None	226	1.04	3.82	174
S6-C	RAFP	None	226	0.95	4.18	38
S6-D	RASDFP	None	226	0.59	6.76	9
S6-E	SDFP	None	226	0.59	6.77	10
S6-F	–	ARAF	148	1.10	5.54	244
S6-G	–	ASD	148	1.06	5.73	167
S6-H	None	ATQL	148	1.04	5.87	189

Explanation of symbols:

M: Buffer readout rate in samples per second
$\bar{\rho}$: Ratio of average buffer sample readin rate to readout rate
$\bar{N}$: Sample compression ratio
EFFP: Exponential filtering, fixed-parameter
RAFP: Running-average filtering, fixed-parameter
RASDFP: Running-average filtering, fixed-parameter, with sample deletion
SDFP: Sample deletion only, fixed-parameter
ARAF: Adaptive running-average filtering
ASD: Adaptive sample deletion
ATQL: Adaptive tolerance, queue length monitoring

Table 3-8

ASSIGNED TOLERANCES FOR S-6 DATA

Sensor No.	Tolerance (counts)	Remarks
1	0.0	Digital
2	0.0	
3	16.0	
4	4.0	
5	4.0	
6	4.0	
7	4.0	
8	0.0	Digital
9	4.0	
10	4.0	
11	4.0	
12	0.0	
13	4.0	
14	4.0	
15	2.0	
16	2.0	
17 – 25, 58 – 64	8.0	Average value on subcom sensors
26 – 41	26.0	Average value on subcom sensors
42 – 57	45.0	Average value on subcom sensors

The probable reason for the poor performance of exponential filtering in the static tests is that sudden waveform jumps are transformed into exponentials, as shown in Fig. 3-22. Because the compression algorithm attempts to fit straight lines to the data waveform, it often can compress the original waveform more efficiently. Because of its poor performance, it was decided not to test exponential filtering on an adaptive basis.

The rms errors observed in the reconstructed data, compared with the original unfiltered data, from Sensors 9, 3, and 11 are shown along with individual sample compression ratios for the runs in Table 3-3. The table shows that among the three processes which increased the combined compression ratio (thus excluding exponential filtering) sample deletion, without filtering, yielded the lowest rms errors. Thus, it appeared from these static tests that, for S-6 data, adaptive sample deletion would be the most promising of the queuing control techniques. However, the rms errors observed were extremely high in most cases, for the relatively small values of T and β which were used.

Summaries of the two runs employing queuing control are shown in Table 3-7. The first, Run S6-F, used running-average filtering, and the second, Run S6-G, employed sample deletion. The control parameters were the same on both runs, and were chosen to match, as closely as possible the parameters used in Run S6-H, which employed an adaptive tolerance control system without integral feedback. In both runs, the feedback constant β_{Ri} was set for all analog sensors so that T_i would change from 2 to 3 at a queue length of 150 samples. The queue length threshold, $(n_{min})_i$, was set to 20 samples. The buffer readout rate was made equal to 148 samples per second, which, without control, would make $\bar{\rho}$ equal 1.5, causing the buffer to fill continuously.

In Run S6-H, the control parameters were chosen for each sensor so that, below a queue length of 20 samples, the tolerance equaled the value shown in Table 3-8. For queue lengths above 20, the tolerance for each sensor was made to equal twice this value at a queue length of 75 samples. The variation of tolerance with queue length was linear above 20 samples. The buffer readout rate was set at 148 samples per second.

Table 3-9

RESULTS BY INDIVIDUAL SENSOR, ADAPTIVE FILTERING AND SAMPLE DELETION, S-6 DATA

Sensor	Run	Filter	Queuing Control System	Maximum Tolerance (counts)[a]	Rms Error (counts)	Sample Compression Ratio
2	S6-A	None	None	4.0	1.31	1.89
	S6-G	—	ASD	4.0	75.86	2.67
	S6-H	None	ATQL	16.≥	5.57	2.54
3	S6-A	None	None	16.0	6.66	4.57
	S6-G	—	ASD	16.0	41.38	6.27
	S6-H	None	ATQL	63.7	25.14	6.14
	S6-A	None	None	4.0	1.50	2.16
	S6-B	EFFP[b]	None	4.0	45.09	1.90
	S6-C	RAFP	None	4.0	86.40	2.06
	S6-D	RASDFP	None	4.0	93.99	4.07
	S6-E	SDFP	None	4.0	85.05	4.07
	S6-F	—	ARAF	4.0	129.37	2.22
	S6-G	—	ASD	4.0	61.75	3.36
	S6-H	None	ATQL	15.≥	6.37	3.14
9	S6-A	None	None	4.0	1.82	18.48
	S6-B	EFFP	None	4.0	6.90	20.89
	S6-C	RAFP	None	4.0	13.28	22.88
	S6-D	RASDFP	None	4.0	12.67	30.03
	S6-E	SDFP	None	4.0	11.75	28.26
	S6-F	—	ARAF	4.0	28.83	26.29
	S6-G	—	ASD	4.0	6.75	22.88
	S6-H	None	ATQL	16.0	5.93	36.02
10	S6-A	None	None	4.0	1.73	17.78
	S6-G	—	ASD	4.0	19.05	20.00
	S6-H	None	ATQL	16.0	4.60	20.00
11	S6-A	None	None	4.0	2.26	3.64
	S6-B	EFFP	None	4.0	5.14	4.07
	S6-C	RAFP	None	4.0	7.99	4.17
	S6-D	RASDFP	None	4.0	8.21	5.05
	S6-E	SDFP	None	4.0	3.23	4.80
	S6-F	—	ARAF	4.0	38.94	4.40
	S6-G	—	ASD	4.0	2.39	4.49
	S6-H	None	ATQL	16.0	7.95	6.06

(a) Full scale = 511 counts
(b) See explanation of symbols in Table 3-7

Queue length time histories are shown for portions of Runs S6-G and S6-H in Fig. 3-20. The time periods shown occurred near the middle of the runs, after the queue lengths had reached equilibrium levels. Note that these levels are very nearly the same for both runs. Thus, the adaptive sample deletion and the adaptive tolerance control systems were about equally effective in controlling the queue length, with the buffer readout rate equal to 148 samples per second.

The queue length in the run employing running-average filtering (Run S6-F) increased in about 3 seconds to nearly the maximum value of 244 shown in Table 3-7, and maintained this approximate value throughout the run. This rapid rise in queue length is caused by the following phenomenon. The simulated control system is designed so that each time a new sample from a given sensor arrives at the filter the queue length is monitored, and the value of the averaging period T is selected according to the current queue length. Because of the multiplexing action, it is possible for the queue length to change sufficiently between samples from a given sensor to change T appreciably. For an already active sensor, such as those found in the S-6 satellite, this variation in T causes unnatural jumps in data amplitude from one sample to the next, which materially reduces the compression ratio. Thus, the effect of the control is opposite to that desired. The averaging period for Sensor 7 is shown as a function of time in Fig. 3-25. The curve shows the manner in which T quickly increases to its maximum value of 20 as the queue length increases. This amplitude-jumping effect could probably be reduced if a different-shaped T-versus-queue length control curve were used, but it may always be a problem with multiplexed telemetry data of this type.

The control effectiveness of the queuing control system employing sample deletion (Run S6-G), however, compared favorably to the adaptive tolerance system used in Run S6-H, as previously noted. It will now be interesting to compare the two systems in reconstructed data fidelity. Table 3-9 presents the values of rms error observed for the two runs in the reconstructed data from six S-6 sensors. Note that for all sensors except Sensor 11, the rms errors with adaptive sample deletion were greater than those found with adaptive tolerance control. In four of the six sensors the rms error was appreciably greater for sample deletion.

Reconstructed data from Sensor 7 are shown for Run S6-G in Fig. 3-26. Figure 3-27 shows the same portion of the Sensor 7 reconstructed data waveform for Run S6-H. Although much of the data waveform reconstructed from Run S6-G follows the original more closely than that reconstructed from Run S6-H, the large deviations, caused by sample deletion, at the transition points could cause gross misinterpretations of the data by the analyst. Hence, the waveform reconstructed from the run employing a adaptive tolerance queuing control appears to be the more desirable.

The test results thus indicated that the adaptive tolerance system performed better than any of the adaptive filtering or sample deletion systems tested, on S-6 telemetry data. While the adaptive sample deletion system equaled the adaptive tolerance system in control effectiveness, it proved to be inferior in reconstructed data fidelity. The adaptive filtering systems which were examined completely failed to match the adaptive tolerance system in performance.

3.2.3.3 <u>Tests Using S-49 Telemetry Data</u>. To study adaptive filtering and sample deletion queuing control with S-49 telemetry data, a total of eight simulated data compressor runs were made. The first three were static runs, in which filtering or sample deletion was used, but their parameter values were fixed throughout the run. The remaining five runs were dynamic; i.e., they employed queuing control. On all the runs except the last two, the capacity of the simulated buffer was made infinite so that there was never any chance of buffer overflow. On the last two runs the buffer capacity was deliberately set to a low value so that the effects of overflow could be observed. The data specimen was taken from GSFC S-49 data tape No. 1125, between 338.4 and 342.4 seconds from the beginning of the tape. Approximately 28,400 samples were processed. Filtering, deletion, and control were applied only to the analog sensors.

The eight runs are summarized in Table 3-10. Run S49-B, also shown in Table 3-1, was added to Table 3-10 for comparison purposes. On all the runs the buffer readout rate was 64 to 544 samples per second. This readout rate produced a maximum queue length, with no filtering and no control, of 200 samples (Run S49-B). On almost all

Table 3-10

COMPRESSOR RUN SUMMARY, ADAPTIVE FILTERING AND SAMPLE DELETION, S-49 DATA

Run	Filter	Queuing Control System	Buffer Capacity (samples)	M	$\bar{\rho}$	Combined $\overline{N}$	Maximum Queue Length (samples)	Samples Lost By Buffer Overflow
S49-B	None	None	∞	544	0.96	13.32	200	0
S49-K	EFFP	None	∞	544	0.91	14.00	183	0
S49-L	RAFP	None	∞	544	0.90	14.14	183	0
S49-M	SDFP	None	∞	544	0.91	13.96	189	0
S49-N	–	AEF	∞	544	0.94	13.66	193	0
S49-O	–	ASD	∞	544	0.94	13.64	196	0
S49-P	–	ARAF	∞	544	0.94	13.62	199	0
S49-Q	–	AEF	50	544	0.95	13.42	50	160
S49-R	–	ARAF	50	544	0.96	13.35	50	160

Explanation of symbols:

M: Buffer readout rate in samples per second
$\bar{\rho}$: Ratio of average buffer sample readin rate to readout rate
$\overline{N}$: Sample compression ratio
EFFP: Exponential filtering, fixed-parameter
RAFP: Running-average filtering, fixed-parameter
SDFP: Sample deletion only, fixed-parameter
AEF: Adaptive exponential filtering
ASD: Adaptive sample deletion
ARAF: Adaptive running-average filtering

the runs the tolerance was set for each sensor to the value shown in Table 3-2. The only exceptions were in Runs S49-Q and S49-R, where the tolerance was set equal to 2 counts for Sensor 50.

The first two of the study runs shown in Table 3-10, Run S49-K, employed exponential filtering. The exponential smoothing factor b_i was set to 0.38 for all the analog sensors in operation, so that the time constant equaled one sample period. The second, Run S49-L, used running-average filtering. The averaging period T_i equaled three sample periods for each analog sensor. Run S49-M deleted all but every third sample of each analog sensor, without any prior filtering.

Queue length time histories for the three sample runs so closely resemble the time history for Run S49-B in Fig. 3-1 that they are not shown. The only major difference was that the maximum queue lengths were slightly lower on the sample deletion and filtering runs. Table 3-10 shows that running average, fixed-parameter filtering (RAFP) performed slightly better than either fixed-parameter sample deletion (SDFP) or exponential, fixed-parameter filtering (EFFP), as far as compression ratio was concerned. Note, however, that neither filtering nor sample deletion increased the combined sample compression ratio or decreased the maximum queue length to any large extent.

The rms errors observed in the reconstructed data from Sensors 14, 27, and 50 are shown along with the sample compression ratios obtained for the three sensors in Table 3-11. (In reconstructing deleted samples, first-order interpolation was used.) The table shows RAFP to cause the highest rms error along with the highest sample compression ratio in most cases.

Note that in the static run the change in compression ratio obtained with SDFP was slightly lower than that observed with either type of filtering. This result is exactly opposite to that obtained with S-8 data (Par. 3.2.3.2). The probable reason for this is that the S-6 data are highly active, and are characterized by generally low compression ratios in the absence of precompression filtering, while the S-49 analog data

Table 3-11

RESULTS BY INDIVIDUAL SENSOR, ADAPTIVE FILTERING AND SAMPLE DELETION, S-49 DATA

Sensor	Run	Filter	Queuing Control System	Tolerance (counts)(a)	Buffer Capacity (samples)	Rms Error (counts)	Sample Compression Ratio
14	S49-B	None	None	2	∞	1.01	12.40
	S49-K	EFFP(b)	None	2	∞	1.14	24.72
	S49-L	RAFP	None	2	∞	1.35	27.81
	S49-M	SDFP	None	2	∞	1.06	17.12
	S49-N	–	AEF	2	∞	1.03	15.89
	S49-O	–	ASD	2	∞	1.08	12.36
	S49-P	–	ARAF	2	∞	1.10	13.09
	S49-Q	–	AEF	2	50	1.00	15.89
	S49-R	–	ARAF	2	50	1.04	11.71
37	S49-B	None	None	2	∞	1.01	35.60
	S49-K	EFFP	None	2	∞	1.35	37.00
	S49-L	RAFP	None	2	∞	1.91	44.40
	S49-M	SDFP	None	2	∞	1.00	37.00
	S49-N	–	AEF	2	∞	1.33	37.00
	S49-O	–	ASD	2	∞	(c)	37.00
	S49-P	–	ARAF	2	∞	1.88	37.00
	S49-Q	–	AEF	2	50	7.42	37.00
	S49-R	–	ARAF	2	50	16.57	37.00
50	S49-B	None	None	4	∞	1.99	24.27
	S49-K	EFFP	None	4	∞	2.16	27.75
	S49-L	RAFP	None	4	∞	2.38	27.75
	S49-M	SDFP	None	4	∞	2.11	27.75
	S49-N	–	AEF	4	∞	1.95	27.75
	S49-O	–	ASD	4	∞	1.94	27.75
	S49-P	–	ARAF	4	∞	1.94	27.75
	S49-Q	–	AEF	2	50	1.08	5.55
	S49-R	–	ARAF	2	50	1.06	5.84

(a) Full scale = 255 counts for analog sensors
(b) See explanation of symbols in Table 3-9
(c) Unavailable

waveforms are, in general, relatively smooth, and yield high compression ratios with no prior filtering. Because the compressor removes a larger number of S-49 data samples without deletion, the effect of sample deletion before compression should be less.

Also, because of the relative smoothness of the S-49 data, coupled with the presence on some waveforms of very low-level noise, precompression exponential filtering increased the combined sample compression ratio for the S-49 data, while it decreased the combined compression ratio for the highly active S-6 data (Table 3-7).

Reconstructed data from Sensor 37 are shown for Runs S49-K, S49-L, and S49-M in Figs. 3-28, 3-29, and 3-30, respectively. The data waveform reconstructed from the run employing sample deletion (Run S49-M) appears smoother than the other two waveforms. Note that none of the filtering or sample deletion methods appreciably alter the waveform, as is the case with Sensor 7 in the S-6 data.

In each of the five dynamic runs the queue length threshold, $(n_{min})_i$, was set to 20 samples. The first dynamic run, S49-N, employed an adaptive exponential filtering system. The feedback constant β_{Ei} was set for all analog sensors so that the exponential time constant would equal three sample periods (or the filtering factor b_i would equal 0.717) when the queue length reached 200 samples.

The second dynamic run, S49-O, employed adaptive sample deletion, and the third run, S49-P, used adaptive running-average filtering. On both runs the feedback constant β_{Ri}, was set for all analog sensors so that the averaging or deletion period T_i would equal 9 when the queue length reached 200 samples.

On the fourth and fifth dynamic runs, S49-Q and S49-R, all parameters were identical to those used on Runs S49-N and S49-P, respectively, except that the simulated buffer capacity was set to 50 samples, as shown in Table 3-10.

The maximum queue lengths observed on the first three dynamic runs are shown in Table 3-10, along with the values of combined sample compression ratio. The queue lengths for these runs are also shown as functions of time in Fig. 3-31. Although the effect of all three control systems with the S-49 data is almost identical, the system employing exponential filtering performed slightly better than the other two systems for the parameters chosen. Note that, while the queuing control performance of adaptive running-average filtering was not quite as good as adaptive sample deletion with the same control parameter values, it did not lower the average compression ratio and increase the queue length, as it did with the S-6 data. Thus, with the S-49 data there is no evidence of the phenomenon of amplitude-jumping, as with the S-6 data. Again, this is probably due to the relative smoothness of the S-49 data.

The most important result of these runs is that none of the adaptive filtering or sample deletion systems tested were effective in controlling the queue. The probable reason for this, as previously stated, is that control was not applied to the digital sensors, and controlling the analog sensors alone is not sufficient to control the queue.

Rms errors and individual sample compression ratios obtained on Sensors 14, 37, and 50 for the five dynamic runs are shown in Table 3-11. Reconstructed data from Sensor 37 are shown for Runs S49-N and S49-P in Figs 3-32a and 3-33a respectively. The corresponding exponential factor and averaging period are shown as functions of time for the two runs in Figs. 3-32b and 3-33b.

As stated above, the major purpose of the last two runs (S49-Q and S49-R) was to examine the effects of allowing the output buffer of the S-49 data compressor to overflow. The buffer capacity for these runs was set to 50 samples. The total number of samples lost by buffer overflow is shown for the two runs in Table 3-10. Figure 3-34 shows the queue length as a function of time for Run S49-R, and also shows how the sample loss was distributed in time. (Each dot on this chart indicates the number of samples lost during the time it took to input the previous 60 samples to the compressor.)

Figures 3-35 and 3-36 show reconstructed data waveforms under overflow conditions. In Fig. 3-35 one pair of nonredundant samples, representing one line segment of the

reconstructed waveform, was lost because of buffer overflow. This segment began at approximately 340.74 seconds. The reconstructor operates as though only the final time of each line segment is transmitted along with the sample magnitudes representing the two ends of the line. Because the reconstructor would have no information about the existence of the missing line segment, it would position the beginning point of the following line segment at the time the missing segment was supposed to begin. Hence, the reconstructed waveform shape shown in Fig. 3-35 would be obtained. In Fig. 3-36 two line segments are missing, indicating that two Sensor 37 sample pairs were lost during the time period shown because of buffer overflow.

3.2.4 Conclusions

Paragraph 3.2 has described an empirical investigation into the performance of several types of queuing control systems, used to alleviate the problem of buffer overflow in a multichannel telemetry data compressor. Methods of precompression adaptive filtering and sample deletion were examined, and compared with a method which varies the tolerance, or maximum allowable reconstructed waveform deviation. In addition a detailed examination of the adaptive tolerance system, both with and without integral feedback was made. System performance was observed for various combinations of control parameter values, and values yielding the highest performance, according to chosen criteria were found.

3.2.4.1 Evaluation With S-6 Data. The results obtained from the experiments using S-6, Explorer XVII, satellite telemetry data are directly applicable only to these data. However, to the extent that other data are similar to those of the S-6, the results, and the conclusions derived therefrom, are generally applicable to the other data as well.

The following conclusions are derived from the experimental results:

- Although the adaptive sample deletion method is a more effective means of controlling the buffer queue length than any of the adaptive filtering techniques, it does not perform as well as the technique using adaptive tolerance. Excessive distortion in the reconstructed data waveform is the main reason for the poor performance of the sample deletion system.

- In an adaptive tolerance control system, the addition of integral feedback [i.e., $\beta_I \neq 0$ in Eq. (3.1)] definitely enhances overall control system performance.
- For the set of parameter values and performance criteria used in this investigation, maximum adaptive tolerance control system performance is obtained for values of the queue length feedback constant β_Q [compare Eq. (3.1)], in the range of 0.1 to 0.2, combined with values of the integral feedback constant β_I in the range of 0.00005 to 0.0001

3.2.4.2 Evaluation With S-49 Data. In the entire investigation of queuing control using S-49 data, one observation is paramount. If control is applied only to the analog sensors, queuing control is virtually ineffective. This is because (1) a relatively large percentage of the S-49 sensors were digital, and (2) the digital sensors were the sensors that were causing the queue buildup. If the queue length is to be effectively controlled, it is necessary to apply some method of control to the digital sensors. Since it is assumed that a partial degradation of the digital outputs would be unacceptable to the user of the data, it is suggested that sensor outputs, both analog and digital, be completely eliminated on a priority basis as the queue tends to build up. With this method, all the data the user obtained would be acceptable, and the queue length could probably be effectively controlled.

Another way in which the problem of excessive queue buildup might be alleviated is to apply appropriate parameter extraction techniques to as many as possible of the digital sensors. If information about the digital data were compressed using conventional redundancy removal techniques, instead of the data words themselves, there is a good chance that the buffer input statistics would become more stationary. Thus the need is again pointed up, as it was in Par. 2.4, for a study of techniques for combining parameter extraction with redundancy removal.

There is also a possibility that the application of bit plane encoding to the S-49 digital sensors instead of polynomial selectors would alleviate the queue buildup problem. This would be especially true if the increased activity were found to be caused by data word variations only in bits of lower significance.

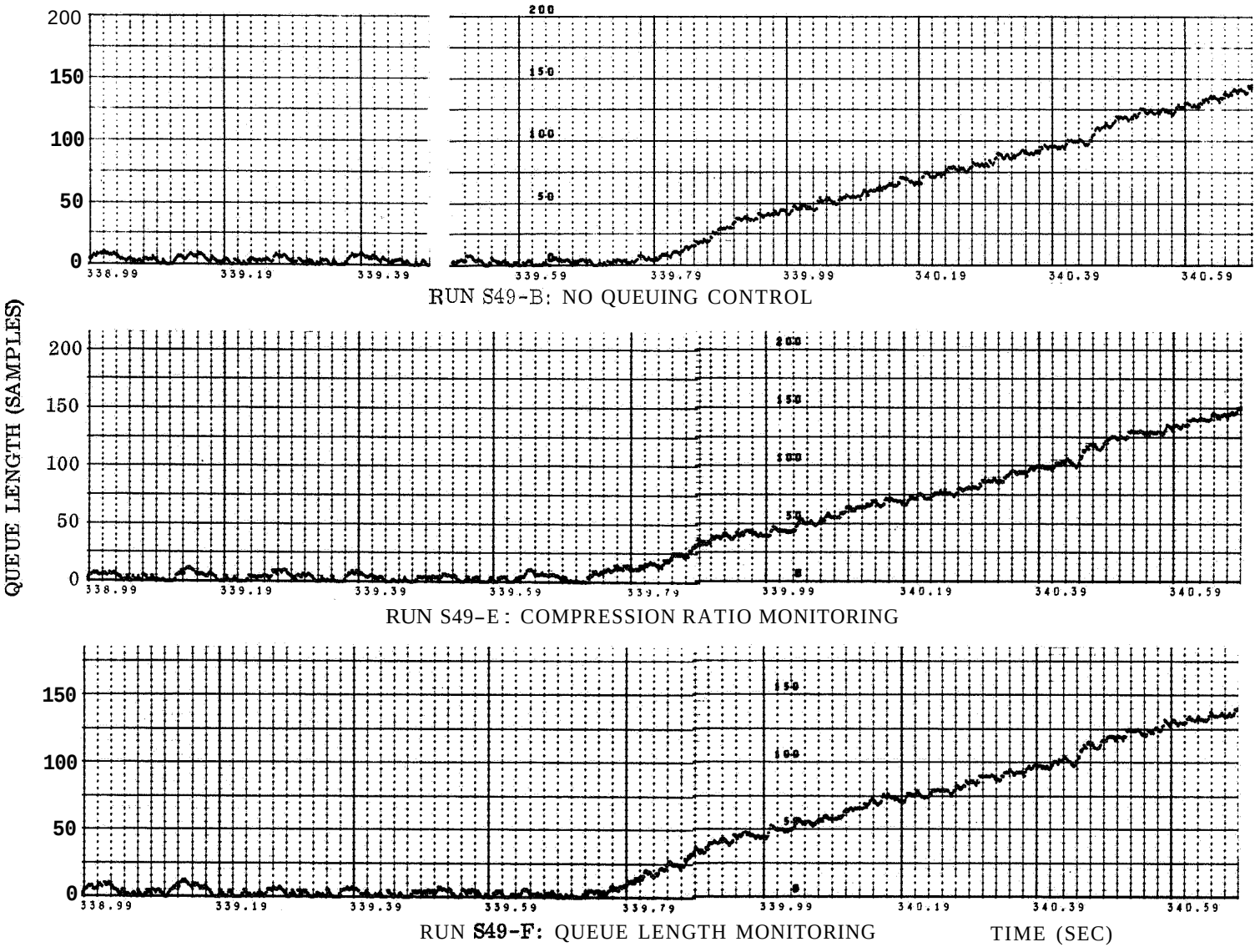


Fig. 3-1 Queue Length Behavior With Adaptive Tolerance, S-49 Data

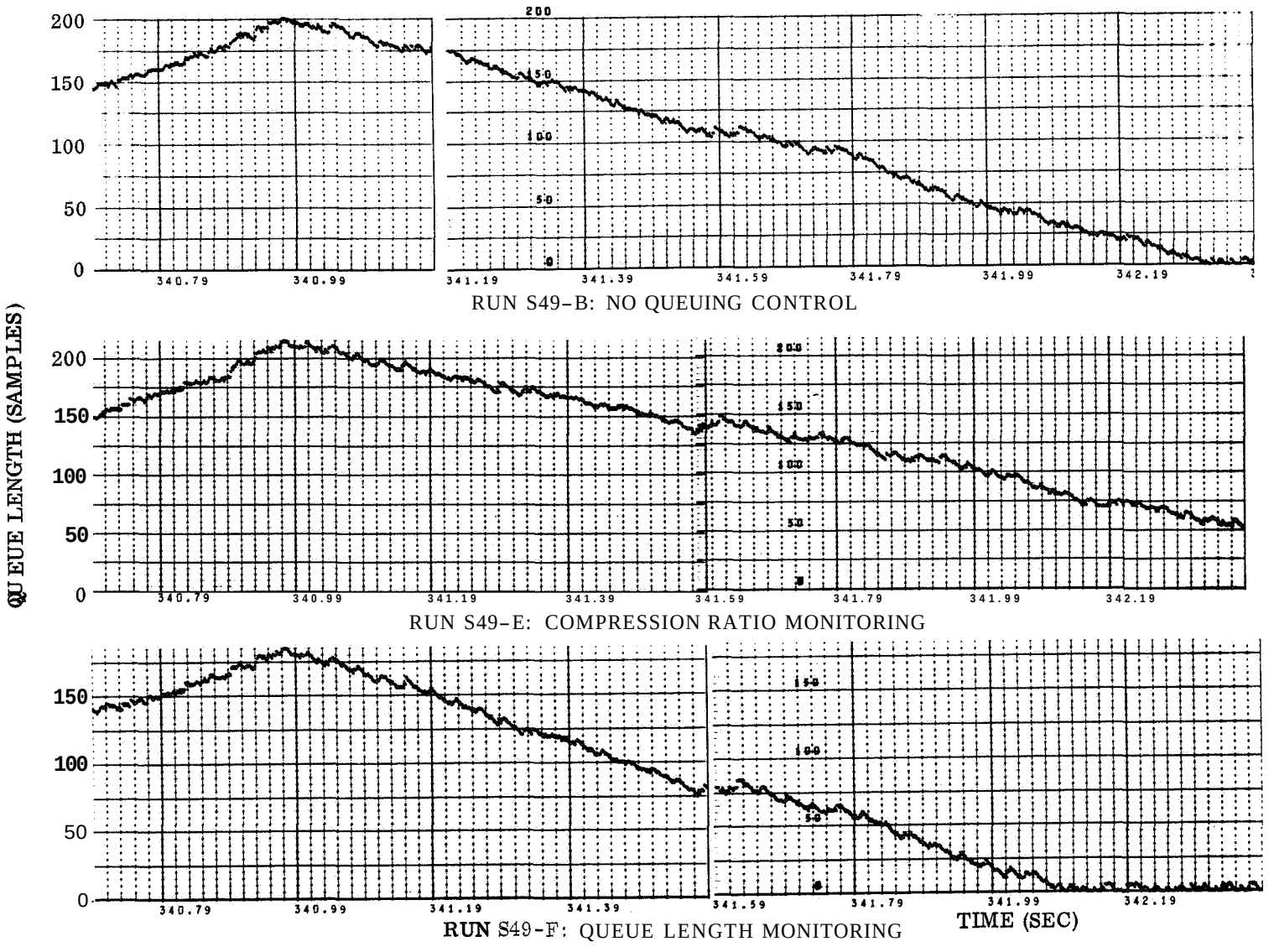


Fig. 3-1 Queue Length Behavior With Adaptive Tolerance, **S-49** Data (Cont'd)

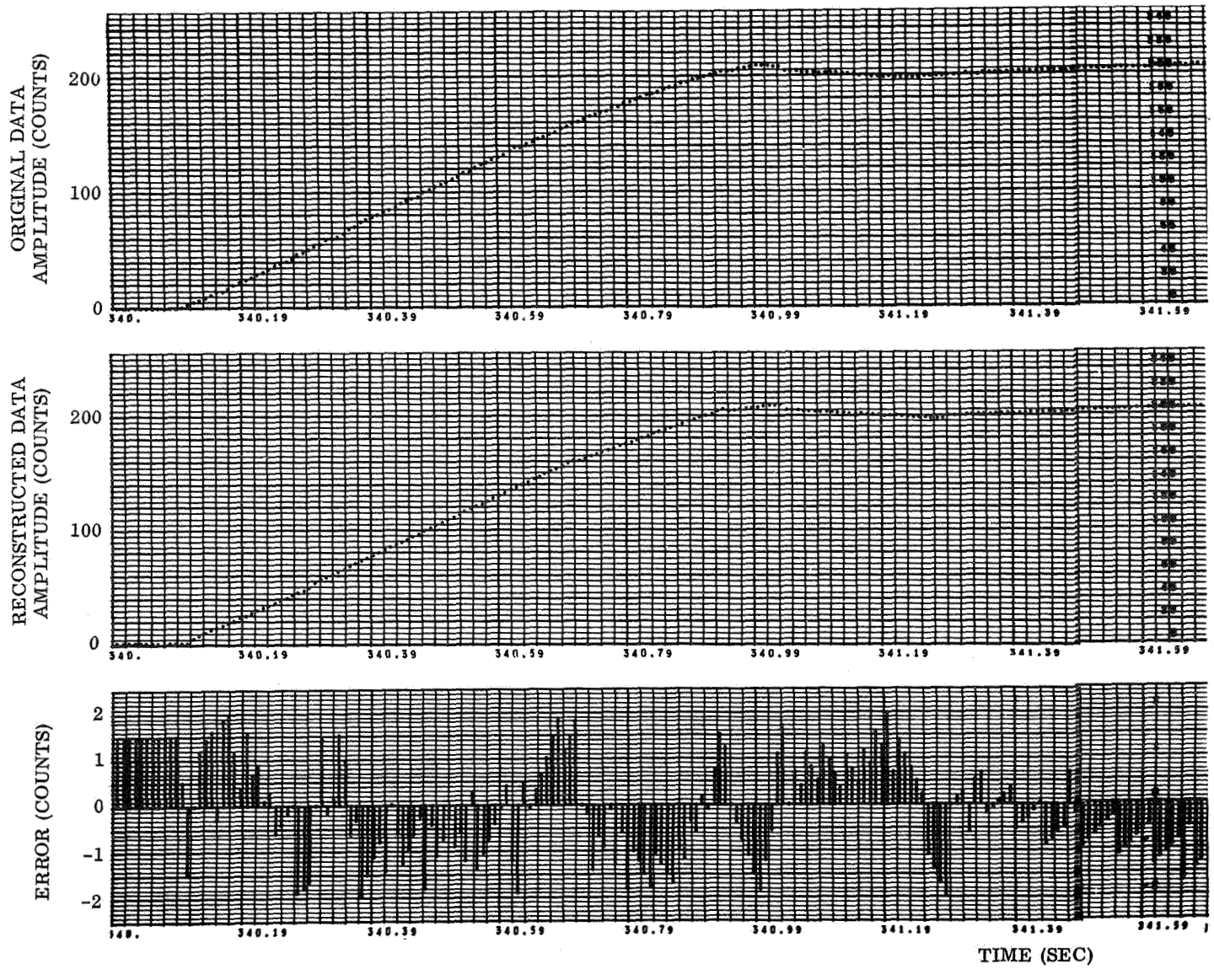


Fig. 3-2 Sensor 37 Reconstructed Data, Run **S49-B:** No Queuing Control

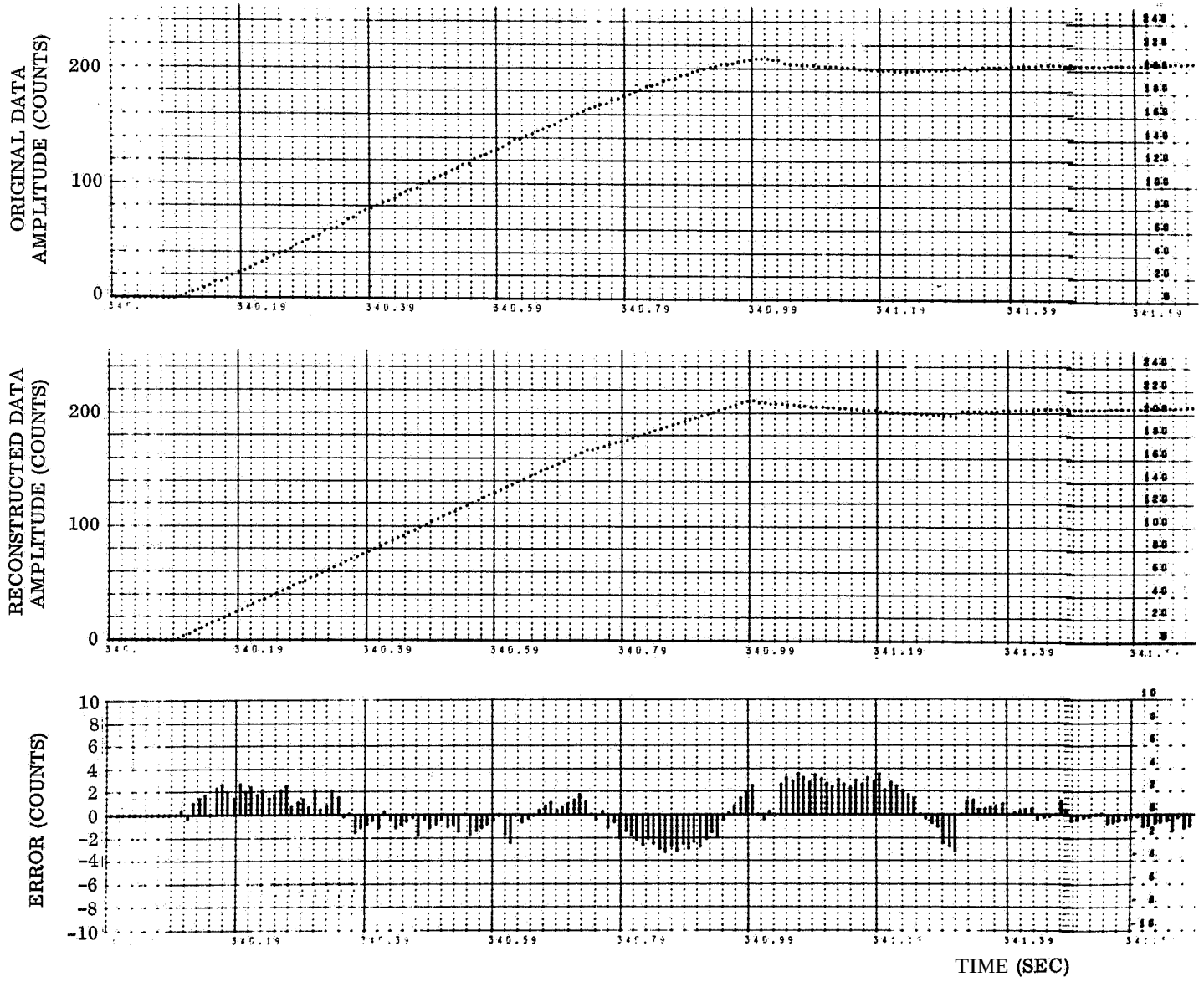


Fig. 3-3a Sensor 37 Reconstructed Data, Run S49-E: Adaptive Tolerance, Compression Ratio Monitoring

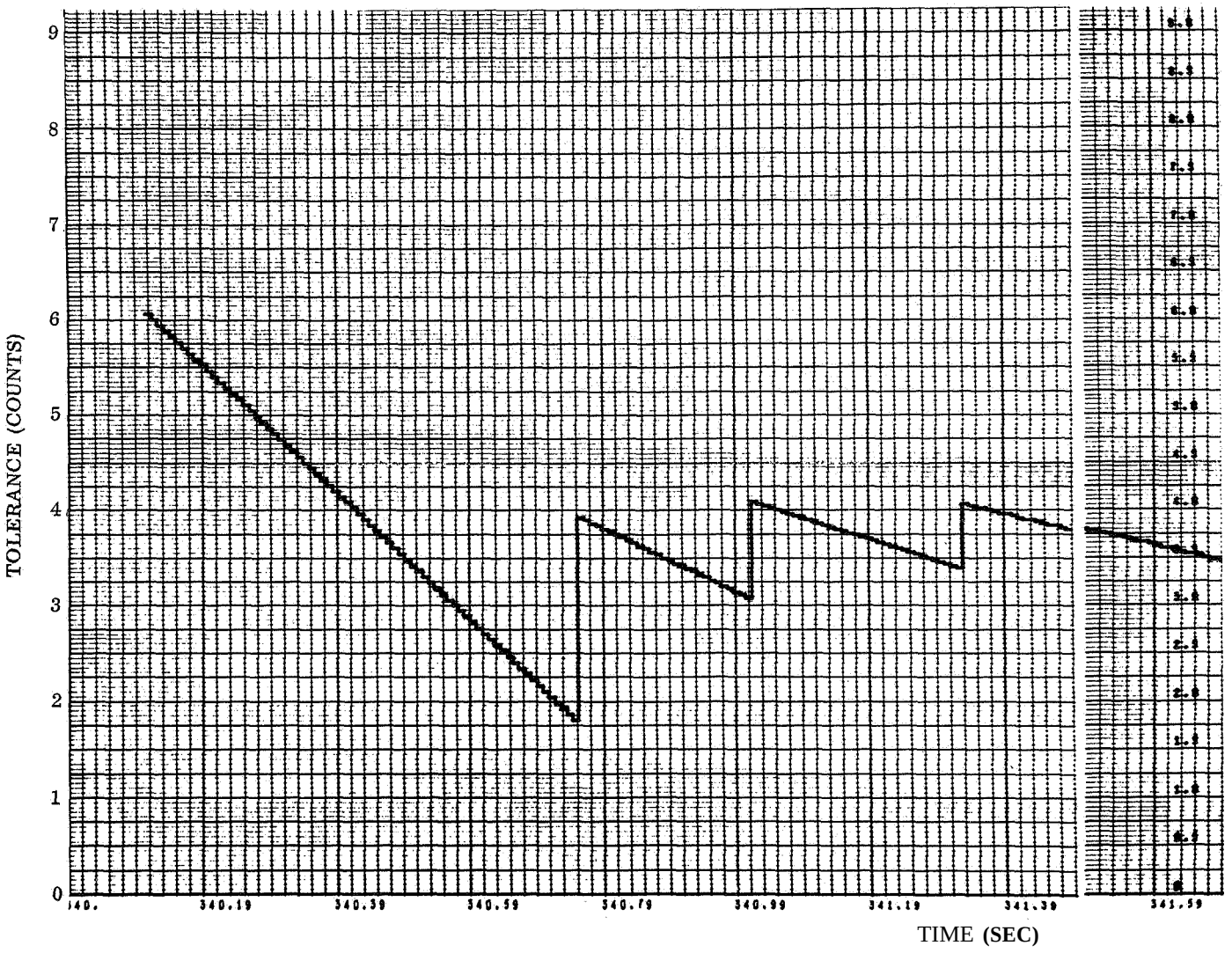


Fig. **3-3b** Sensor 37 Tolerance Variations, Run **S49-E:** Adaptive Tolerance, Compression **Ratio** Monitoring

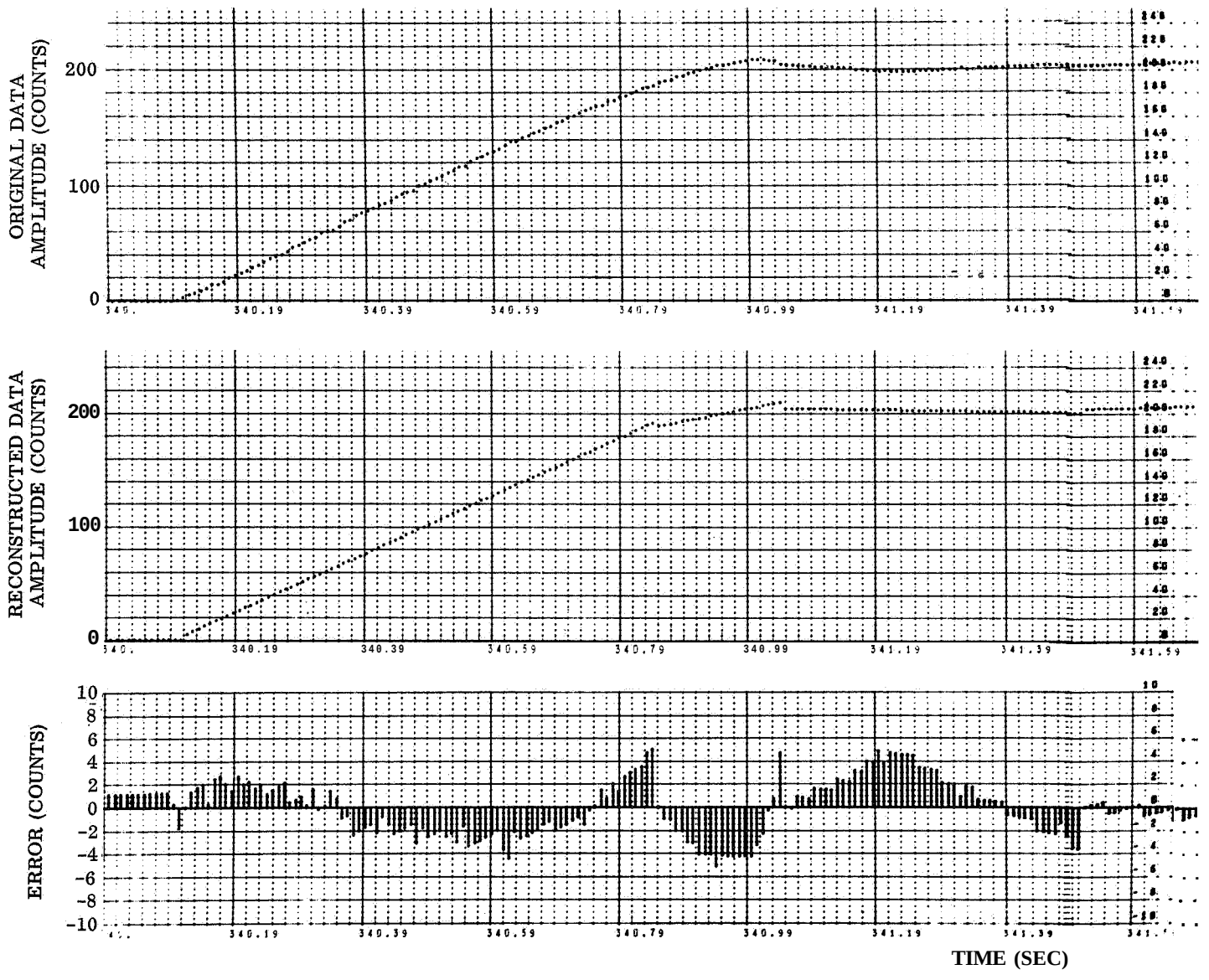


Fig. 3-4a Sensor 37 Reconstructed Data, Run S49-F: Adaptive Tolerance, Queue Length Monitoring

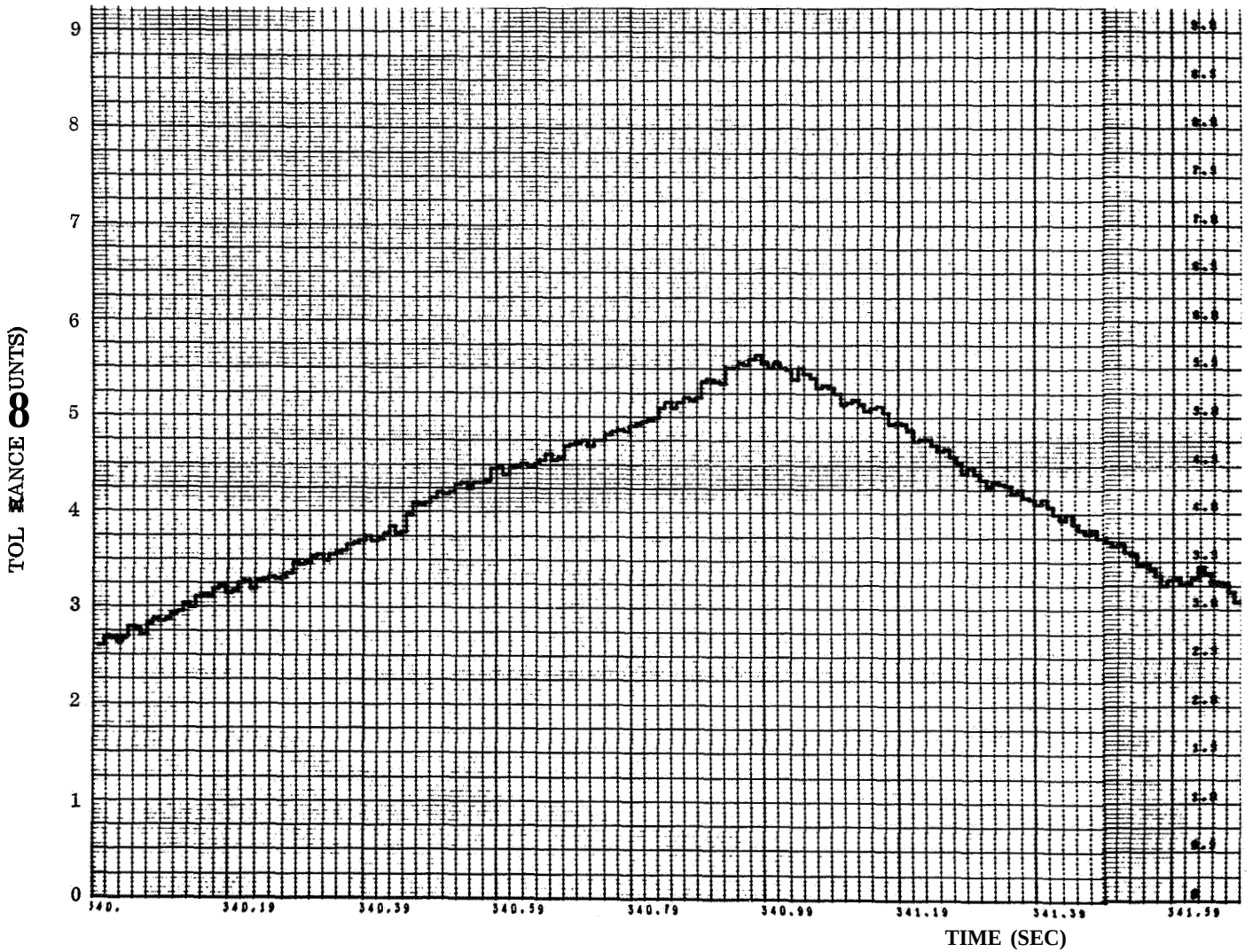
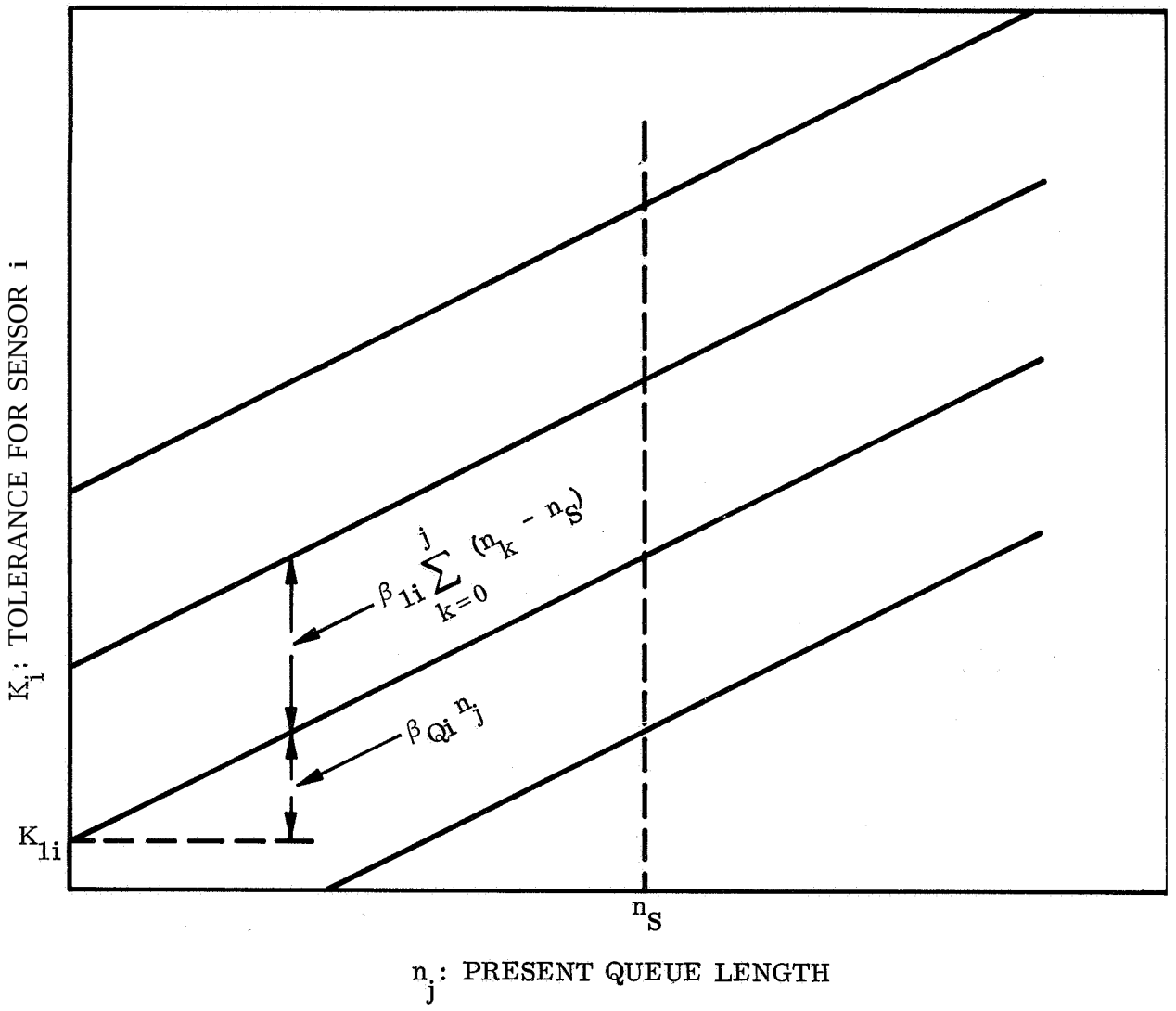


Fig. 3-4b Sensor 37 Tolerance Variations, Run S49-F: Adaptive Tolerance, Compression Ratio Monitoring



K_{1i}	=	INITIAL TOLERANCE FOR SENSOR i
β_{Qi} and β_{1i}	=	FEEDBACK CONSTANTS FOR SENSOR i
n_k	=	QUEUE LENGTH AFTER THE kth INPUT SAMPLE TO THE DATA COMPRESSOR; $k \leq j$
n_S	=	DESIRED QUIESCENT QUEUE LENGTH

Fig 3-5 Control Function for Queue/Integral-Queue Control System

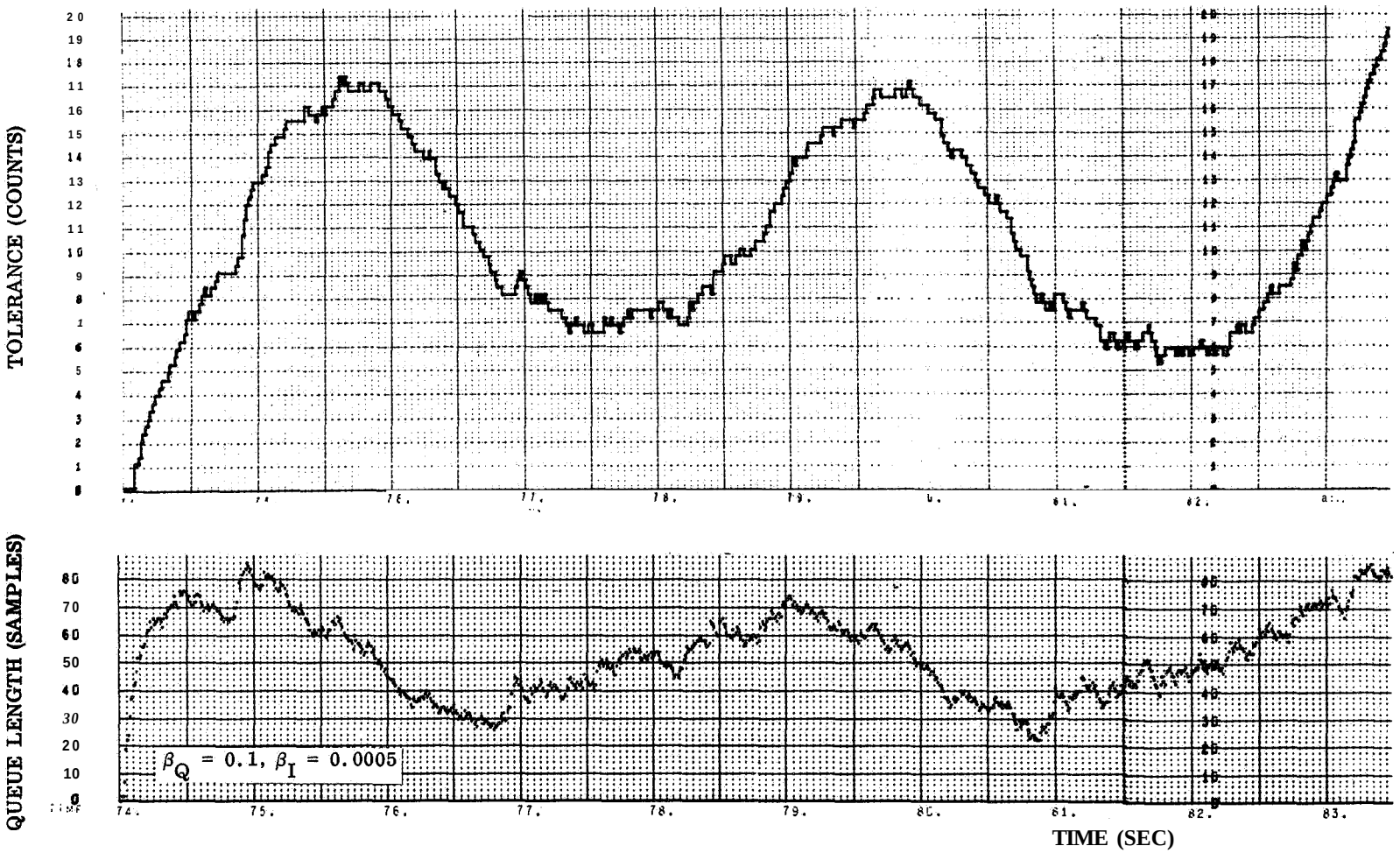


Fig. 3-6 Characteristic Tolerance and Queue Length Behavior, With Integral Feedback Added

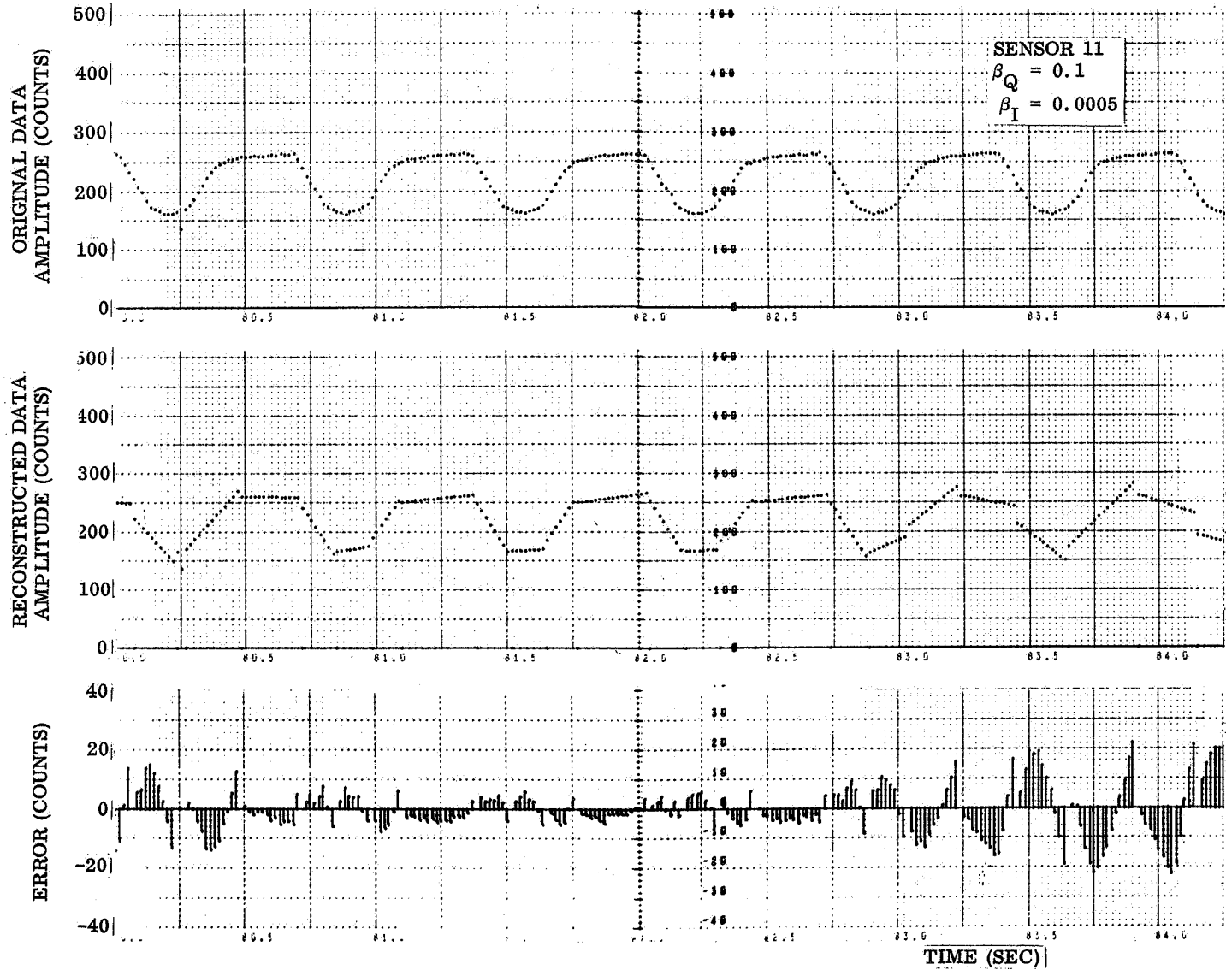


Fig. 3-7 Typical Reconstructed Data, With Integral Feedback Added

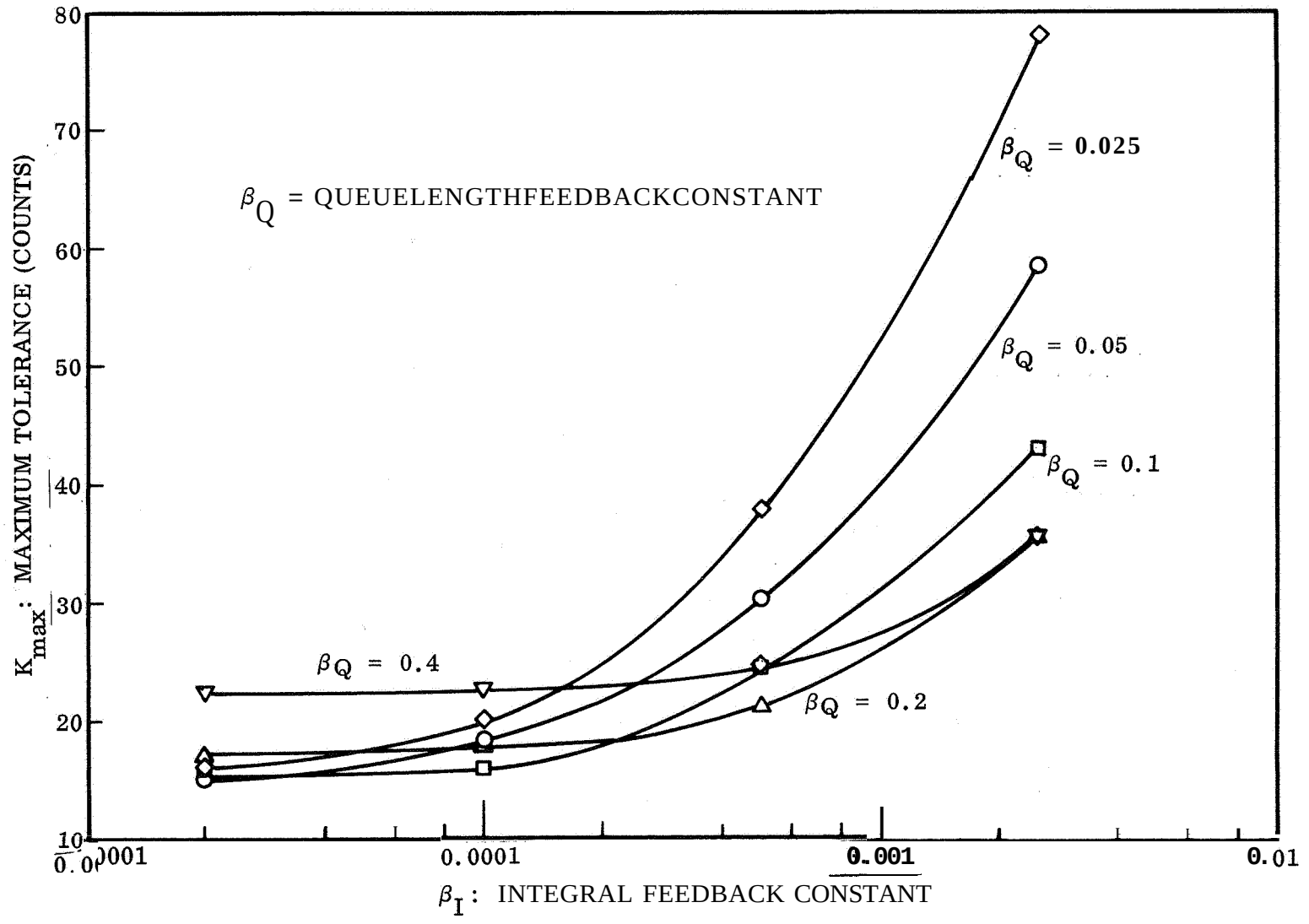


Fig. 3-8 Maximum Tolerance Variations, With Integral Feedback Added

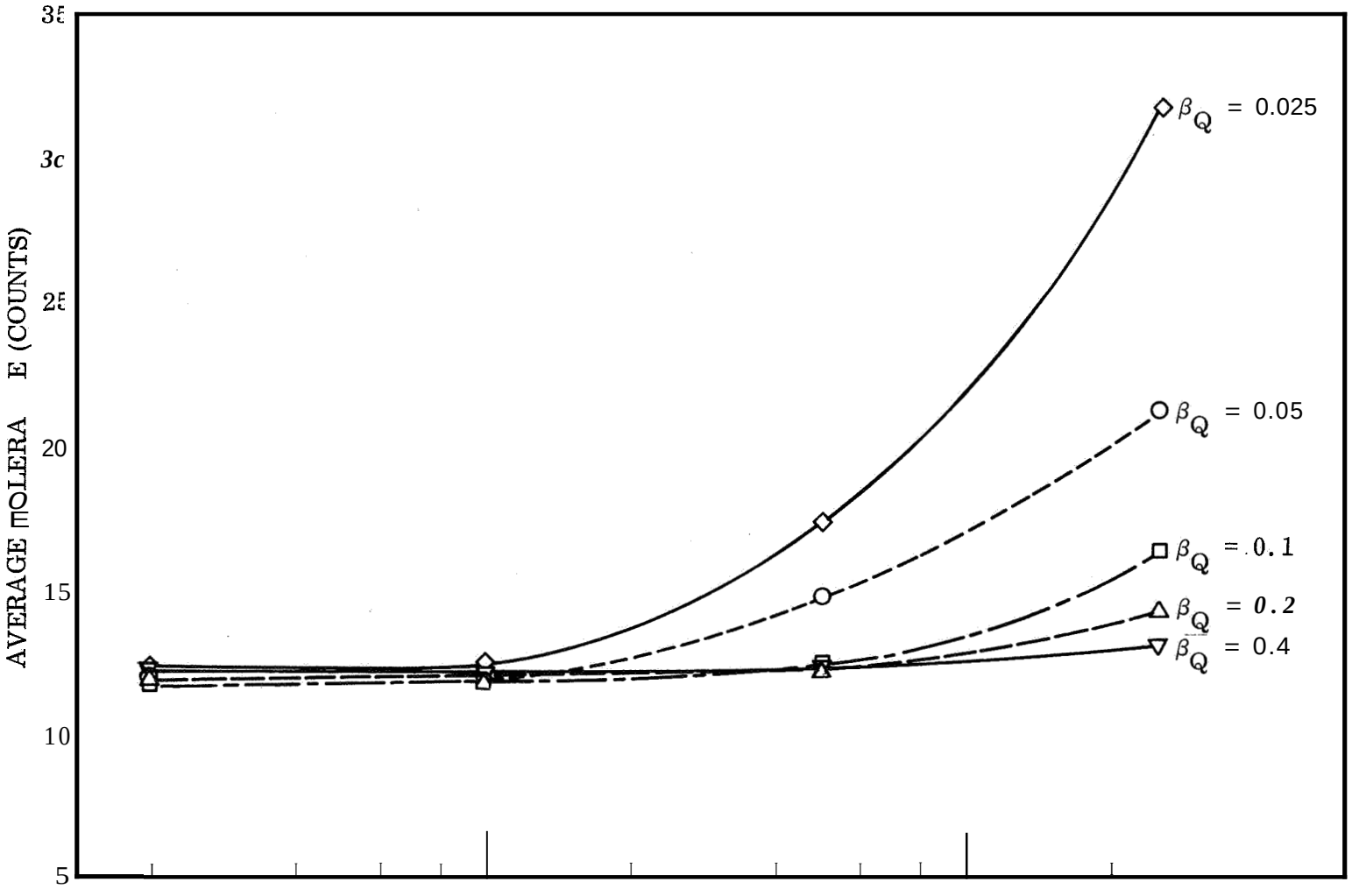


Fig. 3-9 Average Tolerance Variations, With Integral Feedback Added

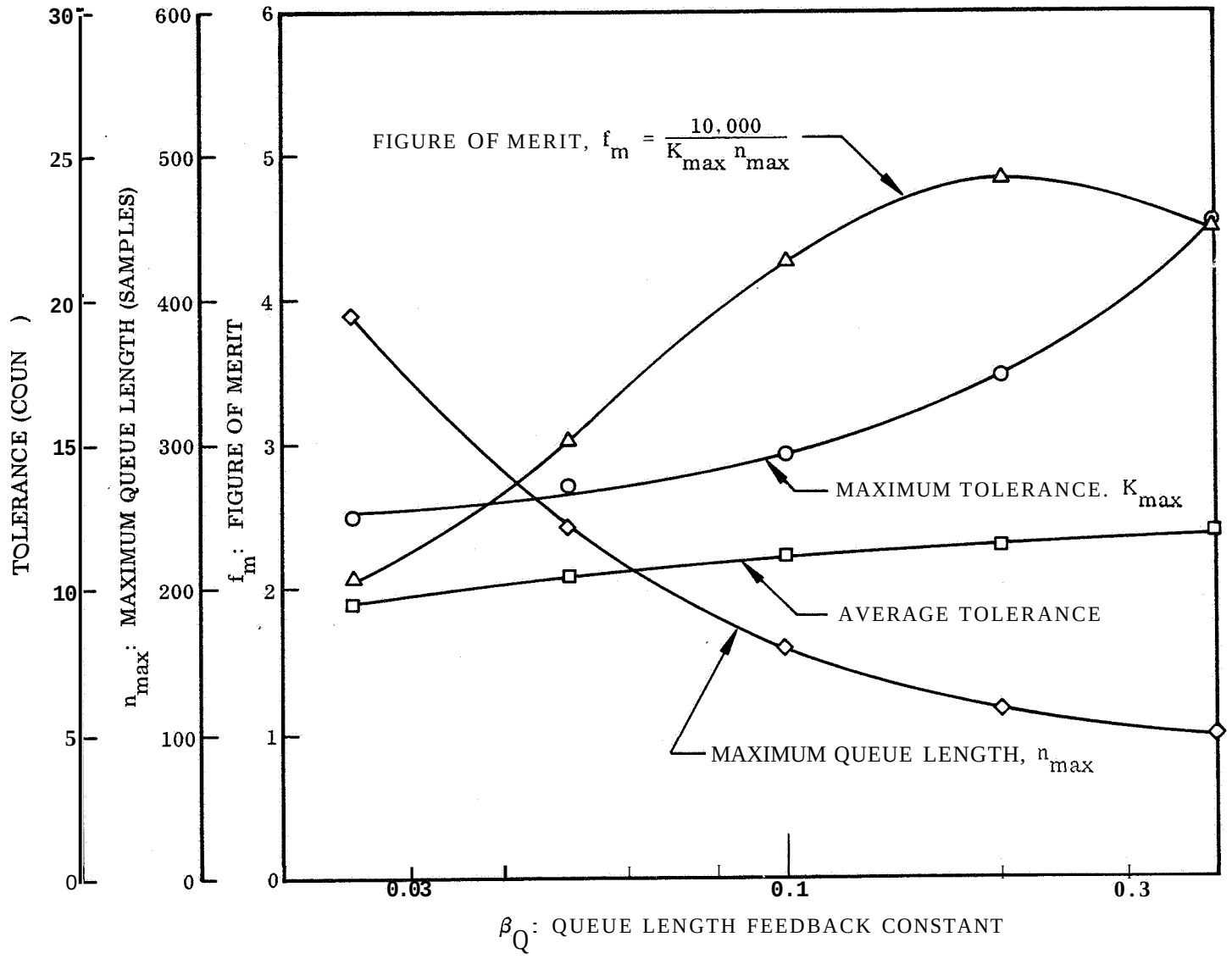
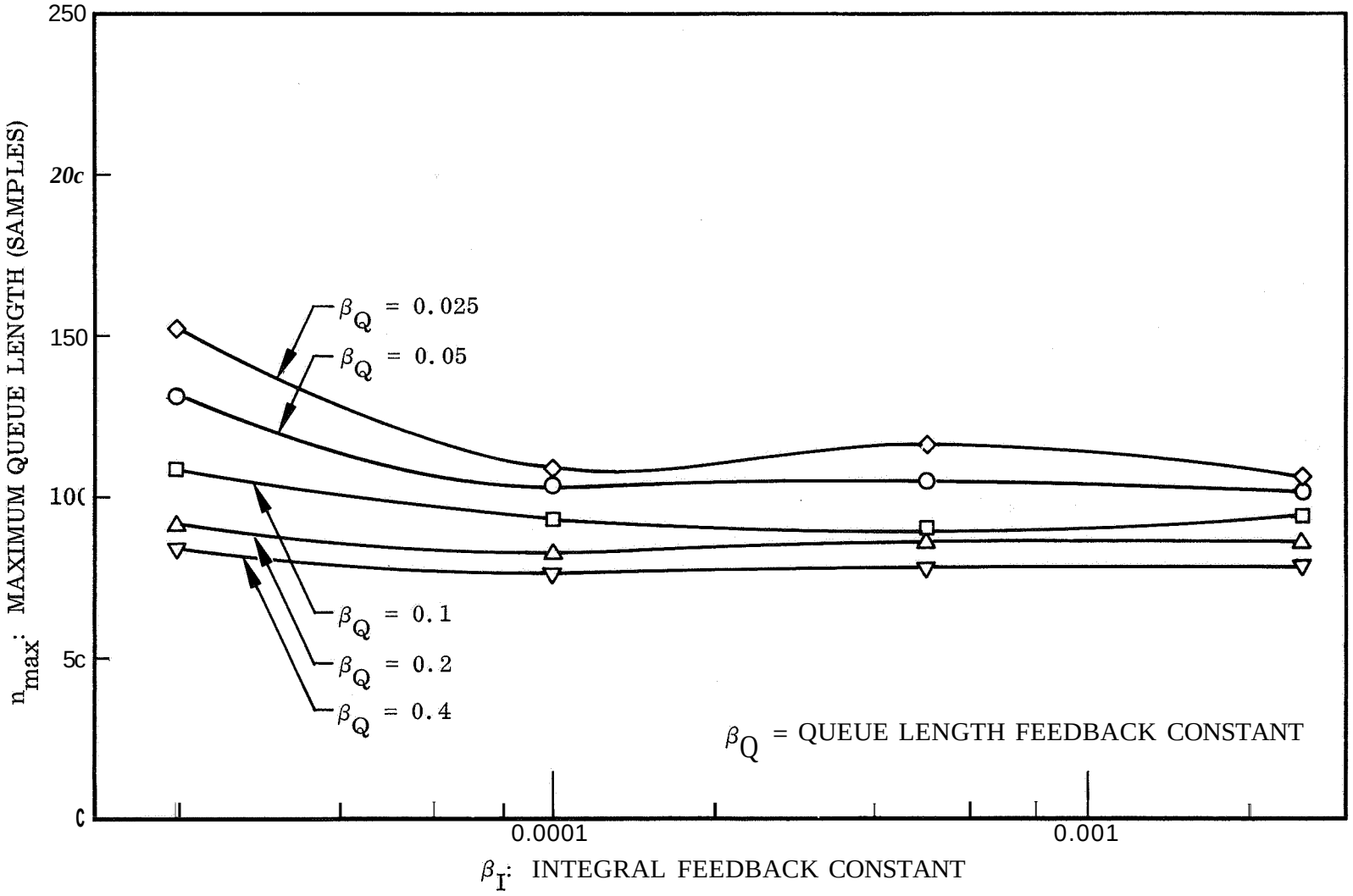


Fig. 3-10 Control System Behavior Without Integral Feedback



Fig, 3-11 Maximum Queue Length Variations, With Integral Feedback Added

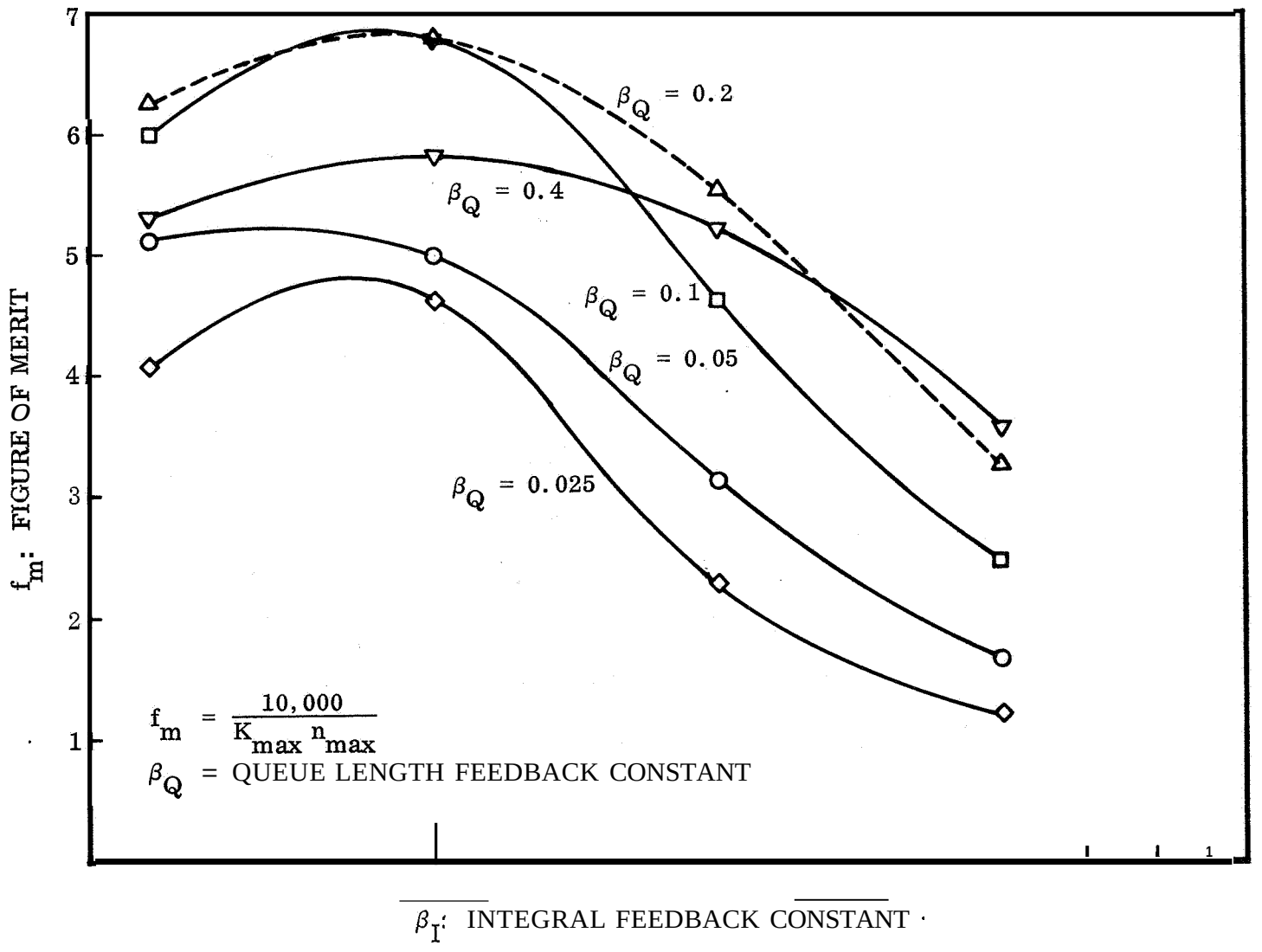


Fig. 3-12 Figure-of-Merit Variations, With Integral Feedback Added

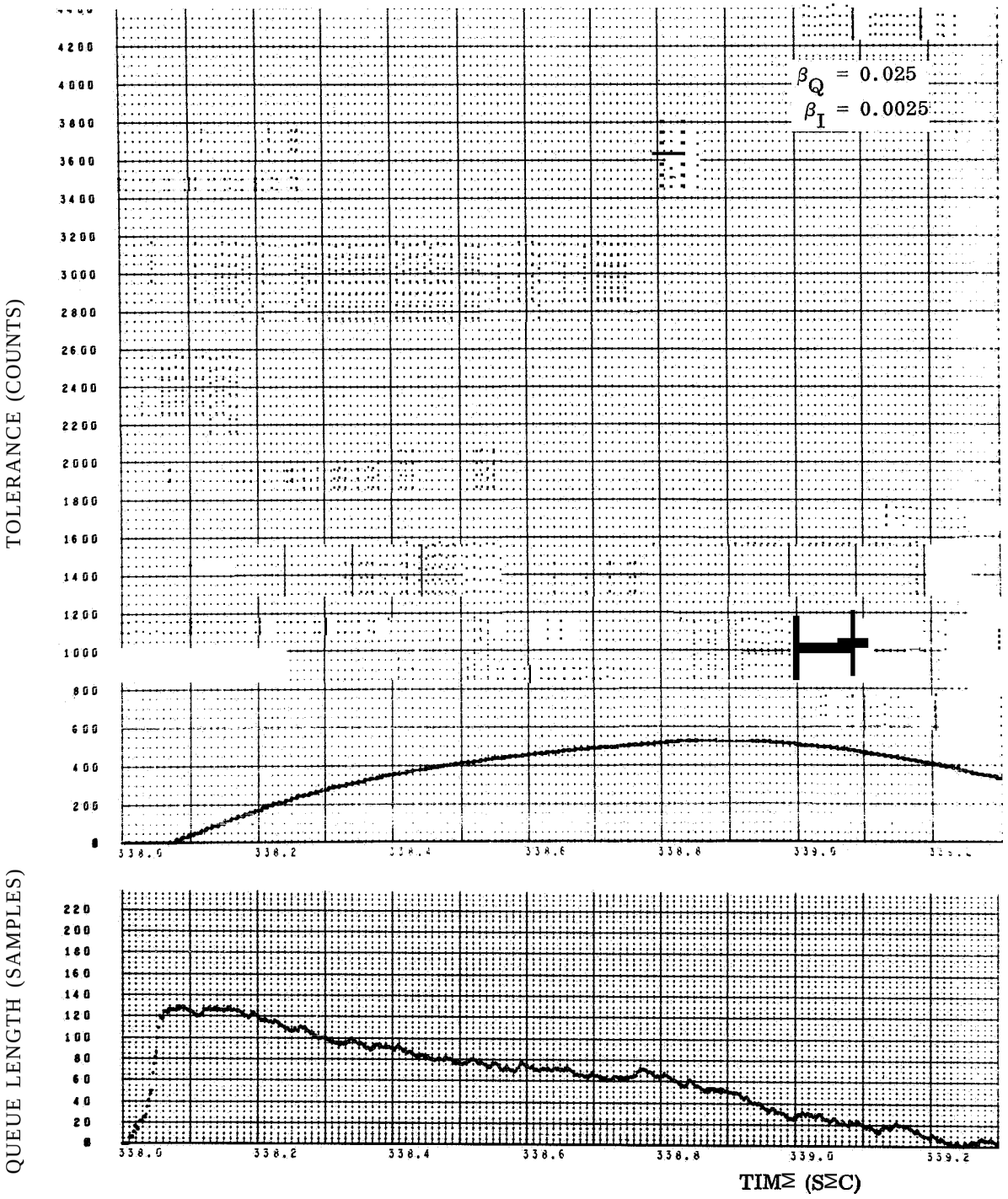


Fig. 3-13 Tolerance and Queue Length Behavior, Run S49-H

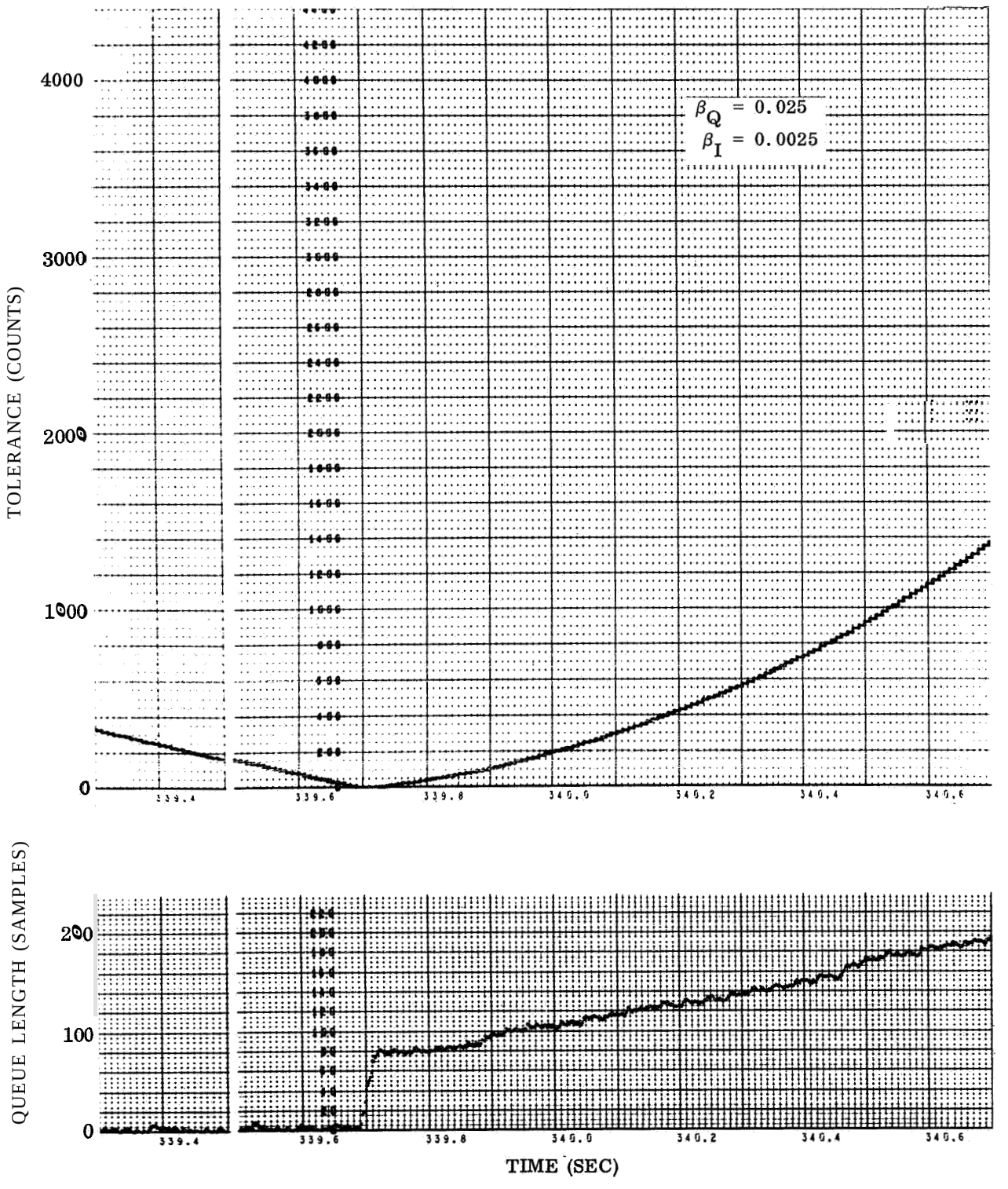


Fig. 3-13 Tolerance and Queue Length Behavior, Run S49-H (Cont'd)

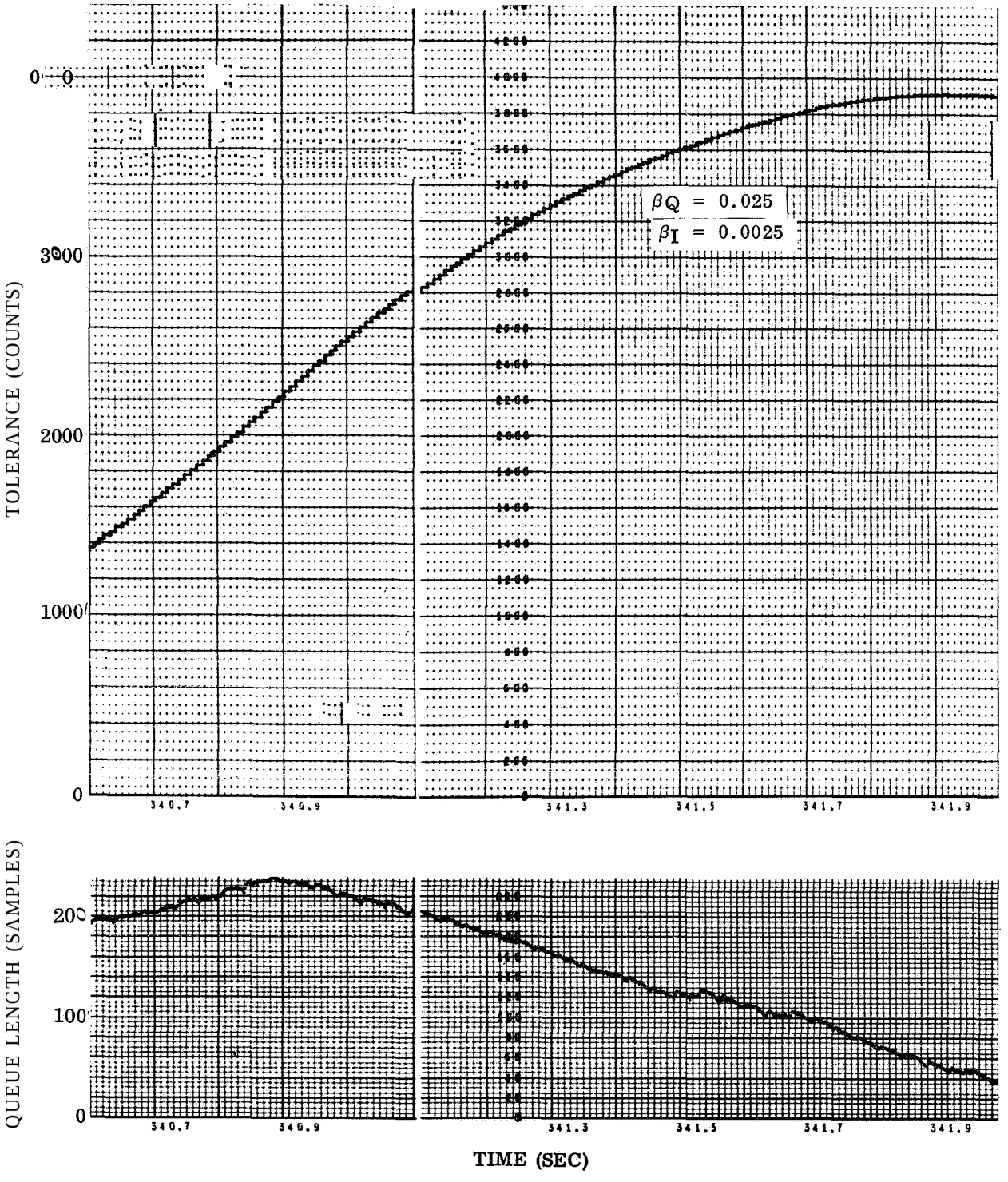
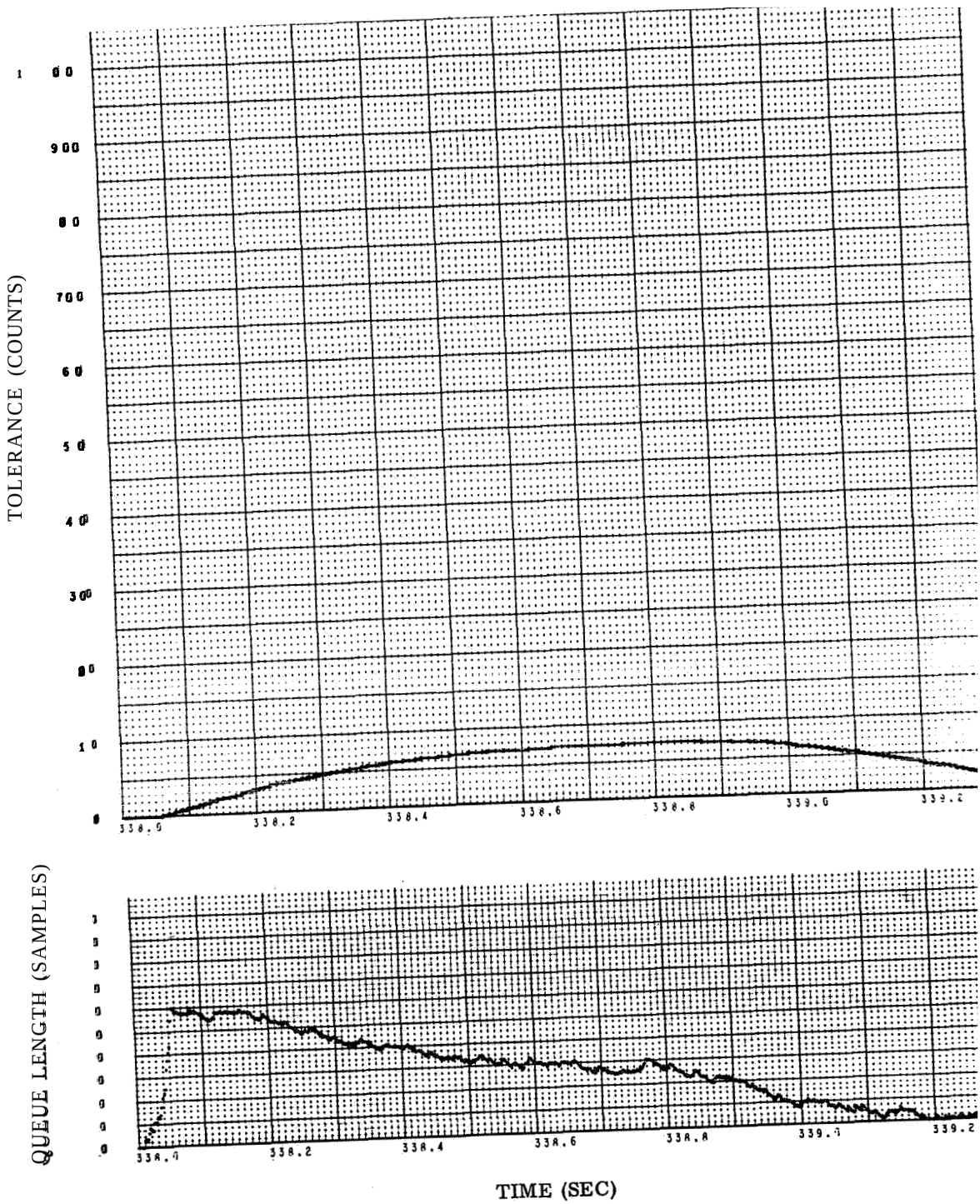


Fig. 3-13 Tolerance and Queue Length Behavior, Run S49-H (Cont'd)



Fi 3-14 Tolerance a Q ue Length Behavior, Run S -I

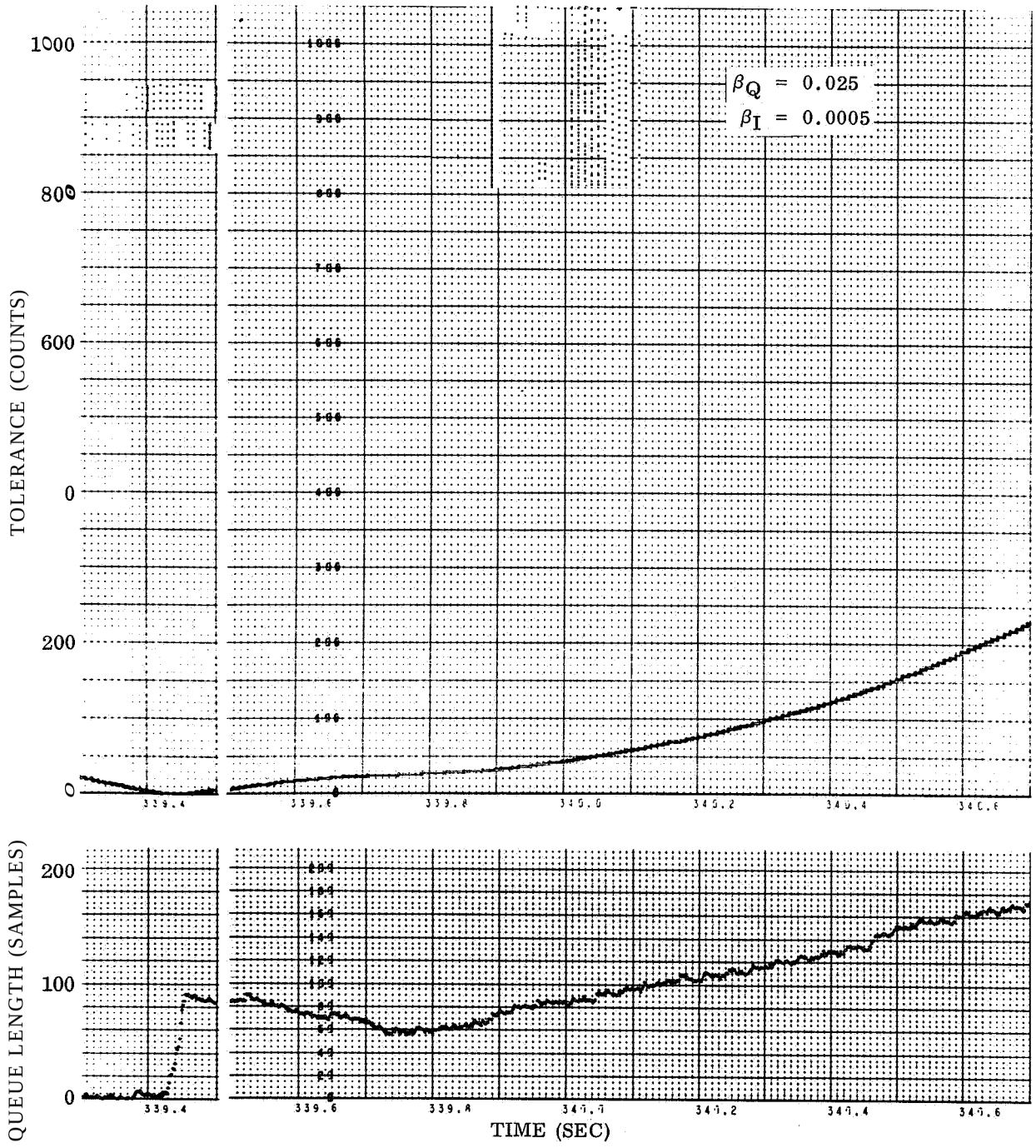


Fig. 3-14 Tolerance and Queue Engt h Behavior, Run S49-I (Cont'd)

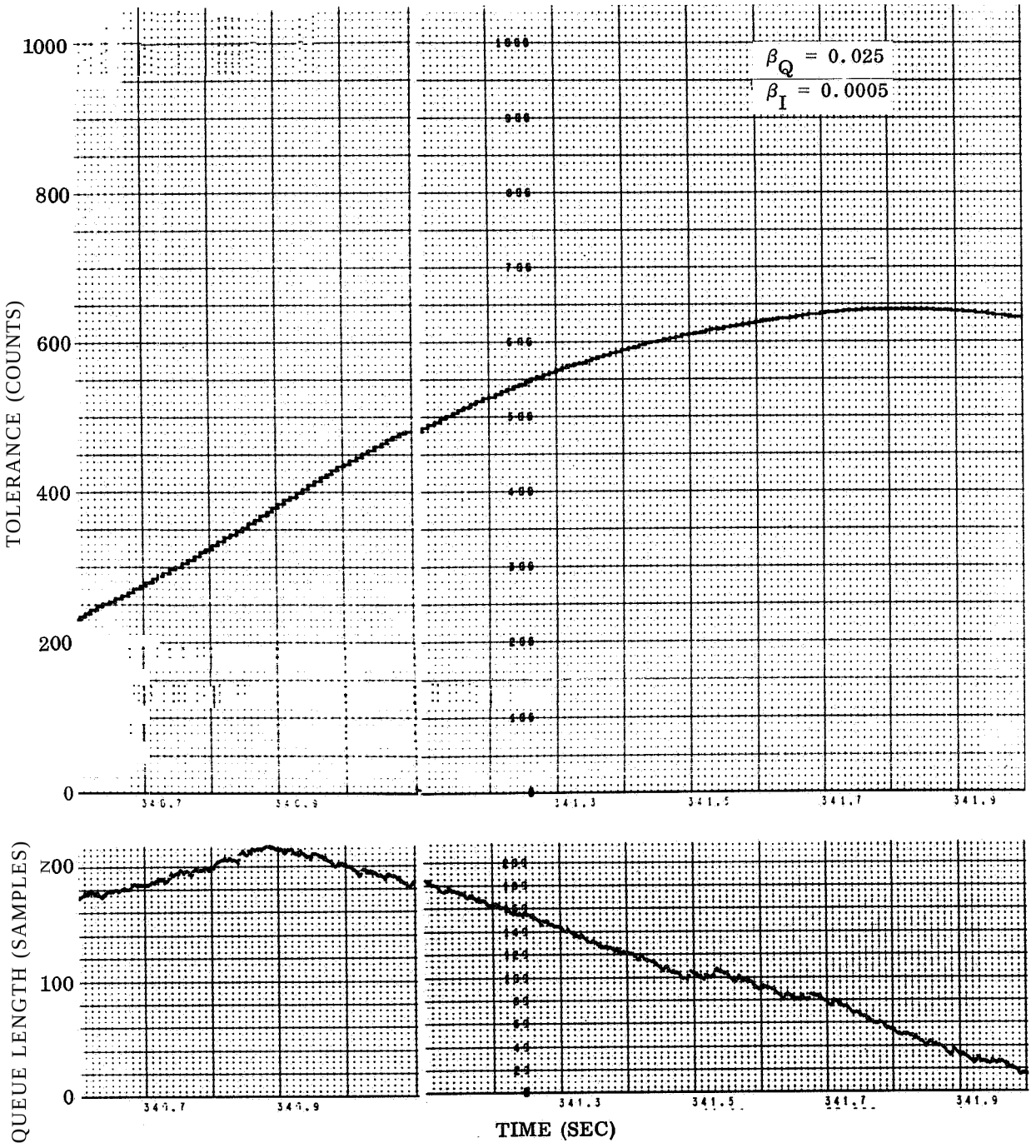


Fig. 3-14 Tolerance and Queue Length Behavior, Run S49-I (Cont'd)

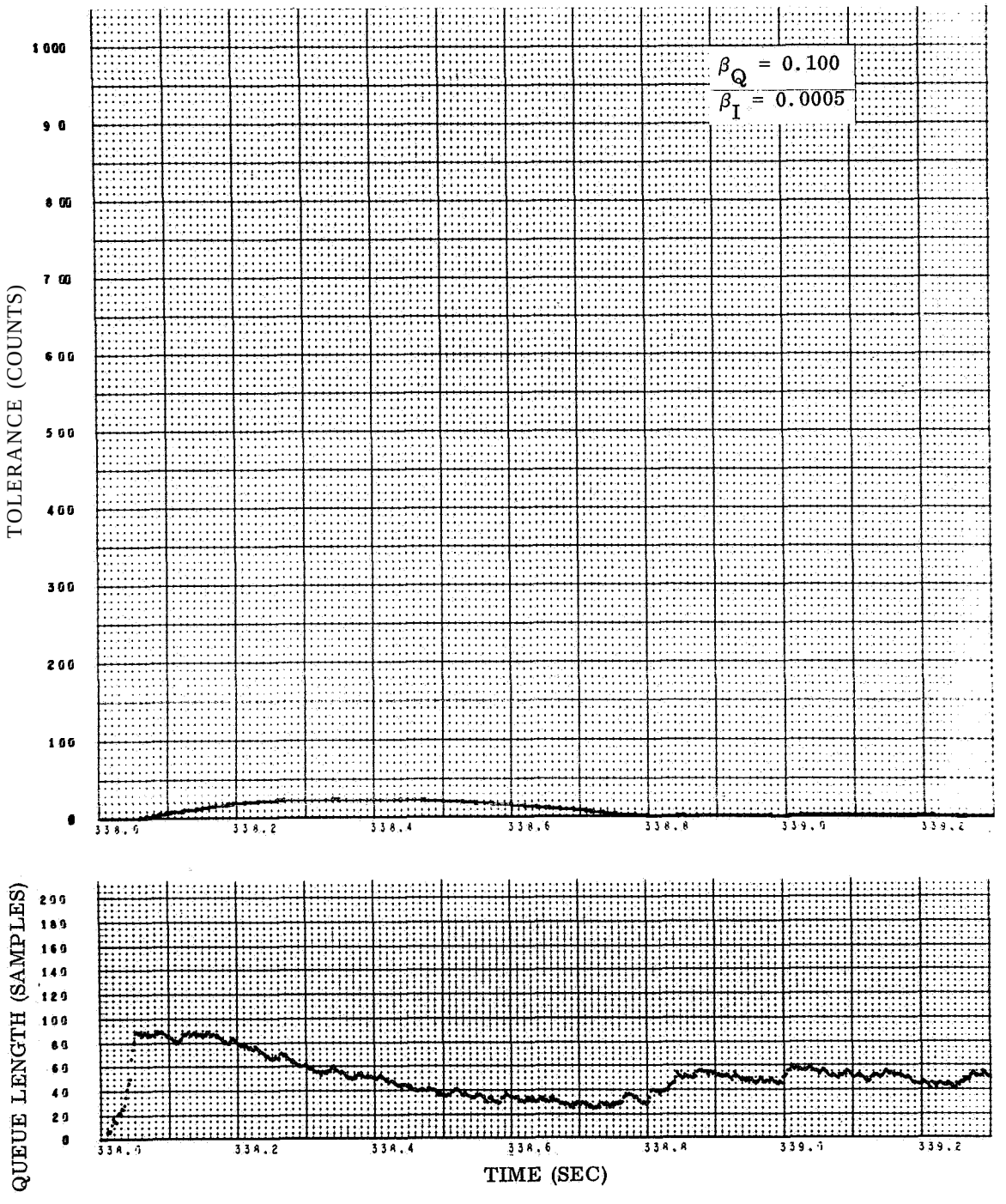


Fig. 3-15 Tolerance and Queue Length Behavior, Run S49-J

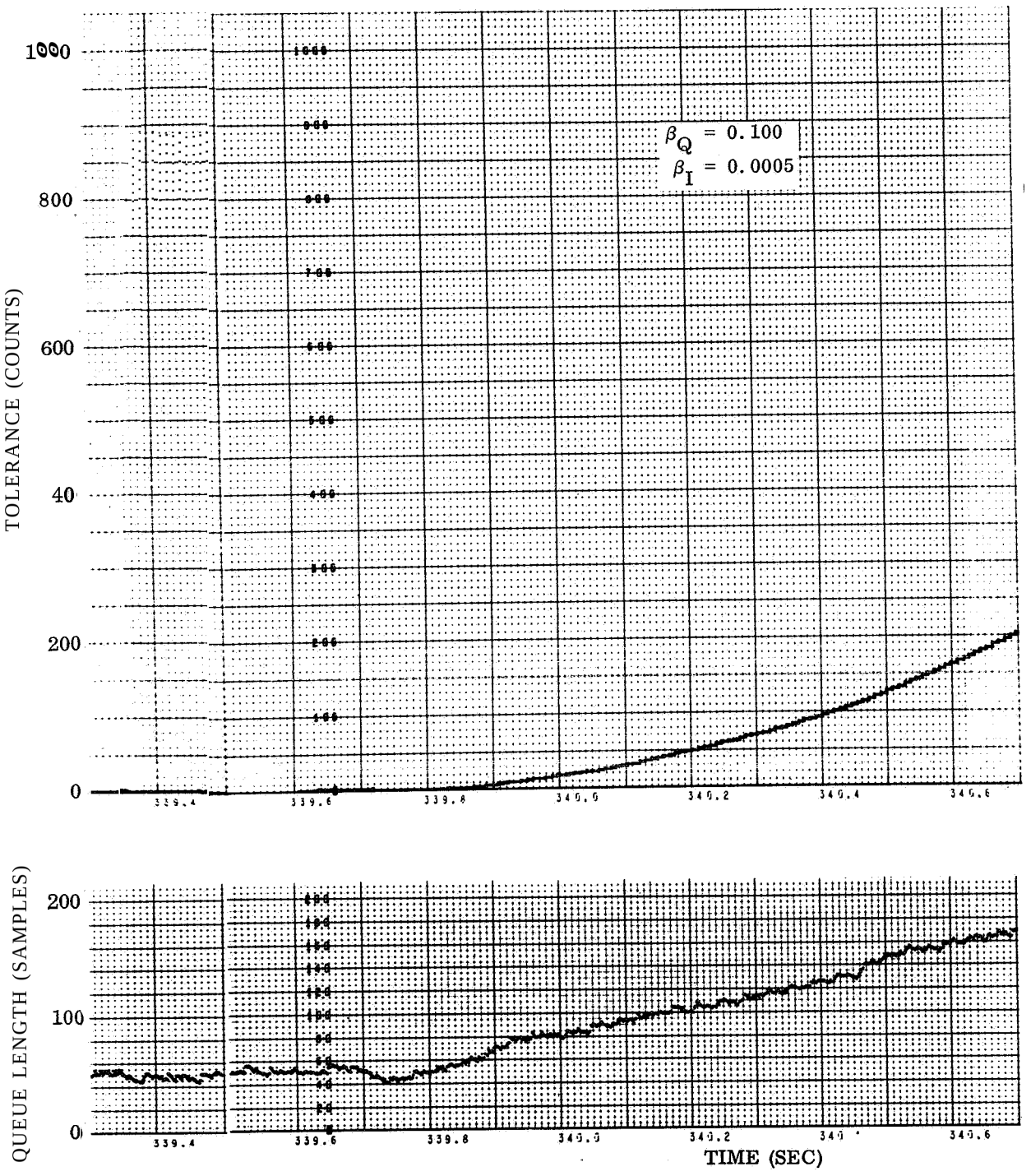


Fig. 3-15 Tolerance and Queue Length Behavior, Run S49-J (Con d)

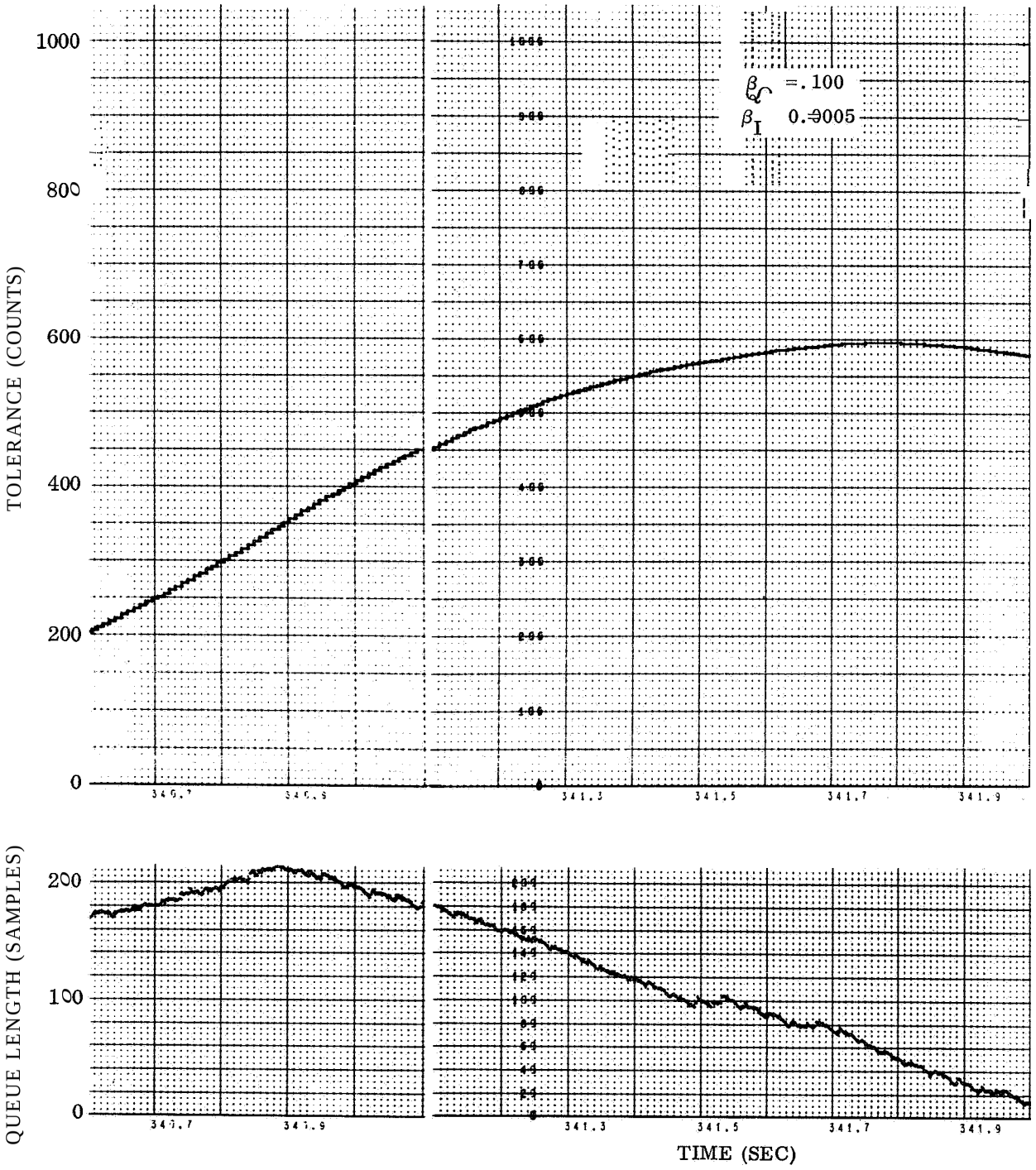
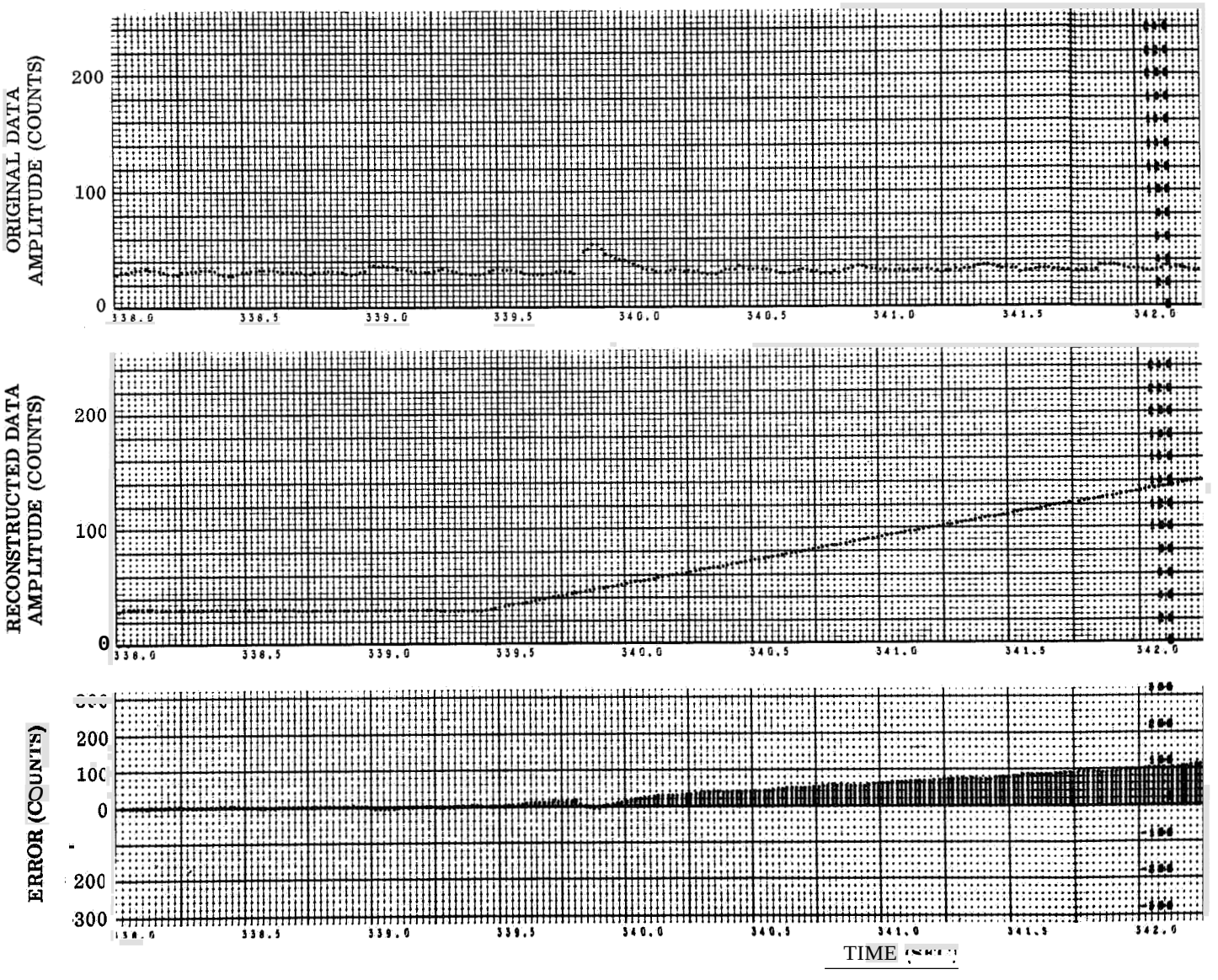


Fig. 3-15 Tolerance and Queue Lengh Behavior Run S49-J (Cont'd)



Fig, 3–16 Sensor **50** Reconstructed Data, Run S49-I: &ueue/Integr al-Queue **Control**

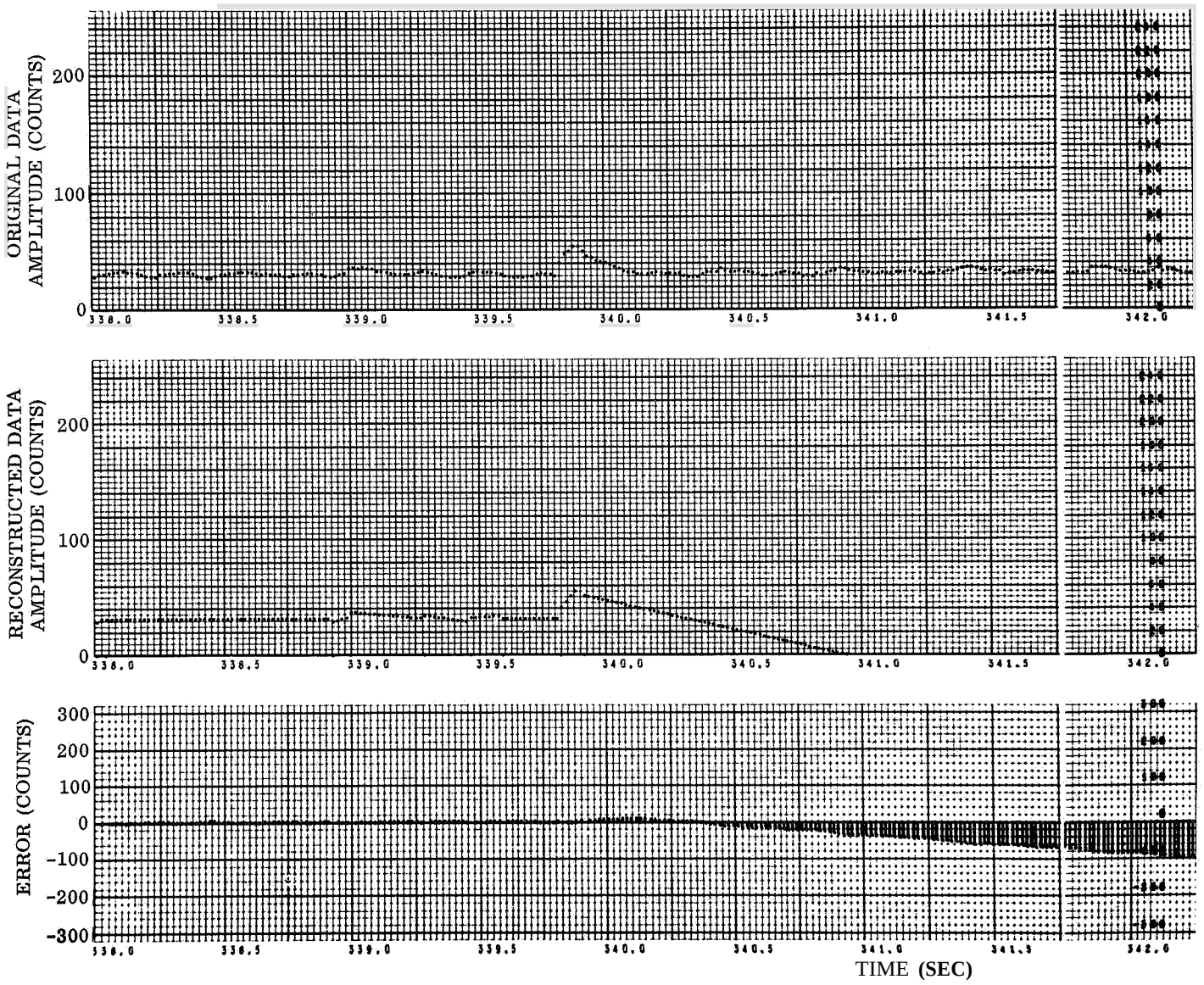
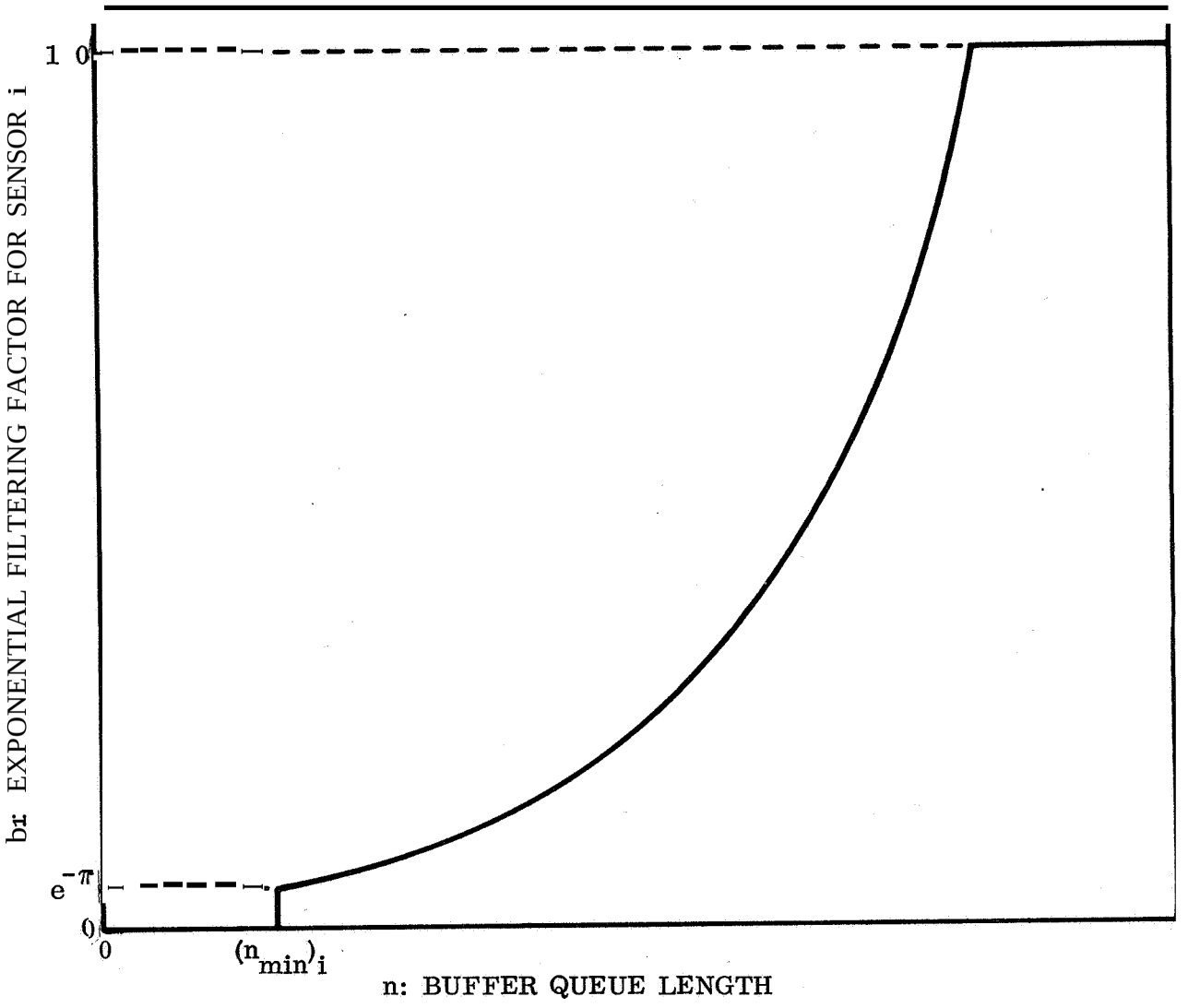


Fig. 3-17 Sensor 50 Reconstructed Data, Run S49-J: Queue/Integral-Queue Control

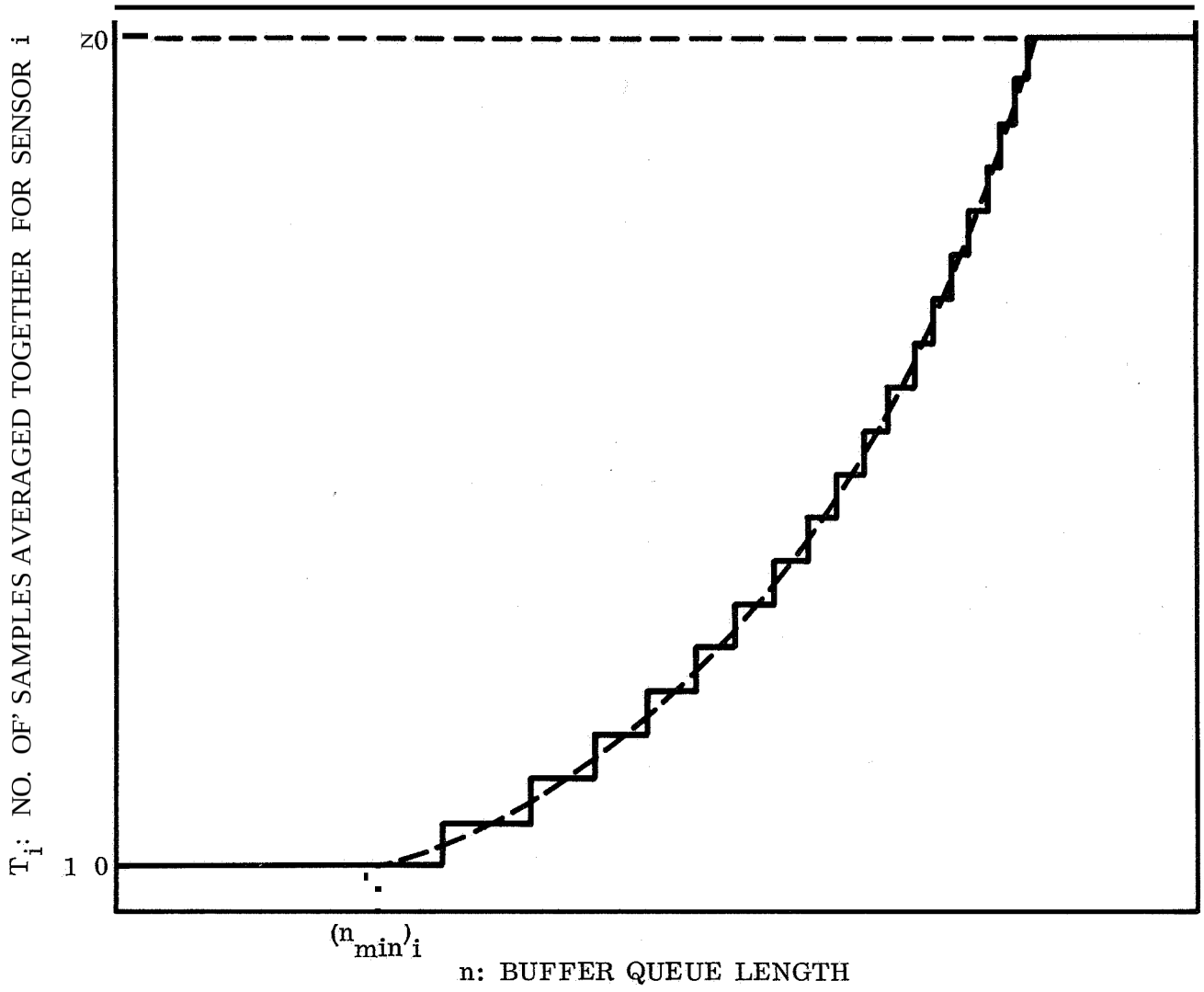
LOCKHEED PALO ALTO RESEARCH LABORATORY
LOCKHEED MISSILES & SPACE COMPANY
A GROUP DIVISION OF LOCKHEED AIRCRAFT CORPORATION



$(n_{min})_i$ = ASSIGNED THRESHOLD VALUE OF QUEUE LENGTH FOR SENSOR i

NOTE: CURVE IS NOT DRAWN TO SCALE.

Fig. 3-18 Exponential Filtering Control Function



$(n_{min})_i$ = ASSIGNED THRESHOLD VALUE OF QUEUE LENGTH FOR SENSOR i

NOTE: CURVE IS NOT DRAWN TO SCALE.

F 3-19 Running-Average Filtering Control Function

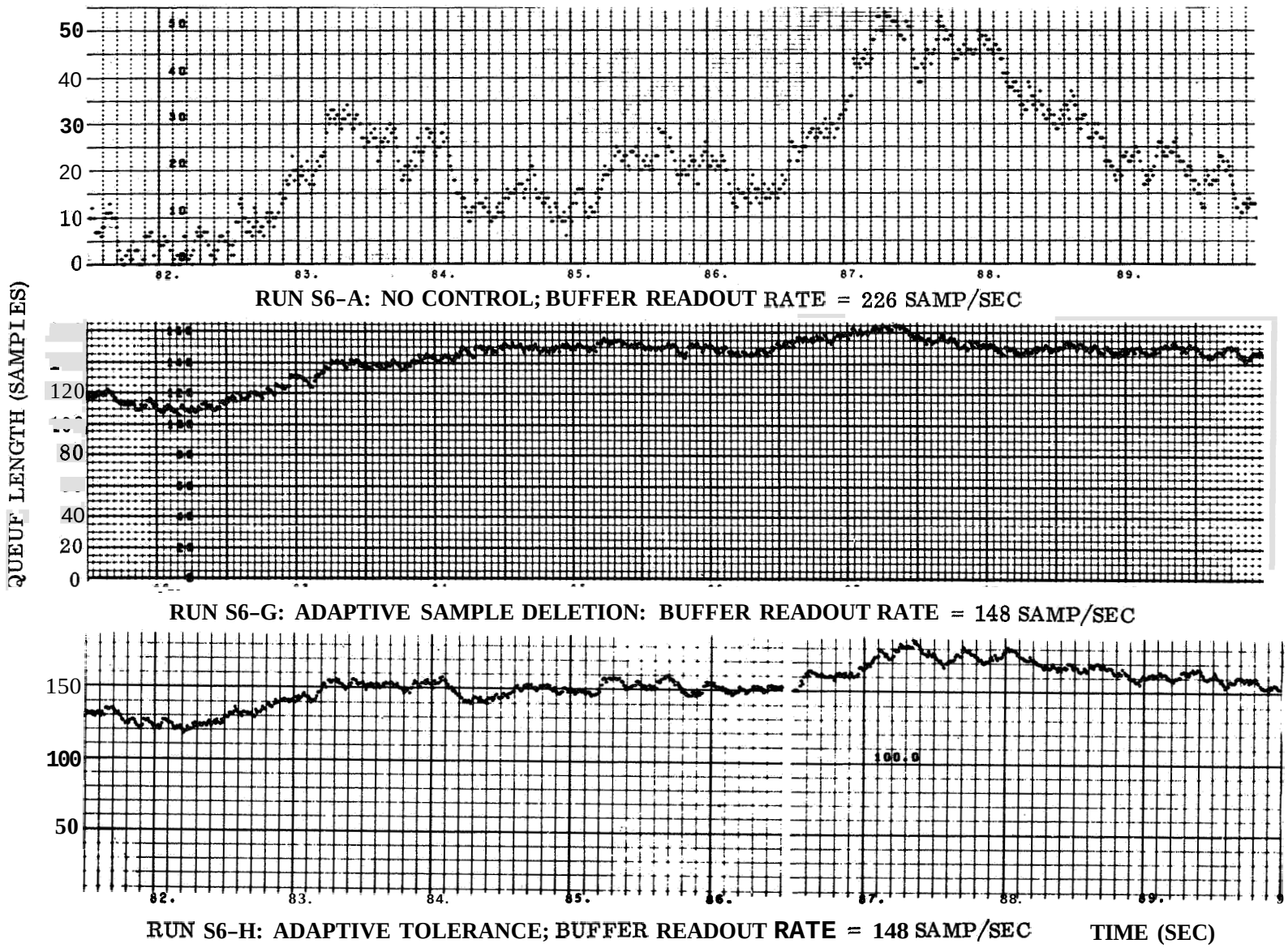


Fig. 3-20 Queue Length Time Histories, Runs S6-A, S6-G, and S6-H

LOCKHEED PALO ALTO RESEARCH LABORATORY
LOCKHEED MISSILES & SPACE COMPANY
A GROUP DIVISION OF LOCKHEED AIRCRAFT CORPORATION

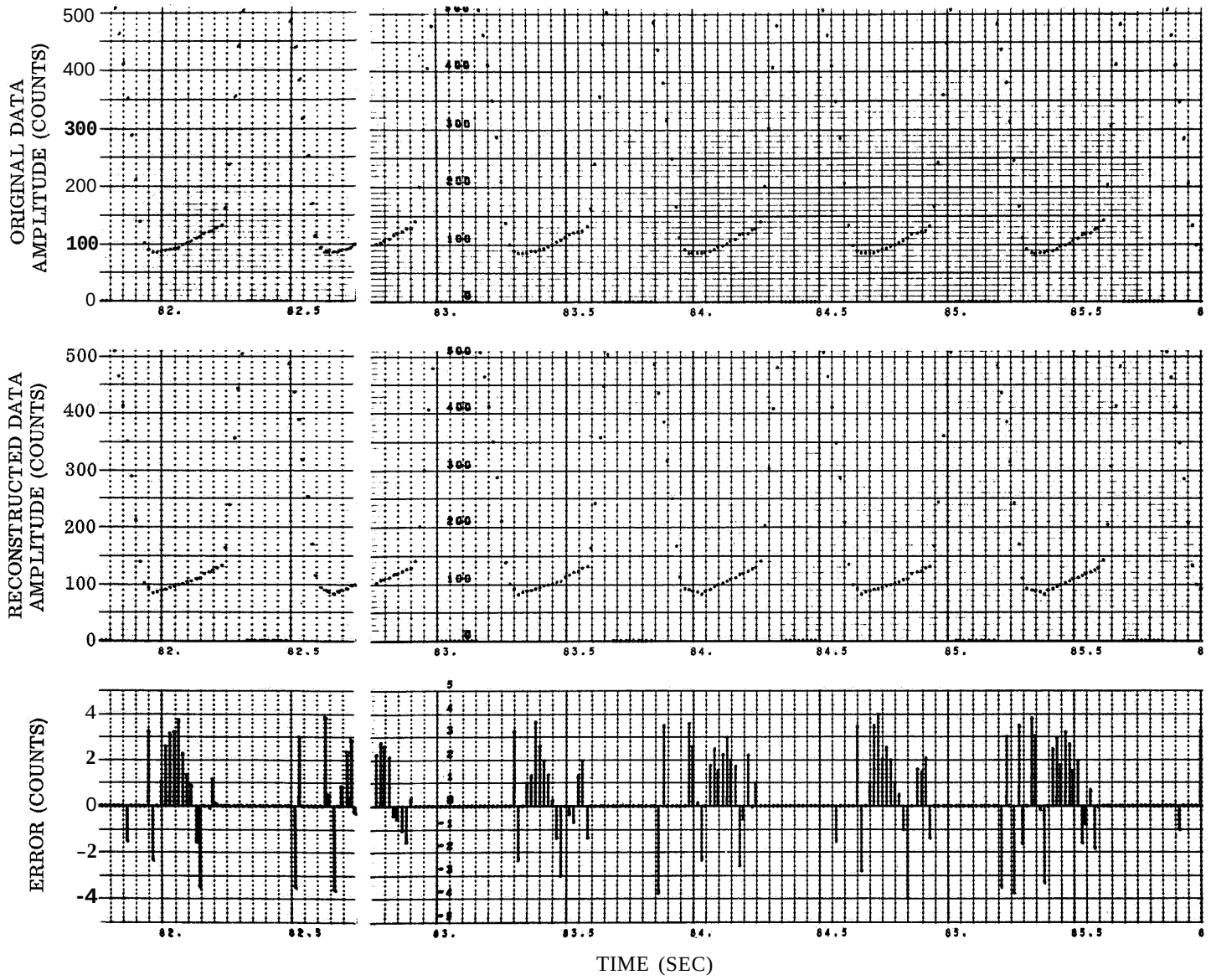


Fig. **3-21** Sensor **7** Reconstructed Data, Run **S6-A:** No Filtering or Queuing Control

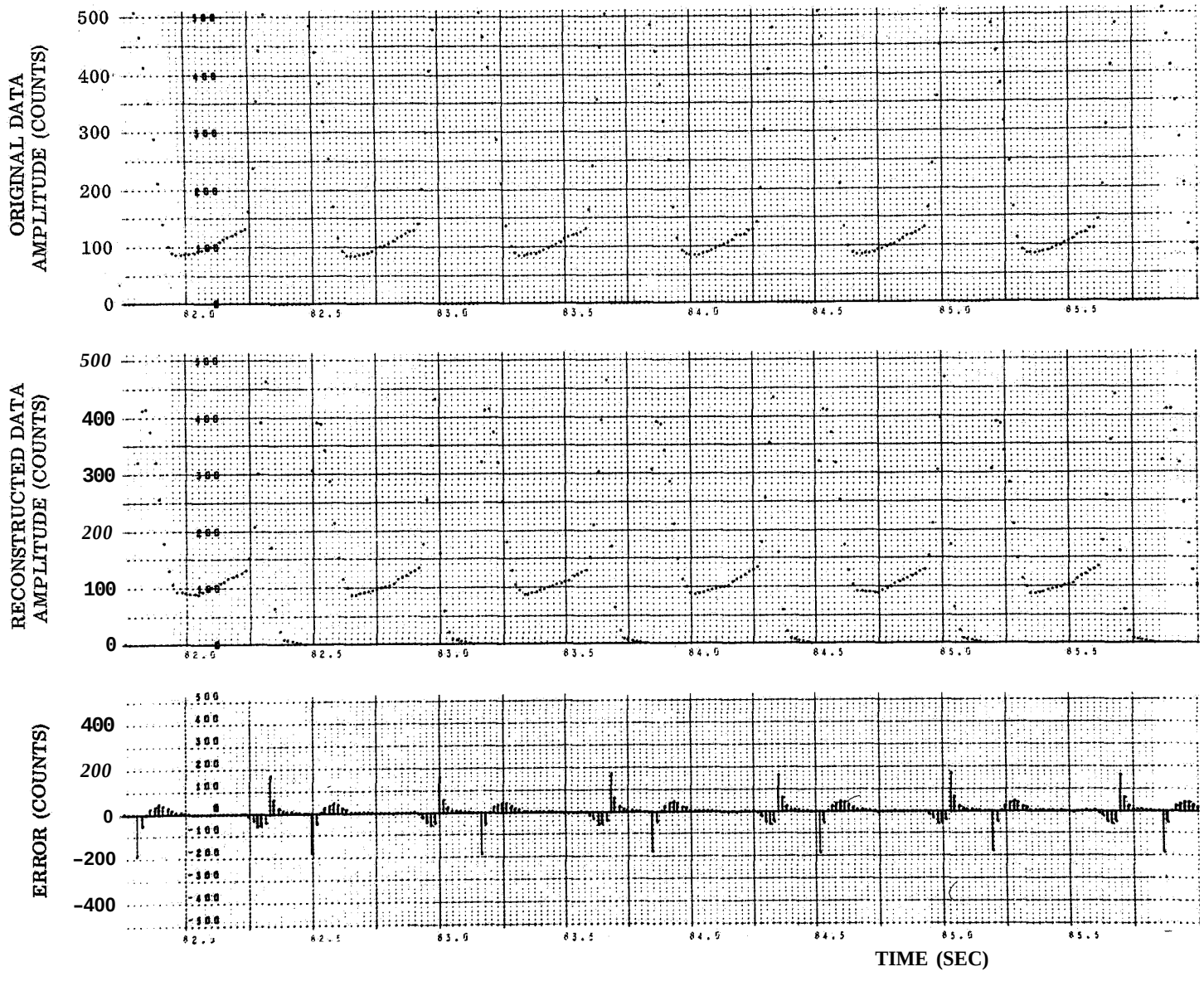


Fig. 3-22 Sensor 7 Reconstructed Data, Run S6-B: Exponential Filtering, Fixed-Parameter

LOCKHEED PALO ALTO RESEARCH LABORATORY
LOCKHEED MISSILES & SPACE COMPANY
A GROUP DIVISION OF LOCKHEED AIRCRAFT CORPORATION

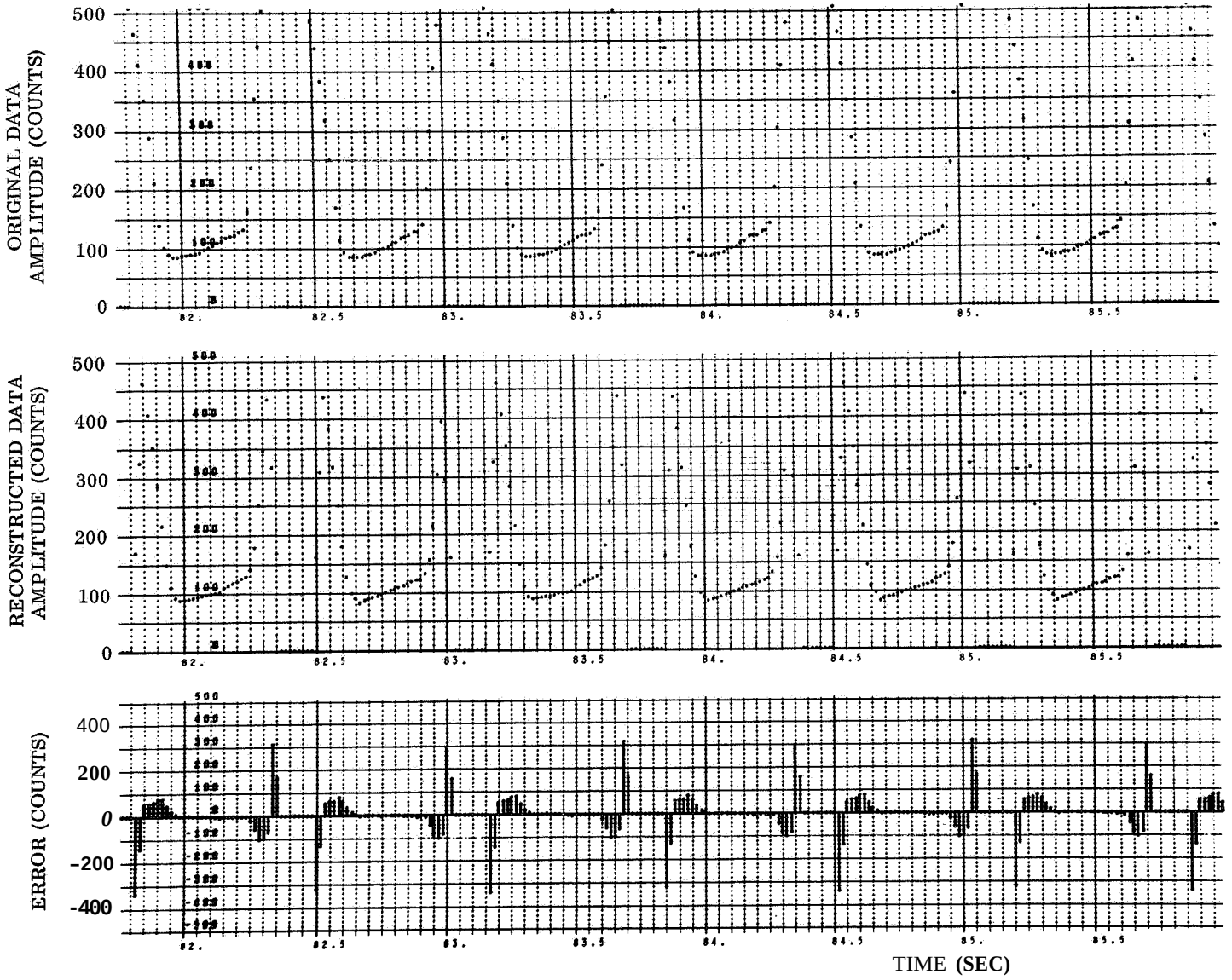


Fig. 3-23 Sensor 7 Reconstructed Data, Run S6-C: Running-Average Filtering, Fixed-Parameter

LOCKHEED PALO ALTO RESEARCH LABORATORY
LOCKHEED MISSILES & SPACE COMPANY
A GROUP DIVISION OF LOCKHEED AIRCRAFT CORPORATION

Fig. 3-24 Sensor 7 Reconstructed Data, Run S6-E: Sample Deletion Only, Fixed-Parameter

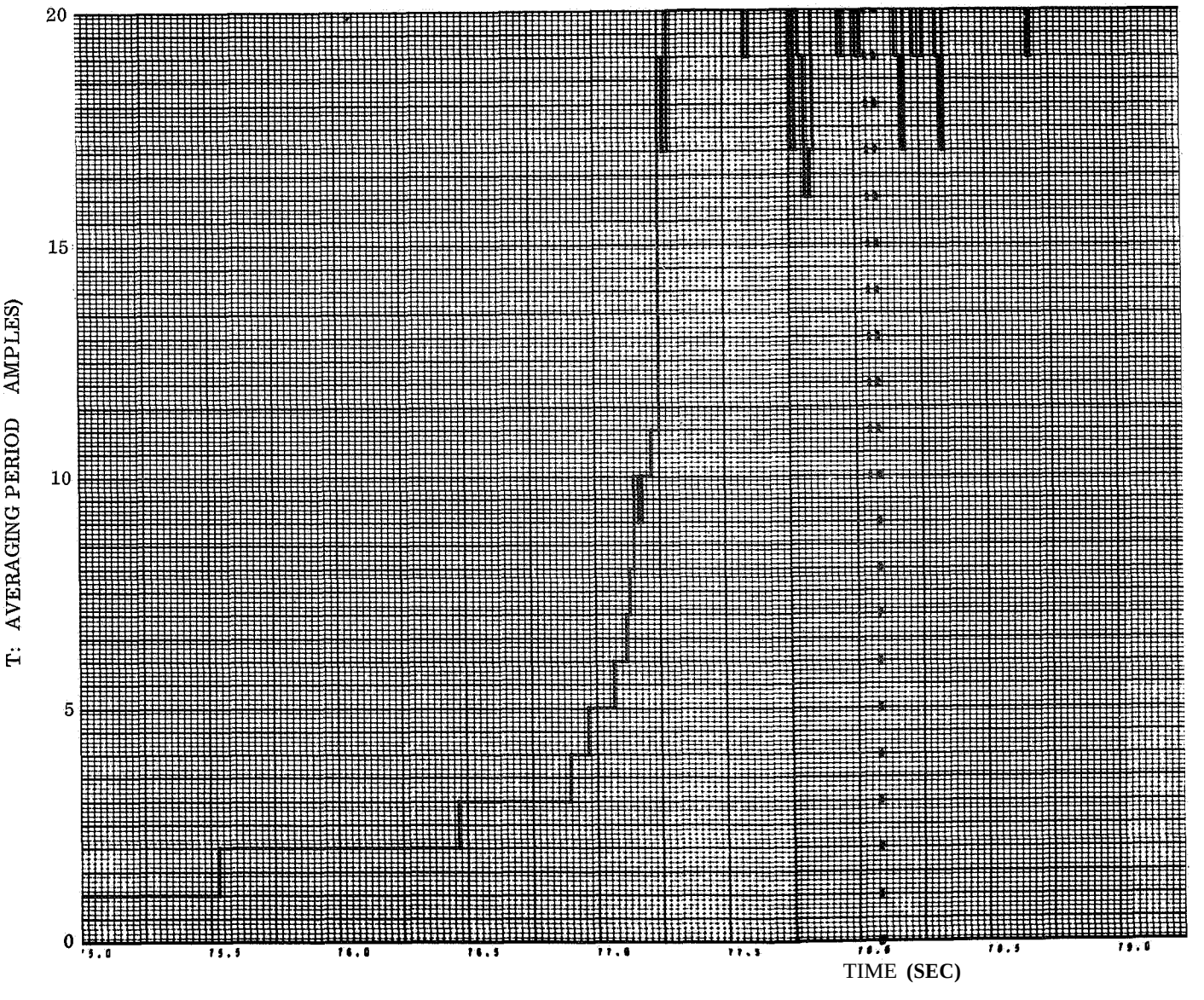
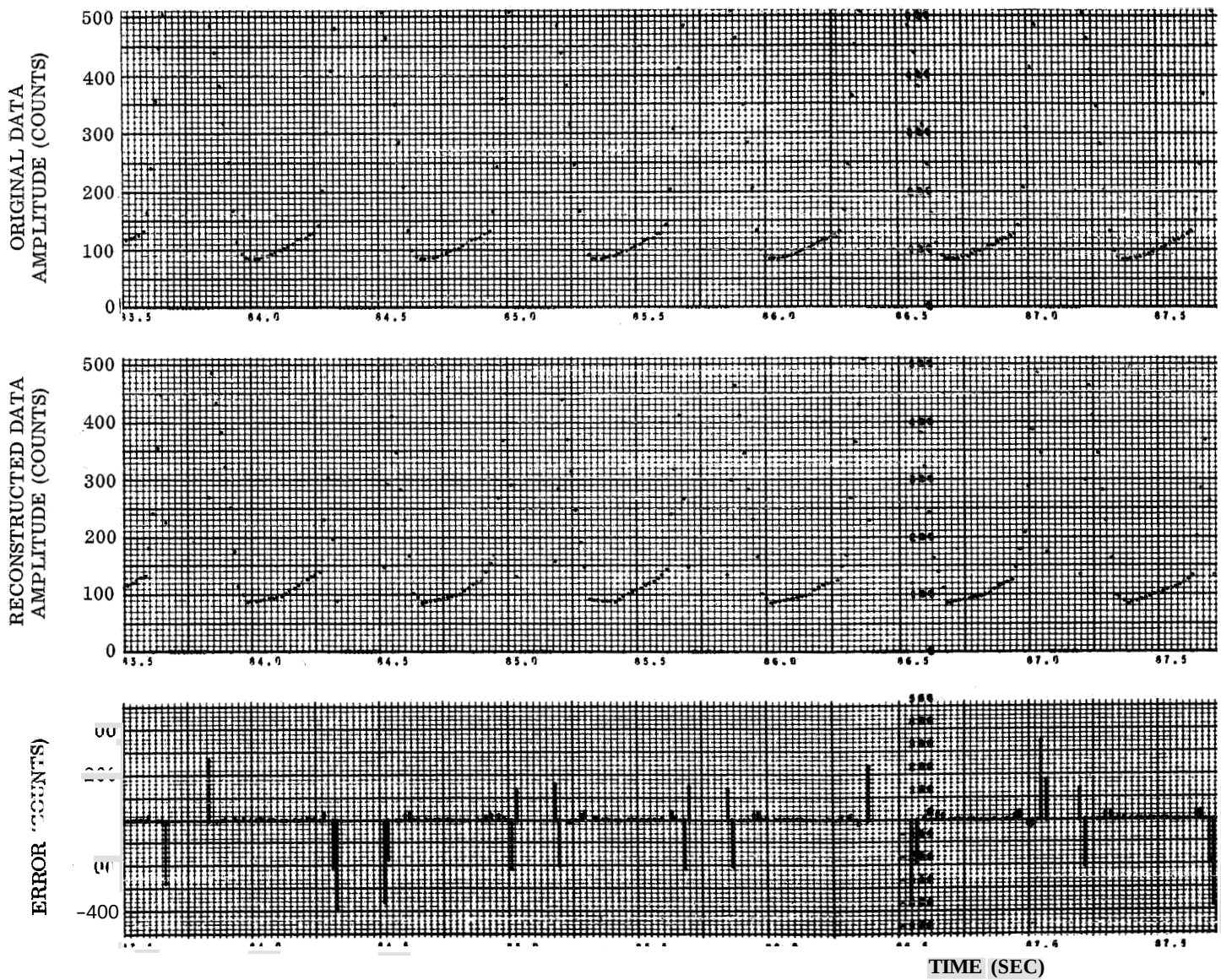


Fig. 3-25 Averaging Period Variations, Run S6-F: Adaptive Running-Average Filtering

LOCKHEED MISSILES & SPACE COMPANY



Fig, **3-26** Sensor **7** Reconstructed Data, Run S6-G: Adaptive Sample Deletion

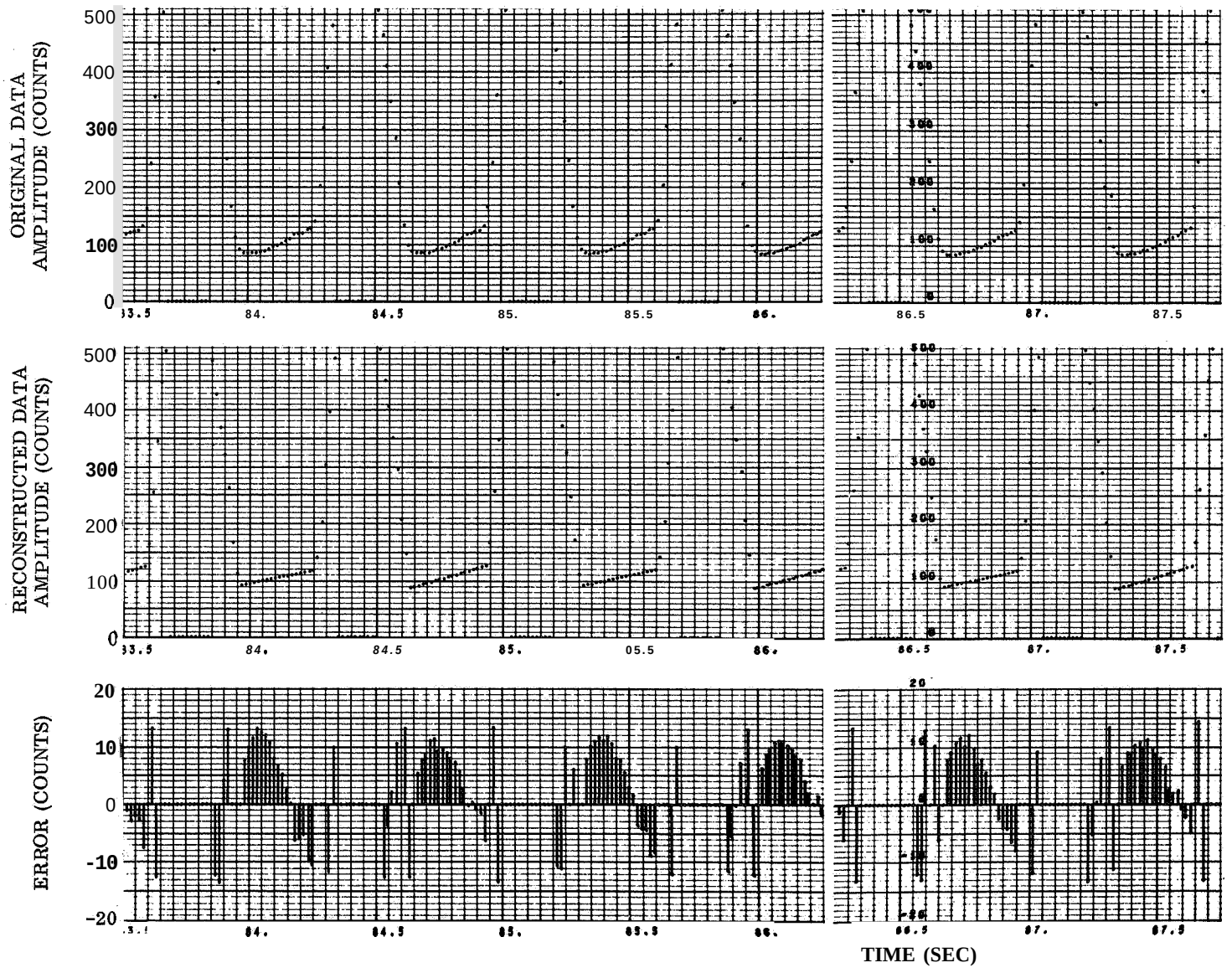


Fig. 3-27 Sensor 7 Reconstructed Data, Run S6-H: Adaptive Tolerance

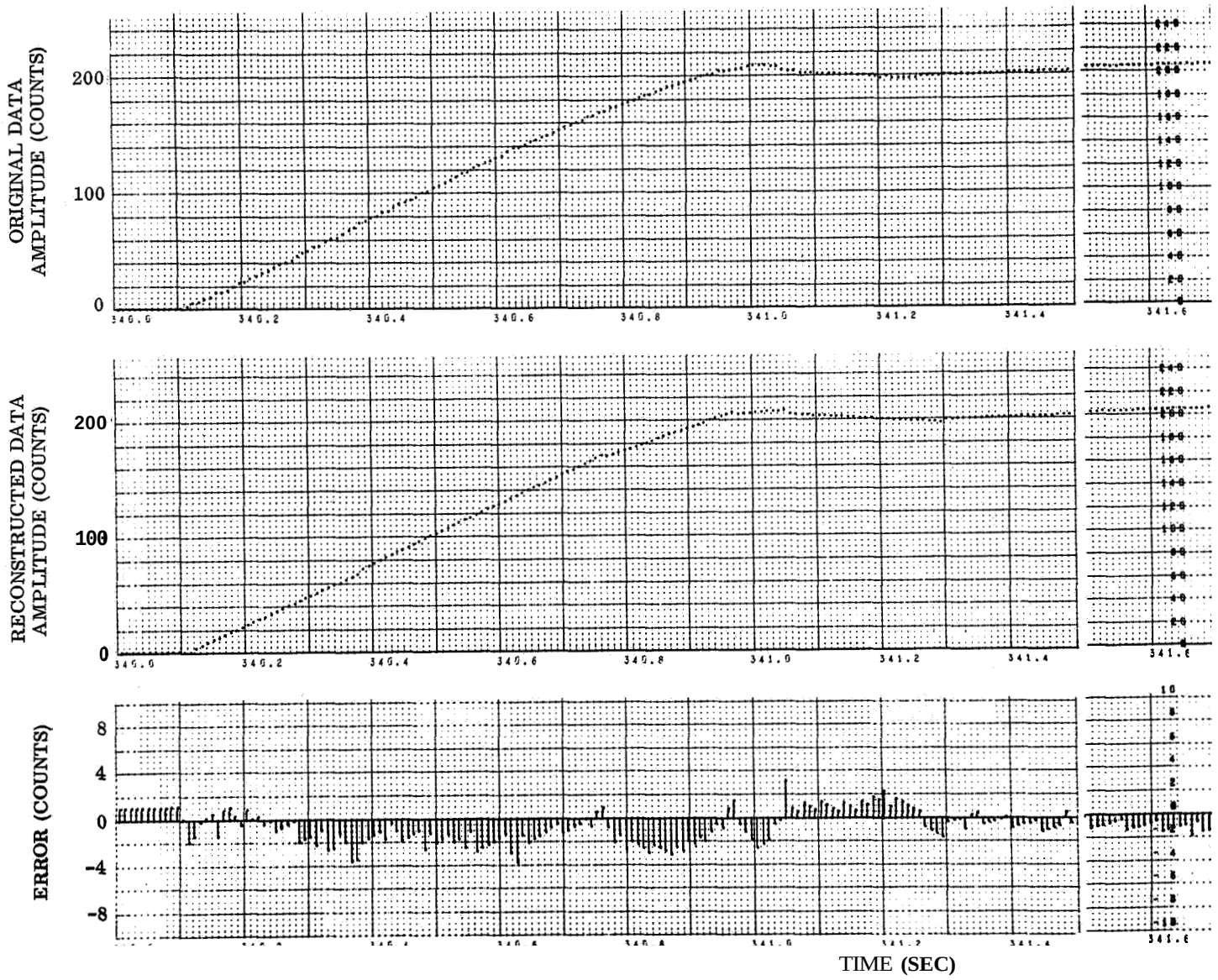


Fig. 3-28 Sensor 37 Reconstructed Data, Run S49-K: Exponential Filtering, Fixed-Parameter

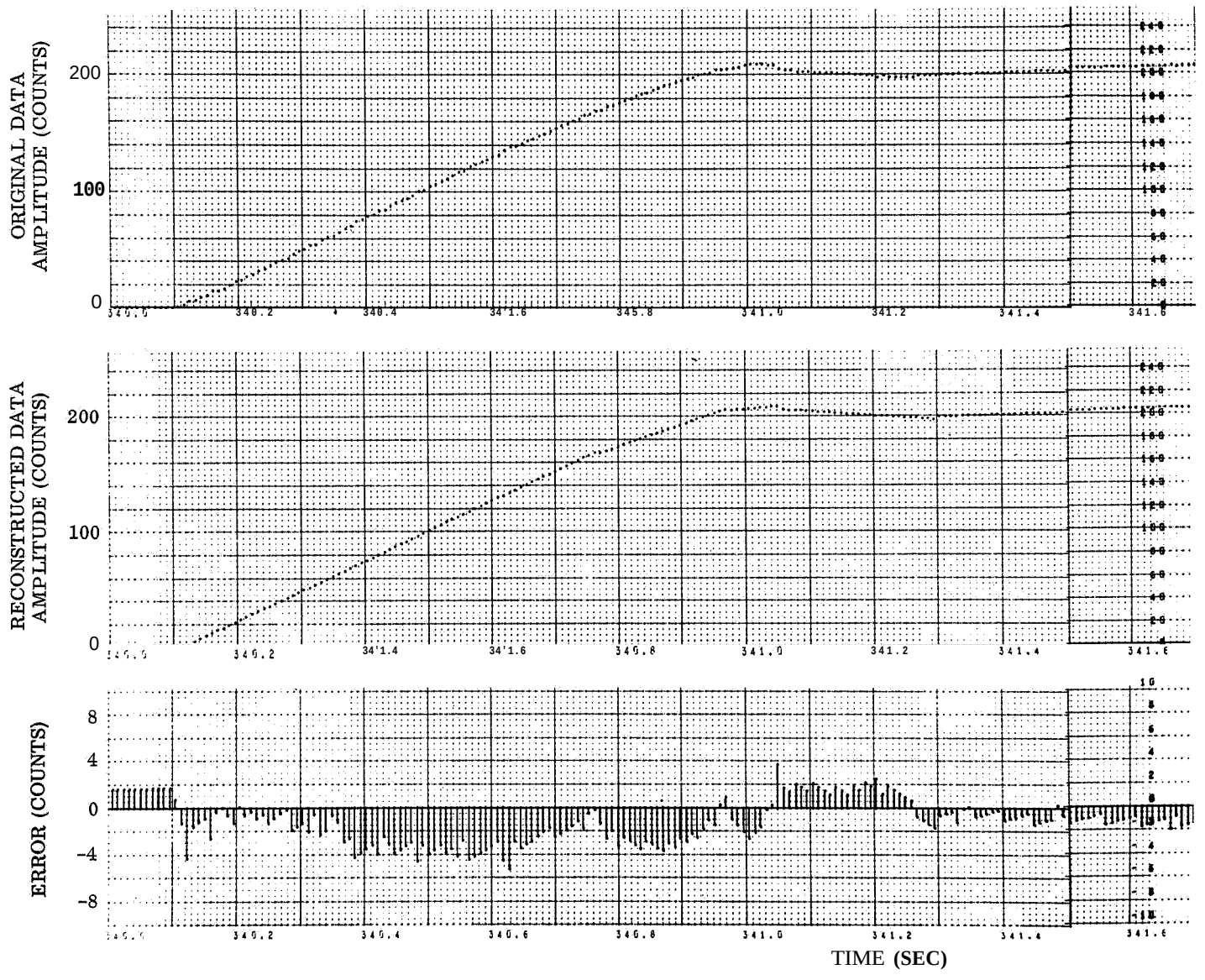


Fig. 3-29 Sensor 37 Reconstructed Data, Run S49-L: Running Average Filtering, Fixed-Parameter

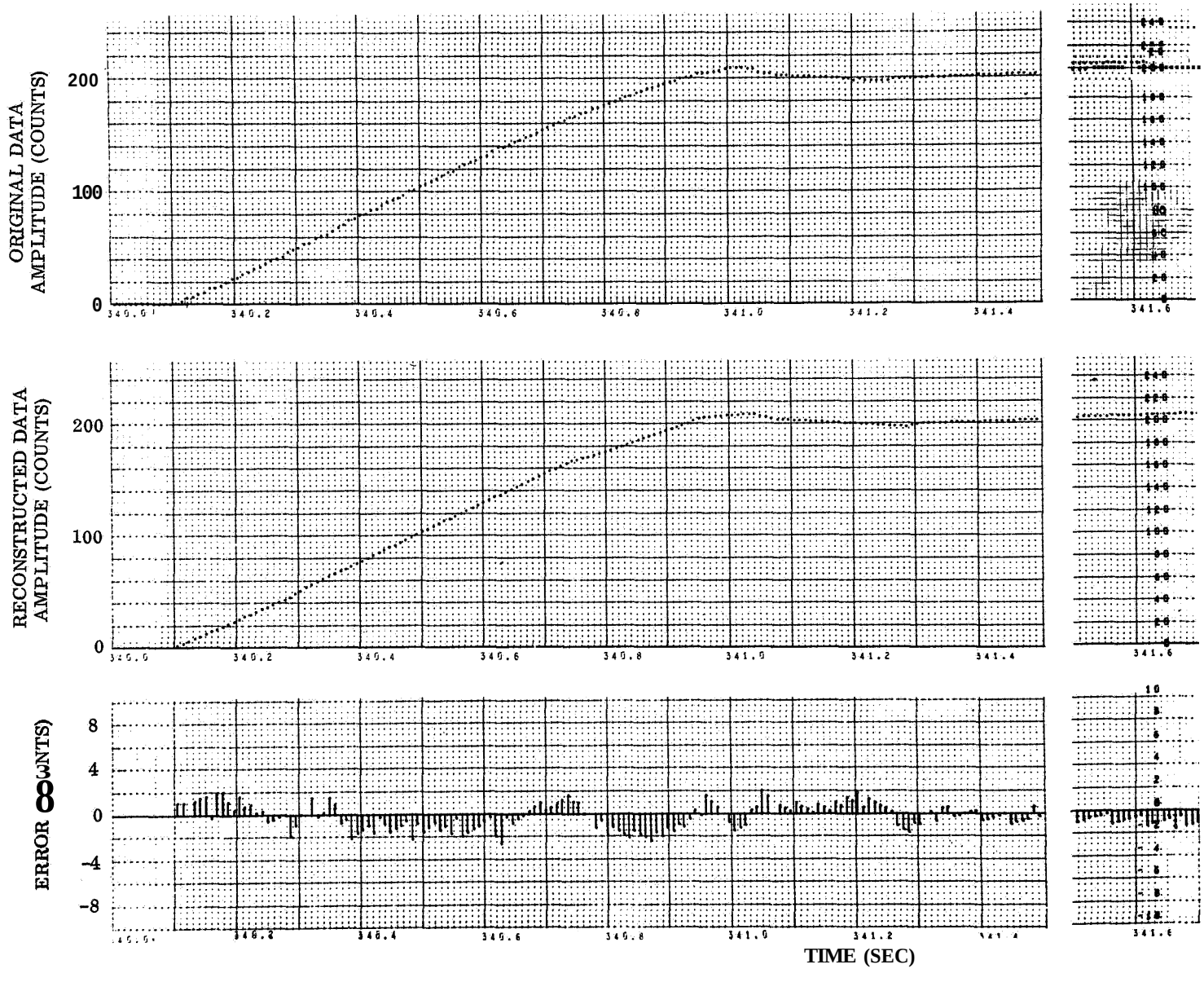


Fig. 3-30 Sensor 37 Reconstructed Data, Run S49-M: Sample Deletion Only, Fixed-Parameter

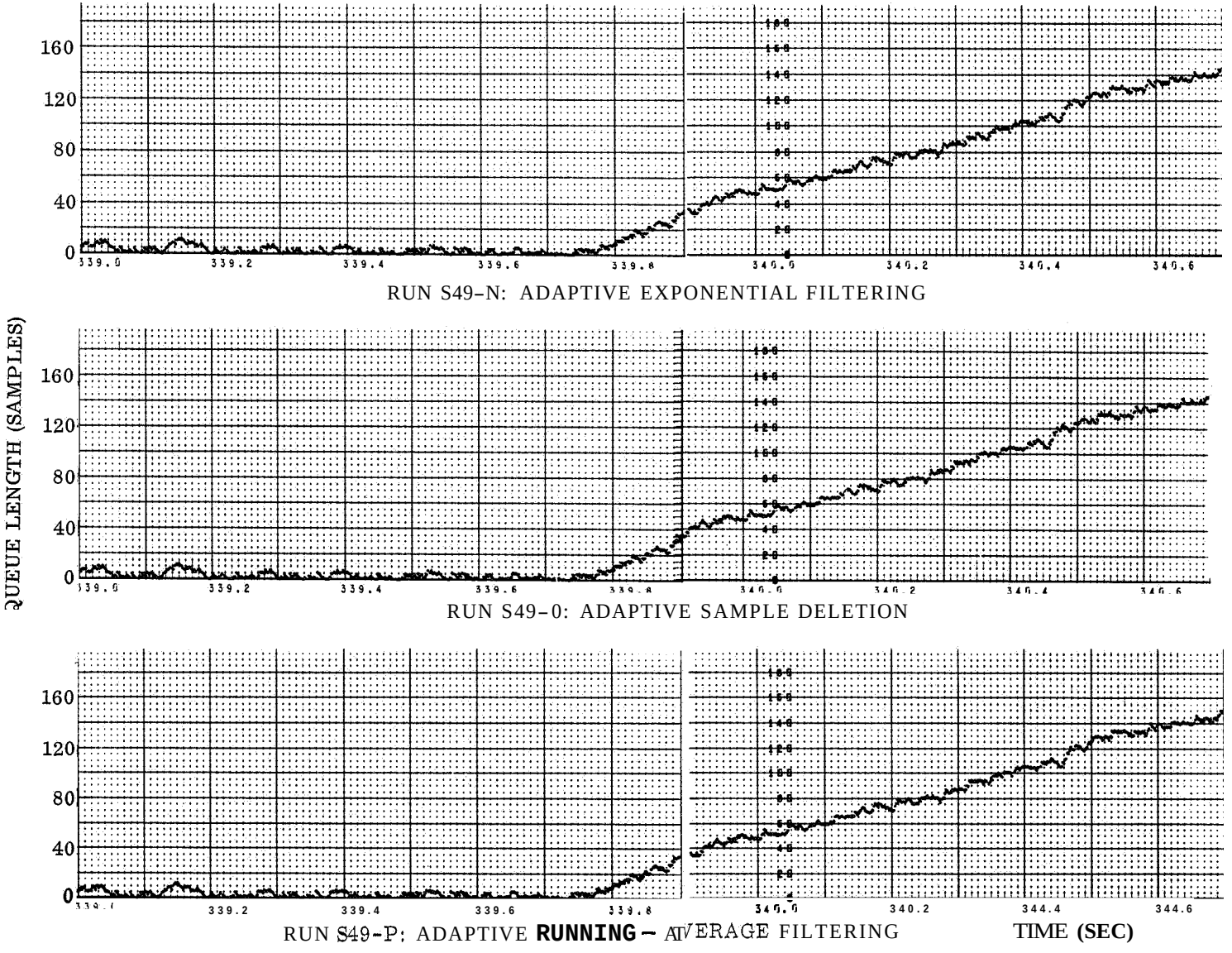


Fig. 3-31 Queue Length Behavior With Adaptive Filtering and Sample Deletion, S-49 Data

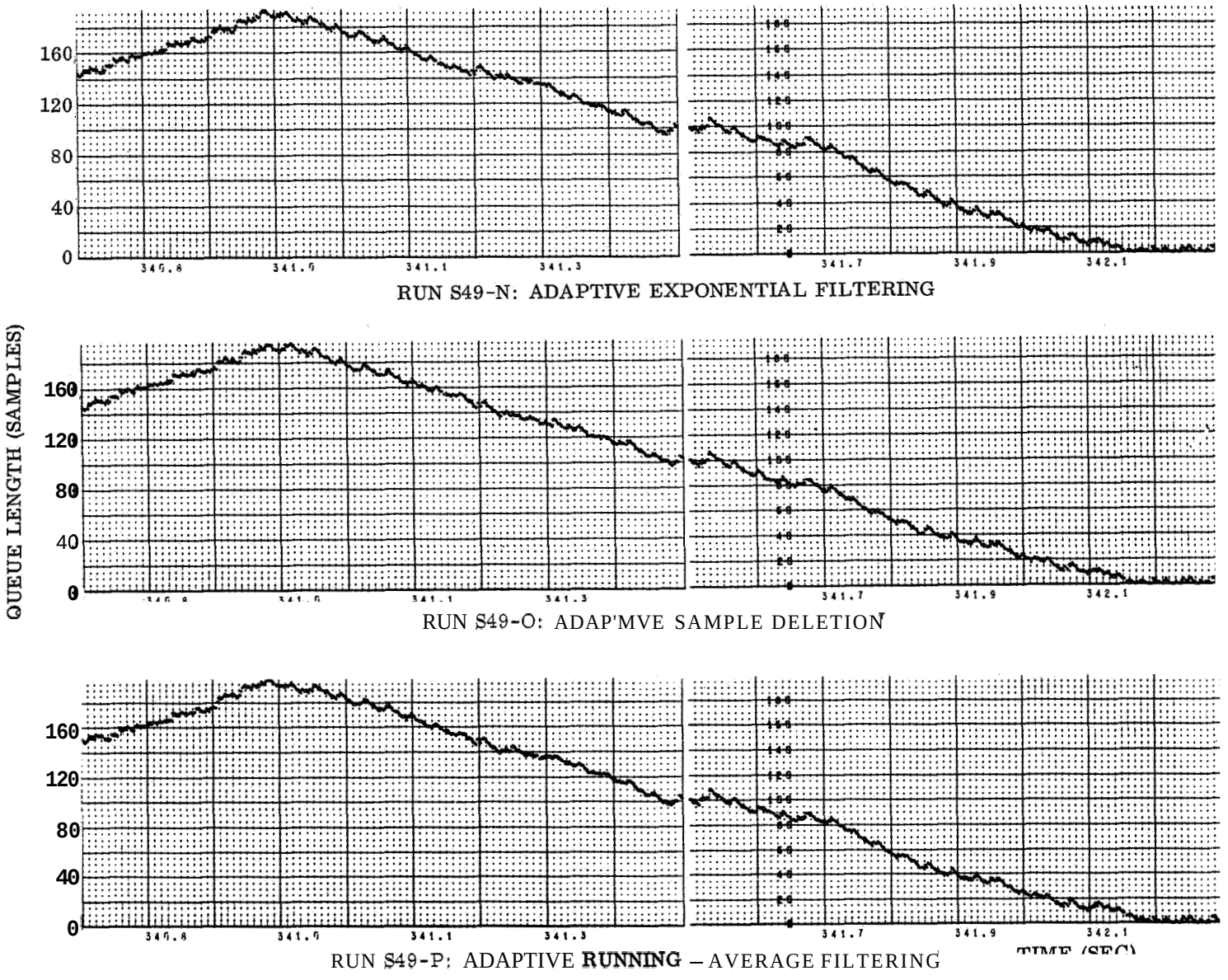


Fig. 3–31 Queue Length Behavior With Adaptive Filtering and Sample Deletion, **S-49** Data (Cont'd)

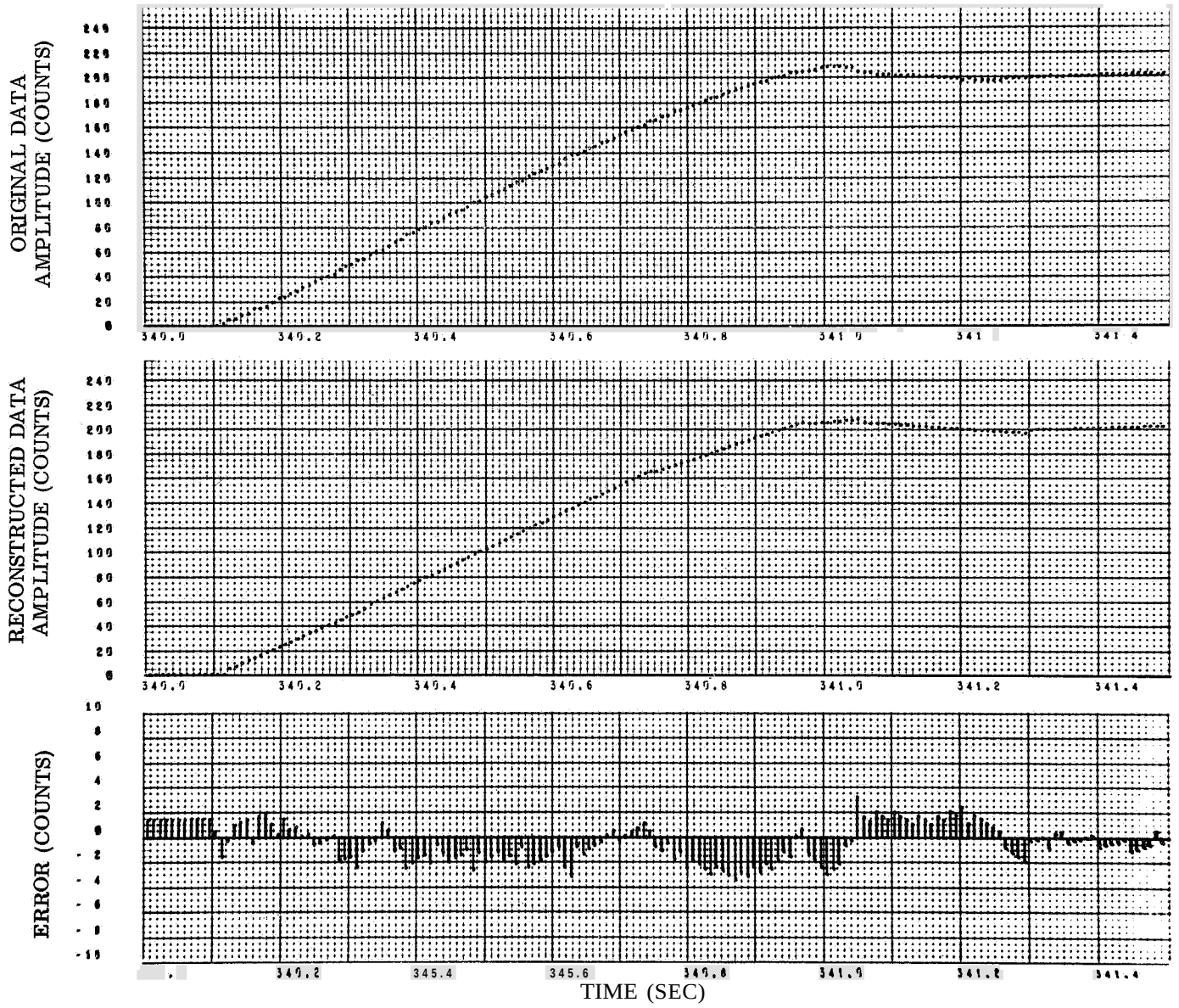


Fig. 3-32a Sensor 37 Reconstructed **Data,** Run **S49**-N: Adaptive Exponential Filtering

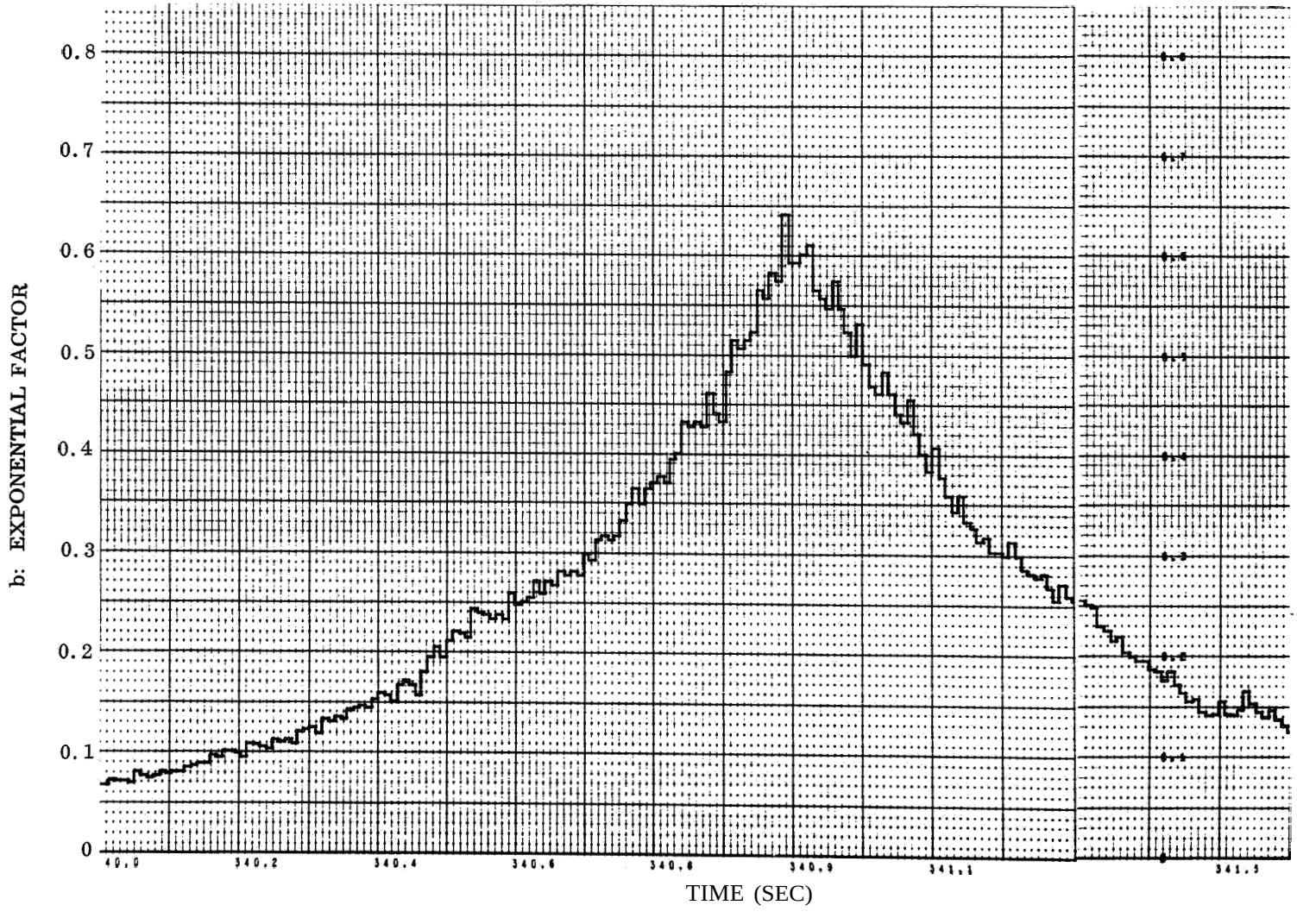


Fig. 3-32b Sensor 37 Exponential Factor Variations, Run S49-N: Adaptive Exponential Filtering

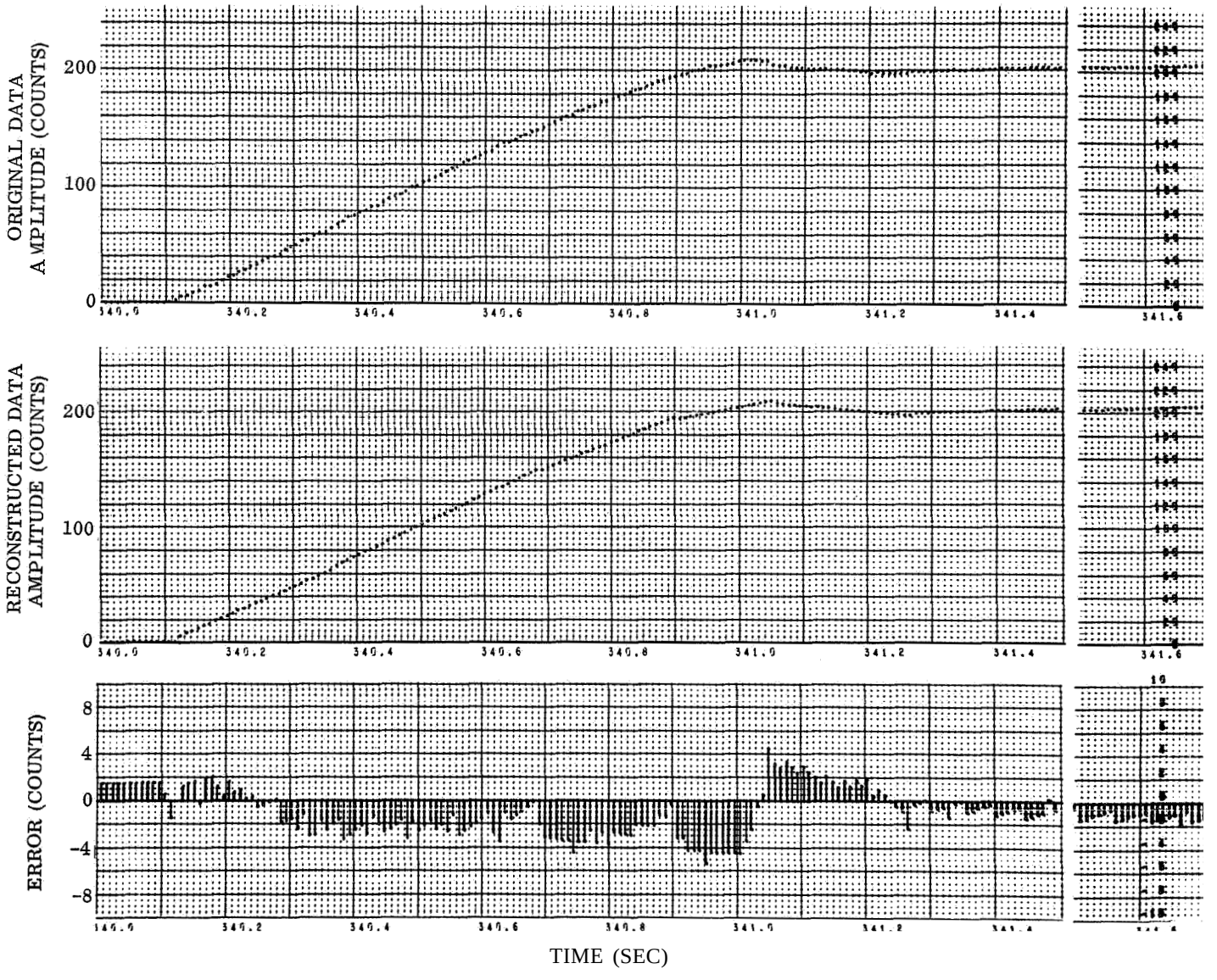


Fig. 3-33a Sensor 37 Reconstructed Data, Run S49-P: Adaptive Running-Average Filtering

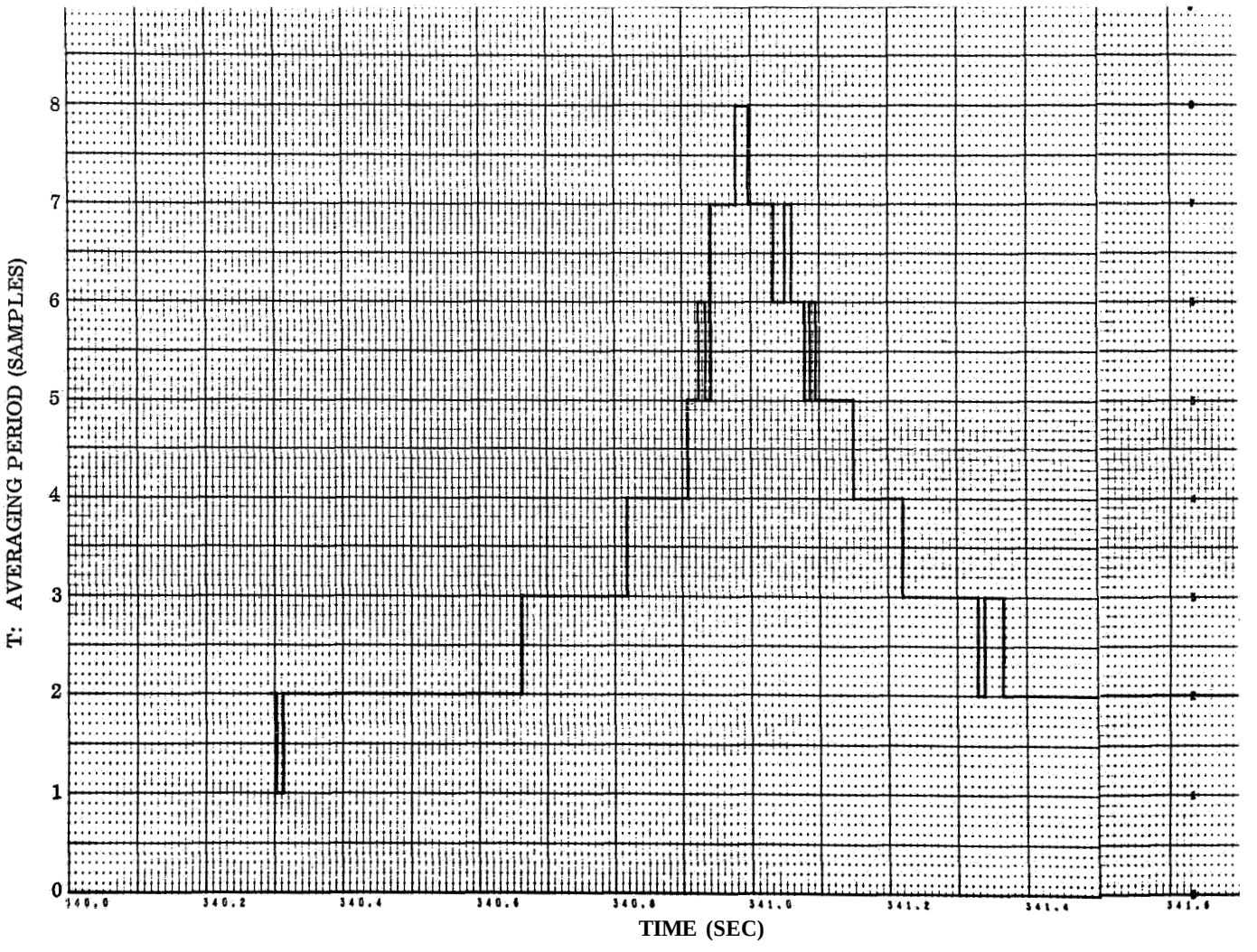


Fig. 3-33b Sensor 37 Averaging Period Variations, Run S49-P: Adaptive Running-Average Filtering

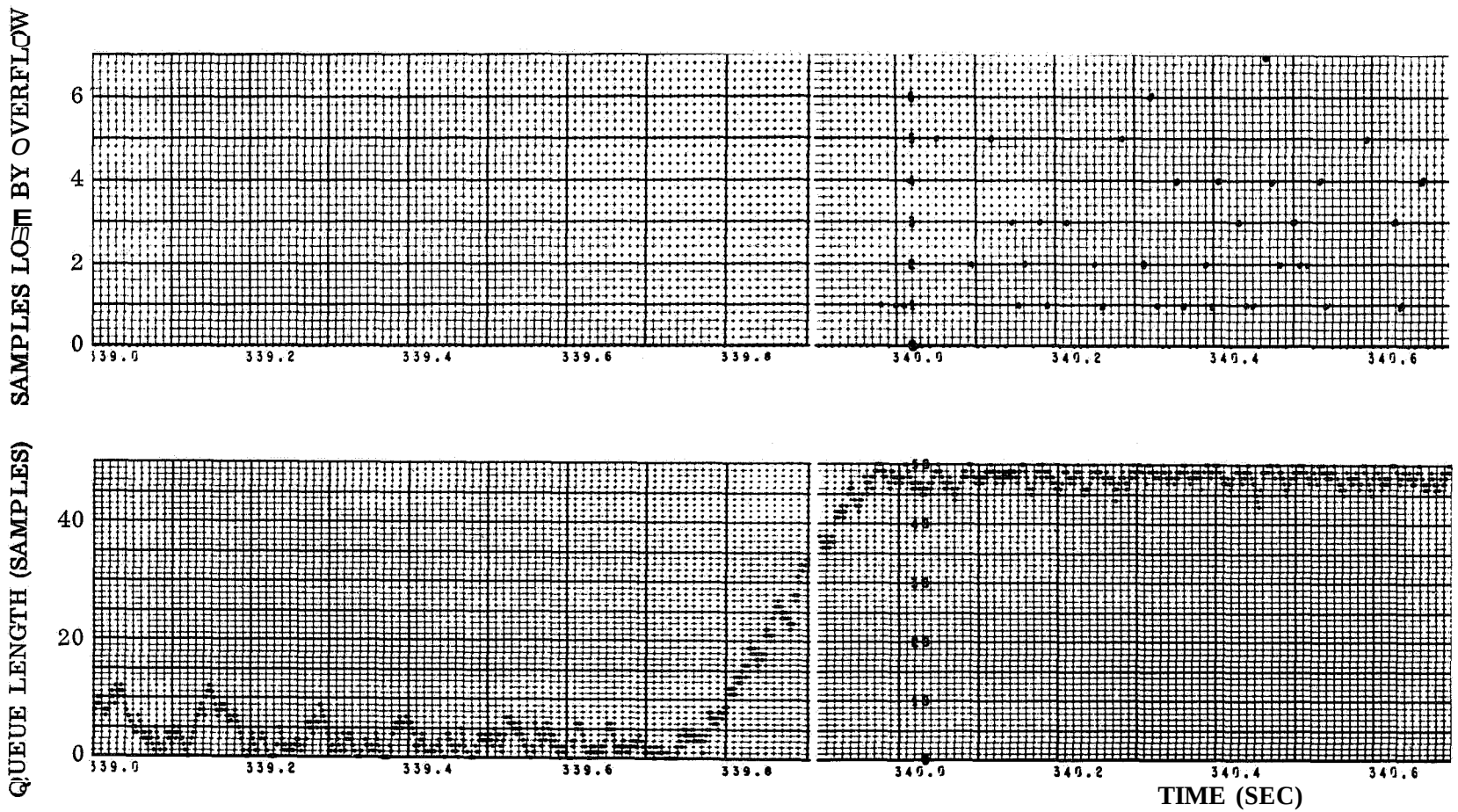


Fig. **3-34** Queue Length Behavior With Finite Buffer, Run S49-R

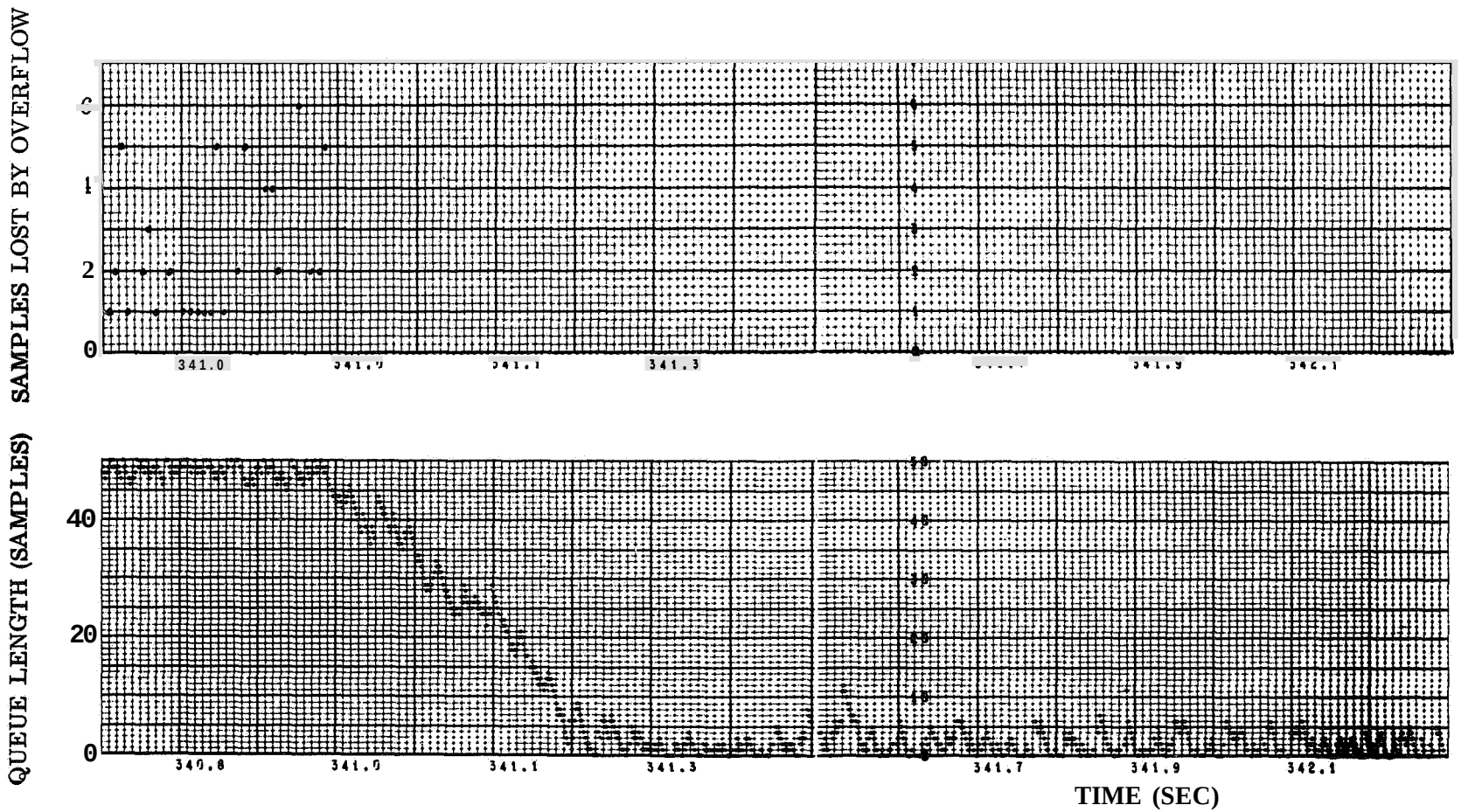


Fig. 3-34 Queue Length Behavior With Finite Buffer, Run S49-R (Cont'd)

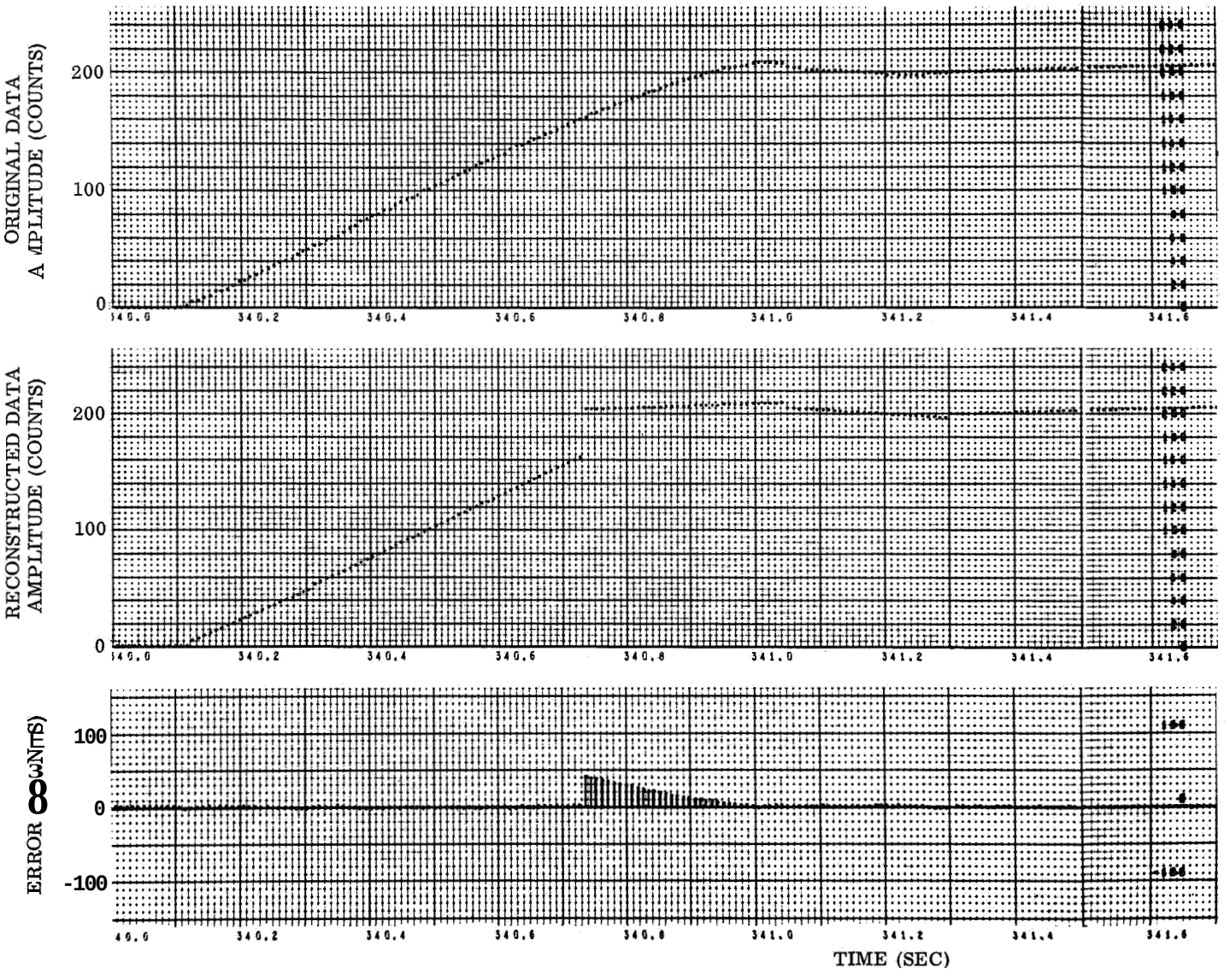
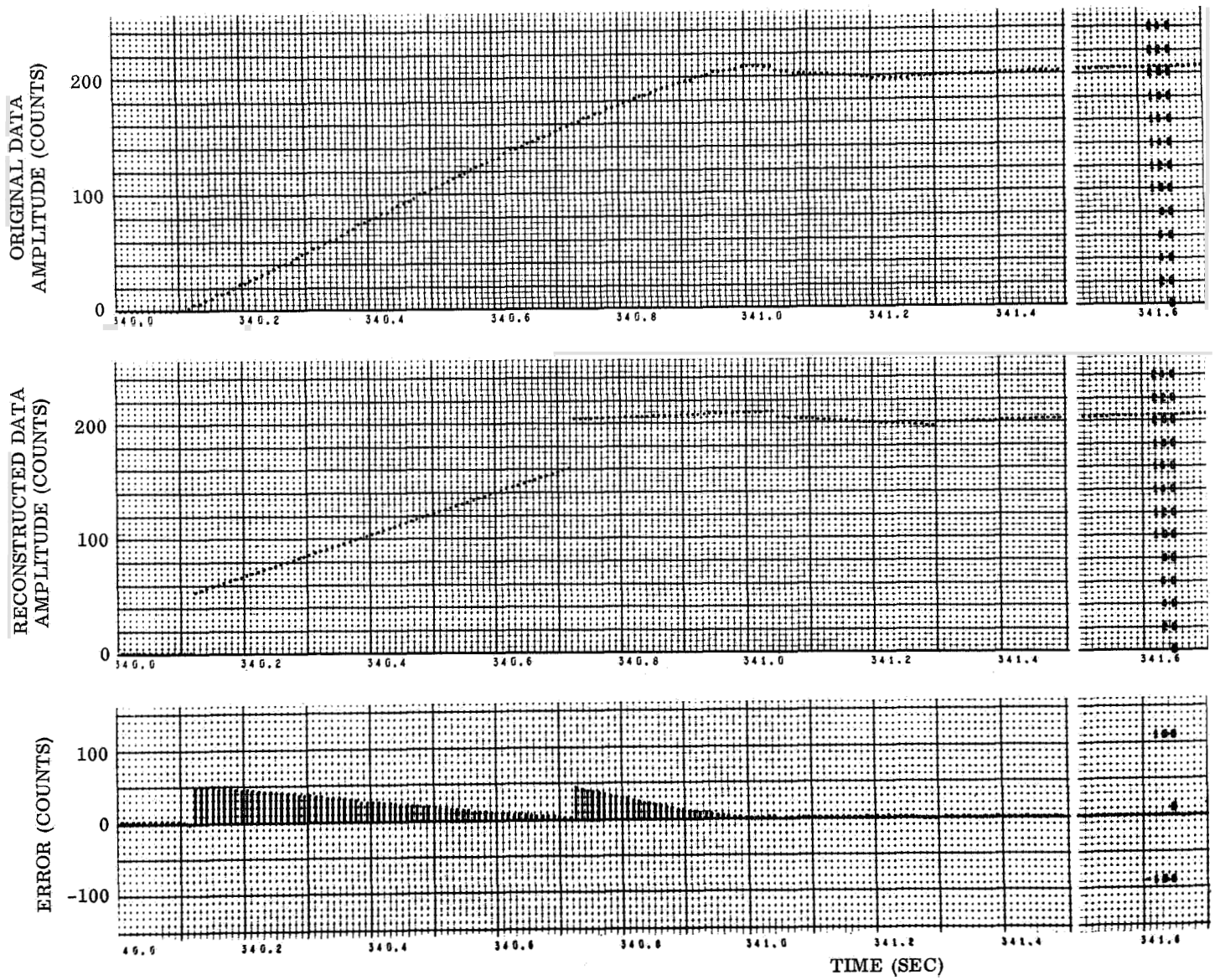


Fig. 3-35 Sensor 37 Reconstructed Data, Run S49-Q: With Buffer Overflow

LOCKHEED MISSILES & SPACE COMPANY



Fig, *3-36* Sensor **37** Reconstructed Data, Run S49–R: With **Buffer** Overflow

Section 4
COMPRESSED DATA TRANSMISSION ERRORS

The purpose of this study was to examine the effect of transmission bit errors on compressed data. To obtain a better understanding of this effect, a detailed theoretical analysis, as well as an experimental analysis, was necessary. While an experimental evaluation must necessarily be limited to a few particular data sets, a theoretical study provides general results which may be applied to any data. The theoretical study developed a formula for rms error for zero and first-order polynomial selectors. In the course of this development, the actual effect of each type of bit error was analyzed; thus, the reasons, and possible remedies, for the fragility of compressed data became evident.

In addition to determining rms error, the experimental analysis provided an opportunity to examine actual data waveforms, reconstructed from compressed data which had been contaminated by simulated ground transmission link bit errors. Thus, one study reinforced and complemented the other.

4.1 THEORETICAL BIT ERROR STUDY

A general formula for the rms error between data reconstructed from a contaminated compressed bit stream and from an uncontaminated compressed bit stream is

$$E_{rms} = \sqrt{\frac{\sum_{i=1}^{N} E_i^2}{N}}$$

where

E_i = error in magnitude observed for the ith sample

N = total number of uncompressed data samples

To obtain an expression for

$$\sum_{i=1}^{N} \varepsilon_i^2$$

the total number of bit errors was first found, along with their distribution among the coded compressed data formats that were used as models. Then the effect of these coded compressed data bit errors on the compressed data magnitudes was found, in terms of errors in magnitude. Finally, the effect of the errors in compressed data magnitudes on the reconstructed data waveform was determined. The specific steps in the analysis, along with the paragraph in which each is described, are as follows:

(1) An expression for the total number of bit errors in a specified uncompressed data time period was derived, along with the error distribution among the magnitude, sensor identification (ID), and flag (which causes errors in sensor ID) portions of the compressed bit stream word (Par. 4.1.1).

(2) The expected value of a data magnitude error was found for two cases: when it is caused by an error in a magnitude word, and by an error in a sensor ID word (Par 4 1.2)

(3) The overall sum of the squares of the errors (during the specified time period) in the magnitudes of the reconstructed data samples was found. These errors are caused by an error in a magnitude word and in a sensor ID word. They were derived first by finding this sum for a single reconstructed line segment of average length, and then by multiplying the answer by the average number of faulty line segments (segments with errors) in the time period (Par 4 1 3)

(4) The rms error was found by dividing the overall sum obtained in step 3 by the total number of reconstructed data samples, and taking the square root of the answer (Par. 4.1.4).

This analysis was performed for both the zero-order (ZOP/ZOI) and first-order (FOIDIS/FOI4DG) polynomial selectors. The compressed data transmission bit stream formats used as models for the theoretical analysis are shown in Fig. 4-1. They are the same formats as those used in the experimental study.

4.1.1 Bit Errors

The first step in analysis was to determine how many bit errors there are in a coded compressed PCM bit stream.

If N_c = total number of compressed words in a given time period

B = number of bits/bit stream record

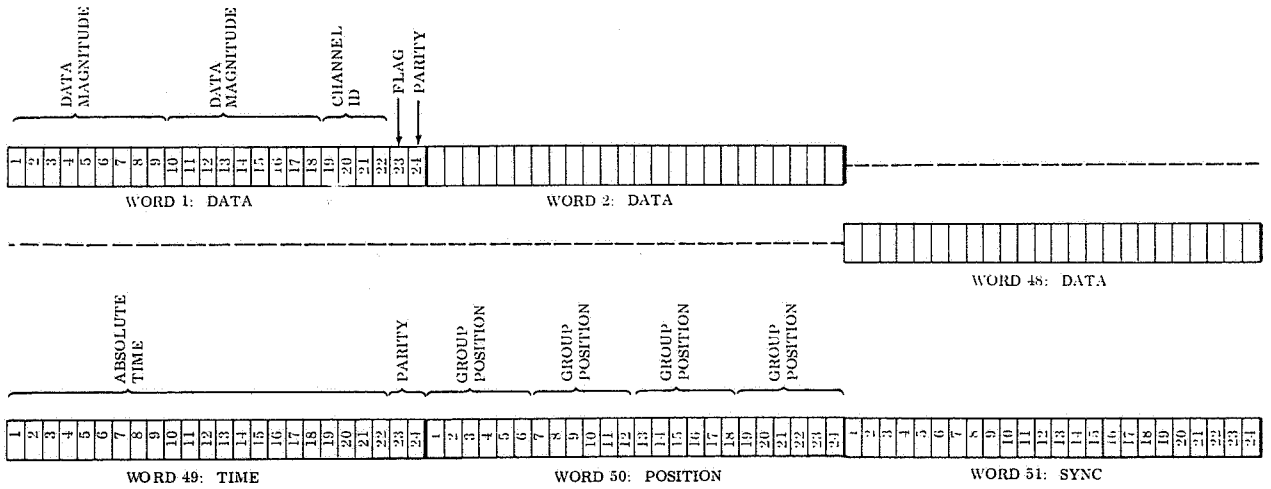
C = number of compressed words/bit stream record

R_E = bit error rate in errors/bit transmitted

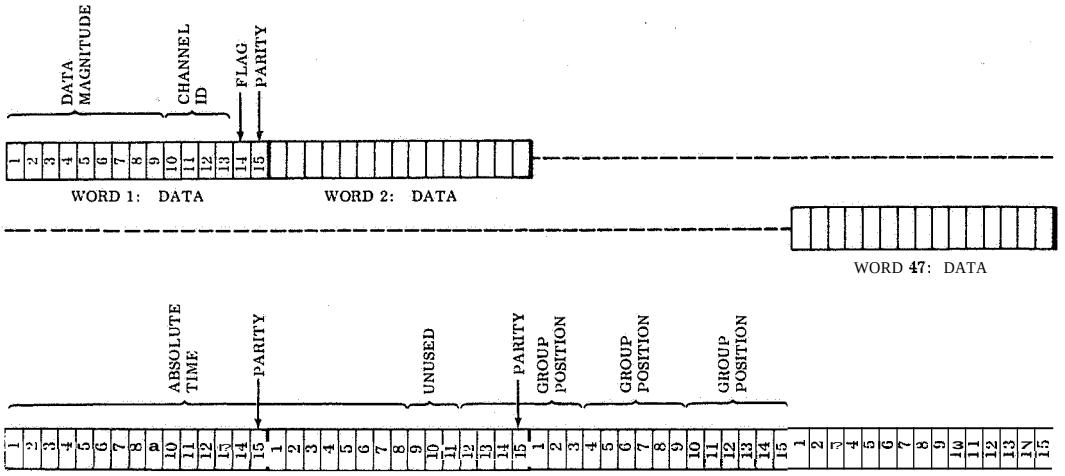
then the total number of bit errors is given by

$$B_T = \frac{N_c \, B \, R_E}{C}$$

Because these bit errors are distributed uniformly in each bit stream record, and the record format is known, the probabilities of occurrence of errors in each portion of the digital word (i.e., magnitude, sensor ID, and flag) can be calculated. These computations were made for the formats shown in Fig 4-1, and the results are presented in Table 4-1.



(a) First-Order Selectors



(b) Zero-Order Selectors

Fig. **4-1** CPCM Bit Stream Record Formats

Table 4-1

BIT ERROR PROBABILITIES

Probability	ZOI/ZOP	FOIDIS/FOI4DG
Probability of magnitude bit errors, $P_M(\epsilon)$	0.553	0.707
Probability of sensor ID bit errors, $P_{ID}(\epsilon)$	.24	.162
Probability of flag bit errors, $P_F(\epsilon)$	.065	.042
Probability of other bit errors	.142	.089

The "other" bit errors, which include time/position, parity, sync, and unused bits, were not considered in this study since their effect upon the bit stream is exceedingly hard to define; it is assumed that effective error elimination techniques could be used. The error elimination procedures are assumed to assure that no error will propagate from one record to the next. Magnitude, sensor ID, and flag errors in the bit stream can be defined fairly easily. These errors were studied in detail. Each bit error in magnitude (assuming one bit error per word for this study*) causes one line segment to be misoriented. Each error in sensor ID causes a line segment to be misassigned, producing error in the waveform where it was mistakenly assigned, as well as in the waveform from which it was taken. A flag error causes the remainder of that bit stream record to be decoded incorrectly, with respect to sensor ID. Thus, a number of sensor ID's are incorrectly assigned, and on the average, half the line segments in a record with a flag error are affected. As will be seen, unless steps are taken to prevent it, a single bit error can do much damage to a compressed PCM bit stream; the extent of this damage is shown in subsequent paragraphs.

*A brief study indicated that the account of more than one bit error per magnitude word would have a negligible effect on the final expression for rms error.

4 1 2 Magnitude of Compressed Data Word Errors

The magnitudes of the errors in the compressed words caused by the individual bit errors are developed in this section.

4.1.2.1 Bit Error in Magnitude Word. If a bit error occurs in a magnitude word, and it is assumed that there is only one bit error per magnitude word, the average value of that error is

$$E_M = \frac{1}{n} \sum_{I=1}^{n} 2^{(I-1)}$$

where n is the number of bits per word. For the purposes of this study n is assumed to be 9. Therefore,

$$E_M = \frac{1}{9} \sum_{I=1}^{9} 2^{(I-1)}$$

$$\approx 57$$

4 1 2 2 Bit Error in Sensor ID or Flag. Assuming a uniform distribution of data magnitudes from 0 to 511 (512 levels) and no dependence in data magnitude among different sensors, the following is true. The average expected error caused by a sensor ID error would be 255.5 if the original point were zero, 255 if the original point were one, and so on. An error in flag introduces this same error for each of the sensor ID's it causes to be incorrectly assigned. If an error in ID occurs, one line segment is affected; if an error in flag occurs, on the average one-half of the line segments in that record are affected.

Figure 4-2 shows the expected error, $g(\epsilon)$, as a function of the original magnitude for a uniform distribution. This figure, together with the following calculation, shows that if a sensor ID is in error or a tag bit is flipped, assuming an correct assignment of compressed data magnitude, the expected error per compressed magnitude word, $\Sigma_{\Sigma D}$, is 191 counts

$$\Sigma_{\Sigma D} = \frac{1}{512} \int_{0}^{255.5} \left(255.5 - \frac{x}{2}\right) dx \approx \frac{1}{512} \int_{255.5}^{511} \frac{x}{2}\, dx$$

$$\approx 191 \text{ counts}$$

4.1.3 Total Error Produced After Reconstruction

To find an overall rms error, the effect on the reconstructed data of errors in bit stream magnitude and sensor ID words was analyzed, and formulas for the overall sum of the squares of the errors in reconstructed data samples were developed. These expressions were derived for both the FOIDIS/FOI4DG and the ZOP/ZOI classes of selectors.

4.1.3.1 Sum of the Squares of the Errors for FOIDIS/FOI4DG. In this paragraph the following notation is used:

L	=	the number of line segments in a bit stream record
N_s	=	sample compression ratio
$2N_s$	=	average length of a line segment
D	=	average difference between the beginning and ending point of a line segment
E_{ID}	=	191 (the average error when a sensor ID is incorrect)
E_M	=	57 (the average error when an error in a magnitude word word occurs)
$\Sigma\Sigma^2_{X1}$	=	the sum of the squares of the errors per line segment when an ID is incorrectly assigned

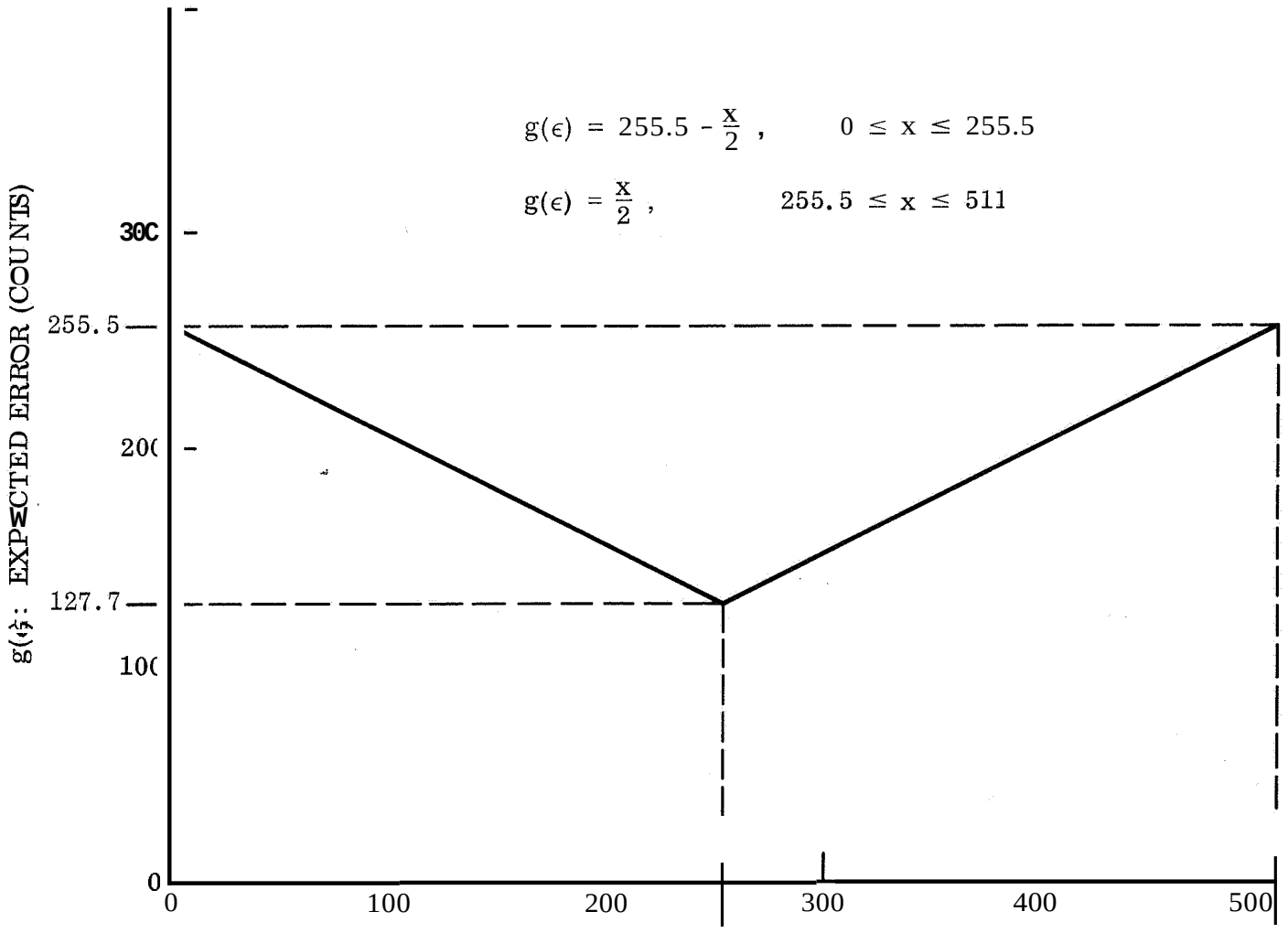


Fig. 4-2 Expected Error Versus Original Data Magnitude

$\sum E_{X2}^2$ = the sum of the squares of the errors per line segment when a magnitude word is in error

$\sum E_T^2$ = the overall sum of the squares of the errors caused by bit error

Sum of the Squares of the Errors Caused by an Error in a Magnitude Word. In Fig. 4-3 the solid line represents the correct reconstructed line segment; the broken line is the incorrect segment. The sum of the errors produced by each incorrect line segment is

$$\sum E_{X2} = \sum_{I=1}^{N_s - 1} \frac{E_M I}{2N_s - 1}$$

The sum of the squares of the errors is then given by

$$\sum E_{X2}^2 = \frac{E_M^2}{(N_s - 2)} \sum_{I=1}^{2N_s - 1} I^2 = \frac{E_M^2}{6}\left(4N_s + 1 + \frac{1}{2N_s - 1}\right) \qquad (4.1)$$

Sum of the Squares of the Errors Caused by a Bit Error in Sensor ID. Figure 4-4 shows what happens, in the average situation, to compressed data when a sensor ID is incorrect. Figure 4-4a shows the effect on the waveform from which the line segment was taken; Fig. 4-4b shows the effect on the waveform to which the line segment was added. When a sensor ID is incorrectly assigned, a line segment, along with the ending time of that segment, is eliminated from the waveform of the original sensor (segment A_1, A_2 in Fig. 4-4a). Because no information is thus available to the reconstructor about the correct beginning time of the next line segment, the magnitude A_3 (Fig. 4-4a) is shifted in time from T_3 to T_0. This causes errors in simple magnitudes to occur between times T_0 and T_5 in Fig. 4-4a. The type of errors caused by the removal of a line segment is designated "Error Category 1." The notation for the sum of these errors in a line segment of average length is $\sum E_1$. The notation for the sum of the squares of the errors is $\sum E_1^2$.

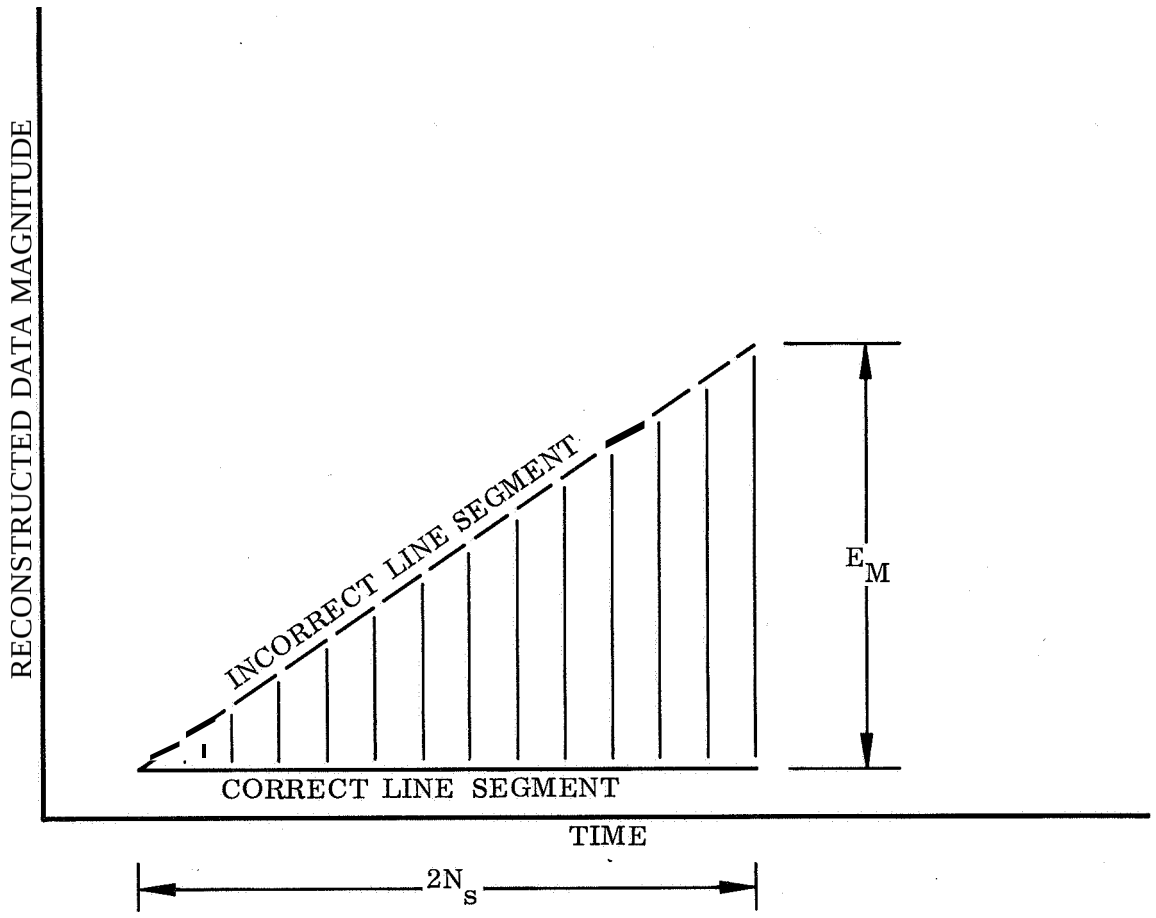
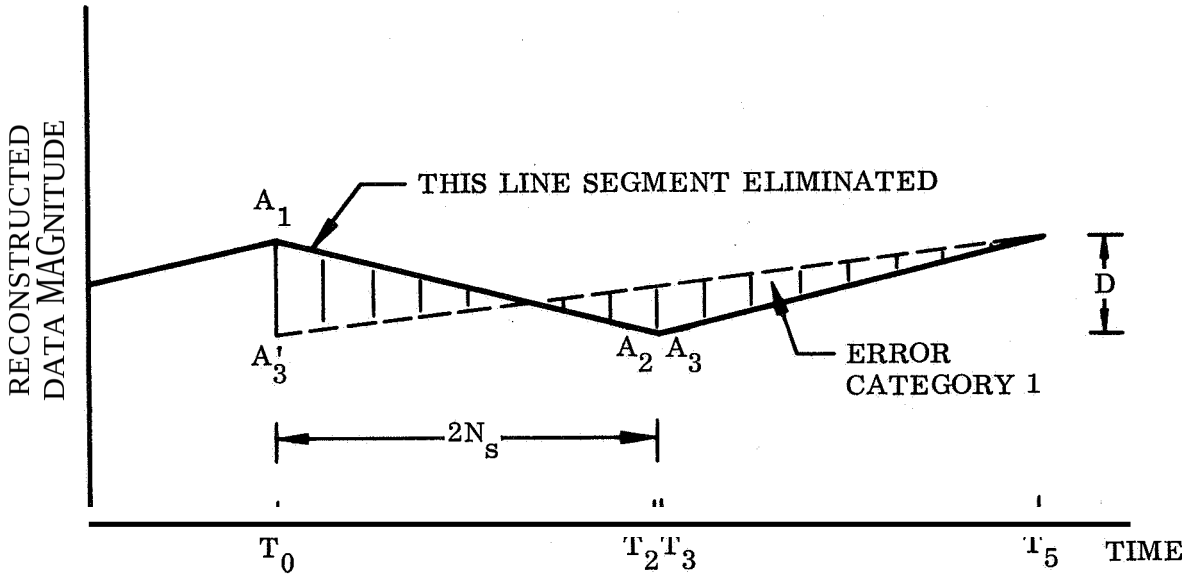
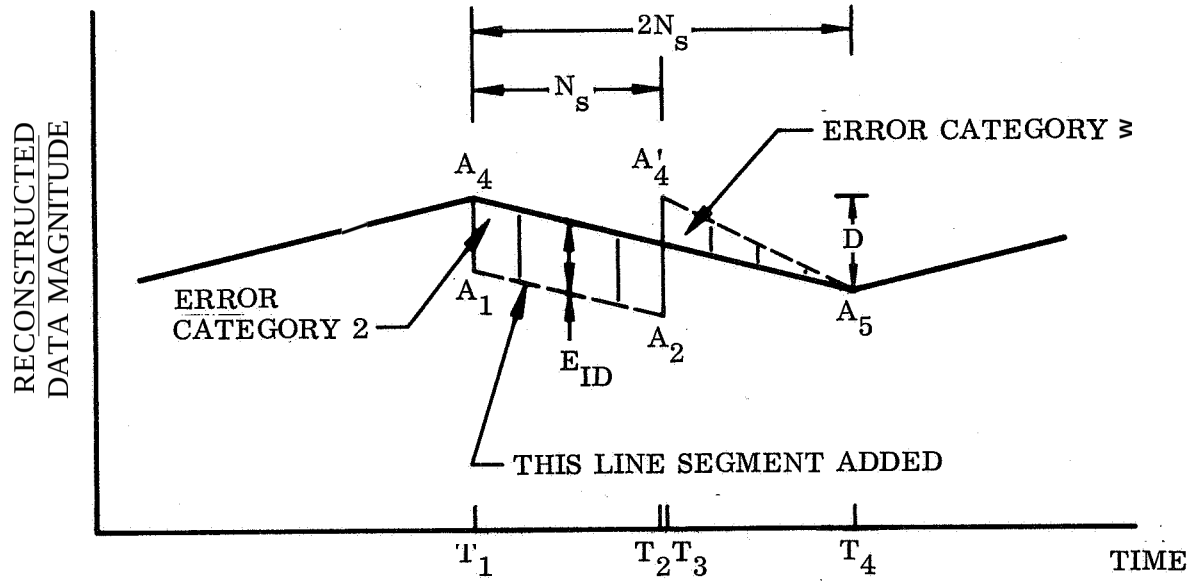


Fig. 4-3 Error in Reconstruction Caused by Faulty Bit Stream Magnitude Word, First-Order Selectors



(a) Waveform From Which Line Segment Is Eliminated



(b) Waveform to Which Line Segment Is Added

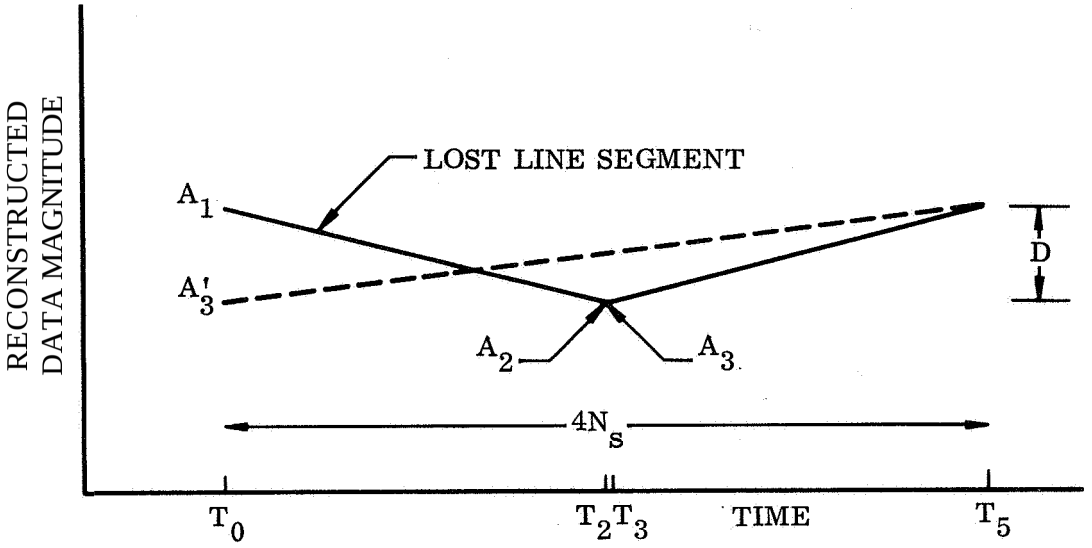
Fig. 4-4 Errors in Reconstruction Caused by Faulty Sensor ID, First-Order Selectors

The incorrect assignment of a sensor ID adds a line segment to the waveform of a new sensor. Again, operating with the only information available to it, the reconstructor erroneously positions the sample magnitude A_4 at time T_3, as shown in Fig. 4-4b. This causes two types of errors to occur in the reconstructed data. The first type, designated "Error Category 2," is caused by the addition of the line segment to the new waveform. The second, called "Error Category 3," is caused by the time shift of the beginning of the correct line segment from time T_1 to T_3 in Fig. 4-4b. The notation for the sum, per line segment of average length, of the errors in Category 2 is $\sum E_2$, and in Category 3, $\sum E_3$. The notation for the sum of the squares of the errors in each category is $\sum E_2^2$ and $\sum E_3^2$.

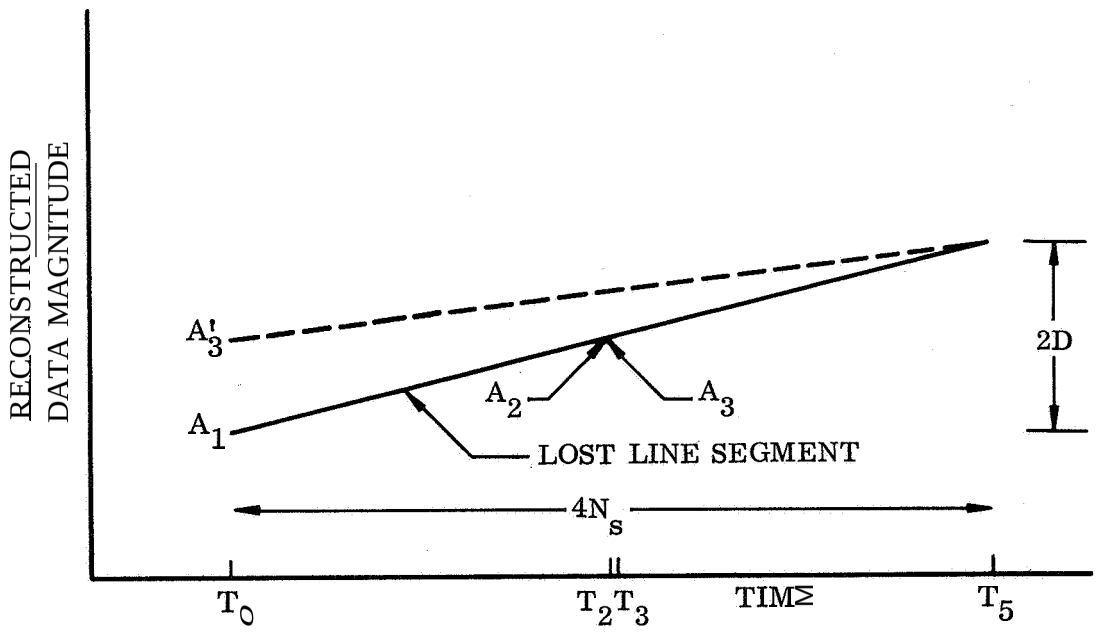
Note that this analysis is based entirely on an <u>average</u> situation. This means, for example, that the basic length of a line segment is $2N_s$, and that the basic length of an erroneously added segment is half this value, or N_s, as shown in Fig. 4-4.

<u>Error Category 1</u>. Assuming that the average difference between the beginning and end point of a line segment is D, two consecutive line segments may have one of two configurations. The two possibilities are shown in Fig. 4-5, with the broken lines showing the incorrectly reconstructed data. Thus, Error Category 1 is further divided into Categories 1a and 1b, as indicated in Fig. 4-5. Let the sum of the errors in a line segment of average length for the two subcategories be given by $\sum E_{1a}$ and $\sum E_{1b}$, and the sum of the squares of the same errors be givey by $\sum E_{1a}^2$ and $\sum E_{1b}^2$. Then, from inspection of Fig. 4-5,

$$\sum E_{1a} = \sum_{I=1}^{2N_s-1} \left| \frac{\quad}{2(2N_s - 1)} - \left(\frac{-DI}{2N_s - 1} + \alpha \right) \right| + \sum_{I=1}^{2N_s-1} \frac{DI}{2N_s - 1} - \frac{DI}{2(2N_s - 1)}$$



(a) Error Category 1a



(b) Error Category 1b

Fig. 4-5 Sensor ID Errors Caused by Missing Line Segments, First-Order Selectors

$$\Sigma E_{1a}^2 = \sum_{I=1}^{2N_s-1} \frac{9D^2I^2}{4(2N_s - 1)^2} + \frac{3D^2I}{(2N_s - 1)} + D^2 + \sum_{I=1}^{2N_s-1} \frac{D^2I^2}{4(2N_s - 1)^2}$$

$$= \frac{5D^2}{12(2N_s - 1)} \left(8N_s^2 - 2N_s\right) + 5N_sD^2 + D^2$$

$$\Sigma E_{1b} = \sum_{I=1}^{4N_s-1} \frac{2DI}{4N_s - 1} - \frac{DI}{4N_s - 1}$$

$$\Sigma E_{1b}^2 = \frac{D^2}{3(4N_s - 1)} \left(16N_s^2 - 2N_s\right)$$

Since each configuration is equally likely,

$$0.5\ \Sigma E_{1a}^2 + 0.5\ \Sigma E_{1b}^2 = \Sigma E_1^2$$

or

$$\Sigma E_1^2 = \frac{5D^2\left(8N_s^2 - 2N_s\right)}{24(2N_s - 1)} + \frac{D^2\left(8N_s^2 - N_s\right)}{3(4N_s - 1)} + 2 \cdot N_sD^2 - 0 \quad 2$$

Error Category 2. As shown in Fig. 4-4b, the average length of the added line segment is N_s. Hence, from inspection of Fig. 4-4b,

$$\Sigma E_2 = N_sE_{ID}$$

$$\Sigma E_2^2 = N_sE_{ID}^2$$

Error Category 3. Figure 4-4b shows the errors falling in Category 3. From inspection of Fig. 4-4b,

$$\sum \Sigma_3 = \sum_{I=1}^{N_s} \left| \frac{D}{N_s} I - \frac{D}{2N_s} I \right|$$

$$\sum E_3^2 = \frac{D^2}{4N_s^2} \sum_{I=1}^{N_s} I^2 = \frac{D^2}{24N_s} \left(2N_s^2 + 3N_s + 1\right)$$

Categories 1, 2, and 3, Combined. The combined sum of the squares of the errors per faulty line segment, i.e., per error in sensor ID, is

$$\sum E_{X1}^2 = \sum E_1^2 + \sum_2^2 + \sum E_3^2$$

$$= \frac{5D^2 \left(8N_s^2 - 2N_s\right)}{24(2N_s - 1)} + \frac{D^2 \left(8N_s^2 - N_s\right)}{3(4N_s - 1)}$$

$$+ N_s \left(2\ 5D^2 + \Sigma_{ID}^2\right) + \frac{D^2}{24N_s} \left(2N_s^2 + 3N_s + 1\right) - 0\ \Sigma D^2 \qquad (4.2)$$

Overall Sum of the Squares of the Errors. To determine the overall sum of the squares of the errors, the number of faulty line segments must be found. The number of wrong line segments caused by magnitude error equals the number of bit errors multiplied by the probability that a bit error is an error in magnitude. Thus,

$$L_M = \frac{N_c B R_E}{C} P_M(\epsilon) \qquad (4\ 3)$$

where L_M is the number of wrong line segments caused by magnitude error.

For errors in sensor ID, the expression is quite similar. The number of faulty line segments caused by sensor ID error is

$$L_I = \frac{N_c BR_E}{\mathcal{C}} P_{ID}(\epsilon) \tag{4 4}$$

For errors in flag, on the average, half the line segments in a record (assuming a uniform distribution of flag errors within a record) will be assigned incorrect sensor ID. Thus, the total number of faulty line segments caused by flag error becomes

$$L_F = \frac{N_c BR_E}{C} \frac{L}{2} P_F(\epsilon) \tag{4 5}$$

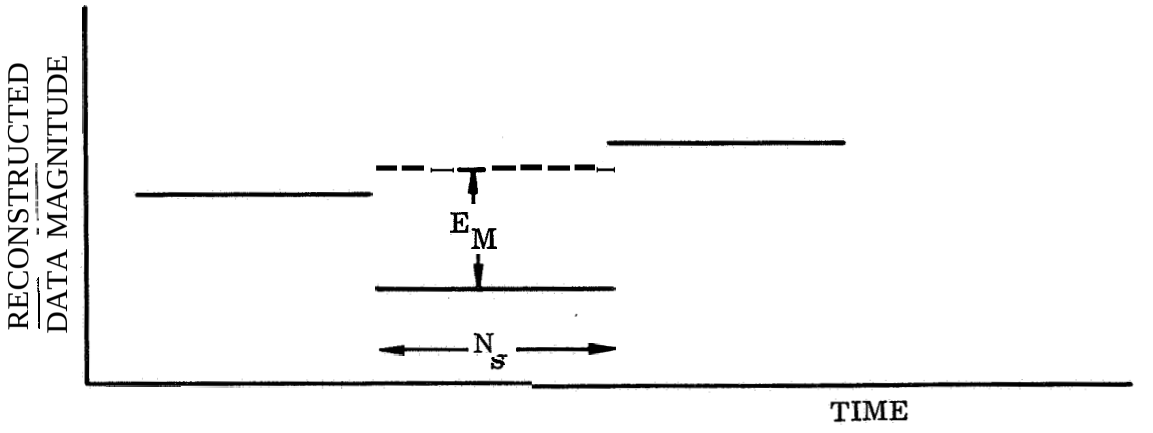
Using the information derived above, the total sum of all the errors squared for the FOIDIS/FOI4DG format, shown in Fig. 4-1, is

$$\Sigma E = L_M \sum E_{X2}^2 + (L_I + L_F) \sum E_{X1}^2 \tag{4 6}$$

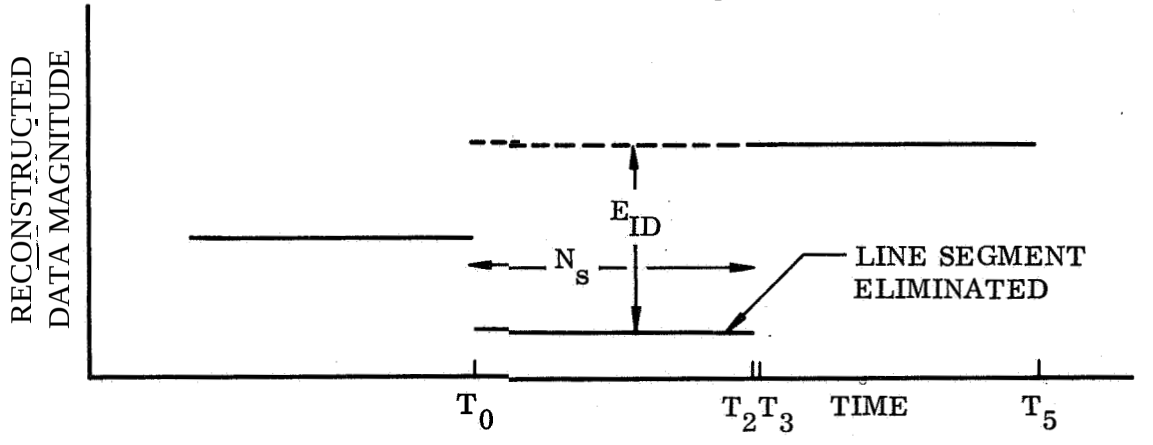
$$= \frac{N_c BR_E}{C} \left\{ \frac{P_M(\epsilon)\, E_M^2}{12} (8N_s + 6 + 1/N_s) + \left[P_{ID}(\epsilon) + \frac{L}{2} P_F(\epsilon) \right] \left[\frac{D^2}{24} (26N_s + 6 + 4/N_s) + N_s E_{ID}^2 \right] \right\} \tag{4 7}$$

4.1.3.2 Sum of the Squares of the Errors for ZOP/ZOI. In this paragraph the symbols used are the same as in Par. 4.1.3.1, except that each line segment is now N_s samples long. This is true for the zero-order selectors, since only one data sample is transmitted per line segment instead of two.

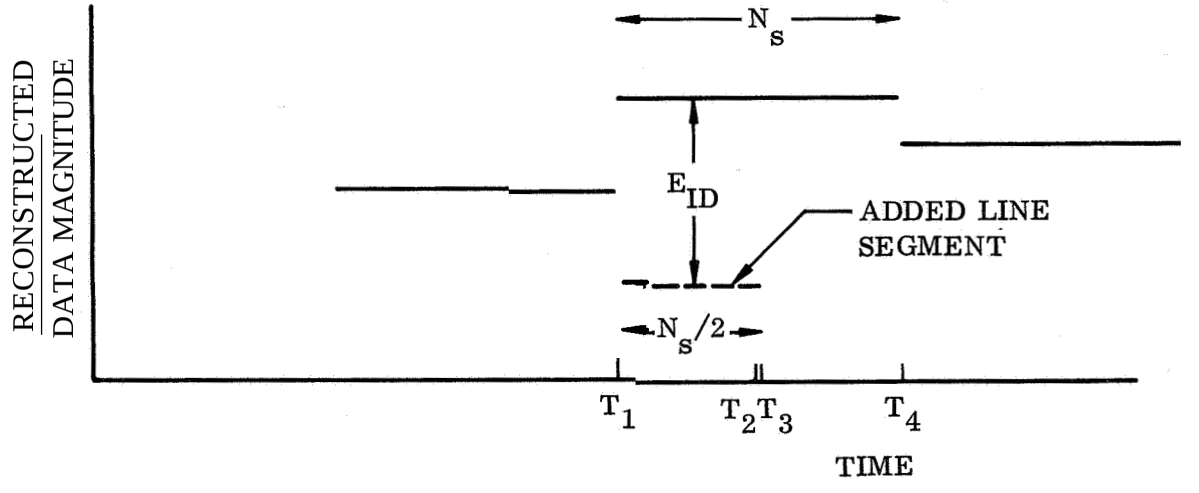
Error Caused by an Error in Magnitude. Figure 4-6a shows a typical reconstructed waveform where a magnitude error has occurred. The broken line is the incorrect segment. From inspection of this figure, it is seen that the error is simply



(a) Error in Bit Stream Magnitude Word



(b) Sensor ID Error Caused by Missing Line Segment



(c) Sensor ID Error Caused by Added Line Segment

Fig. 4-6 Error Due to Faulty Line Segments in the ZOP/ZOI Format

$$\sum E_X = N_s E_M$$

$$\sum E_X^2 = N_s E_M^2 \qquad (4\ 8)$$

Error Caused When a Sensor ID Error Has Occurred. Figures 4-6b and 4-6c show what happens when an ID is incorrectly assigned. Figure 4-6b shows the waveform from which the line segment was taken, and Fig. 4-6c shows the waveform in which it was added. Because the magnitude and ending time of a line segment were eliminated from the waveform shown in Fig. 4-6b, the reconstructor, operating on the only information available to it, begins the next line segment at time T_0, instead of T_3, which is the correct time. Thus, all the reconstructed data samples between T_0 and T_2 are in error.

Likewise, in the waveform to which the line segment is erroneously added, the reconstructor begins the line segment A_2 at the end of the added segment A_1 or at time T_3 shown in Fig. 4-6c, instead of at T_1, which is the correct time. Hence, all the reconstructed data samples between T_1 and T_2 are in error. As in the case of the first-order selectors, note that the analysis is based entirely on an average situation. For example, the average length of an added line segment, as shown in Fig. 4-6c, is $N_s/2$.

From inspection of Figs. 4-6b and 4-6c,

$$\sum E_X = N_s E_{ID} + N_s/2\ E_{ID}$$

$$\sum E_{X1}^2 = 1.5\ N_s E_{ID}^2 \qquad (4\ 9)$$

Overall Sum of the Squares of the Errors. The number of line segments affected is given by Eqs. (4 3), (4 4), and (4 5), and the total sum of the errors squared is given by Eq. (4 6). Thus, from Eqs. (4 3), (4 4), (4 5), (4 8), and (4 9), the overall sum of the squares of the errors for the ZOP/ZOI format shown in Fig. 4-1 is

$$\sum E_T^2 = \frac{N_c N_s B R_E}{C} \left\{ P_M(\epsilon)\, E_M^2 + \frac{3}{2} E_{ID}^2 \left(P_{ID}(\epsilon) + \frac{L}{2} P_F(\epsilon) \right) \right\} \tag{4.10}$$

4 1 4 Rms Error

Once again, the expression for rms error is

$$E_{rms} = \frac{\quad}{\quad} = \sqrt{\frac{\sum E_T^2}{N}}$$

where E_i is the error at every sample and N is the total number of samples. The expressions for $\sum E_T$ have been found in Par. 4.1.3. The number of samples present is simply the number of compressed samples, N_c, multiplied by the sample compression ratio, N_s. Thus, the rms error due to bit errors is

$$E_{rms} = \sqrt{\frac{\sum E_T^2}{N_c N_s}} \tag{4.11}$$

4.1.4.1 <u>Rms Error for FOIDIS/FOI4DG Format.</u> Substituting Eq. (4.7) into (4 11),

$$\Sigma_{rms} = \sqrt{\frac{BR_E}{N_s C} \left\{ \frac{E_M^2 P_M(\epsilon)}{6} \left(4N_s + 1 + \frac{1}{2N_s - 1} \right) + \left(P_{ID}(\epsilon) + \frac{L}{2} P_F(\epsilon) \right) \left[\frac{5D^2 \left(8N_s^2 - 2N_s \right)}{24(2N_s - 1)} + \frac{D^2 \left(8N_s^2 - N_s \right)}{3(4N_s - 1)} + 2.5 N_s D^2 - 0.5D \right] \right\}} \tag{4 18}$$

For the FOIDIS/FOI4DG bit stream discussed in Refs. 1 and 2,

$$B = 1224$$

$$C = 36$$

$$P_M(\varepsilon) = 0\ 707$$

$$P_{ID}(\varepsilon) = 0.164$$

$$P_F(\varepsilon) = 0.042$$

$$L = 48$$

$$D = 63^*$$

Substituting these values into Eq. (4.12), a curve of $E_{rms}/\sqrt{R_E}$ versus N_s was obtained. This curve is presented in Fig. 4-7. The largest value of E_{rms} is for $N_s = 1$ ($181.2\sqrt{R_E}$, % full scale) and as $N_s \rightarrow \infty$, $E_{rms} \rightarrow 180.8\sqrt{R_E}$, % full scale. Figure 4-7 indicates that above sample compression ratios of about 5, the rms error is almost independent of compression ratio. Also, for compression ratios below 5, the rms error actually decreases with an increase in compression ratio. Because this result is not intuitively evident, it emphasizes the value of theoretical analyses such as this one.

4.1.4.2 Rms Error for ZOP/ZOI Format. Substituting Eq. (4.10) into (4.11),

$$E_{rms} = \sqrt{\frac{BR_E}{C}\left\{P_M(\epsilon)\,E_M^2 + \frac{3}{2}\,E_{ID}^2\left(P_{ID}(\epsilon) + \frac{L}{2}\,P_F(\epsilon)\right)\right\}} \qquad (4.13)$$

*This number was obtained experimentally using a sample size of 52,000 line segments.

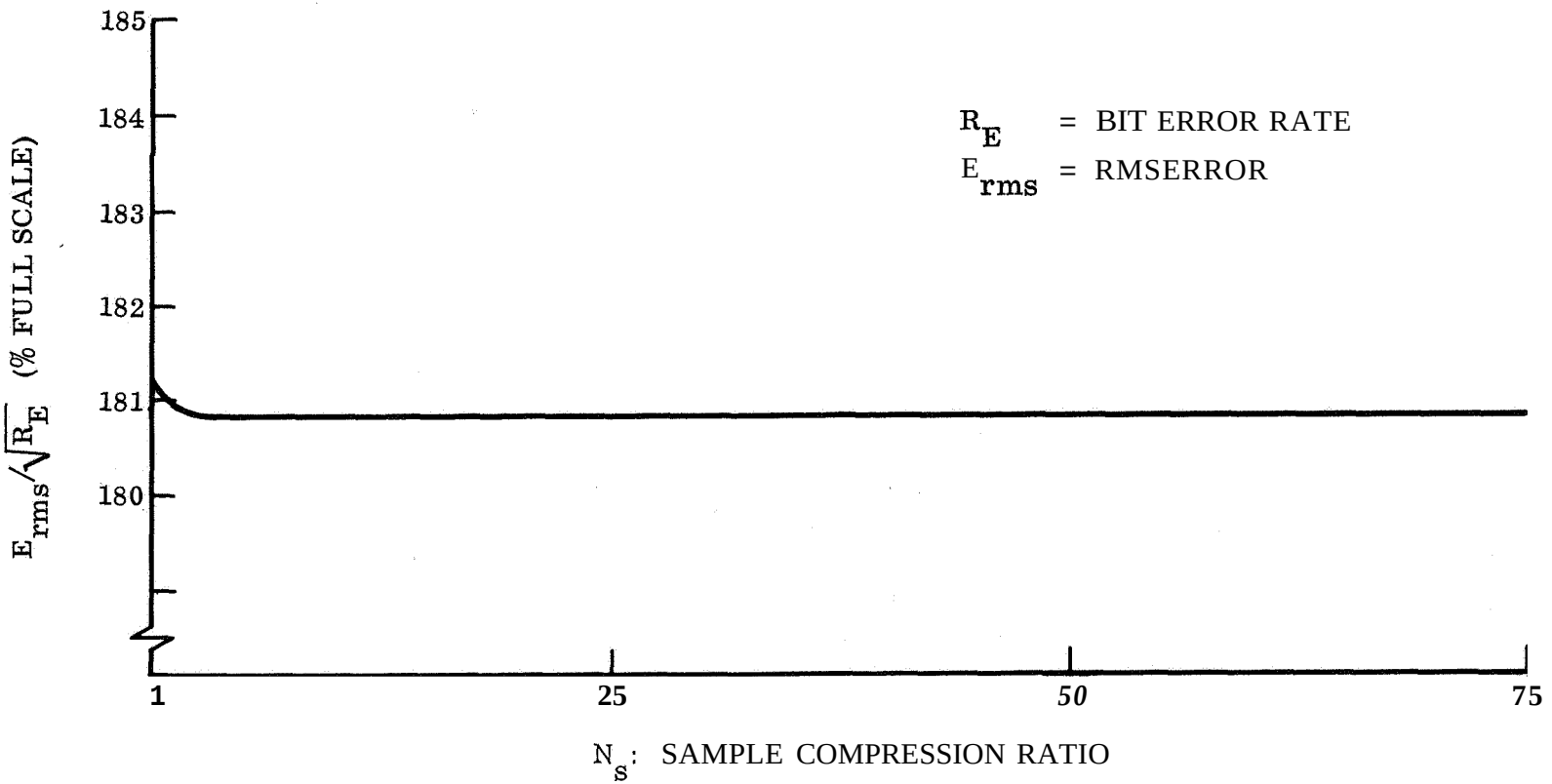


Fig. 4-7 Rms Error/$\sqrt{R_E}$ Versus Sample Compression Ratio

For the ZOP/ZOI bit stream format,

$$B = 765$$

$$C = 48$$

$$P_M(\epsilon) = 0.553$$

$$P_{ID}(\epsilon) = 0.240$$

$$P_F(\epsilon) = 0.065$$

$$L = 48$$

Substituting these values into Eq. (4.13),

$$\sigma_{rms} = 248 \sqrt{R_E}\ , \text{ \% full scale} \qquad (4.14)$$

Note that Eq. (4.14) is independent of N_s – a result which as in the case of the first-order selector, is not intuitively evident.

4.1.4.3 Rms Error Due to Bit Errors in Uncompressed Data. An expression for rms error in PCM words due to bit errors is (Ref. 5)

$$\sigma_{rms} = \sqrt{\frac{P(\epsilon)}{3}\left(1 - \frac{1}{2^{2n}}\right)} \cdot 100, \text{ \% full scale}$$

where

$P(\epsilon)$ = bit error probability

n = number of bits in the PCM word

4.1.4.4 A Comparison of Theoretical Rms Errors. Table 4-2 shows the theoretical rms errors for both compressed and uncompressed data for two values of bit error rate, R_E.

Table 4-2

THEORETICAL RMS ERRORS

W rror te	α (a)	Rms Error (% full scale)		
		FOIDIS/FOI4DG-Compressed	ZOP/ZOI-Compressed	Uncompressed
1.35×10^{-4}	0.3	2.1	2.88	0.67
4.00×10^{-4}	0.4	3.62	4.96	1.155

(a) The quantity α is the Pareto noise model clustering factor, which determines the bit error rate (Ref. 6).

As expected the compressed data has a higher rms error, demonstrating that it is more fragile than compressed data. However, the difference is not as large as might expected and with additional error elimination techniques built into the coded bit stream the overall rms error could made comparable to the rms error in the compressed case.

4 1 5 Observations and Conclusions

The following points are considered worthy of note:

- The rms error for the FOIDIS/FOI4DG selectors varies with compression ratio, but only very slightly.
- The rms error for ZOP/ZOI selectors is much higher than for the FOIDIS/FOI4DG owing to the fact that every error associated with the ZOP/ZOI selectors is equal to of the expected errors E_M or E_{ID}; while in the FOIDIS/FOI4DG case, the errors sometimes decrease from E_M or E_{ID} to 0 as a faulty line segment is scanned.
- The effect of flag error is predominant in both error models. For FOIDIS/FOI4DG the errors caused by flag errors account for 5 6 percent of the faulty line segments; while in the ZOP/ZOI case the figure is 66.4 percent. The total effect of flag errors on the rms error is further magnified the fact that errors in magnitude words are smaller deviations in line segments

tha error in sensor ID Thus, a reduction in the number of flag errors, perhaps th ough redundant transmission of the flag bit, is a first step in greatl redu g the overall rms error.

4.2 EXPERIMENTAL BIT ERROR STUDY

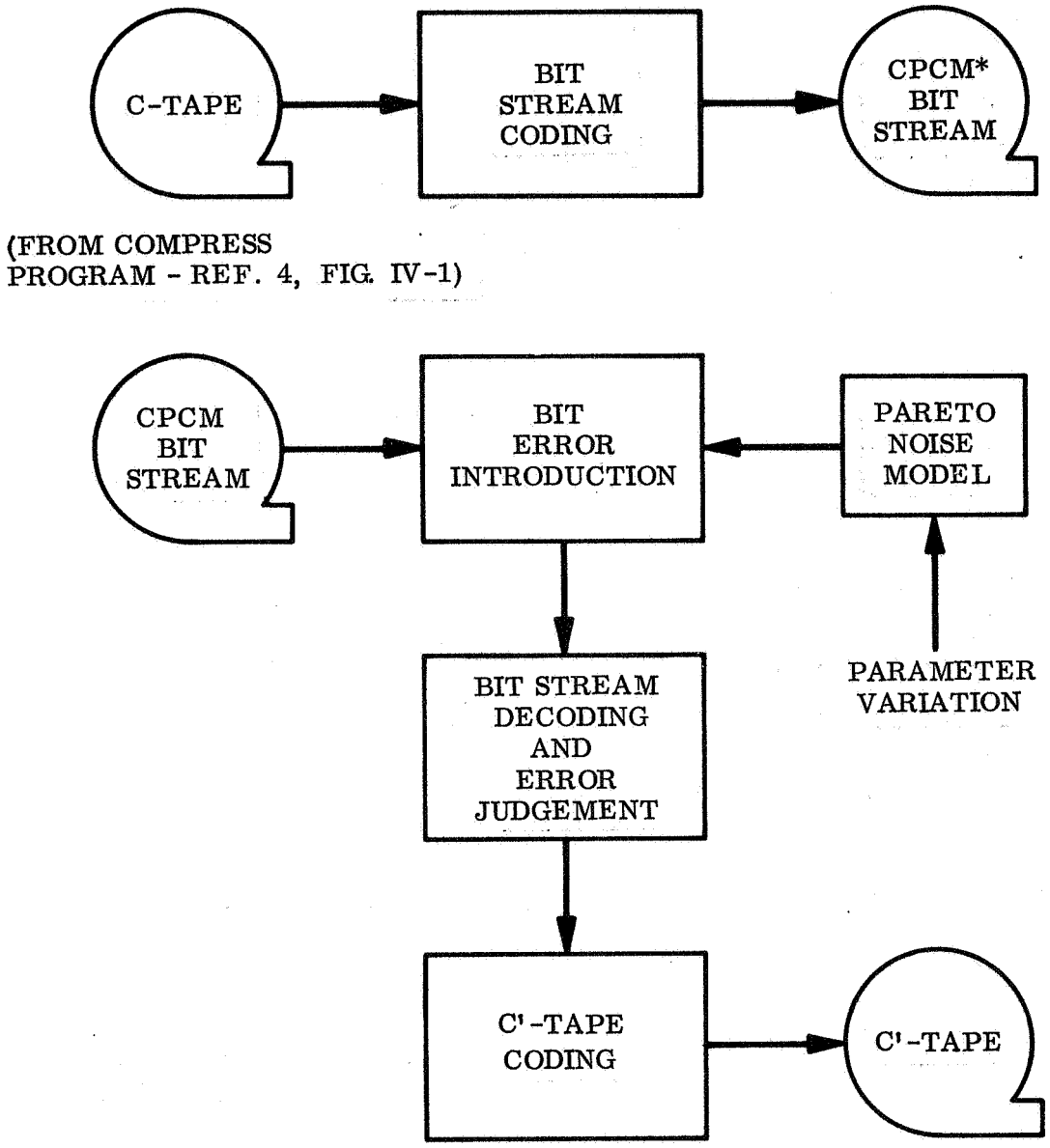
The purpose of this study was to investigate, by computer-simulatio , the effect of ground transm ission link noise on compressed data. The general approac was as f ows. Firs , data were compressed using the multichannel compression program. Next, the compressed data were coded into compressed transmission link bit streams. Errors were then ad ed to the bit streams, with the number of bits between errors following a Pareto distribution (Ref. 6) The contaminated data were then decoded into the original compressed data fo at and reconstructed These operations are shown in Figs 4-2, 4-3 and 4-10. The formats of the compressed transmitted data bit streams that were used a experimental model are s wn in Fig. 4-1. The major criteria used to evaluate the effec of link noise on compressed data were rms error, on a per-s sor basis, and the visually observed fid ity of the reco st ted data waveforms after contamination.

4.2.1 Total Rms Error for the Compressed Data

The error study was c ucted using S-6 data Three data compression selectors, ZOI, FOIDIS, and FOI4DG, were examined with two values of the Pareto noise model clustering fac or α The entire 241 seconds of GSFC S-6 data tape No. 53 (described in Sect. 2) was co pressed, coded, contaminated, and decoded Sensors 7 and 3 (Table 2-1) were reconstructed The rms errors observed in the contaminated data are shown in Table 4-3*

These errors seem quite high — much higher than the theoretical rms errors in most cases This is because the statistical sample size, was e fairly adequate when

*These figures include both the errors caused by noise-contamination and those caused by compression; Appendix I shows that, for the bit error rates examined, the error caused by compression is negligible compared with the total rms error.



*CPCM: COMPRESSED PULSE CODE MODULATION

Fig. 4-8 CPCM Transmission Link Simulation

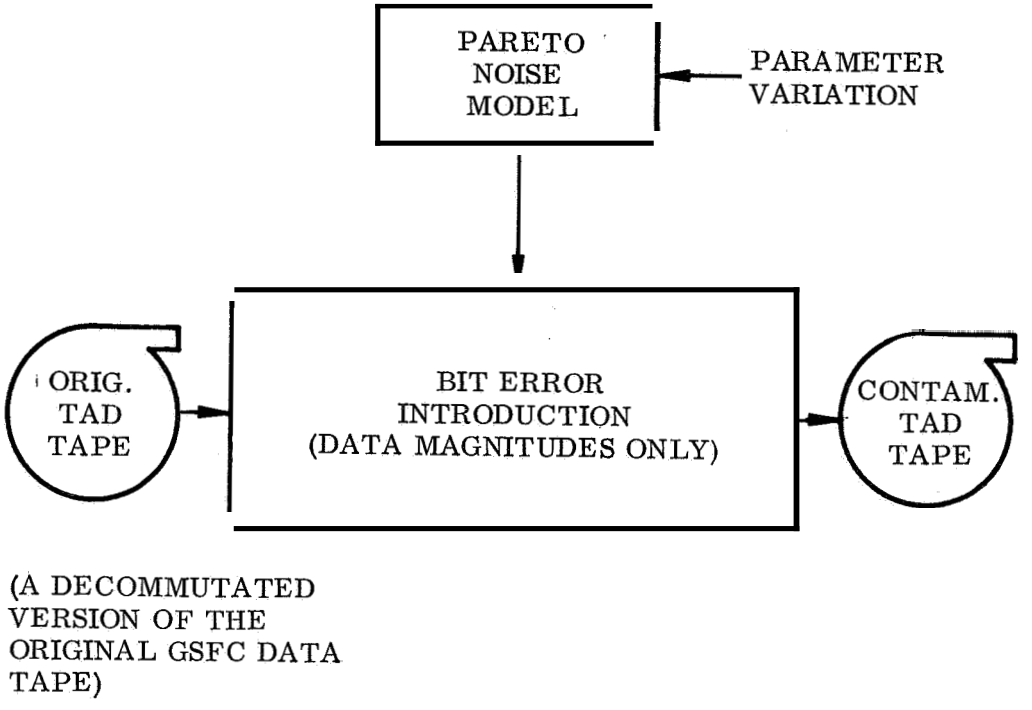
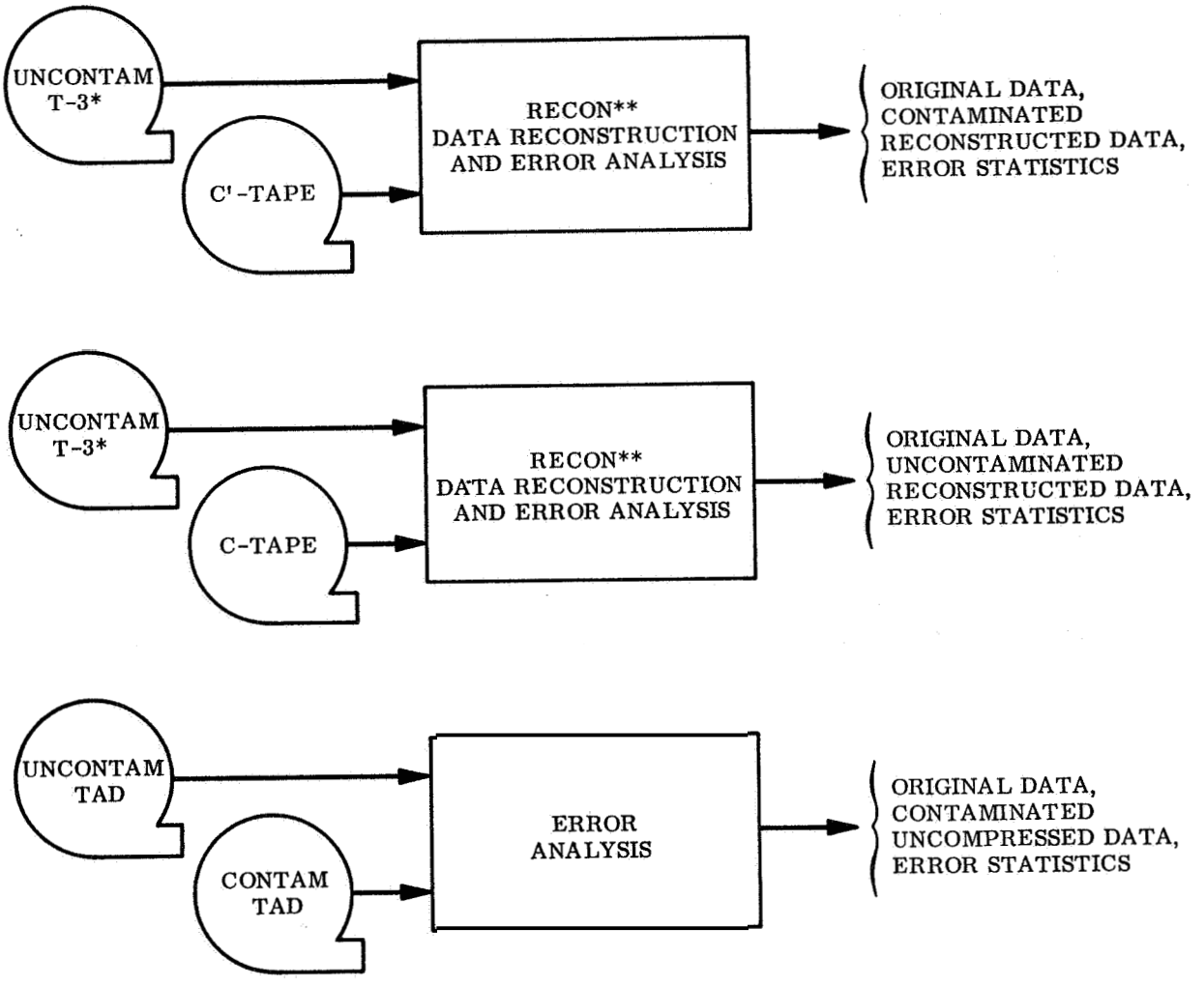


Fig. 4-9 PCM Transmission Link Simulation



NOTES: *OBTAINED FROM THE TAD TAPE, THE T-3 PROVIDES DATA FROM A SINGLE SENSOR ON A TIME/DATA MAGNITUDE BASIS
**COMPUTER PROGRAM PREPARED DURING PHASE I (REF. 4, FIG. IV-1)

Fig. 4-10 Data Reconstruction and Error Analysis

Table 4-3

RMS ERRORS FOUND IN RECONSTRUCTED DATA

Clustering Factor[a]	Bit Error Rate	Rms Error (% full scale)					
		Sensor 9			Sensor 11		
		ZOI	FOIDIS	FOI4DG	ZOI	FOIDIS	FOI4DG
0.3	1.35×10^{-4}	7.7	4.88	10.52	4.28	3.3	1.79
0.4	4.00×10^{-4}	9.2	6.38	8.00	4.06	3.54	3.94

(a) The maximum gap, L , was made equal to 200,000 bits.

considering the whole compress run, is simply not adequate when a single sensor is considered. This is demonstrated by the fact that the rms errors vary markedly from Sensor 9 to Sensor 11. Also, the rms error should increase when α is increased from 0.3 to 0.4, since the bit error rate increases. However, in two of the six cases shown above, the opposite occurs. This rms error reversal is seen to be independent of selector or sensor, which further emphasizes the inadequacy of sample size. If the sample size were increased by reconstructing all the data on the S-6 tape, much closer agreement with the theoretical values would seem likely.

4.2.3 Actual Faulty Line Segments

Various errors in reconstructed data were shown in an idealistic way in Par. 4.1 This paragraph will illustrate real-life situations, so that the ever-popular "average" situation may be complemented. Figure 4-11 shows a first-order interpolator-compressed and reconstructed waveform in which one endpoint on one of the line segments is incorrect. Notice the rather slight effect on the overall waveform. Figure 4-12 shows a first-order interpolator-compressed and reconstructed waveform where line segments were improperly inserted, and Fig. 4-13 shows a waveform from

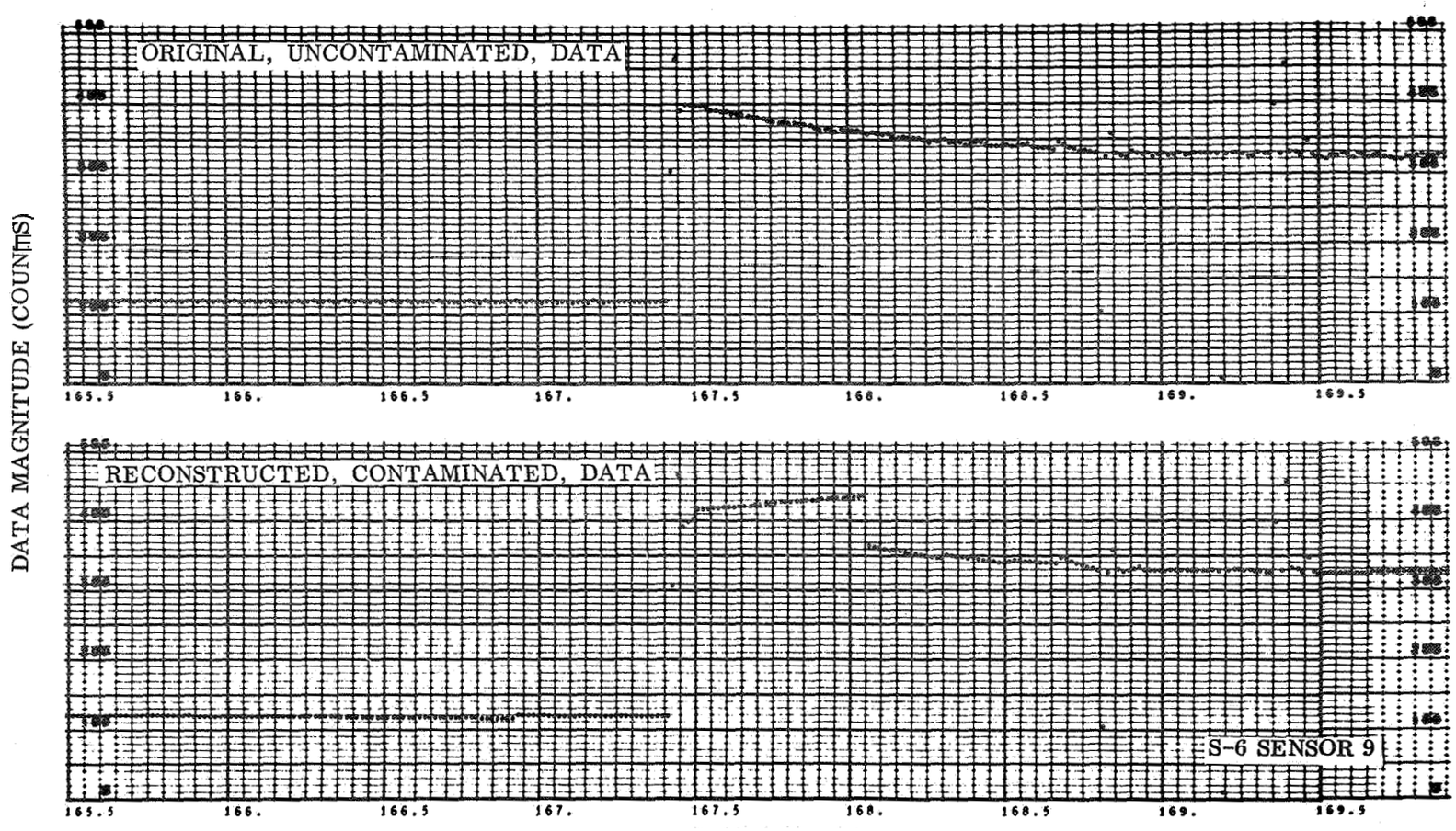


Fig. 4-11 Error Caused by Magnitude Bit Error, First-Order Interpolator

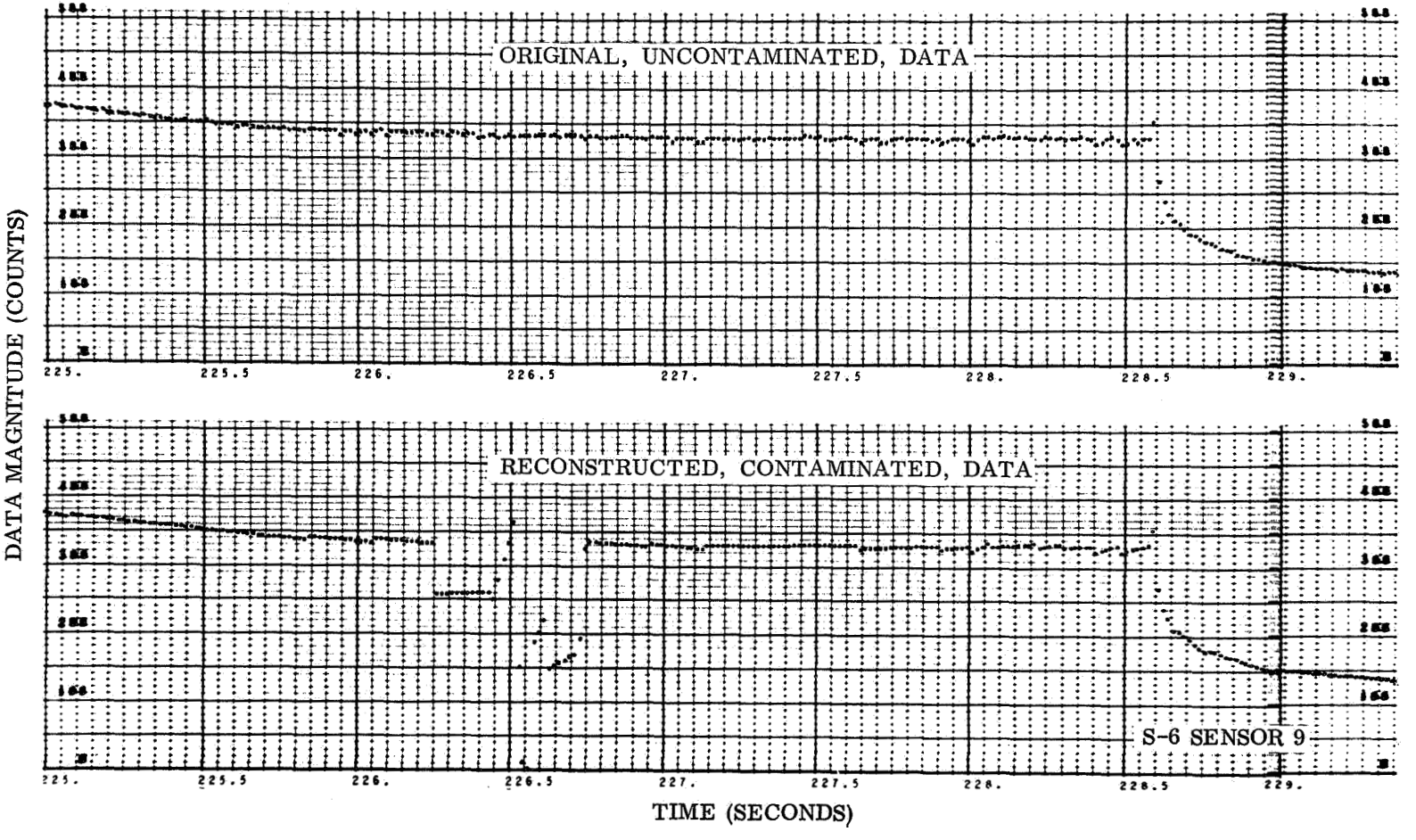


Fig. **4-12** Error Caused by Line Segment Additions With Wrong ID Assignments, First-Order Interpolator

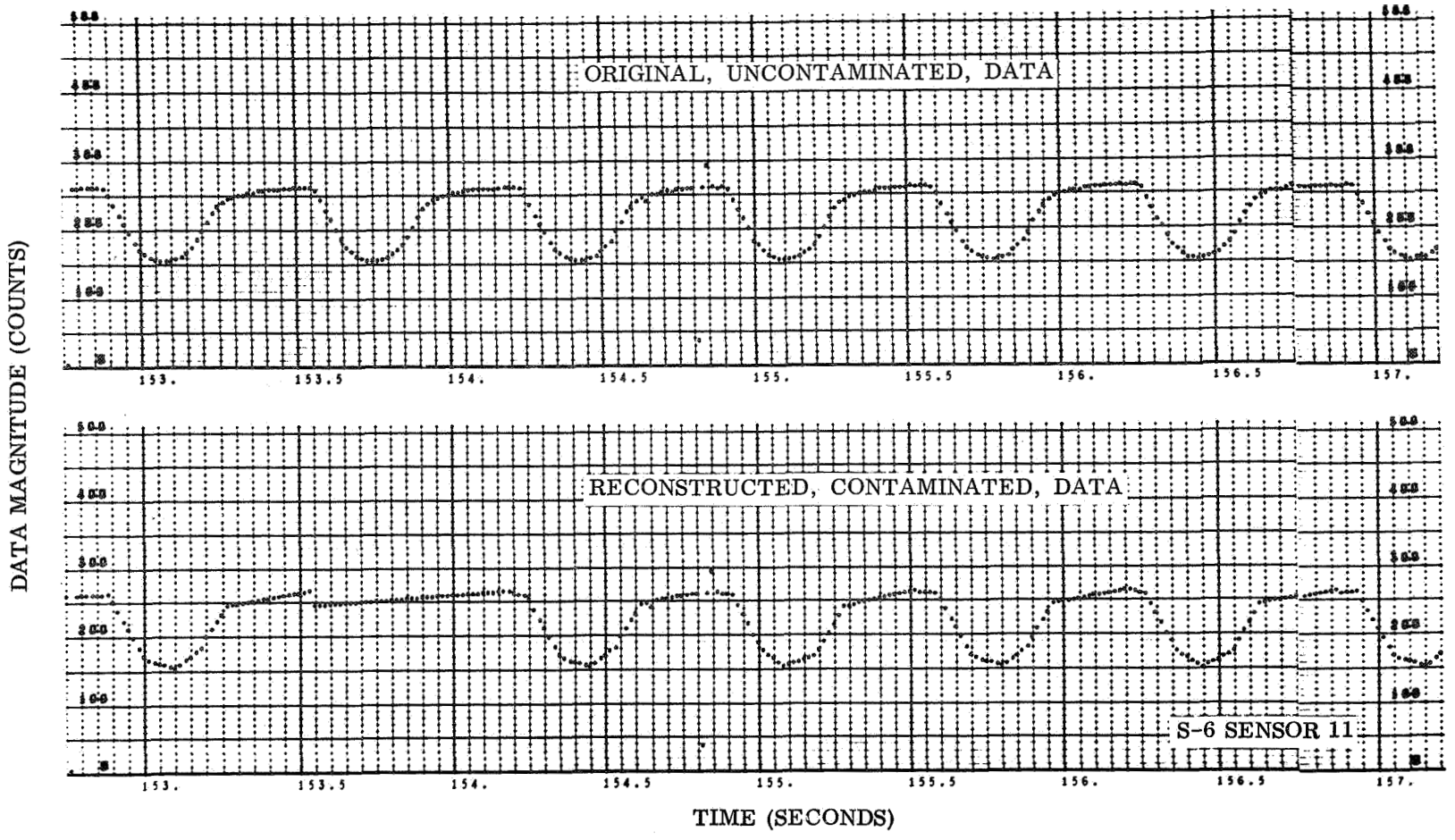


Fig. **4-13** Error Caused by Line Segment Eliminations With Wrong ID Assignments, **First-Order Interpolator**

which line segments were improperly eliminated.* Note that the overall effect is much greater when there is an error in ID rather than magnitude. This is true for two reasons: (1) the expected error in magnitude is greater when it is caused by a sensor ID error than when it is caused by an error in a bit stream magnitude word, and (2) a bit error in sensor ID may affect two line segments, while a bit error in magnitude affects only one. Figures 4-14 through 4-16 show the same type of error in the zero-order-compressed and reconstructed data. These six figures demonstrate what was pointed out in Par. 4.1. First, errors in magnitude have little effect compared with errors in sensor ID. Second, the error for ZOP/ZOI selectors is larger because every erroneously reconstructed data point has a large error, while in the case of the FOIDIS/FOI4DG selectors the individual error may approach zero as one end of the line segment is reached.

Extensive visual observations of a large number of reconstructed waveforms from the noise-contaminated data indicate that the aberrations that occurred in the waveforms were generally not as severe as were indicated by the relatively high value of rms error shown in Table 4-3. The apparent reason for this is that many of the erroneous line segments deviate from their true values by very large amounts. Thus, the contaminated waveforms were characterized by relatively few (compared with the total number) erroneous line segments that deviated markedly from their original positions.

4.2.3 Bit Errors in the Uncompressed Data

To complete the experimental study, the effect of bit errors on uncompressed data was considered. Bit errors with the Pareto distribution were added to the original uncompressed data, using two values of the clustering factor α. The rms error for the entire S-6 data as well as for Sensor 9 and 11 was found, and is shown in Table 4-4.

---

*The probable reason for the addition or elimination of more than one line segment in Figs. 4-12 and 4-13 was that a large number of consecutive sensor ID's in a bit stream record were incorrectly assigned, because of a flag bit error which occurred early in the record. This incorrect assignment could involve a single active sensor many times in one record.

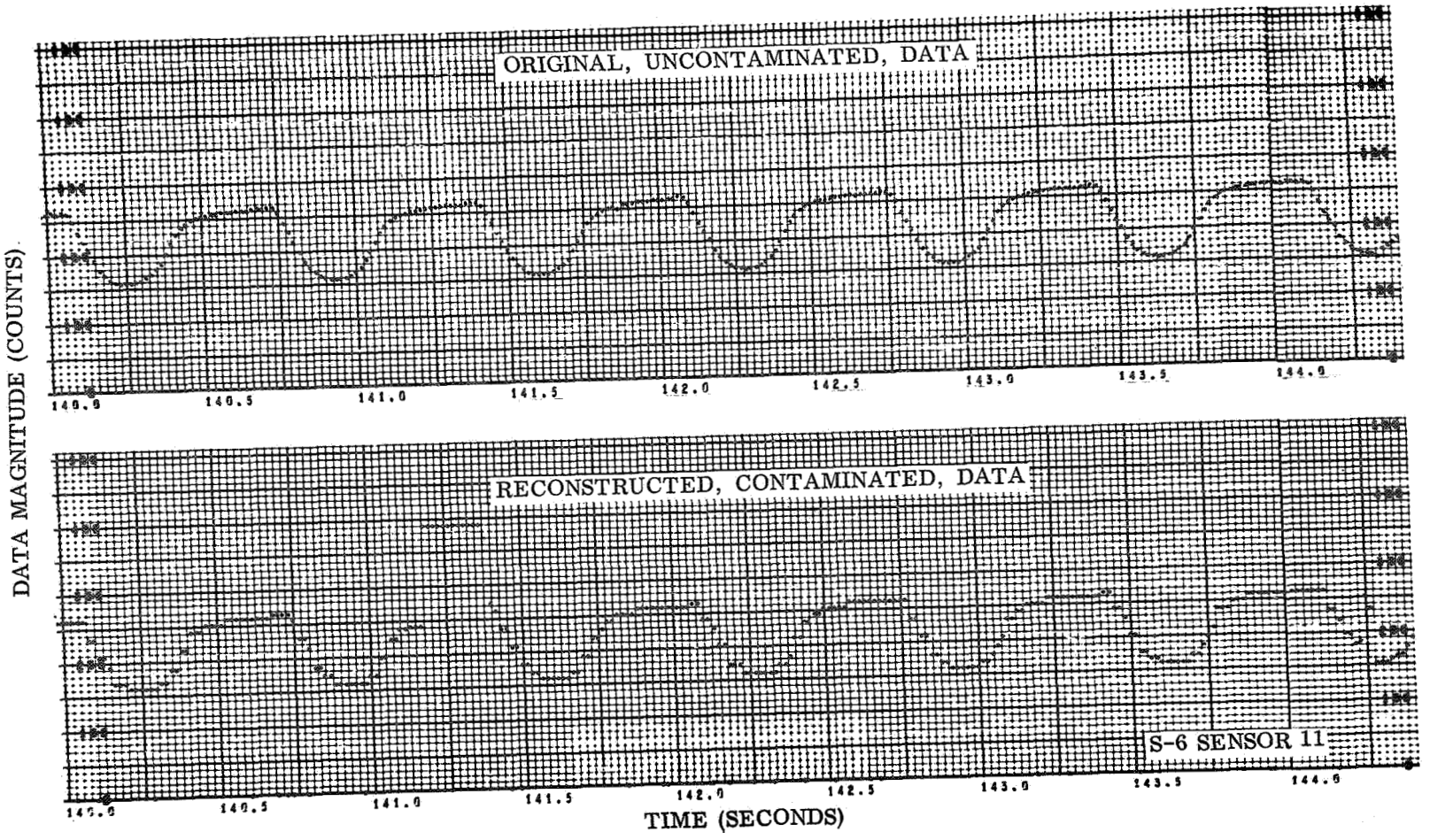


Fig. 4-14 Error Caused by Magnitude Bit Error, Zero-Order **Selector**

LOCKHEED MISSILES & SPACE COMPANY

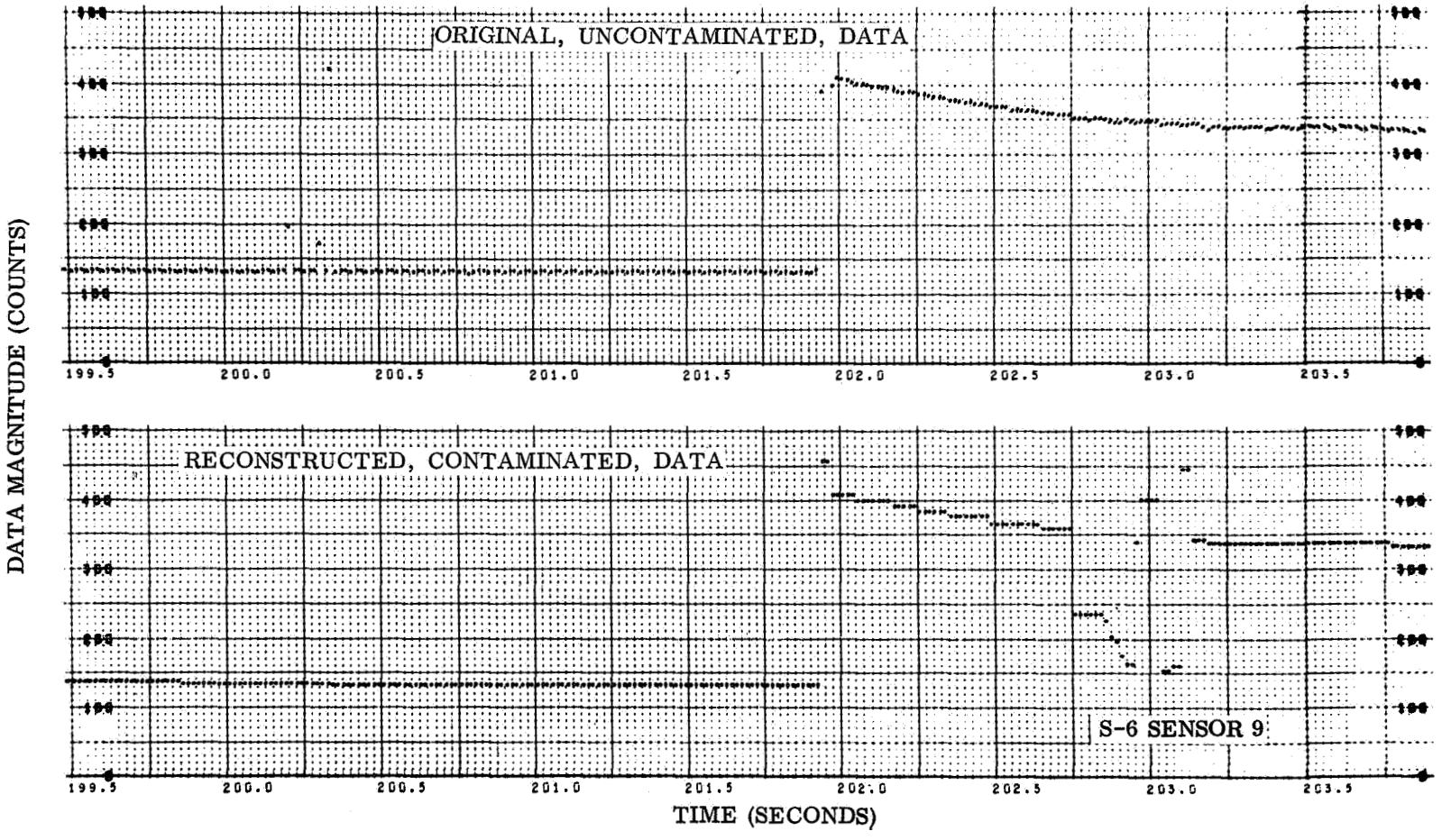


Fig. 4-15 Error Caused by Line Segment Additions With Wrong ID Assignments, Zero-Order Selector

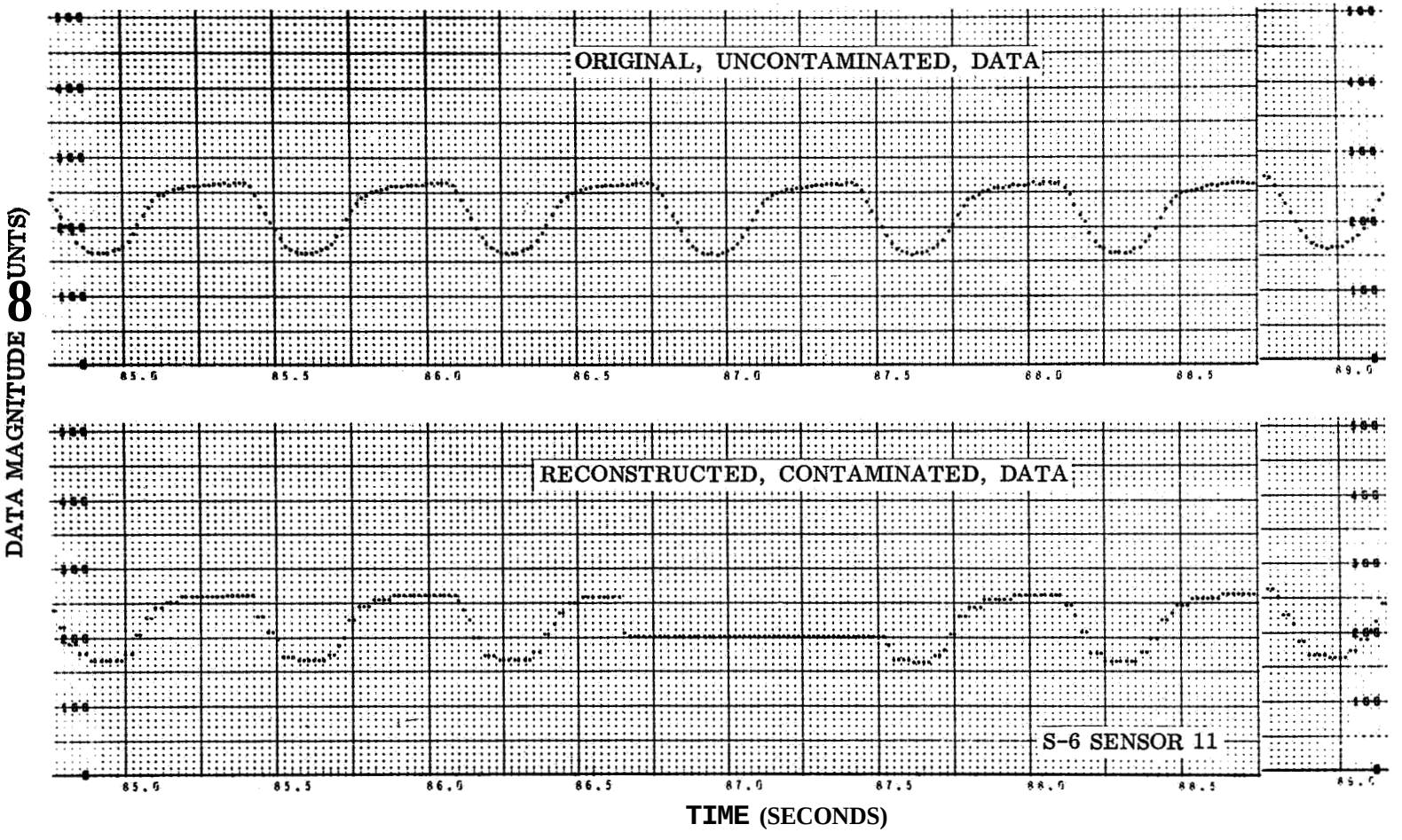


Fig. 4-16 Error Caused by Line Segment Eliminations With Wrong ID Assignments, Zero-Order Selector

Table 4-4

RMS ERRORS FOUND IN UNCOMPRESSED DATA

Clustering Factor	Bit Error Rate	Rms Error (% full scale)		
		Overall Error	Sensor 9	Sensor 11
3	1.38×10^{-4}	0.674	0.54	0.685
4	4.00×10^{-4}	1.21	1.13	1.46

These values agree very well with the theoretical values — much better than in the case of the errors in the compressed data. However, in the case of the uncompressed data the sample size was larger by a factor of the compression ratio. Hence, closer agreement is expected.

4.3 THEORETICAL-EXPERIMENTAL COMPARISON

Paragraphs 4.1 and 4.2 discussed theoretical and experimental error due to bit errors, with results presented in tabulated form. This paragraph combines this information into one graph for comparison and study. Figure 4-17 shows rms error versus bit error rate; the solid lines represent the theoretical results* and the individual points the experimental output.

It is also worthy of note that Fig. 4-17 is plotted for bit error rates on the order of 10^{-4} errors/bit, so that a comparison could be made between the two phases of the study. A more realistic bit error rate of 10^{-6} would produce errors of 0.181 percent for FOIDIS/FOI4DG and 0.248 percent for ZOP/ZOI, based on the theoretical study. The error in the uncompressed case would be 0.0577 percent. It is evident that with realistic bit error rates the error in reconstructed data caused by errors in the compressed PCM bit stream would be of the same order of magnitude as the error

*The compression ratio is assumed to be greater than 5, so the error encountered with the FOIDIS and FOI4DG selectors is almost invariant with compression ratio.

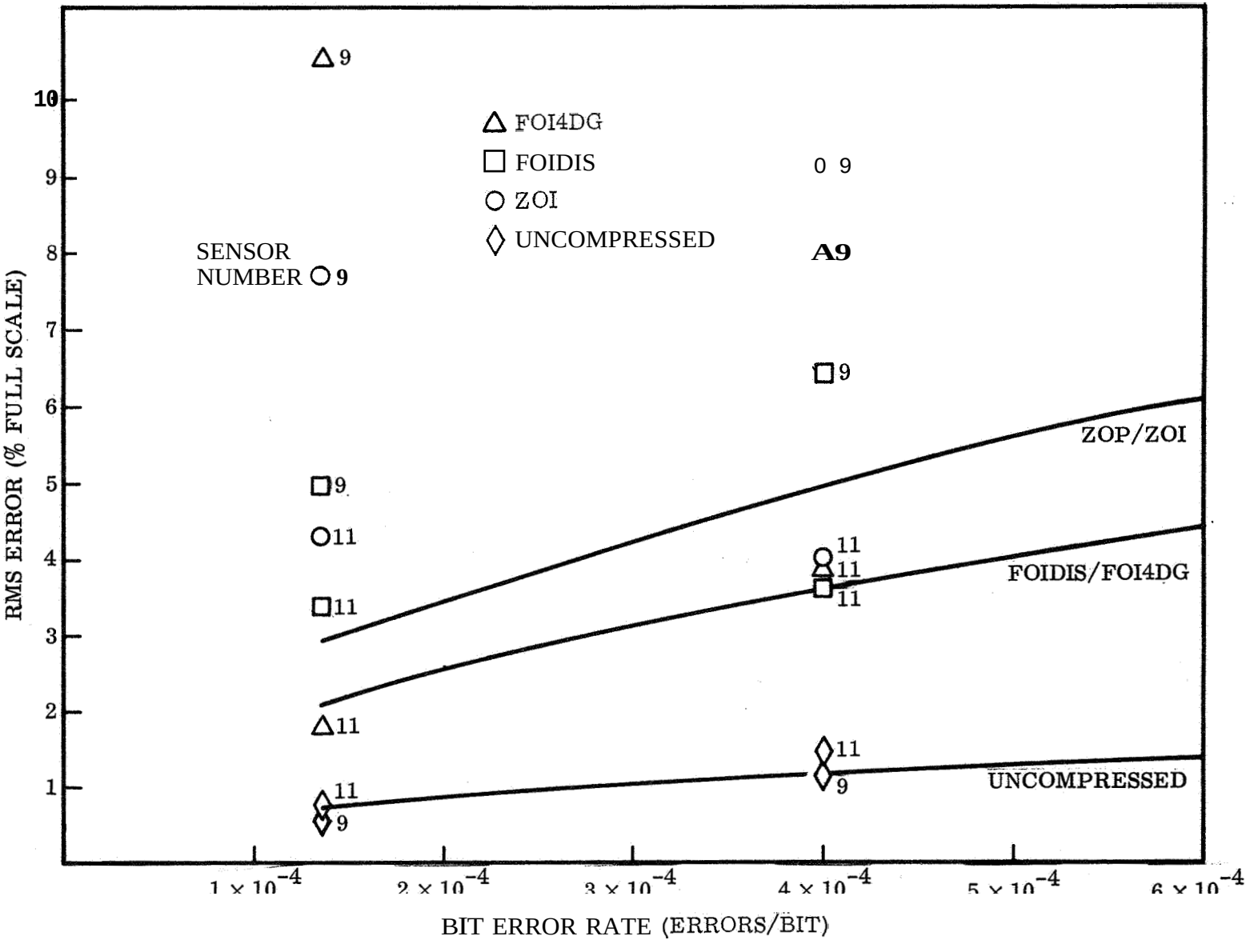


Fig. 4-17 Rms Error Versus Bit Error Rate

caused by compression itself. The abnormally high bit error rates used in the empirical study were chosen because of the limited statistical sample size.

4.4 CONCLUSIONS AND RECOMMENDATIONS

The following conclusions and recommendations are presented as a result of the transmission link error study.

- From an extensive visual examination of a large amount of reconstructed data from contaminated bit stream, it is concluded that for the bit stream formats and error rates used in this study, contaminated reconstructed waveforms are characterized by relatively few (compared with the total number) erroneous line segments that deviate markedly from their true positions. This characteristic generally accounts for the relatively high rms errors observed.
- Realistic bit error rates on the order of 10^{-6} errors per bit would produce rms errors of the same order of magnitude as the error caused by compression itself. Furthermore, even with very high rms errors, relatively few line segments are actually in error. Therefore, it is concluded that at normal operating signal-to-noise ratios, ground transmission link noise will not degrade the compressed data to any appreciable extent, if compressed PCM formats and bit stream decoding procedures are designed to minimize the effect of link noise.
- In an empirical analysis of this type, rms error alone by no means presents a complete picture of the extent of data waveform degradation caused by link noise; visual observation of the reconstructed waveform is also necessary. However, the calculation of theoretical rms error is worthwhile, so that comparisons can be made among compression methods, compressed data formats, and decoding techniques.
- Since errors in flag bits cause many line segments to be affected, it is strongly recommended that the flag bit be transmitted redundantly, as was done in the empirical study with the time-position word. Although this would decrease the bit compression ratio slightly, it would greatly reduce the effect of bit errors on compressed data.

Section 5
COMPRESSION MODEL TRADEOFF ANALYSIS

Reference 4, Par. 6.4, presented a tradeoff analysis involving six selectors, which was performed to determine the most desirable data compression model. During Phase II, existing selectors were evaluated with new telemetry data, new selectors were examined, and a better understanding of the performance criteria was acquired. The new selectors included the exponential interpolator (EXPINT) and the short-term frequency component selector (STFC). The new selectors are combined with the zero-order predictor (ZOP), the zero-order interpolator (ZOI), three first-order interpolators, FOIDIS, FOIOOT and FOI4DG, and bit plane encoding (BPE), to provide a total of nine data compression selectors for analysis. (Separate evaluations are made for binary codes and Gray-codes BPE.)

The performance criteria used for the analysis in Ref. 4 included:

- Bandwidth compression ratio
- Reconstructed data fidelity
- Sensitivity to data class
- Sensitivity to sensor noise
- Effectiveness with adaptive queuing control
- Sensitivity to link noise
- Selector complexity
- Ease of reconstruction

All of these criteria except sensitivity to data class are used in this study. This criterion was eliminated because considerable doubt has arisen concerning its validity as a measure of performance. The same general method of performance factor application described in Ref. 4 was used in this analysis. The seven performance criteria are weighted for both ground-based and spaceborne data compression; the method of this weighting is described. Next, performance factors are assigned to each selector,

with respect to each criterion. Finally, weighted performance factors for the nine selectors are determined for both ground-based and spaceborne compression, by summing the products of the weightings and performance factors over all the criteria. Thus, by this method the selectors are ranked.

5.1 PERFORMANCE CRITERIA

In this paragraph the method of determining the various performance factors is shown. If the method is identical to that used in Ref. 4, only the formula is presented; a detailed presentation may be found in Ref. 4, Par. 6.4. If the performance criterion has been further developed or modified, a more detailed presentation is given.

5.1.1 Bandwidth Compression Ratio

The formula is identical with that presented in Ref. 4 except that a different number of sensors was used*

$$f_p\,(\bar{N}_B)_i = \frac{0.95 \sum_{\text{sensors}} [(\bar{N}_B)_{K=2} + (\bar{N}_B)_{K=4}]_i}{\sum_{\text{sensors}} [(\bar{N}_B)_{K=2} + (\bar{N}_B)_{K=4}]_{max}}$$

where

$f_p\,(\bar{N}_B)_i$ = bandwidth compression ratio performance factor for Selector i

$\bar{N}_B$ = bandwidth compression ratio

K = tolerance in units

The denominator is the maximum sum obtained for the models tested.

*In every case where a summation over "sensors" is involved and the performance factor was calculated and not estimated, the following sensors were used: S-6 Sensors 2, 3, 7, 9, 11 and S-49 Sensors 14, 37, 50, 60, 63.

5.1.2 Reconstructed Data Fidelity

$$f_p(F)_i = \frac{0.95 \sum_{sensors} [(E_R/K)_{K=2} + (E_r/K)_{K=4}]_{min}}{\sum_{sensors} [(E_R/K)_{K=2} + (E_R/K)_{K=4}]_i}$$

where

$f_p(F)_i$ = fidelity performance factor for Selector i

E_R = rms error, in units

K = tolera e in its

The numerator is the minimum sum obtained for the models tested.

5 1 3 Sen ity to Sensor Noise

$$f \quad S)_i = \frac{0.95 \sum_{sensors} [(\bar{N}_B)_{K=4} - (N_B)_{K=2}]_{min}}{\sum_{sensors} [(\bar{N}_B)_{K=4} - (N_B)_{K=2}]_i}$$

where

$f_p(S)_i$ = sensor noise sensitivity performance factor for Selector i

$\bar{N}_B$ = bandwidth compression ratio

K = tolerance, in units

The numerator is the minimum sum obtained for the models tested.

5.1.4 Effectiveness With Adaptive Queuing Control

$$f_p(A)_i = \frac{0.95 \sum_{sensors} [(\bar{N}_B)_{K=4} - (\bar{N}_B)_{K=2}]_i}{\sum_{sensors} [(\bar{N}_B)_{K=4} - (\bar{N}_B)_{K=2}]_{max}}$$

where

$f_p(A)_i$ = adaptive queuing control effectiveness performance factor for Selector i

$\overline{N}_B$ = bandwidth compression ratio

K = tolerance, in units

The denominator is the maximum sum obtained for the models tested.

5 1 5 Sensitivity to Noise in the Compressed Data Transmission Link

This performance criterion is now better understood because of the transmission link noise studies performed in Phase II. The results of the experimental link noise study varied so much that the theoretical study is used to [illegible] performance factors. Because of the absence of a standard for the other criteria, their performance factors are calculated on relative bases, and are multiplied by 0.95 to allow for a better selector, as yet unknown. However, for this criterion a convenient standard is available: the rms error caused by link noise with no compression. Therefore the performance factor for sensitivity to transmission link noise is

$$f_p(L)_i = \Sigma$$

where

$f_p(L)_i$ = link noise sensitivity performance factor for Selector i

E_{NC} = rms error due to link noise with no compression

Σ_i = rms error due to link noise for Selector i

5.1.6 Selector Complexity

The same estimates as those used in Ref. 4 were used here

5.1.7 Ease of Reconstruction

Estimates of these performance factors have been revised due to increased knowledge of the problems of reconstruction. The problems are notably in the areas of time delay and buffer capacities needed when attempting to reconstruct data which have been compressed using an interpolator selector. Because of these problems, the performance factors for all the interpolator selectors have been reduced.

5.2 WEIGHTINGS OF PERFORMANCE CRITERIA

To obtain a measure of overall performance, it was necessary to weight the seven performance criteria for both ground-based and vehicle-borne data compression. This was done by first breaking the seven performance criteria into three groups: those which predominately affect (1) the data user, (2) the communication link, and (3) the economic considerations. The three measures of error, reconstructed data fidelity, sensitivity to sensor noise, and sensitivity to link noise, were assigned to the data user. The two measures of channel capacity, bandwidth compression ratio and effectiveness with adaptive queuing control, were assigned to the communication link. Finally, complexity and ease of reconstruction were classed as economic considerations. Each of these criteria was rated within each group for both ground-based and vehicle borne applications, as shown in Fig. 5-1b. Next, relative weights were assigned to each of the three main groups, as shown in Fig. 5-1a. The products of the group weighting factors and the criterion weighting factors within these groups yielded the overall weighting factors which were used in the final evaluation. Note that the weighting factors assigned to the three main groups may vary from one space project to another, and perhaps even within one project the values may be somewhat arbitrary. However, it is believed that this method of criterion weighting is more accurate than the direct assignment of weighting factors to the performance criteria, as was done in the Phase I analysis.

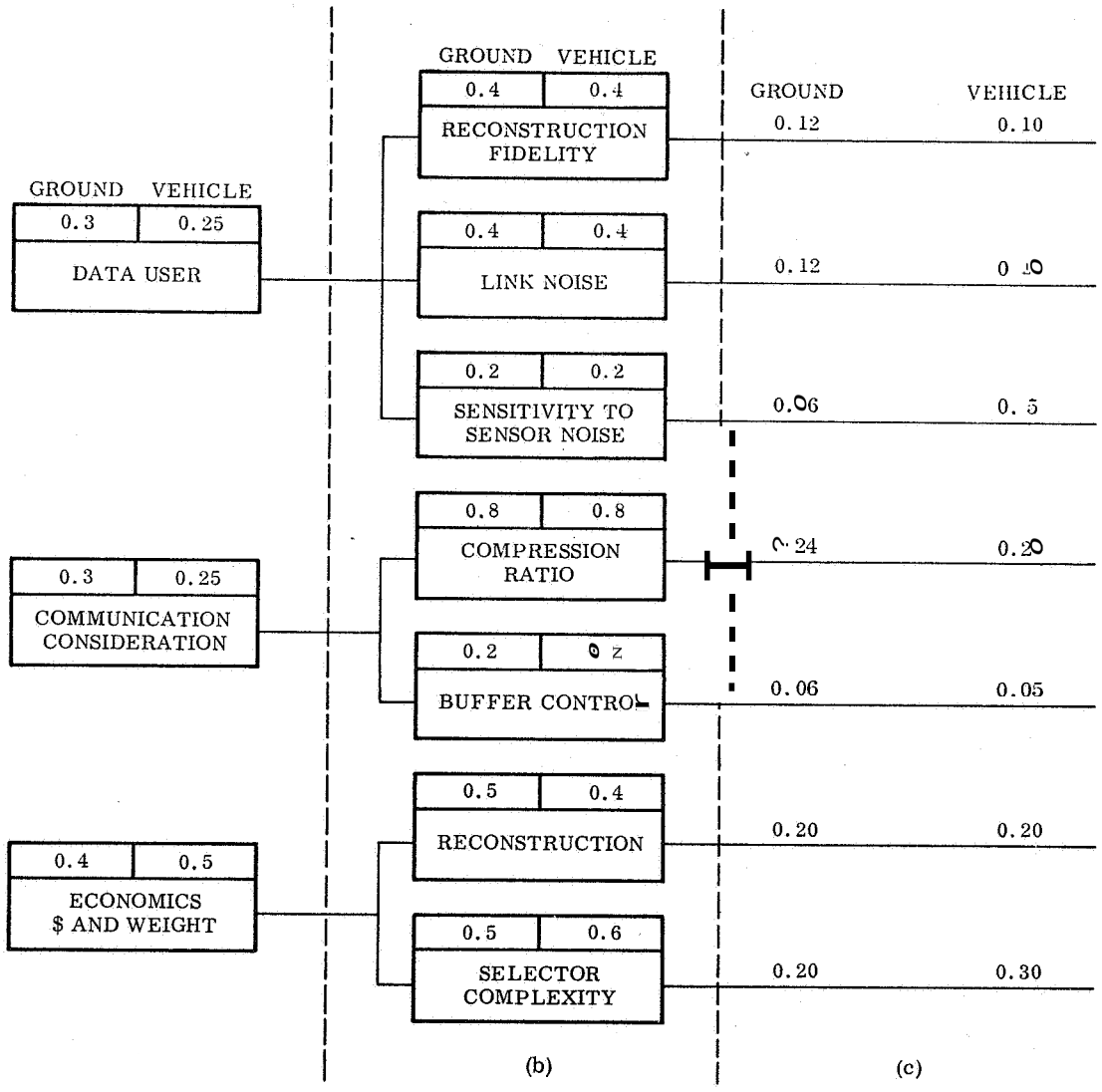


Fig. 5-1 Performance Criteria Weighting

5 3 RELATIVE COMPRESSION MODEL PERFORMANCE

The performance factors which were derived for the nine compression models are shown in Table 5-1. Also shown are the weighting factors derived in Fig. 5-1. Note that, as with the Phase I tradeoff analysis, many of the performance factors were estimated. However, for Phase I, 52.1 percent of the performance factors were estimated, while in the present study only 43 percent of the values were estimated. Clearly, although progress has been made, there is still much work needed to obtain more measures of performance, particularly in the area of complexity and ease of reconstruction.

Table 5-1 shows the relative rankings of the nine models for both ground-based and spaceborne data compression. For both applications, the zero-order predictor (ZOP) ranks highest, followed by the FOIOOT and FOIDIS selectors. The FOI4DG selector ranks fourth in the ground-based application but ranks fifth for spaceborne use behind the ZOI selector

It is interesting to note that the ZOP selector, which ranked last among the seven models evaluated in terms of combined bit compression ratio and rms error (Table 2-8), ranks first in Table 5-1, while the FOI4DG selector, which was No. 1 in Table 2-8, shows up poorly in Table 5-1. The reason, of course, lies mainly in the simplicity of the ZOP selector and in the relative ease of reconstructing ZOP-compressed data*

It should be noted that the EXPINT performance factors for bandwidth compression ratio, sensor noise sensitivity, and queuing control effectiveness were derived by relating its performance with $b_1 = b_2 = 60$ to that of FOIDIS on a proportion basis

*Note that the ZOP selector did not rank as high as a result of the Phase I tradeoff analysis (Ref. 4, Par. 6.4) as it does in Table 5-1. The two main reasons for its increased performance are a new evaluation of predictors and interpolators with regard to reconstruction, and increased weightings assigned to the reconstruction criterion.

Table 5-1

COMPRESSION MODEL PERFORMANCE FACTORS

Spaceborne Weighting Factor	0.20	0.10	0.05	0.05	0.10	0.30	0.20	Overall Weighted Performance Factor	
Ground-Based Weighting Factor	0.24	0.12	0.06	0.06	0.12	0.20	0.20		
Compression Model	Performance Factor							Ground-Based	Space-borne
	Bandwidth Compres-sion Ratio	Recon-structioin Fidelity	Sensor Noise Sensi-tivity	Queuing Control Effec-tiveness	Link Noise Sensi-tivity	Selector Complexity	Ease of Recon-struction		
ZOP	0.228	0.950	0.328	0.250	0.233	0.950(a)	0.950(a)	0.611	0.668
ZOI	0.466	0.609	0.150	0.541	0.233	0.850(a)	0.450(a)	0.514	0.557
FOIDIS	0.700	0.608	0.112	0.736	0.366	0.700(a)	0.350(a)	0.546	0.559
FOIOOT	0,950	0.536	0.088	0.929	0.300(a)	0.700(a)	0.350(a)	0.599	0.604
FOI4DG	0.830	0.577	0.086	0.950	0.366	0.500(a)	0.350(a)	0.544	0.532
Bit Plane Encoding (Binary)	0.261	0.404	0.950	0,086	0.200(a)	0.600(a)	0.850(a)	0.488	0.514
Bit Plane Encoding (Gray)	0.370	0.404	0.701	0.116	0.200(a)	0.600(a)	0.850(a)	0.500	0.525
EXPINT	0.725	0.625(b)	0.134	0.615	0.366(a)	0.550(a)	0.350(a)	0.518	0.517
STFC	0.150(a)	0.350(a)	0.900(a)	0.058(a)	0.250(a)	0.400(a)	0.450(a)	0.336	0.348

(a) Estimated value.
(b) Based on error values shown for $b_1 = b_2 = 60$ in Table 2-6.

for the sensors and tolerances shown in Table 2-6, and then extrapolating to the remaining sensors and tolerances, using the FOIDIS values shown in Table 5-1.

5 4 CONCLUSIONS AND RECOMMENDATIONS

The main conclusion that can be drawn from this study is that the zero-order predictor (ZOP) is the best all-around selector, predominantly because of its simple implementation and ease of reconstruction. The more exotic selectors perform much better in compression efficiency, but they have the disadvantages of complexity and reconstruction problems.

Thus, it is recommended, for the performance weightings chosen, that the ZOP selector be used for both spaceborne and ground-based compression. However, it must be pointed out that, in a system where the communications link and the data user were given more consideration and economic factors considered less, a more exotic first-order interpolator may be preferred over the ZOP selector.

Section 6
REAL-TIME COMPUTER ANALYSIS

6 1 INTRODUCTION

The purpose of this study was to analyze the effect of data compression when utilized at a remote or central control station for real-time operations. Breifly, the primary elements of the task were:

- Determine the current and anticipated requirements of real-time computer operations, data processing, and compression, which were to include
  - Type, quantity and quality of measurements
  - Complexity, combination and quantity of calculations
  - Modes of display
- Compare differences in calculation times of a medium-size computer utilizing compressed and actual data formats. The data were assumed to be stored in the computer
- Compare differences in handling compressed and standard real time inputs for a medium sized computer

6.1.1 Spacecraft Selection

The orbiting Geophysical Observatory (OGO) was selected by the Goddard Space Flight Center (GSFC) as the model for this study because it is typical of a family of general-purpose spacecraft designed specifically to carry a number of related and/or unrelated experiments on a variety of space missions. In addition, PCM Telemetry is utilized, which is typical of the systems in use today and expected to be in use for the next few years.

6.1.2 Spacecraft Telemetry System

The OGO spacecraft uses a PCM/FM telemetry system with a minor frame of 128 words (9-bits/word). The information is transmitted from the spacecraft at a 64 kb/sec rate in the normal operational mode, and at a 128 kb/sec rate in the playback mode (on-board tape recorder). Three 128-word (9-bit) subcommutators complement the telementry configuration. A minor frame switching array allows up to 34 different formats on command. (See Table 6-1.)

Table 6-1
MINOR FRAME FORMAT MODES

Standard
Accelerated subcommutator
Experimenter

TOTAL

[illegible] (ASC) switches the information contained on subcommutator Number 2 (Subcom 2), containing critical spacecraft systems meas[illegible] ments, onto the main frame. The ASC mode is used, primarily, during launch, an[illegible] for spacecraft command and controlling after orbital insertion. This mode is also used for periodic status checks.

[illegible] periods of experiment measure-
ment. [illegible] modes are utilized during optimum periods of experiment measure-

6.1.3 Telemetry Measurement Assignment Schedule

The types and quantity of measurements for a typical OGO spacecraft are presented in Table 2-2. Characteristics of each measurement were not available at GSFC.

6 2 RΣA

[illegible] real-time command and control ope[illegible]
Center [illegible]

[illegible] after insertion is to: [illegible] into planned orbit. The f[illegible]

- Command and control the spacecraft in the proper attitude [illegible] required by the experiment and/or [illegible]
- Monitor the spacecraft's systems operation.

The layout of this center is diagrammatically sho[illegible]
6-3 are sket[illegible]

[illegible] center equipment and Table 6-3 presents the names of the software [illegible] operation. Table 6-2 is a list [illegible] utilized in conjunction with real-time operations.

The real-time operation [illegible] all phases from the soft[illegible]

- [illegible]
  - [illegible] equipment checkout, diagnostic, and verificatic [illegible] ams
  - Real time software
    - MONITOR – Controls I/O, limit checks, and scl [illegible] eervice and special purpose (SSP) operations.
    - SERVICE AND SPECIAL PURPOSE (SSP) –
      - Updates MONITOR subprograms
      - Spacecraft system test
      - Experiment tests

At the time this study was conducted, GSFC was contemplating the replacement of the current SDS-920 computer with the SDS-930. The study therefore examined the effect of compressed data on the real-time analysis system, utilizing either the 920 or the 930.

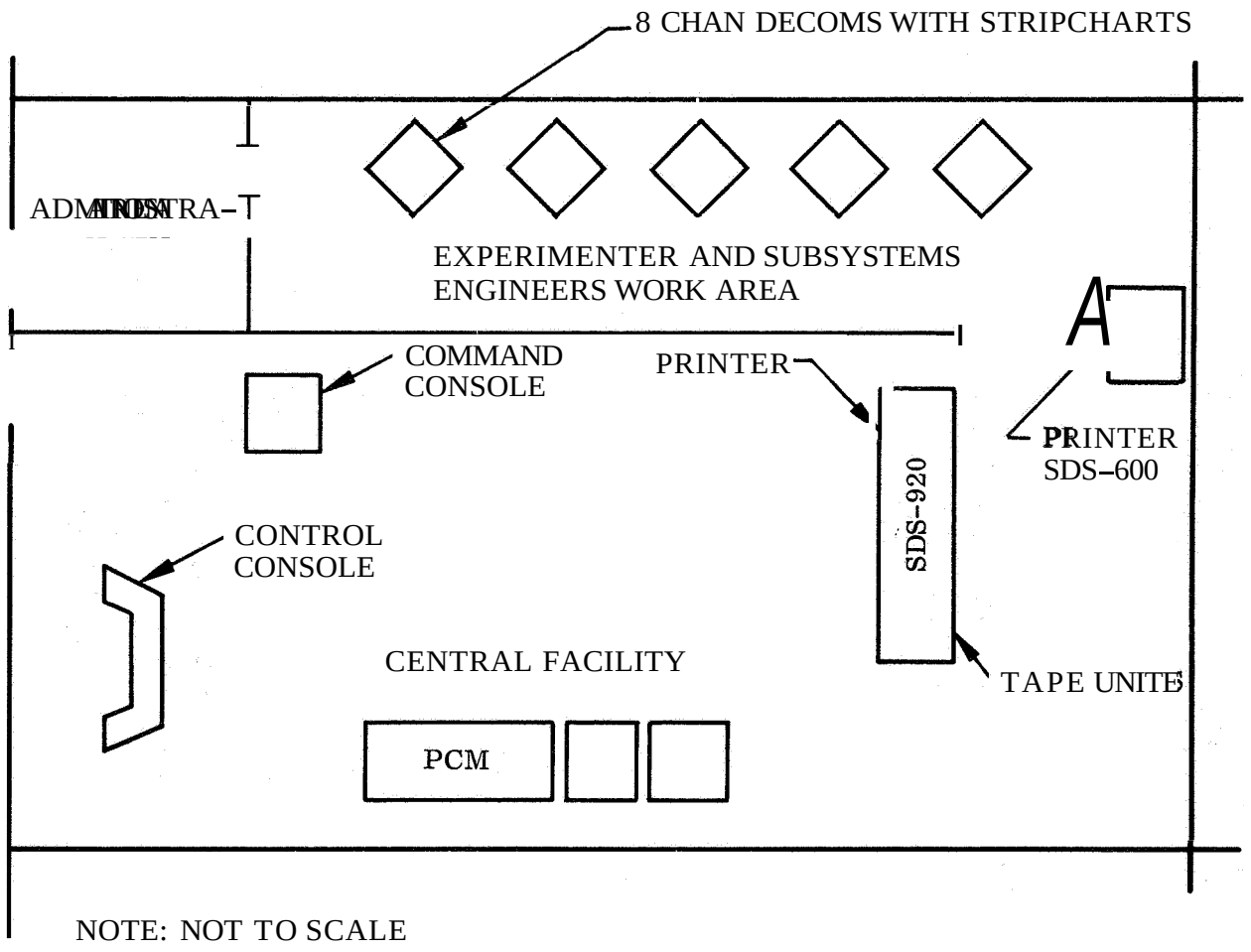


Fig. 6-1 Layout of OGO Operations Control Center

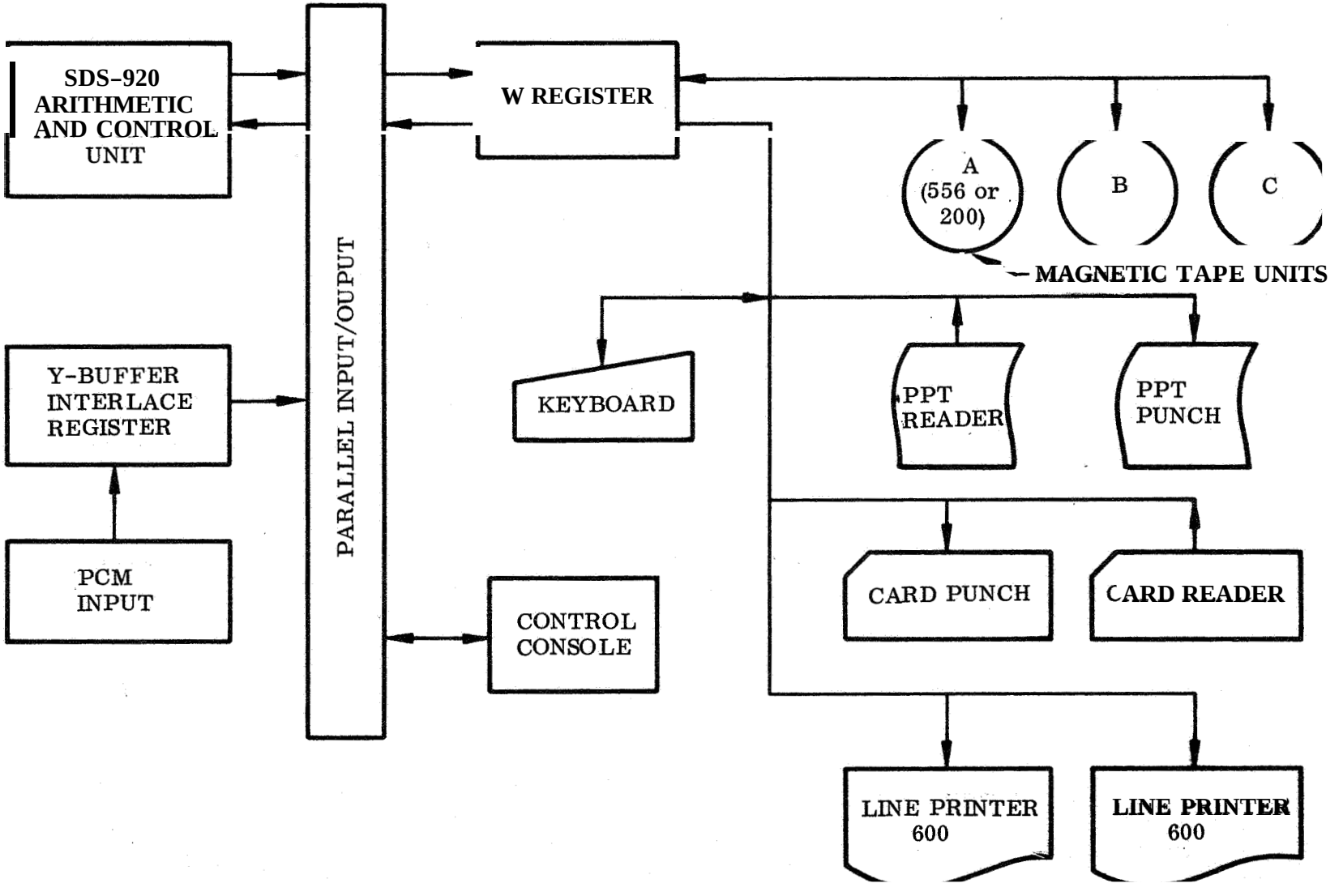


Fig. 6-2 Input/Output Device for OGO Real-Time Computer System

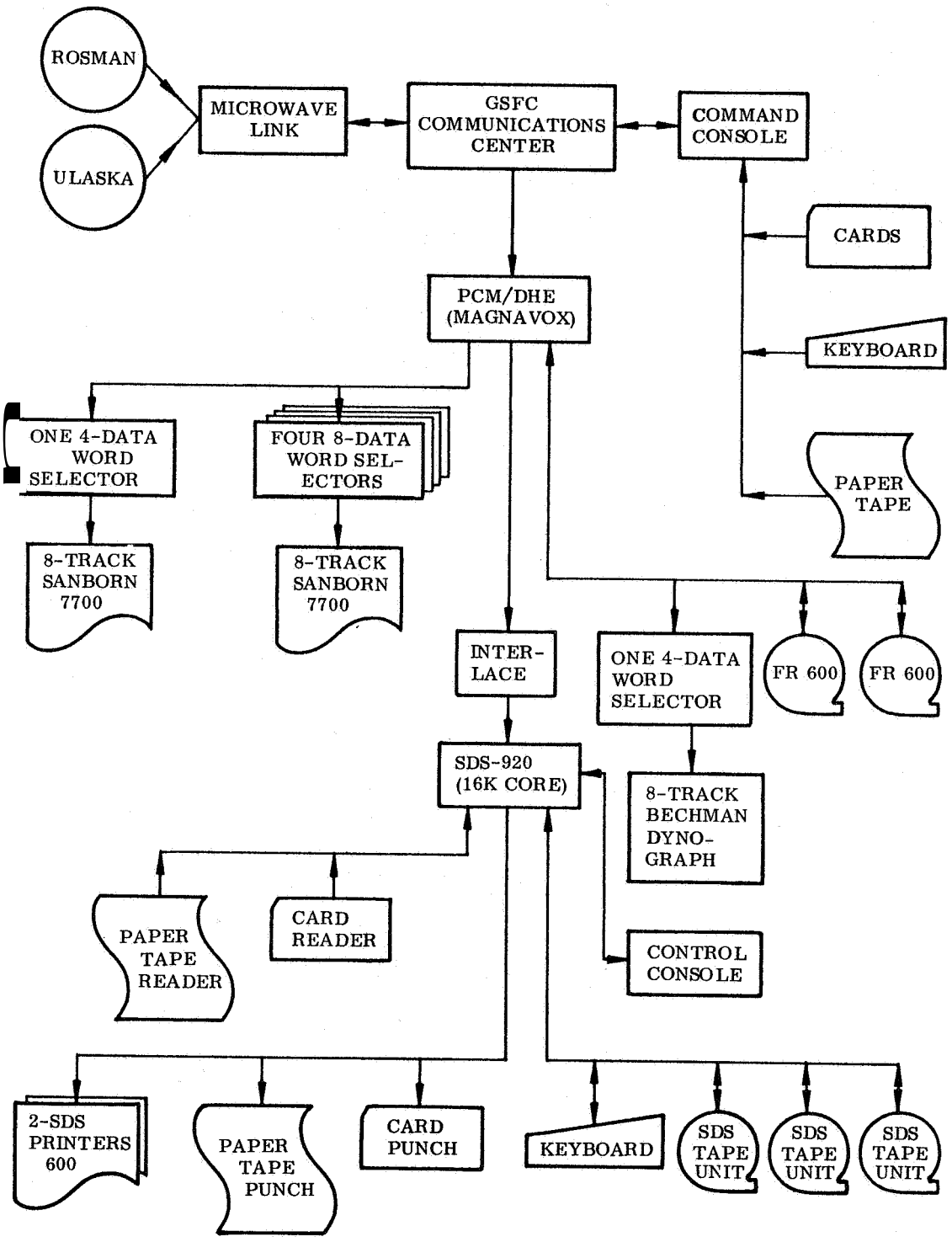


Fig. 6-3 Block Diagram of OGO Operations Control Center Equipment Layout

Table 6-2

PERTINENT OGO OPERATIONS CONTROL CENTER EQUIPMENT

Central Facility		Work Area
Computing 1 – SDS 920 (16 K Core) 3 – SDS Tape Units 2 – Printers 1 – Card Reader 1 – Card Punch 1 – Paper Tape Reader 1 – Paper Tape Punch 1 – Keyboard	PCM Station 1 – Magnavox PCM-DHE 2 – Ampex FR-600 Tape Units 4 – Data Word Selectors 1 – 8 Track Bechman - Dynograph	Decom Stations 4 – Consoles Each With: 8-Data Word Selectors 8-Track Sanborn 7700 1 – Console With: 4-Data Word Selectors 8-Track Sanborn 7700
Control Console	Command Console 1 – Data Display, Model No. 123 (Keyboard, Papertape, and Card Entry)	

Table 6-3

SERVICE AND SPECIAL PURPOSE OPERATIONS PERFORMED AT THE CONTROL CENTER

Class	Item No.	Service and Special-Purpose Operations	Running Time or Rate	Real Time Cycles "C"	
				SDS 920	SDS 930
I	1	Static Limit Check (Subcom)	15 ms	3	1
II	2	Raw Data Printout	10 lls/sec	17/line	7/line
IIII	3	Orbit and Attitude Printout	10 to 15 secs	1700 to 2500	700 to 1000
	4	Command Spacecraft Configuration			
	5	Attitude and Control Printout			
	6	Control and Data Handling Printout			
	7	Power Supply Printout			
	8	Thermal Printout			
IV	9	Search Magnometer	2 to 5 mins	20K to 50K	8K to 20K
	10	Ionization Chamber			
	11	Rubidium Magnometer			
	12	Air Glow Photometer			
	13	Bennet RF Mass Spectrometer			
	14	Solar Spectra Photometer			
	15	Measurement of Micrometeorites			

6.2.1 PCM Conversion Station and Computation System Interface

The telemetry information is received from the communications center and converted by the PCM station to a serial bit train and buffered directly via the interlace register to the SDS-920 computer. The serial bit train word size is 36 bits and the word structure is as shown in Fig. 6-4.

0 – 8	9 – 17	18		20	21 – 23	24 – 30	31 – 35
DATA WORD (N+1)	DATA WORD (N)	F^*_1	F^*_2	F^*_3	N.U.**	WORD COUNT	N.U.**
Flags	Not Used						

Fig. 6-4 36-Bit Word From PCM Station

A complete word block, (or main frame) contains 16,384 (128 × 128) words. Table 6-4 presents the pertinent times for the SDS-920 and the proposed SDS-930.

Table 6-4

APPROXIMATE REAL TIME DATA PROCESSING TIMES

Computer	Computer Cycle Time (μsec)	Times (Approx.)		
		Telemetry (128 wds)	Computation (Est.)	
			Monitor	SSP
SDS-920	8	18 msec	12 msec	15 msec to 5 min
SDS-930	1.75	18 msec	3 msec	4 msec to 1.5 min

By use of the interlace system independent input and compute operations are allowed, and the data are stored in memory as follows (Refs. 7 and 8):

- Minor frame buffers

  Two 64-word (24 bits) arrays
- Subcommutator buffers

  Three 128-word (24 bits) arrays

The dual minor frame buffers are used to take advantage of the multiprogramming capability of the computation system. While one buffer is being filled via the interlace buffer, the compute operation is using the alternate data buffer (Par. 6.3.2.2).

The word structure of the minor frame buffers is as shown in Fig. 6-5.

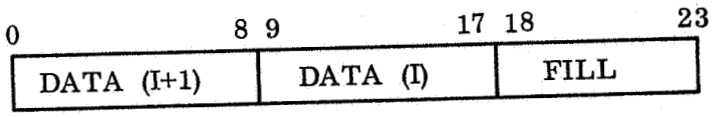


Fig. 6-5 Minor Frame Computer Word Format

The word structure of the subcom arrays is shown in Fig. 6-6.

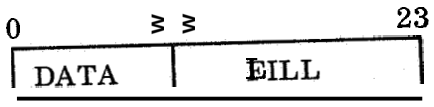


Fig. 6-6 Subcommutator Computer Word Format

6.2.2 Real-Time Operational Control

Active real-time command and control operations are conducted at the OGO control center, GSFC, after the spacecraft is inserted into the planned orbit. The function of the center is to check the operational status of the craft's systems, and to command and control the spacecraft into the proper attitude as required by the flight as well as experiment test profile.

6.2.2.1 Pre-Test (or Pre-Pass) Operations. A zero-order fixed aperture limit test (100 parameters maximum) is one of the subprograms of the MONITOR. The limits are loaded during program assembly or can be updated during the flight test.

The experimenter provides GSFC with two sets of calibration data, either coefficients of the polynomial (second degree maximum) and/or a table of quantity versus count. The methods used to quantifying the data are: (1) polynomial expansion (2) table look-up, and (3) a combination of (1) and (2) There is no special calibration tape. Each calibration is tested separately with its parent parameter and loaded during assembly.

There is no attempt to correct the information for signal conditioning errors. However there is a program available that allows for biasing the data (sensor correction) during pre-pass operation An update or limit change capability ( ) is provided as part of the SSP package.

6.2 2 2 Test Operations. The real-time procedure is to have the experimenter view the data at the control center. The experimenter may have up to 36 measurements displayed in strip charts. A prime (SDS-600) is centrally located for convenience in viewing and obtaining computer output data

Discussions with GSFC personnel indicate that the experimenter requests the cognizant control center operations personnel to display the measurement(s) desired. The operators key-in (decommutation selection) the setup at the strip chart display consoles (Fig. 6-3). It is understood that a significant amount of experiment analysis is conducted using strip charts.

During attitude orientation, the minor frame data are replaced by the Subcom 2 spacecraft system data (accelerated subcommutation mode). This mode of operation allows the control station to monitor the spacecraft system response to a particular set of commands. Upon command verification, the system is reset to the normal minor frame format or to any of the 32 flexible formats (experimenter modes).

When a service or special purpose operation is requested the program is loaded and the data are transferred (data freeze) to a computation storage area which thereby permits computation concurrent with continuous updating of the data in the minor frame and subcom storage areas. If the SSP computation on the transferred data

takes longer than the remaining elapsed time to input one minor data frame, the computation is interrupted and the system transfers to the monitor mode. This multiprogramming cycle continues until the SSP computation operation is completed. The output is a printer listing. Figure 6-7 is a functional block diagram of the control center real time processing system. All programs are coded in SYMBOL.

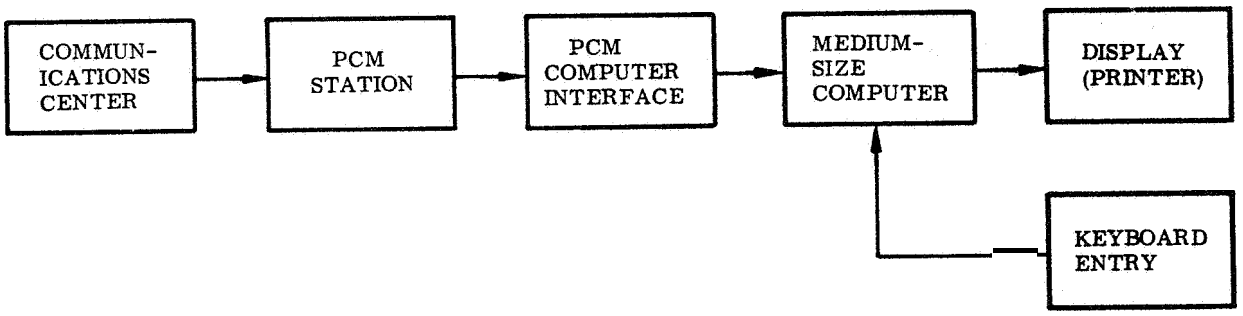


Fig. 6-7 Block Diagram of Real-Time Computation System

There are currently no processing operations external to those contained within the MONITOR that can be truly classed as real time. Some can be operated in a "near" real-time environment. Table 6-3 presents the SSP computer operations capable of being performed by the system as a function of their estimated running time and as a function of real-time cycles. The latter unit is of value when discussing the multiprogramming mode in Par. 6.3 of this report.

6.3 REAL-TIME COMPUTER SYSTEM ANALYSIS

To investigate the effects of data compression on the real-time computer operations, a simplified model is presented for the system currently operational at the GSFC control center. Figure 6-8 is a functional description of the processing system operation.

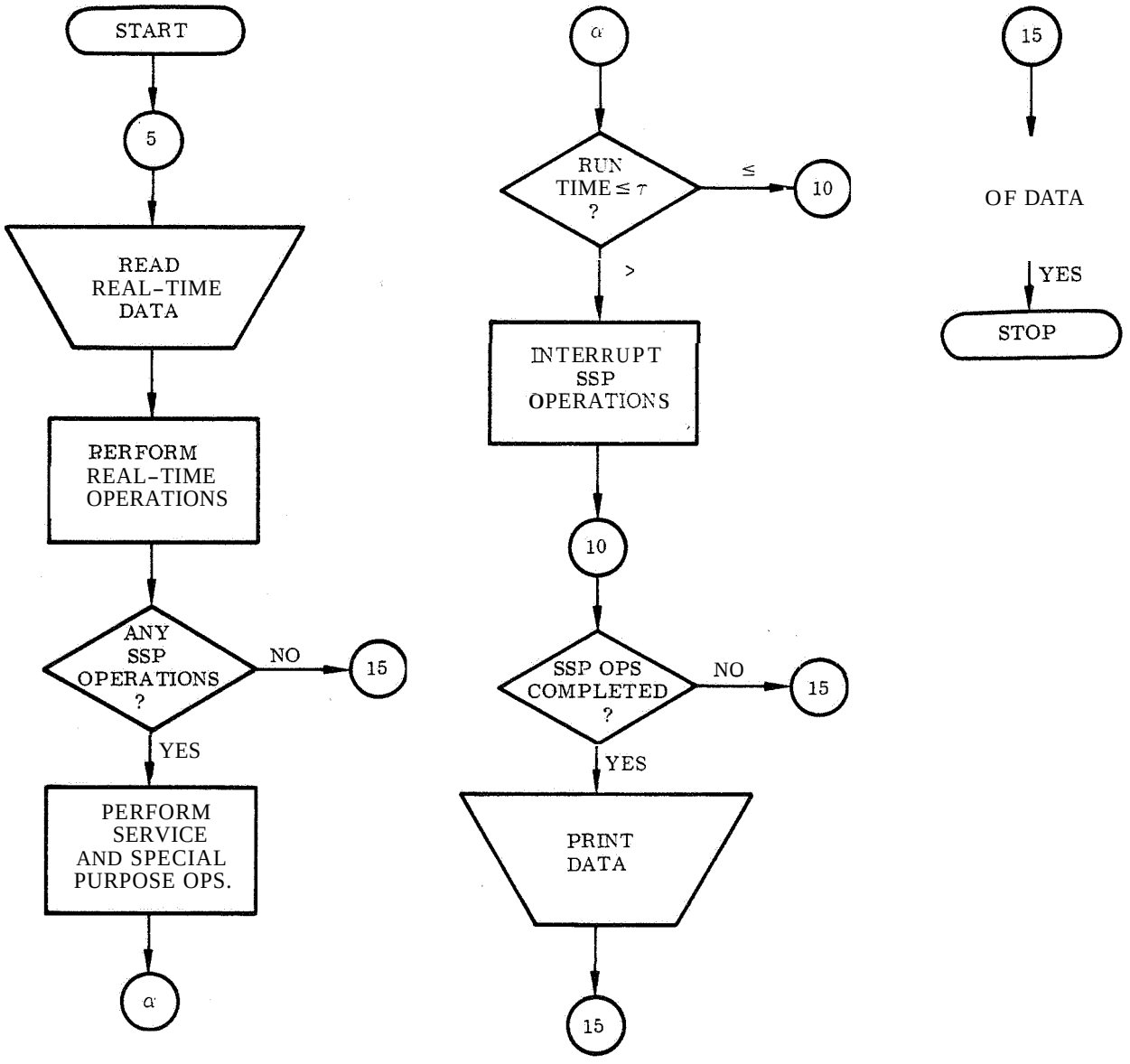


Fig. 6-8 Functional Block Diagram of Control Center Real-Time Processing System

6.3.1 System Considerations

Figure 6-9 presents the timing for critical operations for both the SDS-920 and 930 computation systems. From the figure, it can be seen that the SDS-920 system is capable, via the interlace register, of handling the input at real-time transmission rates (64 kb/sec). The time necessary to transmit one 128-word frame is 18 msec. This data frame is transferred to the computer in approximately 3 msec, elapsed time (interlaced). The zero-order fixed aperture test is then performed on a maximum of 100 parameters. The maximum time for the fixed-limit test with the SDS-920 is approximately 9 msec. If a paremter is out of limits, it is flagged and the value (octal or decimal) displayed on a line printer. A dwell or waiting time of 6 msec is available before the next monitor operation. This time is used when required, on a multiprogramming basis, for SSP operations.

The proposed introduction of the SDS-930 is expected to increase the time for SSP operations from 6 to 15 msec; a 150% increase in SSP capacity. Table 6-5 presents a summary breakdown of system capacity.

Table 6-5

SUMMARY OF REAL-TIME SYSTEM COMPUTATION CAPACITY

Computer	Time (msec)				Percent			
	Transmit	Transfer[a]	Limit Test[a]	SSP	Transmit	Transfer	Limit Test	SSP
SDS-920	18	3	9	6	100	17	50	33
SDS-930[b]	18	0.6	2.0	15.4	100	3	11	86

(a) Monitor Time = Transfer (M_I) + Limit Test (M_A) = M_{Total}
(b) Timing for 930 System Assumed Proportional

The relationship of the computation system input time, I, to the monitor running time, M, shows that the input time must be always greater than or equal to the monitor running time, i.e.,

$$I \geq M \qquad (6\ 1)$$

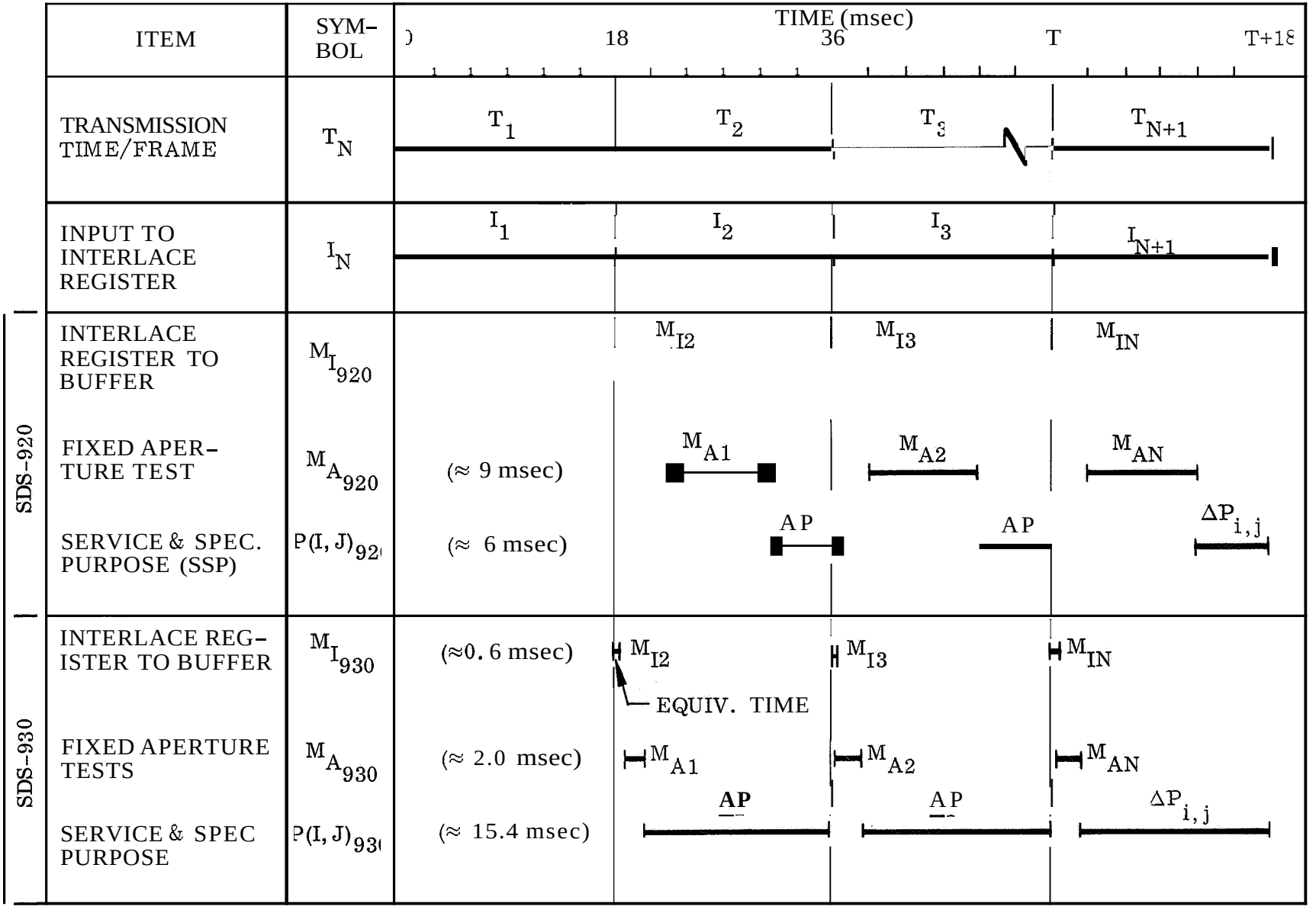


Fig. 6-9 Timing for Critical Operations of Computation System

and,

$$M = M_I + M_A \tag{6.2}$$

where

M_I = input transfer time

M_A = limit test time.

M_I was determined empirically from information obtained from GSFC personnel, such that

$$M_I \geq 3 F_L \tau_T \tag{6.3}$$

where

F_L = frame length, words

τ_T = computer cycle time = constant

$$\tau_T(\geq 20) \geq 8 \times 10^{-6} \frac{\text{sec}}{\text{word}}$$

and

$$\tau_T(\geq 30) \geq 1.75 \times 10^{-6} \frac{\text{sec}}{\text{word}}$$

When the time required to run SSP routines is considered:

$$I \geq M + \Delta P \tag{6.4}$$

or

$$I \geq M + \frac{P}{C} \tag{6.5}$$

where

P = SSP running time,

ΔP = SSP time/cycle, and

C = Cycles(number of frames for a given SSP operation)

The approximate running times for SSP operations (Table 6-5), were obtained from cognizant GSFC contractor personnel. It can be seen that for the same input time a minimization of monitor time maximizes the SSP time. The replacement of the SDS-920 with the faster SDS-930 significantly increases the desired SSP time.

6.3.2 Compressed Real-Time Data Computation

This paragraph examines the changes in data input and computation times that would be necessitated by the use of compressed data instead of the original, uncompressed data. To determine these changes it was first necessary to choose a method of inputting the compressed data to the computer. This method should take full advantage of the input time reduction made possible by the decrease in the number of samples in the compressed data train.

Although the introduction of an external decoder/buffer at the input to the computer is not necessary for proper system operation, to simplify the study the presence of this device was assumed. The function of the decoder/buffer would be to accept the compressed data in their transmitted CPCM format, and to alternately supply non-redundant samples to the two computer input buffers during successive 18-msec minor frame periods. Because only the nonredundant samples from a minor frame would be read into an input buffer, they could not be identified by their position in the buffer; hence, seven bits would have to be added to each nine-bit data word for channel identification (128 channels). The buffer could thus contain as few as zero sample words (signifying that all the samples from that original frame were redundant), and as many as 128 words (no redundant samples). The time required to read each "minor frame" of data into the computer during each 18-msec frame period would correspond to the number of nonredundant samples in that frame. Since each sample magnitude/ID word would contain 16 bits, only one data word could be fitted into each buffer word, instead of two, as is the case with the original uncompressed data. Hence, the data input time, on the average, would be reduced from the uncompressed case by a factor of half the sample compression ratio. Studies thus far have shown a combined sample compression ratio on the S-49 experimenters' data, using

the FOIDIS selector, of about 13. Thus for these data the SDS-920 input time of approximately 3 msec per minor frame for uncompressed data would be reduced to an average value of about 3/6.5 = 0.46 msec with compressed data. The data at the input to the computer would be delayed, using this system configuration, by a fixed amount, corresponding to the combined waiting times in the buffers at the transmitter and the receiver (Ref. 4 Pag. 5 3 2)

At this writing, the first-order type of interpolator (FOI) has been shown to be the most promising compression technique. The content of the data arriving at the computer is therefore modeled after a FOI output. In particular, a FOIDIS (or FOI4DG) output was assumed.

6.3.2.1 Computation Times. When an interpolator is used for sample selection in a data compressor, it is not possible to reconstruct a line sement for a given sensor until the end point of that line has been transmitted. Thus, the computer must "wait" for the end point before it can determine what action the data have taken during this segment.

T characteristic will be examined from two system viewpoints:

- The effect on limit testing
- The effect on service and special purpose calculations

Limit Testing. The current real-time system is designed to check the limits of specific parameters each time the parameters are received. The amplitude of a selected parameter is compared to the specified lower and upper limit (minimum and maximum values). If the data waveform goes out of bounds, it is flagged, quantified (optional), and an output message is printed.

Because all the data are available at real-time rates, the use of a zero-order fixed aperture (ZOF) test allows the determination of the intercept to an accuracy equal to the time between samples. When the data are compressed by a first-order interpolator, it is necessary to determine the intercept of the limit with an equivalent

accuracy. The problem is graphically illustrated in Fig. 6-10. Therefore, to obtain a precise data-time relationship, some method of reconstruction must be included in the real-time computation scheme.

One of the most direct methods of reconstructing FOI output data is by linear interpolation. Appendix I contains the geometry of the problem, and equates the solution. An LMSC program called INTERP was evaluated to determine the approximate running time of a linear interpolation operation. The FORTRAN equation solved by INTERP is as follows:

$$Y = Y(I) + (X - X(I)) * (Y(I+1) - Y(I)) / (X(I+1) - X(I)) \qquad (6\text{-}6)$$

The number of instructions involved in this solution were counted and the results are tabulated in Table 6-6. Table 6-7 summarizes Table 6-6 and converts the data to equivalent SDS-920 and SDS-930 computation times. Table 6-7 shows that a minimum of approximately 1.7 computer cycles (58 instructions) would be required, and that the computation time for a linear interpolative method would be approximately 1.18 msec and 0.26 msec for the SDS-920 and 930 respectively.

Approximately 100 additional cells of data storage would be required to store the time word of the last data point. Program instructions similar to that shown in Eq. (6-6) would be required in the main program.

Five conditions were examined. The conditions analyzed were:

A Normal subcom mode, SDS-920

B Accelerated subcom mode, SDS-920

C Normal subcom mode, SDS-930

D Accelerated subcom mode, SDS-930

E Experimenter modes, SDS-920 and 930

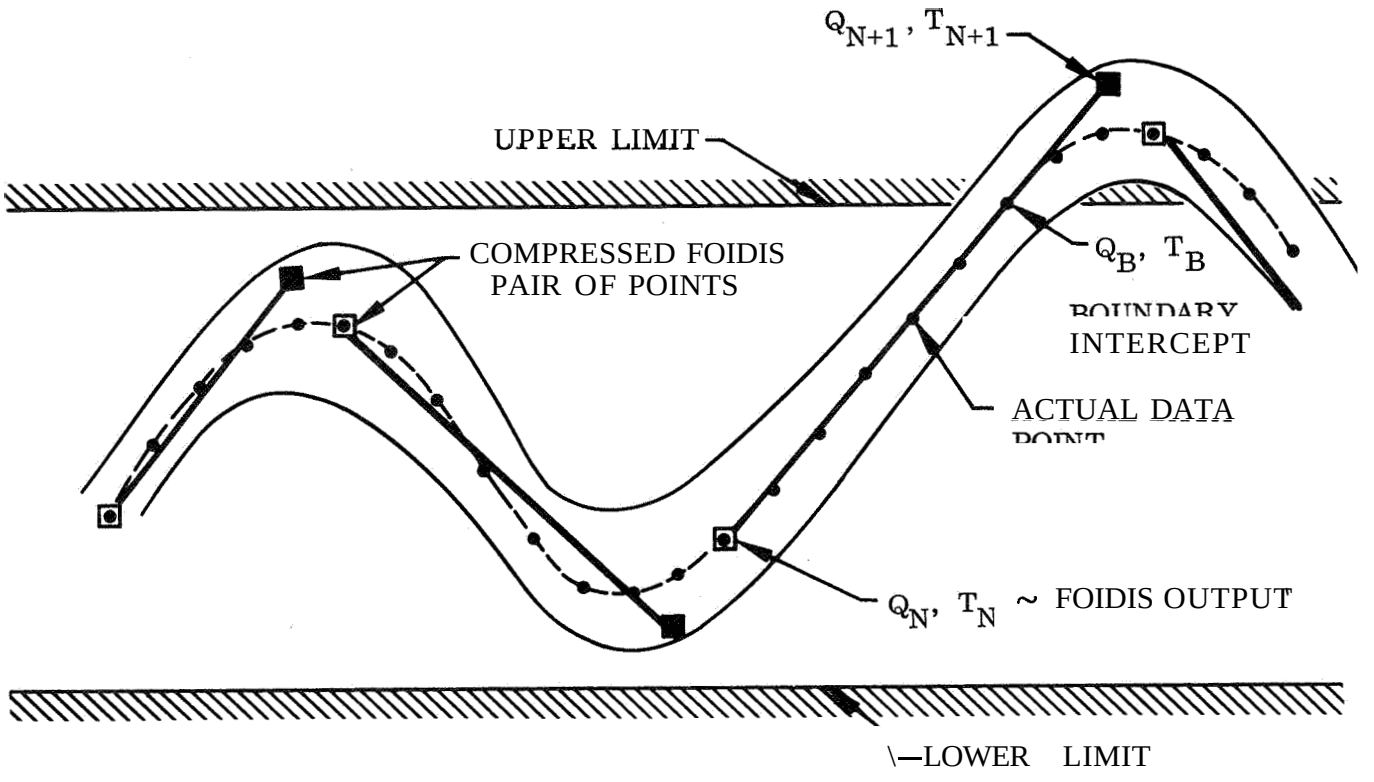


Fig. 6-10 Limit Check Boundary Intercept Determination for FOIDIS-Compressed Data

Table 6-6

TABULATION OF PROGRAM INTERP INSTRUCTIONS

Symbolic Operation	SXA	LXA	CLA	SUB	STD	STO	STA	CAS	TRA	TXI	TXL	FSB	FDP	FMP	FAD
Number of Operations	1	2	15	1	1	8	8	2	9	4	1	3	1	1	1
Timing per Operation (cycles)[a]	2	2	2	2	2	2	2	3	1	2	2	6.4[b]	13[b]	11[b]	6.4[b]
Total Cycles	2	4	30	2	2	16	16	6	9	8	2	19.2	13	11	6.4

(a) IBM 7094 Reference Manual
(b) Average Value; FSB and FAD (6 to 15), FDP (3 to 13), and FMP (2 to 13)

Table 6-7

ESTIMATED COMPUTATION TIME FOR LINEAR INTERPOLATION

Program	Number of Instructions	Average Timing (cycles)	SDS-920		SDS-930	
			Speed (μsec/cycle)	Computation Time (msec)	Speed (μsec/cycle)	Computation Time (msec)
INTERP	58	147	8.0	1.176	1.75	0.2573

The general method of analysis was to find the amount of time necessary for performing the MONITOR functions in one 18-msec minor frame period (and hence the time left for SSP calculations), assuming worst-case conditions, e.g., that both limit checks and interpolations must be made on all nonredundant data samples in the frame. On the average of course, only a fraction of the nonredundant samples should require limit checking, and only a fraction of those that are checked should require interpolations. In this way a lower bound on available SSP calculation times will be obtained.

Condition A: Normal Subcom Mode; SDS-920. The normal subcom mode for the OGO spacecraft places a predetermined array of sensors onto the minor frame (128 word length) such that they are sampled at high rates. As far as it could be determined, none of the 116 minor-frame parameters were of the class which are limit tested during the monitor phase of the real-time data processing. Of the remaining 12 positions, three are used for monitoring Subcoms 1, 2, and 3, and the remaining are used for frame synchronization, spacecraft clock and data-handling status.

Investigation of the OGO-C telemetry schedule revealed that the parameters that are scheduled to undergo limit testing were contained in Subcoms 2 and 3. (Table 6-8 is a listing of critical parameters obtained from GSFC and used in this study.) This being the case, the maximum number of critical parameters expected in each minor frame would be two.

If the absolute worst case is taken, then for a specific 18-msec minor frame period, it must be assumed that the input buffer contains 128 nonredundant samples. Thus a readin time of approximately 6 msec is postulated. Assuming further that on this particular frame both subcom parameters required limit checks and interpolations, 0.141 msec would be required for limit checking,* and 2.35 msec would be needed for interpolation. The total monitor time for this worst case would thus be 8.5 msec. The remaining 9.5 msec appears to be indeed adequate for the minimum value of available SSP calculation time. Assuming a sample compression ratio of 13, the average input time per minor frame would be

*This quantity was obtained by assuming that 128 channels can be checked in 3 msec, and scaling down.

Table 6-8

CONTINUOUS LIMIT CHECK PARAMETERS FOR OGO-C

Item	Limits				Item	Limits			
	Low		High			Low		High	
	Unit	Dec	Unit	Dec		Unit	Dec	Unit	Dec
A1	2500 psi	147	None	None	D8	25 v	156	37.5 v	235
A2	45 psi	116	55 psi	141	D9	25 v	156	37.5 v	235
A4	- 2"	096	+ 2"	158	D10	25 v	177	32.6 v	230
A5	- 2"	095	+ 2"	158	D11	Norm	000		005
A6	≥ 1 Head Sun	244		254	D12	All Norm	000		005
A7	< 4 Heads Earth	244		255	D15	Norm	090		130
A9	20"C	126	0"	193	D16	Both On	049		055
A10	- 5"	101	+ 5"	146	D20	15.74v	171	16.29v	177
A11	- 5"	98	+ 5"	157	D21	8.85v	173	9.21v	180
A23	Not Mode 3S	013		022	D22	4.89v	171	5.09v	178
		145		154	D23		095		104
A27	Gyro A Off	026		085	D32	15.67v	170	16.32v	177
A28	Not Sync	022		040	D33	8.79v	169	9.21v	177
A30	+ 62°C	041	+57°C	046	D34		090		101
A32	+ 60°C	033	+25°C	111	D35		090		101
A33	+ 45°C	187	None	None	D36	15.54v	171	16.45v	181
A34	+ 45°C	187	None	None	D37	8.69v	171	9.30v	183
A37	+ 35°C	086	+ 5°C	176	D38		097		107
A38	Small Earth	000		035	D39	15.5 v	171	16.5 v	182
A39	Small Earth	000		035	D40	8.75v	173	9.26v	183
					D41		097		107
					D54		89		99 } or <10
C1	+ 30°C	098	0"C	193	D53		170		179 } or <10
c2	+ 30°C	098	0"C	193	D55		D8-2 Count		D8+2 Count
c3	+ 25°C	111	+ 0°C	193	D56		D9-2 Count		D9+2 Count
C6		0		77	D24	67.99v	169	72.0lv	179 } or < 20
C8		0		80	D26	67.99v	169	72.0lv	179 } or < 20
					D25	22.40v	169	23.73v	179 } or < 20
					D27	22.40v	169	23.73v	179 } or < 20

Table 6-8 (Cont'd)

Item	Limits				Item	Limits			
	Low		High			Low		High	
	Unit	Dec	Unit	Dec		Unit	Dec	Unit	Dec
D28	15.67v	170	16.32v	177	E23	+25 °C	111	+ 5 °C	176
D29	8.81v	170	9.17v	177 } or 20	E24	+35 °C	086	+ 5 °C	176
D30		091		101 } or 20	E26	+50 °C	058	-10 °C	226
D31		091		102 } or 20	E28	+50 °C	057	-10 °C	227
D42	19.39v	171	20.65v	182	E29	+45 °C	066	+ 5 °C	176
D43	9.69v	171	10.32v	182	E30	+20 °C	126	+ 5 °C	176
D44		096		116					
D45	27.23v	094		100					
D46	112 v	164		210	F1 or F9	9.21v	231	9.81v	246
D3	Norm	000		007	F2 or F10		079		089
D50	+ 40 °C	075		158	F5 or F13	13.0	091	19	133
D51	+ 40 °C	075		158	F8	+25 °C	111	+ 5 °C	176
					F14	+40 °C	075	+10 °C	158
E5	+ 20 °C	126	+ 5 °C	176	F15	+40 °C	075	+10 °C	158
E6	+ 25 °C	111	+10 °C	158	F24*	+35 °C	086	+ 5 °C	176
E7	+ 25 °C	111	+10 °C	158	F25*		000		001
E8	+ 30 °C	098	+15 °C	141	F26*		025		026
E13	+ 75 °C	029	-25 °C	None	F27*		085		086
E14	+ 40 °C	075	-10 °C	226	F28*		132		133
E15	+ 25 °C	111	+ 5 °C	176	F29*		159		160
E16	+ 50 °C	058	-10 °C	226	F30*		206		207
E17	+ 20 °C	126	+ 5 °C	176	F31*		253		254
E19	+ 50 °C	058	-10 °C	226	F16	25 °C	111	5 °C	176
E20	+ 40 °C	075	0 °C	193	F17	40 °C	075	15 °C	141
E21	+ 30 °C	098	+ 5 °C	176					
E22	+ 25 °C	111	+ 5 °C	176					

0.46 msec, as discussed above. This reduction, along with the relatively infrequent need for interpolations, should increase the average available SSP calculation time to more than 17 msec.

Condition B: Accelerated Subcom Mode; SDS-920. The accelerated subcom mode switches Subcom 2 onto the minor frame for the purpose of increasing the sensor sampling rate. When this is done, the number of limit check parameters jumps from 2 (Condition A) to 43 under Condition B. The effect of compressed data on computer operating times is shown in Table 6-3 and in Fig. 6-11. If, as discussed above, it is assumed that on a given minor frame limit checks and interpolations are necessary on all nonredundant samples, then the table and figure show that the maximum number on which limit checks and interpolations can be performed is 14. If there were more samples in the input buffer, they would all be read into the computer, but limit checks and interpolations could not be performed on all of them.) However, if it is assumed that the worst compression ratio is 13,* then the average number of nonredundant samples per minor frame period is approximately 10.

An examination of Fig. 6-10 reveals that unless a single line segment crosses both the lower and upper limits, only one interpolation is required for each pair of nonredundant samples. Thus, neglecting double limit crossings, the average number of interpolations required per minor frame period is 5. The average total monitor time therefore becomes 0.466 msec (for input transfer, 10 samples) + 0.704 msec (for limit checking, 10 samples) + 5.88 msec (for interpolation, 5 sample pairs) = 7.0 msec. The average available SSP calculation time is therefore 11.0 msec. In reality, limit checking should be required on an average of fewer than 10 samples per frame, and interpolations on fewer than 5. Thus the average SSP time should exceed 11.0 msec. Although the currently-available SSP time of 6 msec per minor frame without compression

*Note that the compression ratio of 13 was obtained for experimenters' data. For spacecraft status data in the accelerated subcom mode, one would expect the compression ratio to be much greater than 13.

Table 6-9

OPERATING TIMES WITH COMPRESSED DATA, SDS-920

Number of Nonredundant Samples in Minor Frame	Operating Times[a] (msec)					Minor Frame Time (msec)
	Input Transfer	Limit Check	Interpolation	Total Monitor	SSP	
0	0	0	0	0	18.0	18.00
1	0.0466[b]	0.0704[c]	1.18	1.30	16.7	
2	0.0933	0.141	2.35	2.58	15.4	
4	0.186	0.282	4.70	5.17	12.8	
8	0.372	0.563	9.40	10.34	7.66	
14	0.651	0.985	16.45	18.09	–	

(a) Assuming limit checks and interpolations are made on all nonredundant samples in minor frame (worst-case conditions)
(b) Assuming 2.98 msec to read in 64 words
(c) Assuming 9 msec to check 128 samples

LOCKHEED MISSILES & SPACE COMPANY

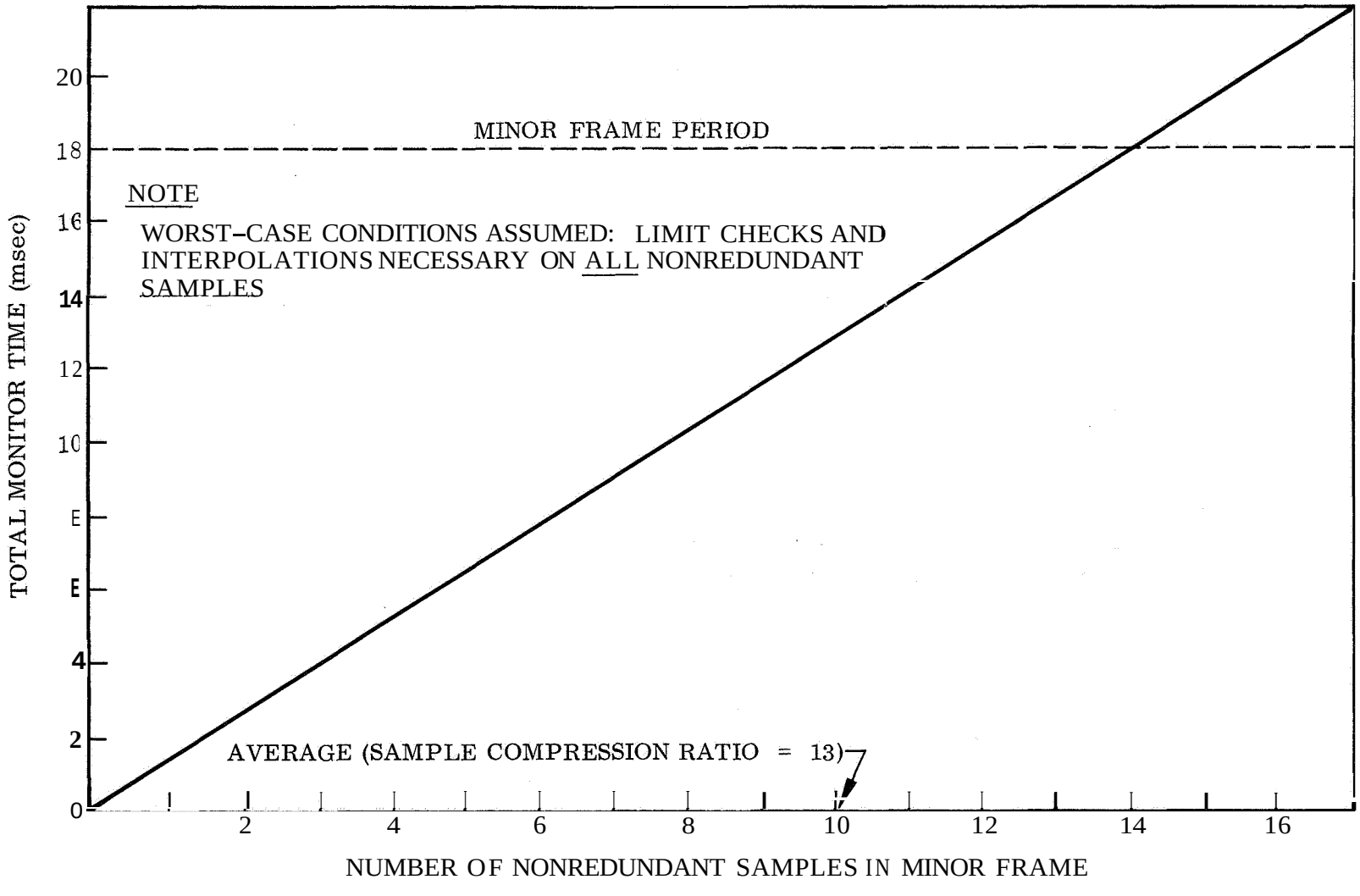


Fig. 6-11 Operating Times With Compressed Data, SDS-920

could possibly also be increased by reprogramming, it is evident that the average available SSP time with compression is roughly at least double that which is currently available.

It is true, however, that a situation could develop with this system model in which all the required limit checks and interpolations could not be made in one minor frame period, since the maximum number that might be required is 43. This unique situation could probably be handled with time-sharing and/or trading techniques, and by delaying limit check operations on some sensors until subsequent e periods.

Condition C: Normal Subcom Mode; SDS-930. As discussed earlier, the SDS-920 may be replaced with a real-time data system based on an SDS-930 computer and peripherals. Using the same approach that was taken for Condition A, the worst-case assumption is made that 128 words, representing nonredundant samples, must be read into the computer. Thus a readin time of 1.06 msec is postulated. Again assuming the necessity for limit checks and interpolations on both subcom channels, 0.0313 msec would be required for limit checking, and 0.514 msec would be needed for interpolation. The total monitor time for this worst case would thus be 1.6 msec. The remaining 16.4 msec is of course adequate for SSP calculations, even in this worst-case situation.

Condition D: Accelerated Subcom Mode; SDS-930. As stated under Condition B, the number of limit check parameters in the accelerated subcom mode is 43. The effect of compressed data on SDS-930 operating times is shown in Table 6-10 and in Fig. 6-12. If it is again assumed that during a given 18-msec minor frame period limit checks and interpolations are necessary on all the nonredundant samples in that frame, then the table and figure show that the maximum number on which limit checks and interpolations can be performed is 64. However, if the word compression ratio is at least 13, then the average number of nonredundant samples per minor frame period is at most 10. The same argument regarding line segment sample pairs that was submitted under Condition B applies here; thus the average number of interpolations required per minor frame period is 5.

Table 6-10

OPERATING TIMES WITH COMPRESSED DATA, SDS-930

Number of Nonredundant Samples in Minor Frame	Operating Times[a] (msec)					Minor Frame Time (msec)
	Input Transfer	Limit Check	Interpolation	Total Monitor	SSP	
0	0	0	0	0	18.0	18.00
1	0.00828[b]	0.0156[c]	0.257	0.281	17.7	
2	0.0166	0.0313	0.514	0.562	17.4	
4	0.0331	0.0626	1.03	1.13	16.9	
8	0.0662	0.125	2.06	2.25	15.7	
16	0.132	0.250	4.11	4.49	13.5	
32	0.265	0.501	8.22	8.99	9.01	
64	0.530	1.001	16.45	17.98	–	

(a) Assuming limit checks and interpolations are made on all nonredundant samples in minor frame (worst-case conditions)
(b) Assuming 0.53 msec to read in 64 words
(c) Assuming 2 msec to check 128 samples

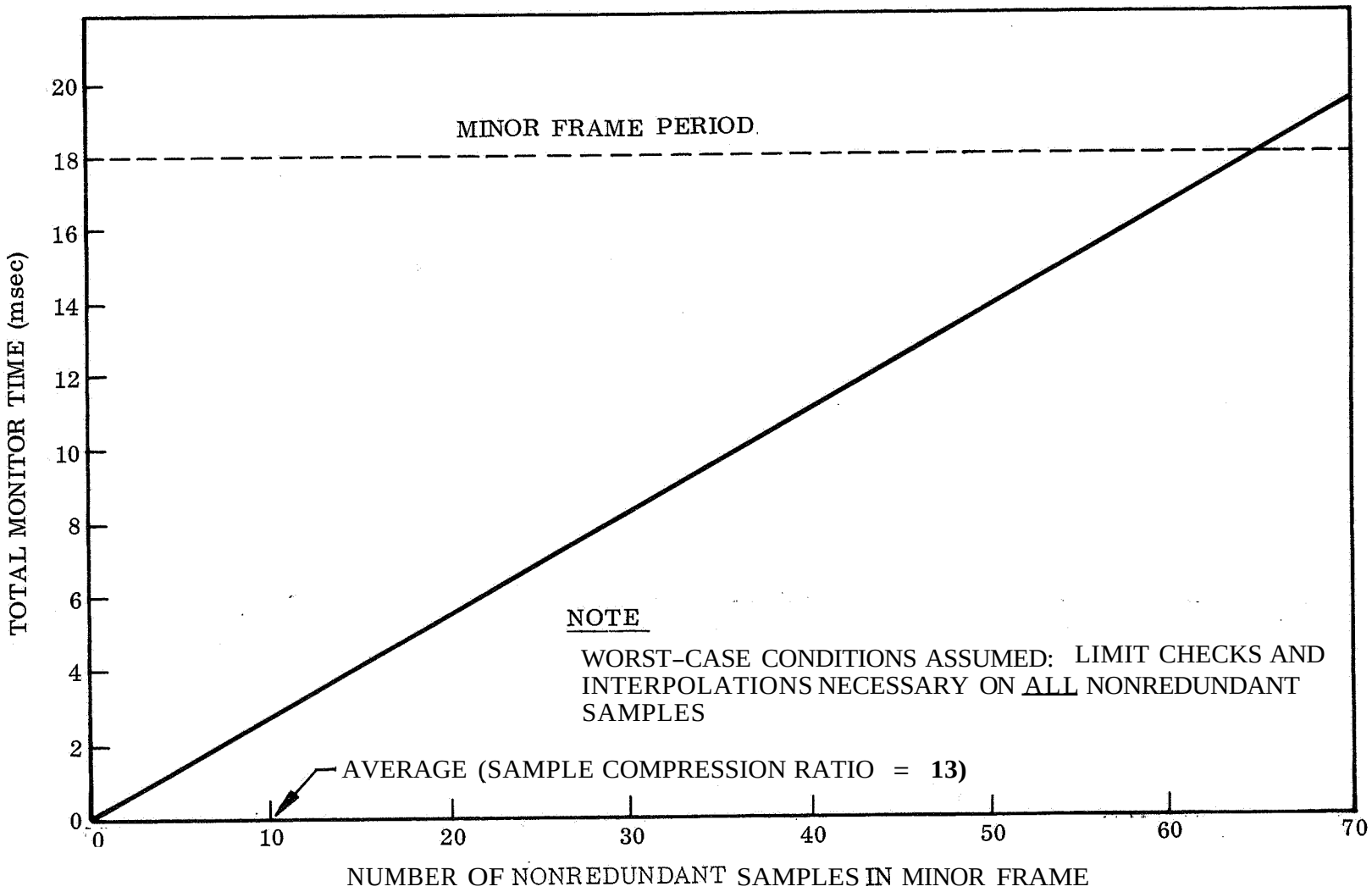


Fig. **6-12** Operating Times With Compressed Data, **SDS-930**

The average total monitor time is therefore 0.0838 msec (for input transfer, 10 samples) + 0.156 msec (for limit checking, [illegible]0 samples) + 1.[illegible] msec (for interpolation 5 sample pairs) = 1.5 msec. This leaves an average of 16.5 msec per minor frame for SSP calculations. As in the case of the SDS-920 (Condition B), the average number of limit checks and interpolations should be less than the above values. Thus the average SSP time should exceed 16.5

With the model chosen for this study, in which there are a total of only 43 limit check parameters, it is evident that the SDS-930 could not be saturated even during maximum conditions, since the [illegible]0 could handle a maximum of about 64 parameters in a frame period.

As is also true with the SDS-920, the computer would always be able to read in as many nonredundant samples as were in the input buffer — even as many as 128. This is because readin is accomplished first in each 18-[illegible] sec period. What the computer could not do would be to also perform limit checks and interpolations on more than about 64 samples, in the case of the SDS-[illegible]0 (or about 14 for the [illegible]0)

Condition E: Experimenter Modes; SDS-920 and SDS-930. The list of critical parameters obtained from GSFC (Table 6-8) did not contain any experimenter mode measurements. Consequently, it was assumed that no limit tests are made on minor frame parameters (excluding [illegible] subcom positions) during the experimenter modes.

Service and Special-Purpose Calculations (SSP). The types of calculations made during real-time computer operations were introduced in Par. 6.2. Table 6-3 is a tabulation of the types used and their approximate running times. The purpose of this section of the report is to discuss the effects of compressed data on the calculation phase of the real-time operation, and to present information that will assist in trade off design decisions as a result of parameter variations. The subject will be discussed in two parts:

- Test for, and assembly of, test data for calculations
- Reconstruction of compressed data for calculations

A simplified block diagram of the operation is presented in Fig. 6-13. In the following discussion it is assumed that a sample output will be forced by the data compressor for each sensor, if a predetermined number of samples have occurred without an output. This operation guarantees a maximum time delay in data reconstruction if an interpolator is used for sample selection. It is depicted graphically in Fig. 6-14.

Test For, And Assembly of, Test Data For Calculations. In the real-time OGO operation a measurement is samples at least once in each 16,384-word block (main frame). However, in a compressed mode of operation a data word is transmitted when the parameter data being compressed takes on a new trend (for example, at time T_I, in Fig. 6-14b), or when an output is forced, if no change in trend has taken place (for example, at time T_F, in Fig. 6-14c).

This characteristic requires that an additional operation be assigned to the pre-calculation operations of parameter transfer and store (data freeze). Whereas, at present, an analog (continuous) function probably requires a linear interpolation operation for determination of the instantaneous value at calculation time T_C, compressed data requires the storage of the most recent data/time value of each critical parameter as well as the knowledge of the next data/time value, in order to bracket the time required for the start of the computation. The amount of additional storage is a function of the reconstruction technique used. A linear interpolative method would require a minimum additional storage for both data and time words equal to the number of analog parameters required in the calculation.

The elapsed time to perform a calculation would be longer, because of the intrinsic delay of waiting for a trend change or forced output of the last (time-wise) parameter. The delay time for each sensor will be a function of its sampling rate and its relative position in the word block. If a trend change does not occur within F_o samples of the last change, the compressor will force an output. The forced output can be adjusted to each sensor's particular sampling

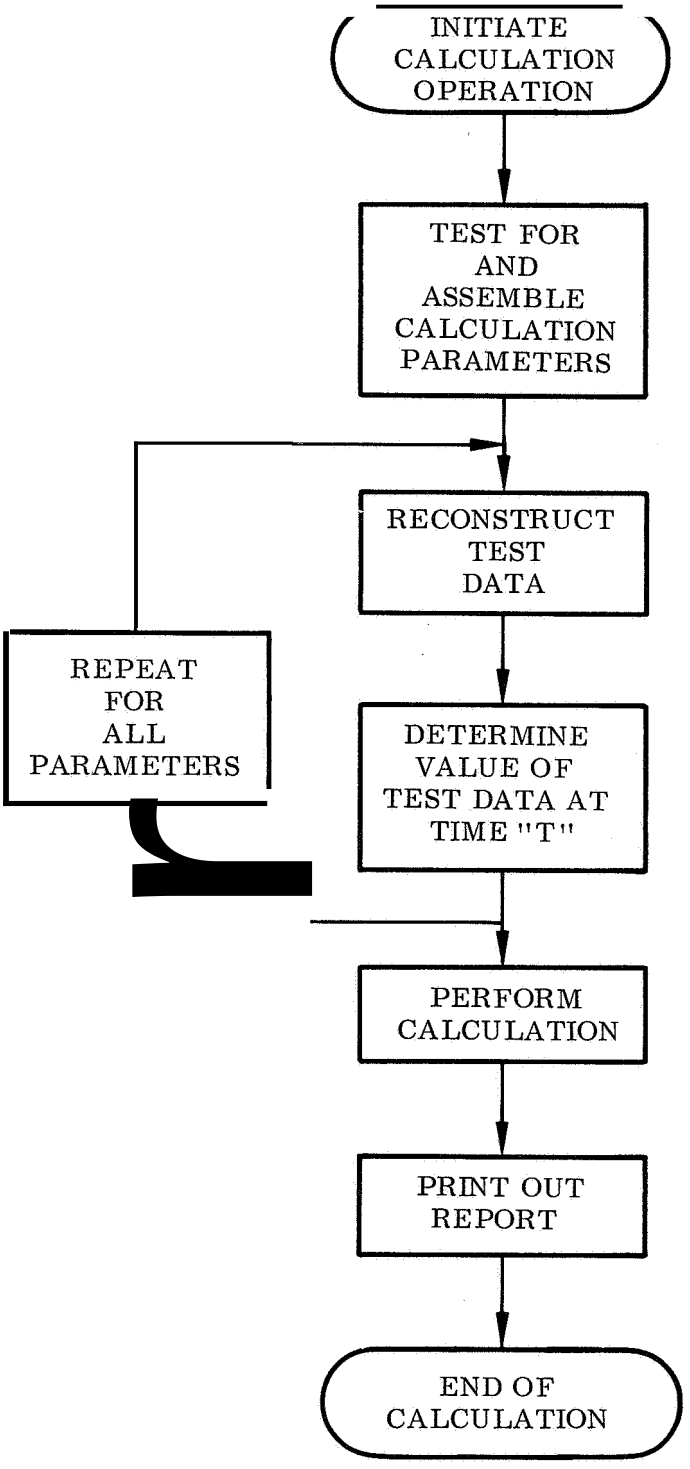
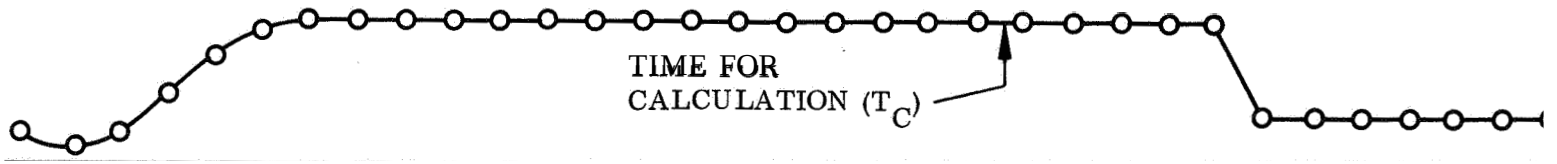
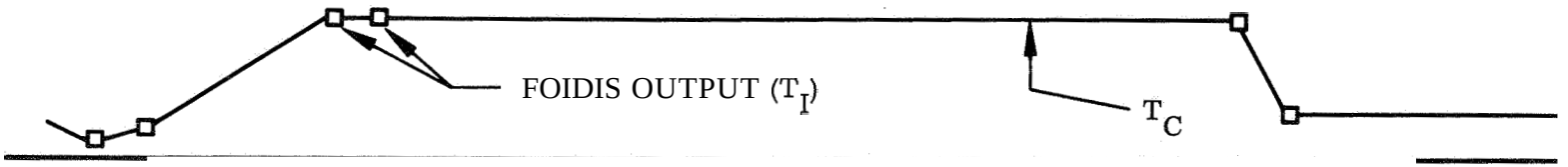


Fig. 6-13 Simplified Block Diagram of Real-Time Calculation Operation on Compressed Test Data



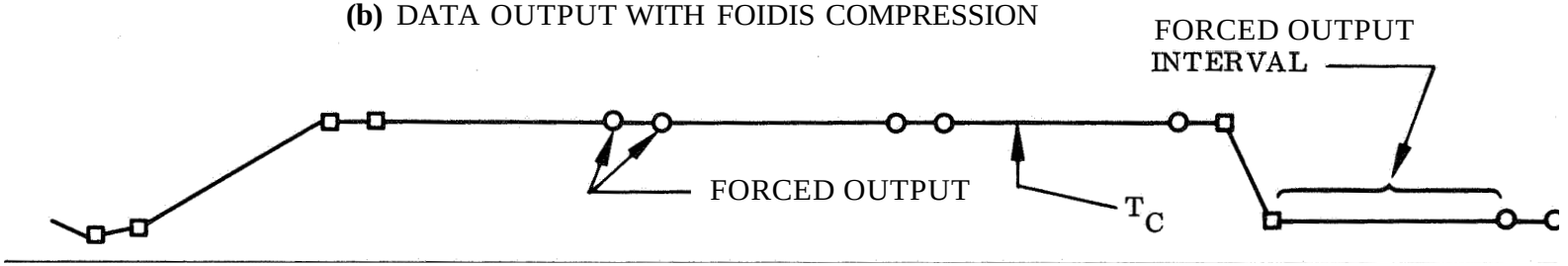
TIME

(a) DATA OUTPUT FROM A SINGLE SENSOR – NO COMPRESSION



TIME

(b) DATA OUTPUT WITH FOIDIS COMPRESSION



TIME

(c) DATA OUTPUT WITH FOIDIS COMPRESSION AND FORCEDOUTPUT

Fig, 6-14 Pictorial Presentation of Compressed Data With and Without Forced Output

requirements The delay time for a particular sensor is given by

$$T_R = \frac{F_o}{R_s} \tag{6.7}$$

where

T_R = delay e

F_o = number of samples since last transm d s at which an output is forced

R_S = sampling rate

The expected delay time for a given calculation is then the largest resultant value. To obtain the maximum expected delay time, the position of the sensors relative to each other in the telemetered wave train must be taken into account. Beginning with the first sensor to be sampled, and assuming that no change in trend occurs on any parameter, the total delay time can be equated as follows

$$T_{DT} = T_P + \left(T_R\right)_{max} \tag{6 8}$$

where

T_{DT} = total delay time

$\left(T_R\right)_{max}$ = largest delay time for y sensor

$$T_P = \frac{D}{R_W} \tag{6.9}$$

where

D = relative location in wor of sensor with largest delay time from first sensor to be sampled

R_W = word train transmission te (128 words per 18 msec)

Reconstruction Time of Compressed Data for SSP Calculations. Because of the desire to base this study on worst-case situations, and since first-order interpolators generally require more sophisticated and time-consuming reconstruction operations than do zero-order selectors, only the interpolation method of reconstruction of compressed data was considered in this report. The time required for reconstruction of parameters for the (1) thermal, (2) power supply, (3) control and data handling, and (4) attitude and control system instrumentation printouts are presented in Table 6-11. The table shows that the maximum time needed for interpolation on any of the four operations listed is less than 0.1 sec. This time appears negligible, compared to the 10 to 15-sec running time listed in Table 6-3.

Table 6-11

TIME REQUIRED FOR RECONSTRUCTION OF PARAMETERS FOR SSP CALCULATIONS

Number	Service and Special-Purpose Calculation	Number of Parameters[a]	Total Interpolation Time (msec)	
			SDS-920	SDS-930
1	Thermal	75	88	19
2	Power Supply	71	83	18
3	Control and Data Handling	64	75	16
4	Attitude and Control System	57	67	15
5				
6				
7				

(a) Determined from actual printouts obtained from GSFC

6.3.2.2 PCM Conversion Station/Computer Interface. The current SDS computer system provides a register and employs a buffer for direct information transfer between the PCM conversion station and SDS computer. The system responsible for the interface transfer is called the buffer interlace system. This system transforms the input character length from, in this application, the 36-bit PCM station input (Fig. 6-4) to a 24-bit buffer output word.

The interlace register is shown in Fig. 6-15.

0 — 11	12 — 25
COUNTER	ADDRESS

Fig. 6-15 Interlace Register

This register consists of a counter and address portion, respectively. The counter for input to the computer initially contains the word size of a memory array (example: 128 words). The address portion initially contains the memory address in which the first output word will be stored. The 14 bits provided allow access to any memory location.

The SDS-920 system currently operates as follows: When the character counter for each data word reaches zero, the computer is halted. The contents of the word register are stored in the memory location specified in the address portion of the interlace register. At the completion of this operation the computer is started again. The register count is decremented by one, and the address incremented by one. This operation requires 16 μsec per word; however, during this period the character register is accepting the next input word from the conversion station.

If the decoder/buffer configuration discussed at the beginning of Par. 6.3.2 is assumed, each computer input word would consist of a channel address in addition to the sample magnitude word. If this address word contained 14 bits, rather than the 7 bits suggested earlier, it could eliminate the need for the address portion of the interlace register.

The last word in the computer input buffer could always be followed by a word containing all zeros, signifying to the computer to stop loading words from its input buffer, and to begin its remaining MONITOR and/or SSP operations.

6 4 CONCLUSIONS AND RECOMMENDATIONS

6.4.1 Conclusions

The following conclusions can be drawn from the study of the effect of compressed data on the GSFC OGO real-time data system.

6.4.1.1 Input Data Handling. To simplify the study, the introduction of an external decoder/buffer at the input to the computer was assumed. This device would accept the compressed data in their transmitted CPCM format, and would alternately supply to the two computer input buffers, during successive 18-msec minor frame periods, only those nonredundant data sample words which correspond to a single original minor frame. The addition of this external device to the system model allowed computer operating times to be based, as they are currently, on the 18-msec minor frame period.

In actuality, however, the computer could be made to accept the nonredundant (compressed) data samples at the constant rate at which they are transmitted (thereby incorporating the buffering action of the decoder/buffer within the computer). If this were done, limit checks and interpolations could be made, when necessary, on each sample as it enters the computer. Because the average compressed data rate would be the same for either model, the choice of models does not materially affect the average available SSP time. This choice would be based on other factors, such as simplicity of operation, storage requirements, etc. These factors were considered in the data compression and reconstruction system configuration study (Section 8).

6.4.1.2 Operating Time. There are three factors that affect the average computer operating time on compressed data

- The need for partial data reconstruction by interpolation (when first-order interpolators are used) will tend to increase the operating time over that needed for uncompressed data.
- Because fewer data samples are transmitted, fewer samples are required to enter the computer. This will decrease the average input time, which will tend to decrease the operating time below that needed for uncompressed data.
- Because it is not possible to perform limit checks on samples which are redundant and thus have not been transmitted, the average time necessary for limit checks will tend to be decreased below that needed for uncompressed data.

When these factors are combined, it appears that the overall average MONITOR time, including interpolations, would be decreased with compressed data, thus increasing the average available SSP time. During periods of abnormally high limit check (critical) parameter activity in the accelerated subcom mode, limit check calculations may have to be delayed on the SDS-920 for several minor frame periods, to allow the computer time to "catch up" to the input data.

On the S -930, limit checks and interpolations could be done on about 64 [illegible] in a single minor frame. Hence, with the 43 critical parameters assumed in the system model, even during periods of absolute maximum activity, all MONITOR functions for a minor frame could be done during its e period [illegible] there would be need for delays in limit check calculations.

The computer time required for interpolation of compressed data on service and special-purpose calculations is negligible, compared with the current running time required for these operations, on uncompressed data.

6.4.2 Recommendations

It is evident from the foregoing discussion that the compressibility of OGO data in the accelerated subcom mode is a highly influential factor in determining actual average operating times with compressed data. The sample compression ratio of 13 used in this study was based on results from compression tests on experimenters' data. It is likely that spacecraft status data, when sampled at minor-frame rates, would yield compression ratios much greater than 13. However, as long as the compressibility of these data are unknown, the study results can only be estimated. It is therefore recommended that computer simulation tests be performed to determine the compressibility of these data.

Section 7
DATA PROCESSING COMPUTER ANALYSIS

7.1 INTRODUCTION

This study analyzed the effect of data compression on data processing at a central facility when utilizing a large-scale computer, namely, the UNIVAC 1107. Briefly, the primary elements of the task were to:

- Investigate the workload and data processing requirements
- Compare differences in calculation times of a large-scale computer utilizing compressed and current data formats
- Compare differences in input/output requirements for transfer of compressed data and current data formats
- Present advantages and disadvantages of using compressed data at a central station

7.1.1 Spacecraft Selection

The Orbiting Geophysical Observatory (OGO) was selected by the Goddard Space Flight Center (GSFC) as the model for this study because it is typical of a family of general-purpose spacecraft designed specifically to carry a number of related and/or unrelated experiments on a variety of space missions. PCM telemetry is utilized, which is typical of the systems in use today and expected to be in use for the next few years.

7.1.2 Spacecraft Telemetry System

The OGO spacecraft uses a PCM/FM telemetry system with a minor frame of 128 words (9 bits/word). The information is transmitted from the spacecraft at 8 and 64 kilobit/second rates, depending on the mode in which the equipment is operating.

Three 128-word (9-bit) subcommutators complement the telemetry configuration. A minor-frame switching array allows up to 34 different formats on command (Table 7-1).

Table 7-1

MINOR FRAME FORMAT MODES

Format Name	Number
Standard	1
Accelerated Subcommutator	1
Experimenter	32
Total	34

The accelerated subcommutator mode (ASC) switches the information contained on subcommutator Number 2 (Subcom 2), containing critical spacecraft systems measurements, onto the main frame. The ASC mode is used primarily during launch and spacecraft command and controlling after orbital insertion. This mode is also used for periodic status checks. Experimenter modes are utilized during optimum periods of experiment measurement.

7.1.3 Telemetry Measurement Assignment Schedule

The types and quantity of measurements for a typical OGO spacecraft are presented in Table 2-2.

7.2 DATA PROCESSING OPERATIONS AT GSFC

Offline PCM data processing for OGO at GSFC consists of four phases:

- Analog-to-digital conversion
- Editing process
- Decommutation
- Subsystem and experiment analysis

In addition, spacecraft command data are processed and spacecraft attitude and orbit (tracking) data are merged.

Table 7-2

CENTRAL STATION COMPUTATION SYSTEMS EQUIPMENT LISTING

COMPUTER SYSTEM	
UNIVAC 1107 (Current)	UNIVAC 1108 (Future)
28 Uniservo III-C at 556 bpi	12 Uniservo VIII-C at 800 bpi
128 K Core Storage	65 K Core Storage
2 H-880 Drums	6 F-432 Drums
Punch Card Subsystem	1 Fastran (22 Megawords-36 bit)
Printer ~ 600 LPM	2 1004 Reader/Printers
Paper Tape Subsystem	Punch Card and Paper Tape Subsystems
Supervisory Console	Printer
	Supervisory Console

A UNIVAC 1107 system is installed at the central facility, and an equipment list is presented in Table 7-2. At the time this study was conducted, GSFC was planning to update the complex to a UNIVAC 1108 system. The proposed equipment is also presented in the table.

Data processing operations at the central facility require the processing of test data in the following programs:

- Intermediate digital computer programs
  - Playback reformat and time correction program
  - Edit program
- Terminal digital computer processing programs
  - Decommutation program
  - Attitude-orbit program
  - Spacecraft command program
  - GSFC experimenters' analysis programs
  - Spacecraft subsystem processing program

The spacecraft PCM test data are received at GSFC from outlying telemeter receiving stations, on analog tapes. These data are then converted from the analog to a digital format. The output tape from the A-to-D process is called a "buffer tape."

The buffer tape has not been edited or time-corrected and is used only for quick-look analysis. The next step in process is edit and time-correct the buffer tape and output and "edit tape." The next step in the process is decommutation.

The decommutation process sorts the data into 21 separate tapes for OGO which are classified as follows:

- Housekeeping tape (1)
- Experimenter tapes (20)

The housekeeping tape contains spacecraft subcommutator information extracted during decommutation and is used as input to terminal-stage processing programs. An experimenter tape contains sorted test data as required for the analysis of a specific experiment. Figure 7-1 is a block diagram showing the processing operations.

The estimated running times for some of the major data processing computer programs are listed in Table 7-3. This information was obtained from cognizant personnel of the GSFC DPL Processing Branch.

Three of the OGO data processing phases are discussed in greater detail below.

7.2.1 Analog-to-Digital Conversion

The aspect of the analog-to-digital conversion operation which is of major concern in this study is the format of its digital output, the buffer tape. This format is shown in Table 7-4 and in Fig. 7-2.

7.2.2 Editing Process

The editing process consists of three phases:

- Edit
- Time Correction
- Quality Control

Both the real-time and playback data are subjected to these processing phases.

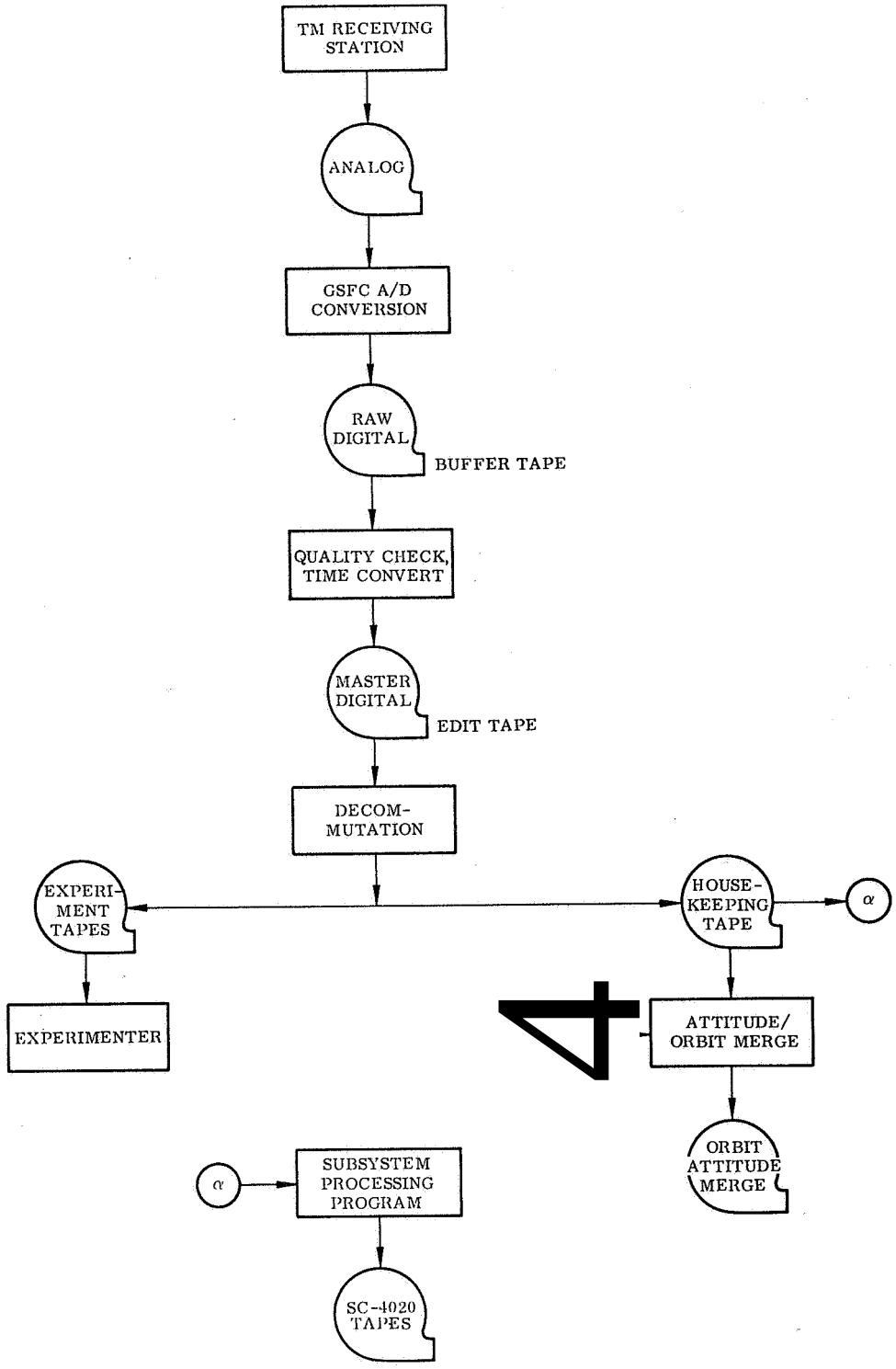


Fig. 7-1 GSFC Data Processing Operations

Table 7-3

ESTIMATED TIMING FOR MAJOR OGO COMPUTER PROGRAMS

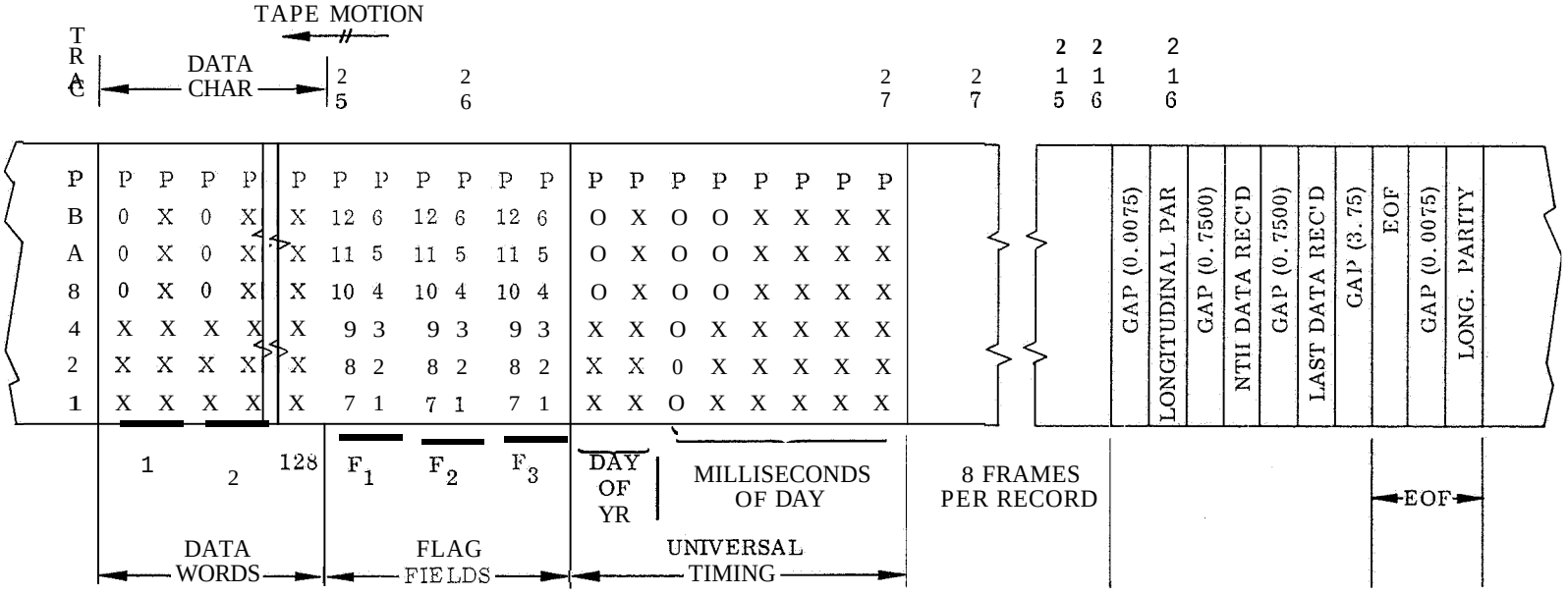
Program Name	Computer	Input		Output		Processor Time (min)	Run Time (min)
		Time (min)	No. of Tape Units	Time (min)	No. of Tape Units		
1. Edit	1107	63	4	57	2	285	285
	1108	50	4	19	2	57	57
2. Time Correction	1107	8	4	Neg.	1	50	40
	1108	6	4	Neg.	1	8	8
3. Decom	1107	57	2	159[a]	21	144	144
	1108	19	2	159[a]	10	29	90
4. Subsystem Status	1107	1	2	3	6	35	35
	1108	1	2	3	6	6	6
5. Experiment Programs (4)[b]	1107	50	(c)	50	(c)	480	480
	1108	35	(c)	35	(c)	80	80
6. Taylor Experiment (2 passes)	1107	21	(c)	16	(c)	555	555
	1108	16	(c)	13	(c)	93	93
7. Orbit Attitude	1107	8	4	3	4	30	30
	1108	4	4	2	4	5	5

(a) Two output channels; one channel ≃ 80 minutes
(b) 120 minutes per experiment
(c) Not available

Table 7-4

BUFFER TAPE FORMAT

Title	Number
1 Buffer Tape File:	
Identification Records	2
Data Records	"N"
2 I.D. Records:	
I.D. Characters	18
File I.D. Characters	18
3. Data Records:	
Data Frames	3
4 Data Frames:	
Data Words	128
Flag Words	3
Time Word	1
5 Data Words:	
Data Characters	2
6. Flag Words:	
Flag Characters	2
7. Time Word:	
Time Characters	3
8. Character:	
Bits	6 (AI)



F_1 IS A CONSTANT (RESERVED FOR COMPUTER-DETERMINED FLAGS).

F_2 IS A FLAG RELATING TO HARDWARE-DETERMINED TIME STATUS.

F_3 IS A FLAG RELATING TO HARDWARE-DETERMINED SYNC STATUS.

Fig. 7-2 Buffer Tape Data Record Format

7.2.2.1 Edit Phase. The primary function of the edit phase is to produce an edit tape which contains files of data in chronological sequence. The only restraint is that playback and real-time data will not be placed on the same tape.

In the processing operation the computer reformats the 8-frame buffer tape record into 128-frame edit tape records. An additional frame is added, so that the total is actually 129 frames per record. Buffer tape I.D. records are carried over and added to the label record of the edit tape. The edit tape format is shown in Table 7-5 and in Fig. 7-3.

Each 8-frame buffer tape record is interrogated for format errors, which includes the verification of word, frame, and record lengths. Frame sync is tested for bit slippage and adjusted when required. Sync errors are listed.

The following checks are made on the data:

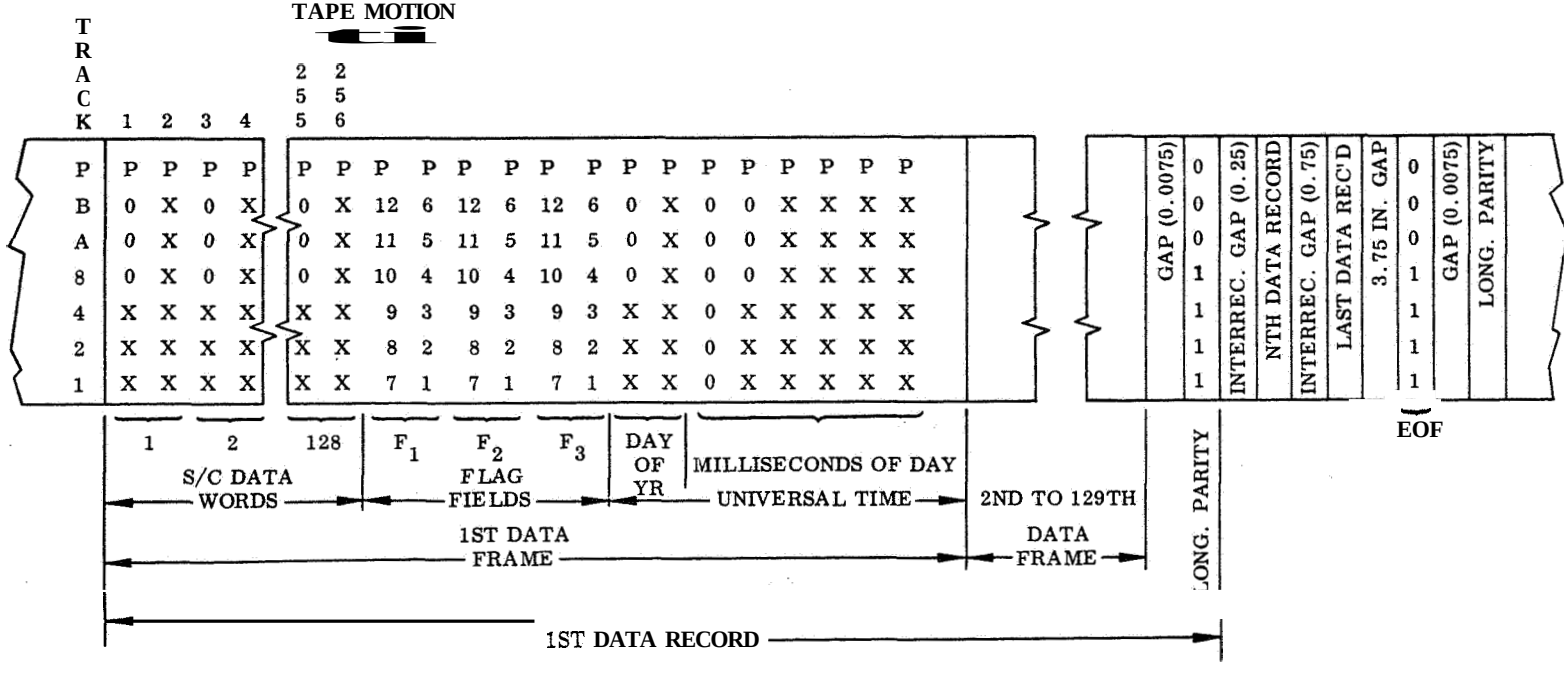
- Parity test
- Bit rate test
- Subcommutator chronology test
- Leading zero test

7.2.2.2 Time Correction Phase. The method of time correction differs quite significantly between real-time and playback data initially, and then reaches a point of commonality further in the correction process. Real-time data are corrected for transmission delays from the T/M station to the spacecraft and the WWV station to the T/M station. The time of transmission of the spacecraft playback data to the T/M receiving station is independent of when the data were recorded on board the spacecraft. Therefore, the spacecraft clock time must be correlated with GMT time determined during real-time processing, and applied to the playback data.

Table 7-5

EDIT TAPE FORMAT

Title	Number
1. Edit Tape File:	
Label Identification Record	1
Data Records	"N"
2. Label I D Records	
I. D. Characters	120
3. Data Records:	
Data Frames	129
4. Data Frames:	
Data Words	128
Flag Words	3
Time Words	1
5 Data Words:	
Data Characters	2
6. Flag Words:	
Flag Characters	2
7. Time Word:	
Time Characters	8
8. Character:	
Bits	6 (All)



F_1 = QUALITY CONTROL F_2 = FRAMESYNC F_3 = TIMEDECODER

Fig. 7-3 Edit Tape Data Record Format

7.2.2.3 Quality Control Phase. Each edit data file is subjected to a quality evaluation, and a listing summarizing the results is printed on-line.* This listing includes such items as:

- File label record
- Start and end times (universal time and spacecraft clock)
- Equipment group
- Flexible format in use
- Average frame period
- Number of frames with sync bit errors

When a file is rejected a card is punched to give the rejected file a tape number.

7 2 3 Decommutation

After the edit tape has been generated, the next step in the process is decommutation of the data into a number of output tapes as required by the testing program. Normally, one tape is formatted to the specifications provided by each experimenter – about 20 per OGO satellite. In addition, one housekeeping tape is generated which contains mostly spacecraft data.

The decommutation process consists of five basic operations, as shown in Fig. 7-4:

(1) Read a data record (129 data frames) into memory
(2) Extract selected portions of the data block
(3) Output selected portions to proper tape unit
(4) Repeat processes (2) and (3) for each experimenter plus housekeeping tape
(5) Repeat process for each data block (1)

The GSFC decommutation operation is as follows. A data matrix is formed in which i denotes the column and j denotes the row, as shown in Fig. 7-5. Each row consists

*Accelerated subcom data are excluded from a quality control evaluation.

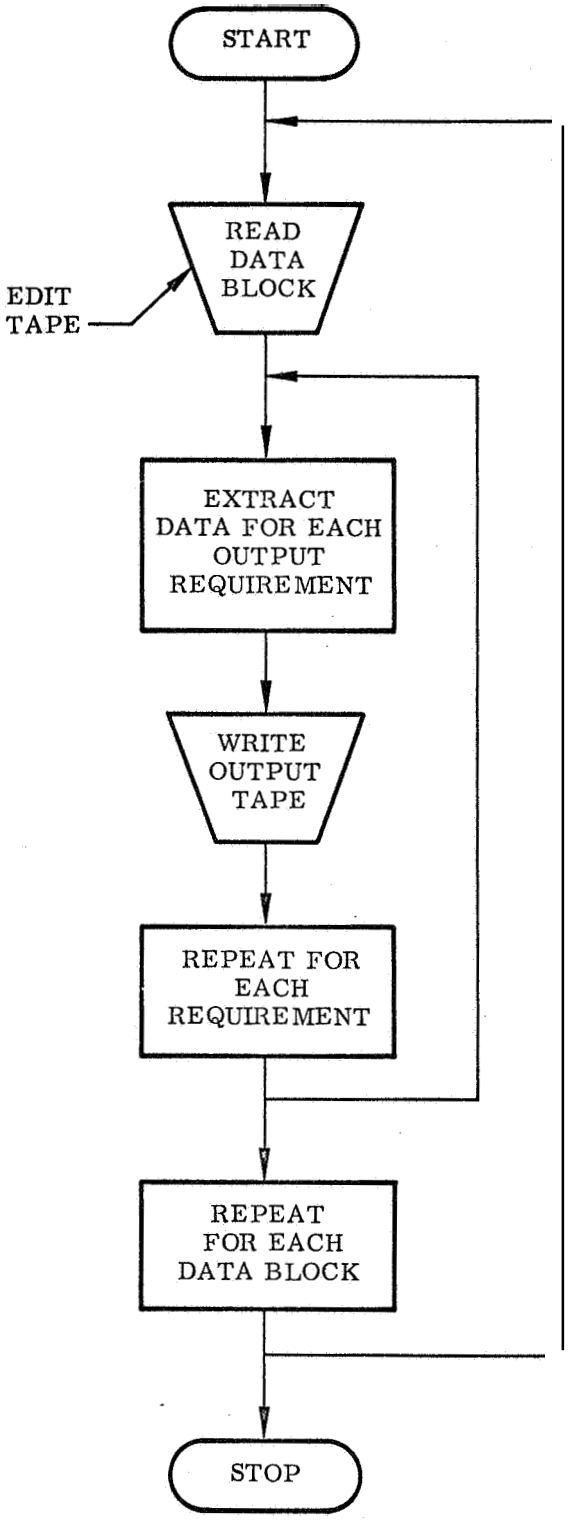


Fig. 7-4 Decommutation Process

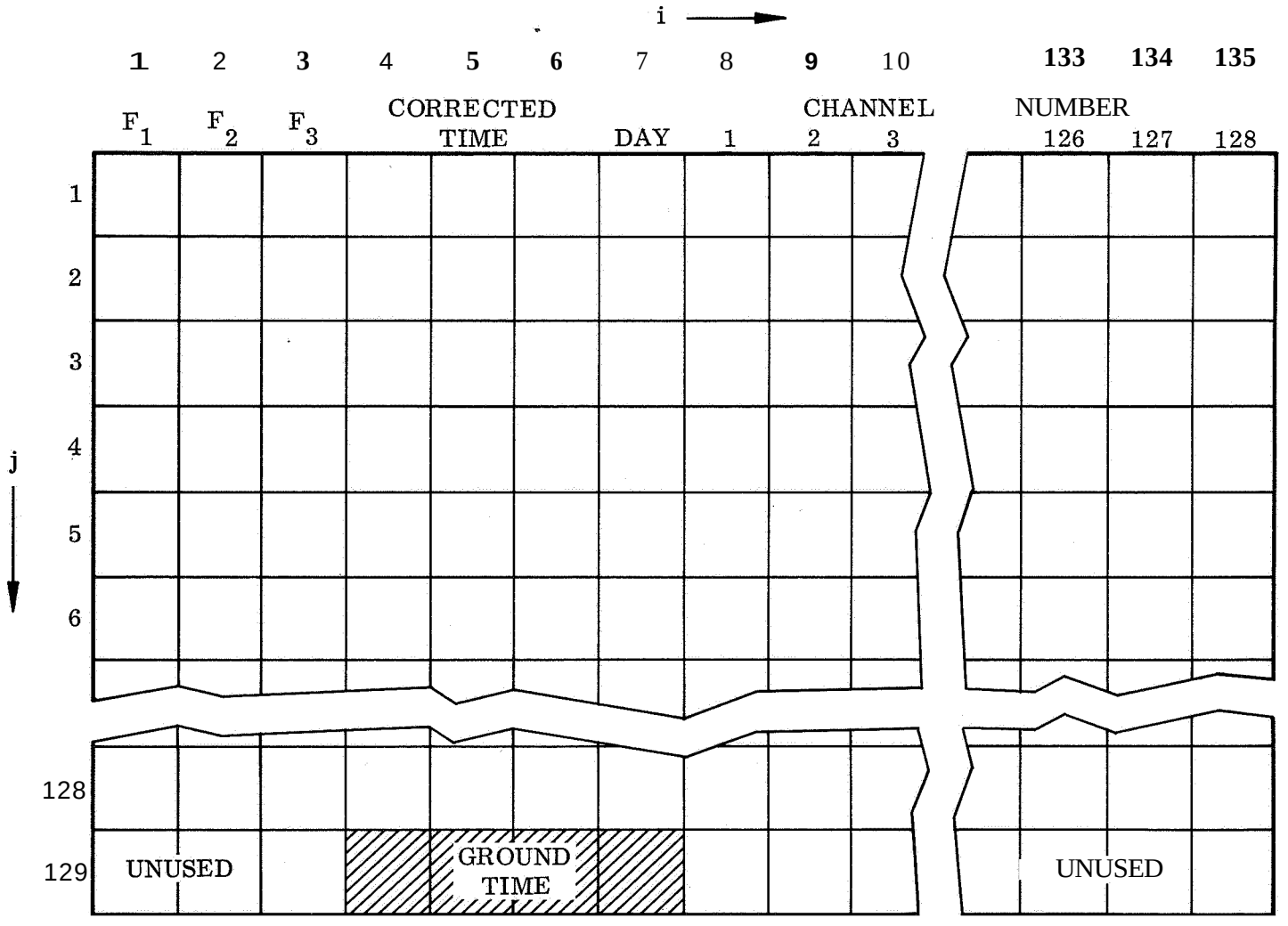


Fig. 7-5 Decommutation Data Matrix

of three flag elements, three corrected universal time elements derived from spacecraft time, one day count of the year element, and the 128 data words from the minor frame ($i \leq 8$ to $i = 135$). The first row includes subcommutator words one. The jth row denotes a subcommutator word number. A value (i, j) designates a word in the telemetry format; a row represents a minor frame; and the entire matrix represents a subcommutator sequence. The ground time box in Fig. 7-5 is also part of the configuration for real-time data. It is the ground time at the data acquisition station which was associated with the beginning of word one of the first minor frame in the matrix. The box is all zero for spacecraft on-board recorded data.

Figure 7-6 is a block diagram of the normal production decommutation process.

7.3 THE IMPACT OF COMPRESSION ON DATA PROCESSING

This paragraph examines the changes that would be involved in the OGO data processing procedure if the input data were in a compressed, instead of the current uncompressed, format. In order to avoid changing all the experimental data analysis computer programs to accommodate data in compressed formats, it was considered advisable to reconstruct the compressed data at some point in the data processing procedure before the experimenters' tapes are generated. This consideration prompted a portion of the study to be made under the assumption that reconstruction was to be done during data processing. The logical time for reconstruction would be during the decommutation process, since no advantage would be gained in reconstructing before this time. Paragraph 7.3.1 describes this study, and presents the results derived therefrom.

Reconstruction, on the other hand, is a consumer of computer operating time. Also, any advantage that could be gained by operating entirely with compressed data would not be realized if the data were reconstructed before the experimenters' tapes were generated. Therefore, expected savings in computer operating time and tape storage requirements, in the event that the data could be delivered to the experimenters in compressed form, were analyzed. The results of this analysis are presented in Par. 7.3.2.

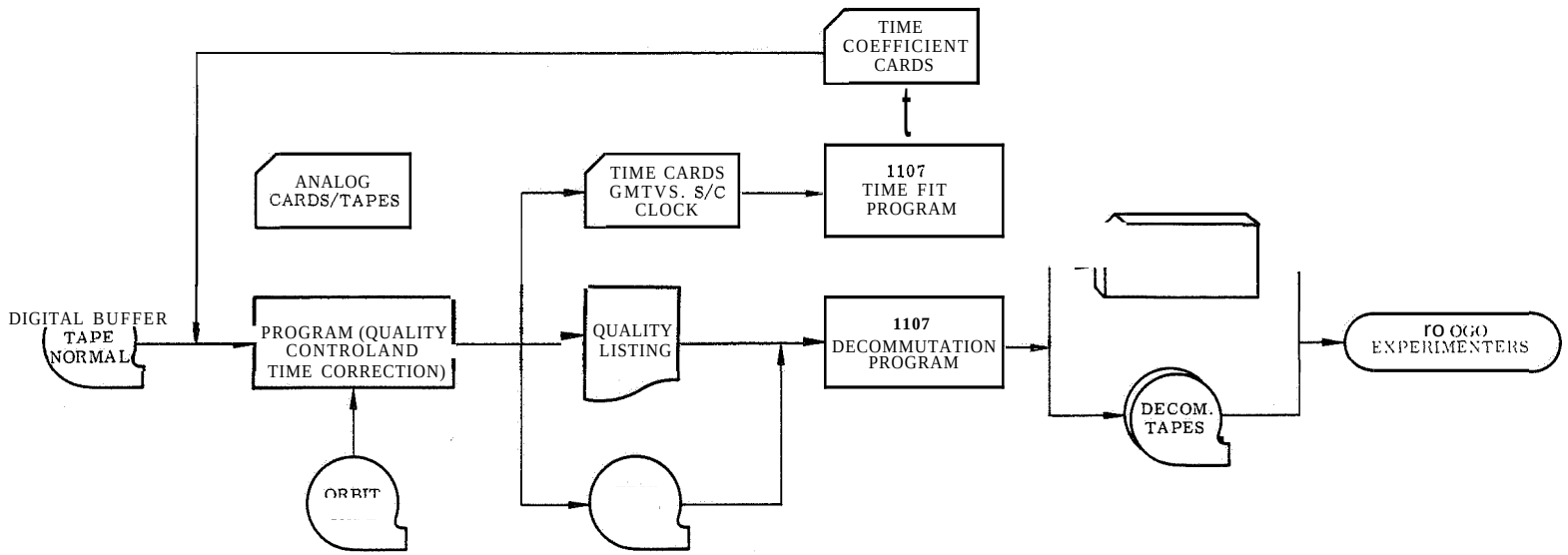


Fig. 7-6 Normal Production Decommutation Process

7.3.1 Decommutation and Data Reconstruction

The reduction of the number of words read into the computer will tend to reduce the decommutation time; on the other hand, the need for reconstruction should increase this time. An analysis of these opposing factors is presented in the following paragraphs.

7.3.1.1 Compressed Data Input Tape Format. In order to proceed with this study, it was first necessary to choose a model format for the compressed data input tape. The model chosen is not necessarily optimum, and was meant only to serve as a method to obtain comparatively rough estimates of computer operating times. The model format is the same as that of the present OGO edit tape, except that only nonredundant data samples are included in each minor frame, along with seven bits for channel identification. Three characters would be required to accommodate the 16 bits needed for data magnitude and identification. Thus, the length of a minor frame could vary between fourteen 6-bit characters (if there were no nonredundant samples) and 398 characters (if all 128 samples in the minor frame were nonredundant). The 14 characters would be for flags and universal time, and would be present in every minor frame on the tape. Because a tape record contains 128 minor frames, the records would be of variable lengths.

Since only 2.3 percent of the data are contained in the subcom channels, and because compression and reconstruction of the subcom channels would require storage capacities many times greater than those needed for the minor frame channels, it may not be profitable to compress the subcom channels in an OGO data compression system. Therefore, it was assumed that no compression was done on the subcom channels. With this assumption, the fixed number of characters is increased from 14 to 23. The frame size is shown in Fig. 7-7.

7.3.1.2 Data Reconstruction. Figure 7-8 provides a pictorial example of a compressed data output for a single sensor, using the FOIDIS selector. It will be noted

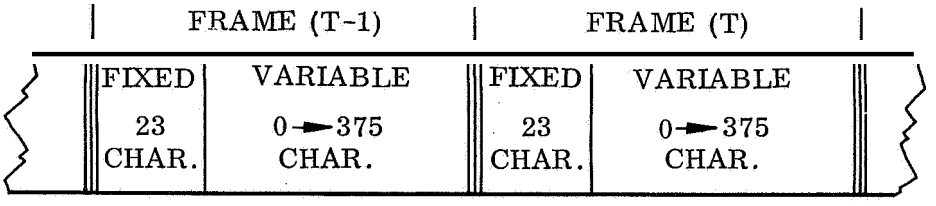
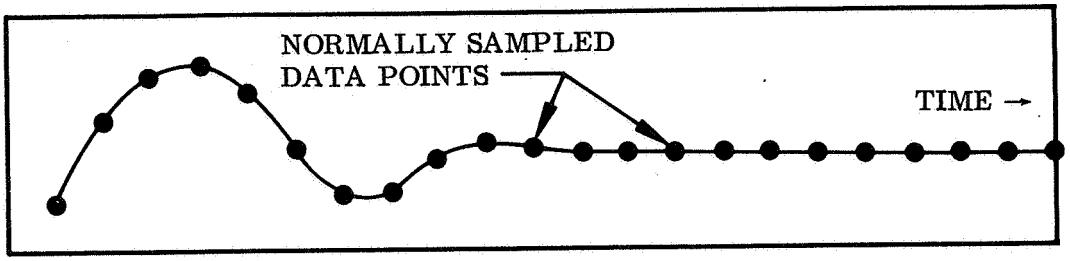
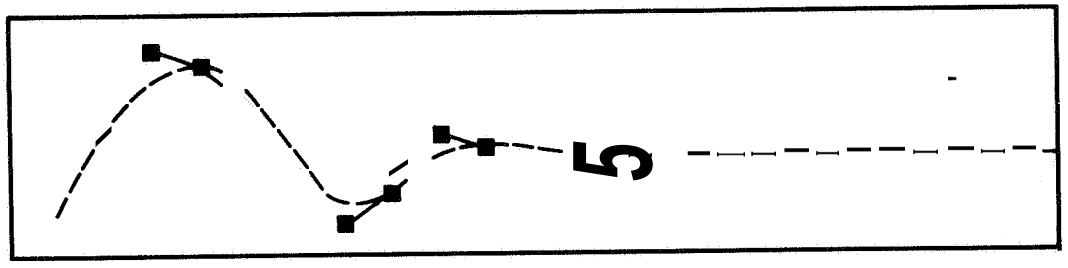


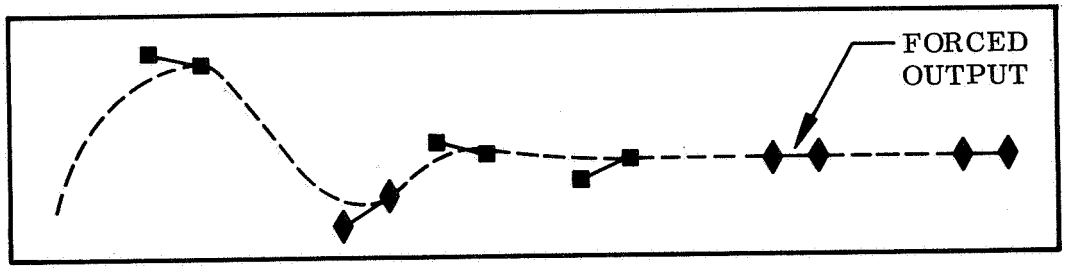
Fig. 7-7 Compressed Tape Frame Size



(a) DATA OUTPUT FROM A SINGLE SENSOR – NO COMPRESSION



(b) DATA OUTPUT WITH FOIDIS COMPRESSION



(c) FOIDIS COMPRESSION WITH FORCED OUTPUT

Fig. 7-E Pictorial Presentation of Compressed Data With and Without Forced Output

that during periods of relatively high data activity, the compressor outputs when a new trend is sensed. Conversely, in periods of low data activity the compressor is forced to output at some preset rate.

The reconstruction procedure is shown in Fig. 7-9. Reconstruction requires the replacement of the redundant samples with restored data words. The operation is accomplished by first determining the slope and intercept of the segment connecting FOIDIS output data. Then, redundant samples are replaced with restored data words by incrementation.

The straight-line interpolation equation is

$$D = \left(\frac{D_{2,1} - D_{1,2}}{T_{2,1} - T_{1,2}}\right)\left(T - T_{1,2}\right) + D_{1,2} \tag{7.1}$$

or

$$D = S(T - T_{1,2}) + D_{1,2} \tag{7.2}$$

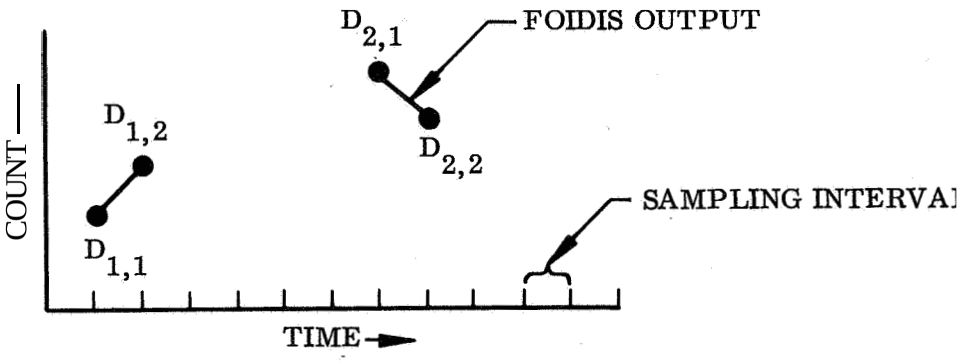
where S is the slope.

An existing LMSC program (INTERP) was used to obtain an approximation of the number of program instructions and running time (Section 6). It was estimated that 147 computer cycles (58 instructions) would be required for each interpolation. Assuming a UNIVAC 1107 cycle time of 2.0 μs and a 1108 cycle time of 0.75 μs, the time to determine the slope and intercept was estimated. The results are presented in Table 7-6.

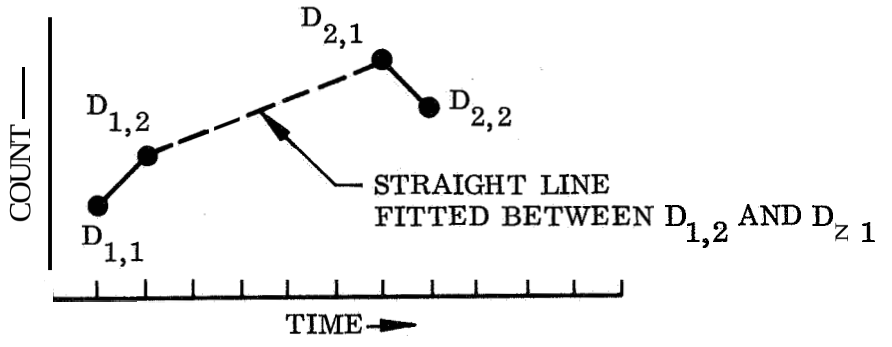
Upon determination of the slope and intercept, the next operation is to replace the intervening redundant words. Because the difference $(T_{2,1} - T_{1,2})$ is a multiple of the sampling rate, the sampling interval is given by

$$\bar{T} = \frac{T_{2,1} - T_{1,2}}{N} \tag{7.3}$$

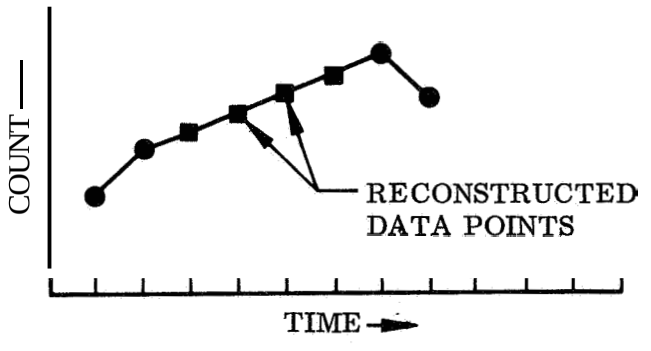
where N is the number of redundant words plus one.



(a) TYPICAL FOIDIS OUTPUT



(b) INTERPOLATION BETWEEN DISJOINED LINED SEGMENTS



(c) RECONSTRUCTED DATA POINTS

Fig 7-3 Compressed Data Reconstruction

Table 7-6

ESTIMATED COMPUTATION TIMES FOR DATA RECONSTRUCTION

UNIVAC Computer	Time To Compute Slope (μs)	Time to Compute N Increments (μs)								
		1	2	4	8	16	32	64	128	256
1107	294	98	196	392	784	1,568	3,136	6,272	12,544	25,088
1108	110	37	74	148	296	592	1,184	2,368	4,736	9,472

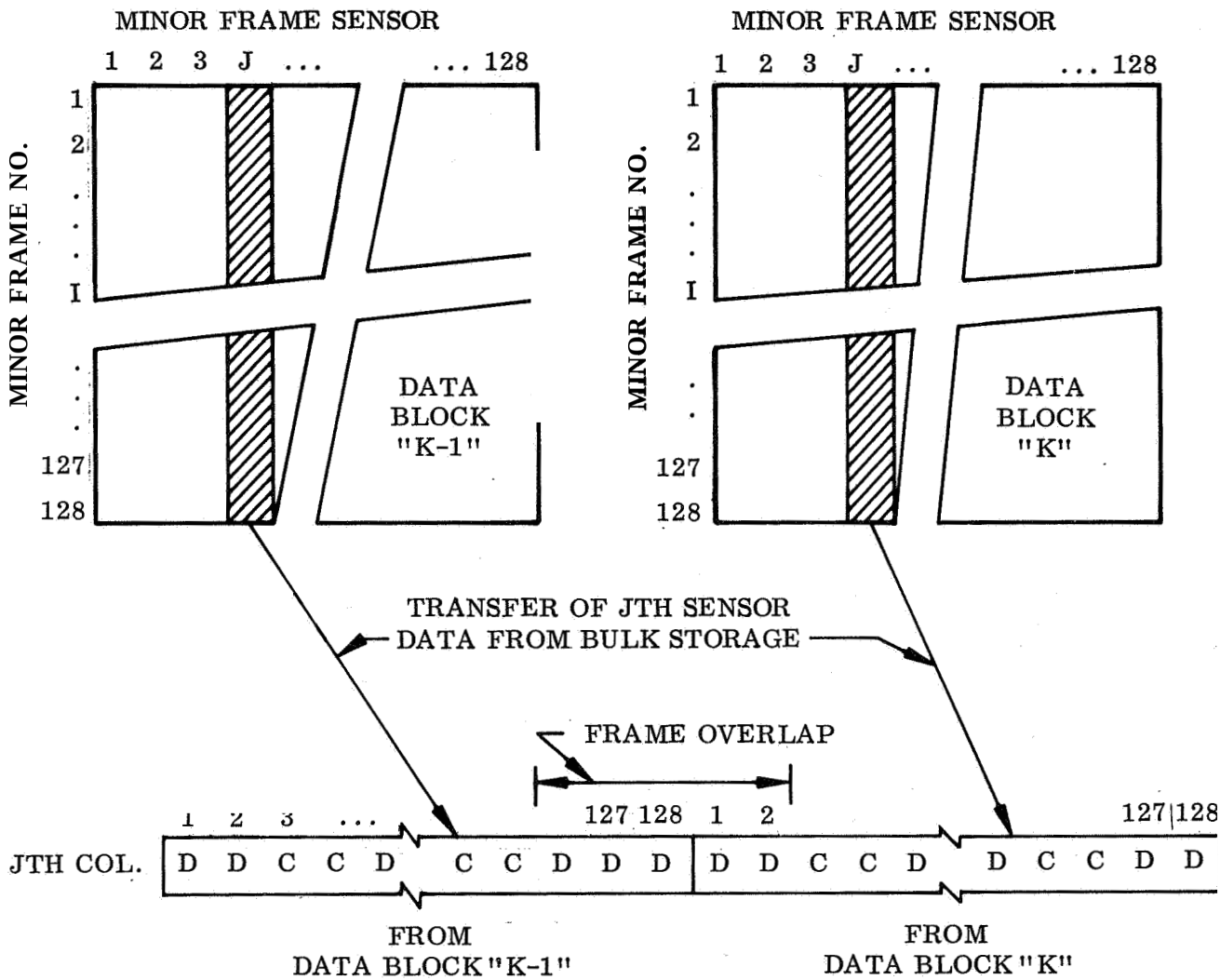
Thus, from Eq. (7 2) the incrementation equation is

$$D_i = i\bar{T}S + D_{1,2} \quad , \quad 1 \leq i \leq N - 1 \tag{7 4}$$

It was estimated that a minimum of one multiplication and three additions would be required to mechanize the replacement loop. However, because of the uncertainties involved, a safety factor of 3 was used, which set the time for incrementing at a maximum of one-third the time it would take to compute the slope. The times required for N increments are also presented in Table 7-6.

Before proceeding further, some attention should be given to the method of organizing the compressed input data in the computer for the purpose of reconstruction. It is assumed that a 128-minor frame data matrix, or data block, similar to the one shown in Fig. 7-5 is formed in the computer from each compressed data tape record. All the nonredundant samples are placed in the matrix according to their channel and minor frame numbers, and all redundant sample locations are initially filled with dummy words. The reconstruction process replaces all the dummy words with restored data words. The following paragraphs discuss this "search and fill" procedure under two operating modes; namely, random forced outputs, without regard to matrix position, and forced outputs at predetermined positions in the matrix.

Random Forced Outputs. The "search and fill" operation would proceed as presented in Fig. 7-10. This method would interrogate the data on a sensor-by-sensor basis



C = COMPRESSED DATA WORD.
D = DUMMY DATA WORD.

Fig. 7-10 Transfer of Data From Bulk to Operating Storage for Data Reconstruction

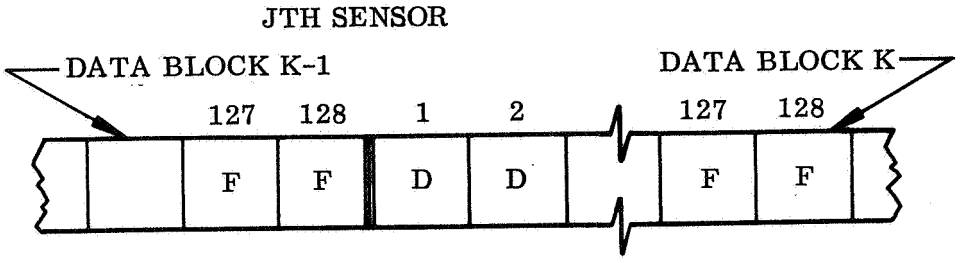
and replace the dummy words with derived data words. The method of derivation would be as follows:

- Sensors sampled once per minor frame: Each sensor would be tested on an individual basis and the dummy words replaced with calculated data words. This would be accomplished by connecting the disjoined samples by a straight line, determining the slope and intercept, and determining the value of the data by incrementation.
- Sensors sampled more [illegible] once per minor frame: The mathematical operation would be similar to that for a single sample; however, an interlacing of the elements of each column would be required. Similarly, the replacement in the memory array would require the adjustment for column positioning in the minor frame.

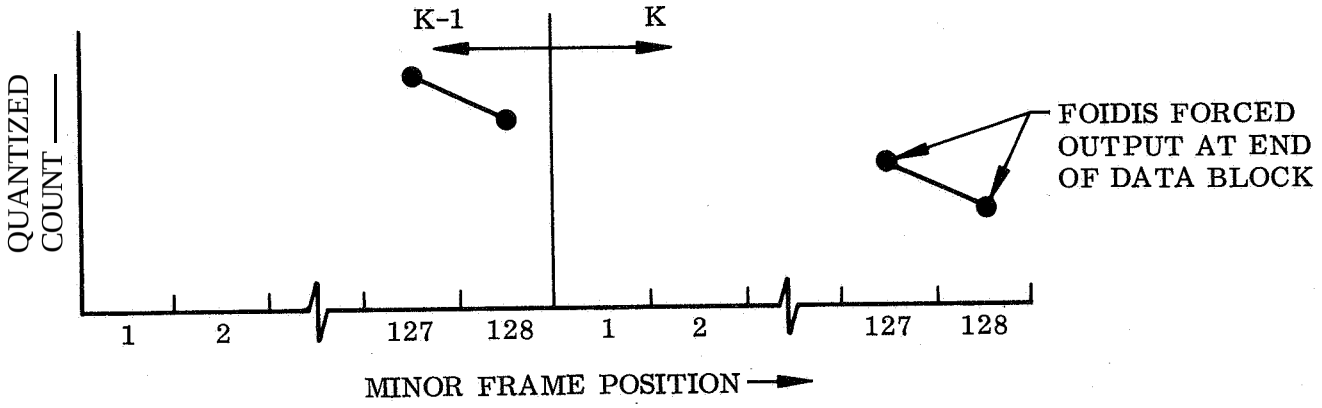
The necessity to store two data blocks simultaneously brings the storage requirements to more than 30,000 words. This would appear to necessitate the use of bulk storage (e.g., a drum). An operating storage capacity of 256 n words would also be required, where n is the maximum number of times a sensor appears on the minor frame.

Predetermined Forced Outputs. The storage requirement for data could be reduced significantly, and the search and fill operation simplified, if the following were incorporated into the data compressor design:

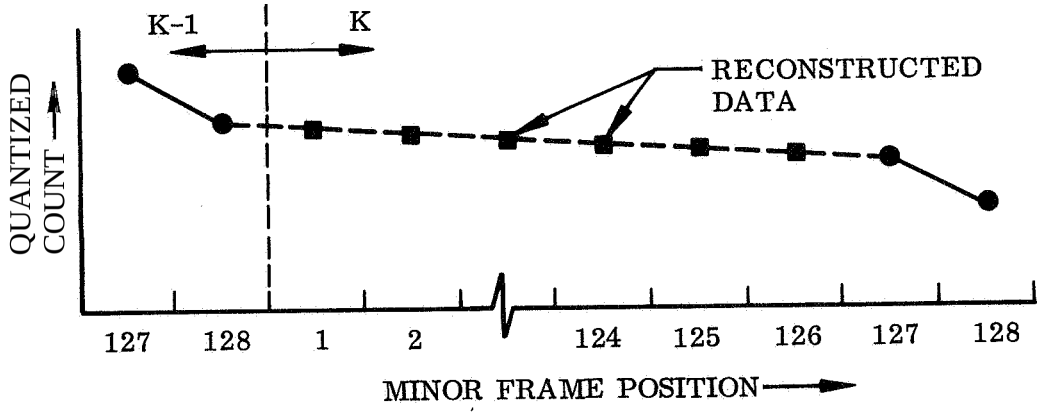
- Force a data word output in the 127th and 128th frame for each minor frame sensor. If this were done, a data storage reduction from about 30,000 to 15,000 words could be realized, because it would be necessary to retain only the forced data words of the last frame (125 cells) of the previous data block (K-1). Figure 7-11 presents a pictorial description of the operation. The reduction in storage might possibly determine where the data block would be stored. Internal storage would have the added advantage of reduced access time.
- Code the forced data word of the 127th minor frame sample so that, upon inspection of the code, it could be determined whether any change in the data had occurred in the data block for that sensor. The search and fill operation for



(a) FORCED OUTPUT IN 127TH AND 128TH FRAMES



(b) FORCED FOIDIS OUTPUT IN 127TH AND 128TH FRAME



(c) RECONSTRUCTED DATA

Fig. 7-11 Predetermined Forced Outputs

static sensors could then be reduced to (1) recognizing a flag, (2) subtracting the 128th sample of the previous block from the 127th sample of the current block, (3) dividing by 127, and (4) replacing the 126 dummy words with data words by incrementation.

7.3.1.3 Computation Time With Compression. Table 7-3 presented the estimated running times of pertinent GSFC data processing programs. This information was used as the basis for the timing graph of the decommutation program, without data compression, presented in Fig. 7-12. The graph indicates that the program is compute-bound on the UNIVAC 1107 and output bound on the UNIVAC 1108. Running times of 144 and 90 minutes for the UNIVAC 1107 and 1108, respectively, are expected.

To estimate the running times associated with data restoration prior to decommutation, it was necessary to make certain assumptions concerning the makeup of the compressed input data. The assumptions were:

- The first data record does not contain compressed data. (This provides an initial and uncomplicated starting point for the reconstruction calculations.)
- All subsequent data records contain compressed data. In addition, each record contains a forced output in the last two frames (i.e., the 127th and 128th frames).
- Reconstruction consisted of fitting a straight line segment between two data points separated in time.
- Restoration of data was by incrementation, and the restored data was placed in the decommutation memory matrix (Fig. 7-5).

To determine reconstruction and restoration times, the number of line segments per data block for all channels was varied from 1 to 42. The initial case (one line segment) represents a data record in which all parameters exhibit no significant change in trend, as shown in Fig. 7-13a. In the other extreme, a redundant or dummy word occurs every third sample, as in Fig. 7-13b. The resulting computation times are presented in Tables 7-7 and 7-8.

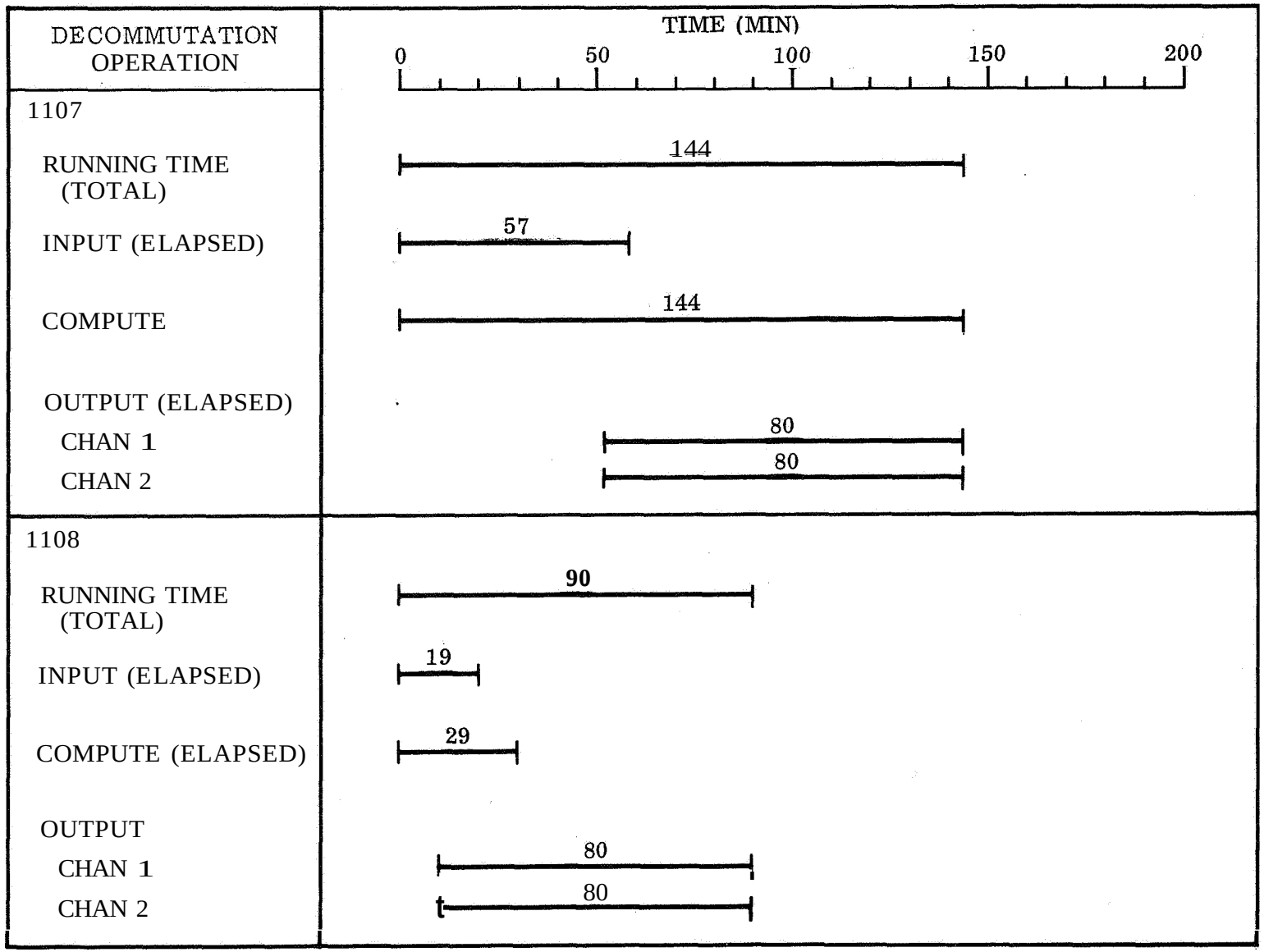
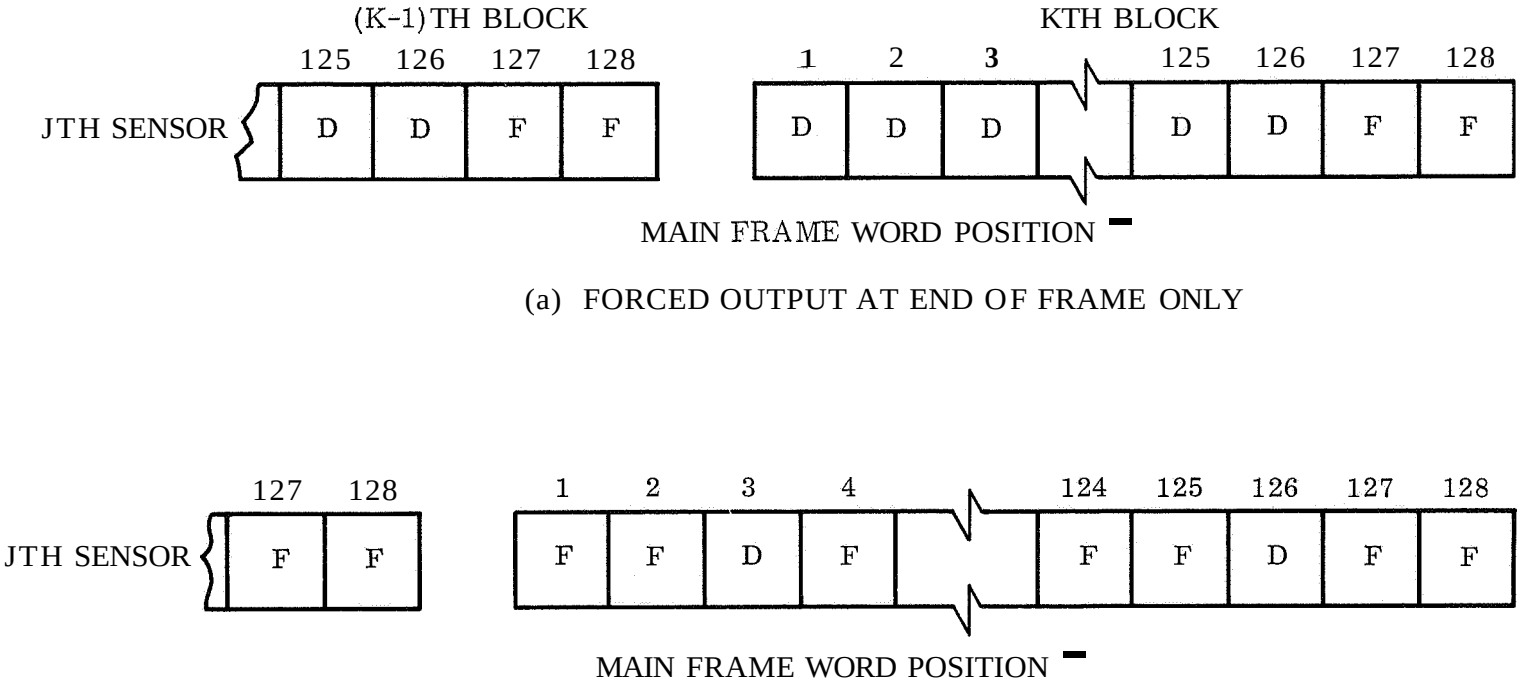


Fig. 7-12 Estimated Timing for Decommutation Program

LOCKHEED MISSILES & SPACE COMPANY



D: DUMMYWORD
F: NONREDUNDANT OR FORCEDOUTPUTWORD

Fig. 7-13 Compressed Data Arrays

Table 7-7

TIME FOR RECONSTRUCTION AND RESTORATION – UNIVAC 1107

Case	Sample Compression Ratio	No. of Nonredundant Sample Pairs in Minor Frame							Number of		Time (ms)(a)			Time Per Tape (min)(a)
		1	2	4	8	16	32	42	Interp	Incr	Interp	Incr	Total	
I	64	X							1	126	0.294	12.348	12.642	6.69
II	32		X						2	124	0.588	12.152	12.740	6.75
III	16			X					4	120	1.176	11.760	12.936	6.85
IV	8				X				8	112	2.352	10.976	13.328	7.06
V	4					X			16	96	4.704	9.408	14.112	7.48
VI	2						X		32	64	9.408	6.272	15.680	8.31
VII	1.5							X	42	44	12.348	4.312	16.660	8.83

(a) For one minor frame channel, per data block
(b) 274 data blocks

Table 7-8

TIME FOR RECONSTRUCTION AND RESTORATION – UNIVAC 1108

Case	Sample Compression Ratio	No. of Nonredundant Sample Pairs in Minor Frame							Number of		Time (ms)(a)			Time per Tape (min)(a)
		1	2	4	8	16	32	42	Interp	Incr	Interp	Incr	Total	
I	64	X							1	126	0.110	4.662	4.772	2.53
II	32		X						2	124	0.220	4.588	4.808	2.55
III	16			X					4	120	0.440	4.440	4.880	2.59
IV	8				X				8	112	0.880	4.144	5.024	2.66
V	4					X			16	96	1.760	3.552	5.312	2.81
VI	2						X		32	64	3.520	2.368	5.888	3.12
VII	1.5							X	42	44	4.620	1.628	6.248	3.31

(a) For one minor frame channel, per data block
(b) 274 data blocks

The results indicate that the additional time required for reconstruction and restoration of minor frame data is not expected to exceed 8 to 9 minutes using the UNIVAC 1107, and 3 to 4 minutes when using the UNIVAC 1108. Because the decommutation is output-bound on the UNIVAC 1108, a significant increase in running time caused by the computation time increase would not be expected. The additional computation on the UNIVAC 1107 does not appear to be significant when compared with the total running time.

7.3.1.4 Input Time With Compression. To obtain a better understanding of the relative lengths of the uncompressed and compressed data tape formats, the record lengths are plotted as functions of sample compression ratio in Fig. 7-14. As expected, at very low compression ratios the compressed format is less efficient than the uncompressed format because of the added identification word burden. However, the crossover point occurs at a compression ratio of less than 2. At ratios between 8 and 16 (a reasonable compression operating range for OGO experimental data), a decrease in average record size of approximately 4 to 1 is achieved over the uncompressed format.

The difference in total tape readin time between the uncompressed and compressed-formatted tape is formulated as follows:

Let

T_1 = total uncompressed tape passing time

T_2 = total compressed tape passing time

Then

$$\Delta T = T_1 - T_2 = CN_R(N_1 - N_2) \qquad (7\ 5)$$

where

ΔT = difference in total tape readin time

C = tape readin time per character

N_R = total number of records on the tape

N_1 = number of characters per uncompressed tape record

N_2 = average number of characters per compressed tape record

For a packing density of 556 characters per inch, the quantity C is assumed to be approximately 0.1 ms/character (Ref. 9).

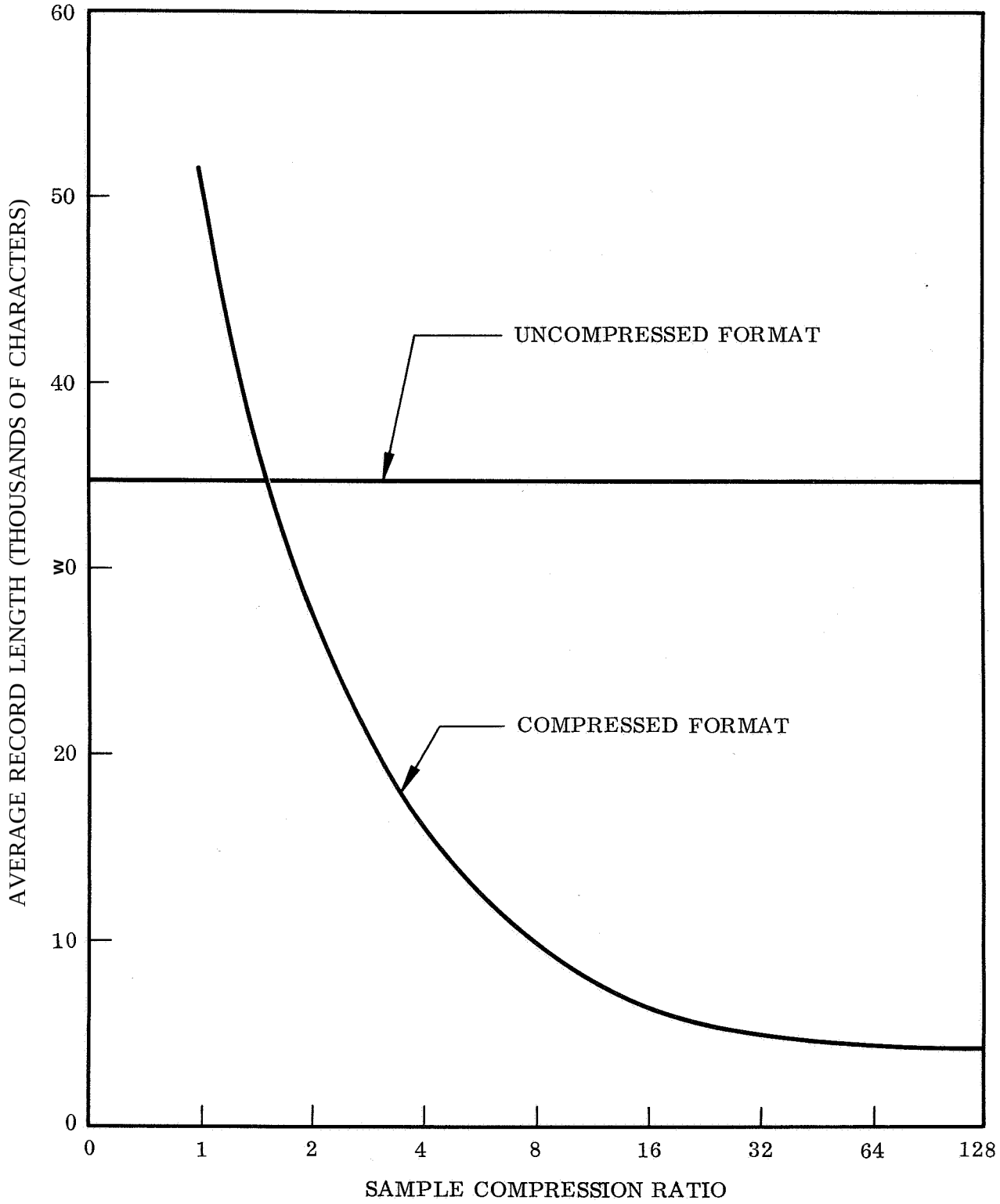


Fig. 7-14 Record Size as a Function of Compression Ratio

Studies thus far have shown a combined sample compression ratio on the S-49 experimenters' data, using the FOIDIS selector, of about 13. For this compression ratio, the average record length of the compressed tape is approximately 6800 characters. The record length of the uncompressed tape is 34,830 characters. Then, for a value of N_R equal to 274 records,

$$\begin{aligned} \Delta T &= 0.1\,(274)\,(34830 - 6800) \\ &= 7.7 \times 10^5 \text{ ms} \\ &= 12.8 \text{ minutes} \end{aligned}$$

Thus, an input tape read time reduction of almost 13 minutes would be expected under the conditions stipulated. Figure 7-15 presents the readin time difference as a function of compression ratio.

7.3.1.5 Reconstruction and Input Time Tradeoff Analysis. The computation times required to reconstruct compressed data were presented in Tables 7-7 and 7-8. A graph showing both of these es and the saving in compressed data tape readin time as functions of compression io is presented in Fig. 7-16. The graph indicates that the saving in tape readin time, in the range of compression ratios between 8 and 16, more than offsets the increase in computation time required for reconstruction, for both the UNIVAC 1107 and the UNIVAC 1108.

It will be recalled (Fig. 7-12) that the UNIVAC 1107 is currently compute-bound; therefore, a saving in tape readin time should not materially reduce the total run time. The total time for this machine would thus still be increased by approximately the amount required for reconstruction.

In the case of the UNIVAC 1108, Fig. 7-12 indicates that this machine is primarily output bound; hence, either a reduction in tape readin time or an increase in computation time would not materially affect the total computer run time.

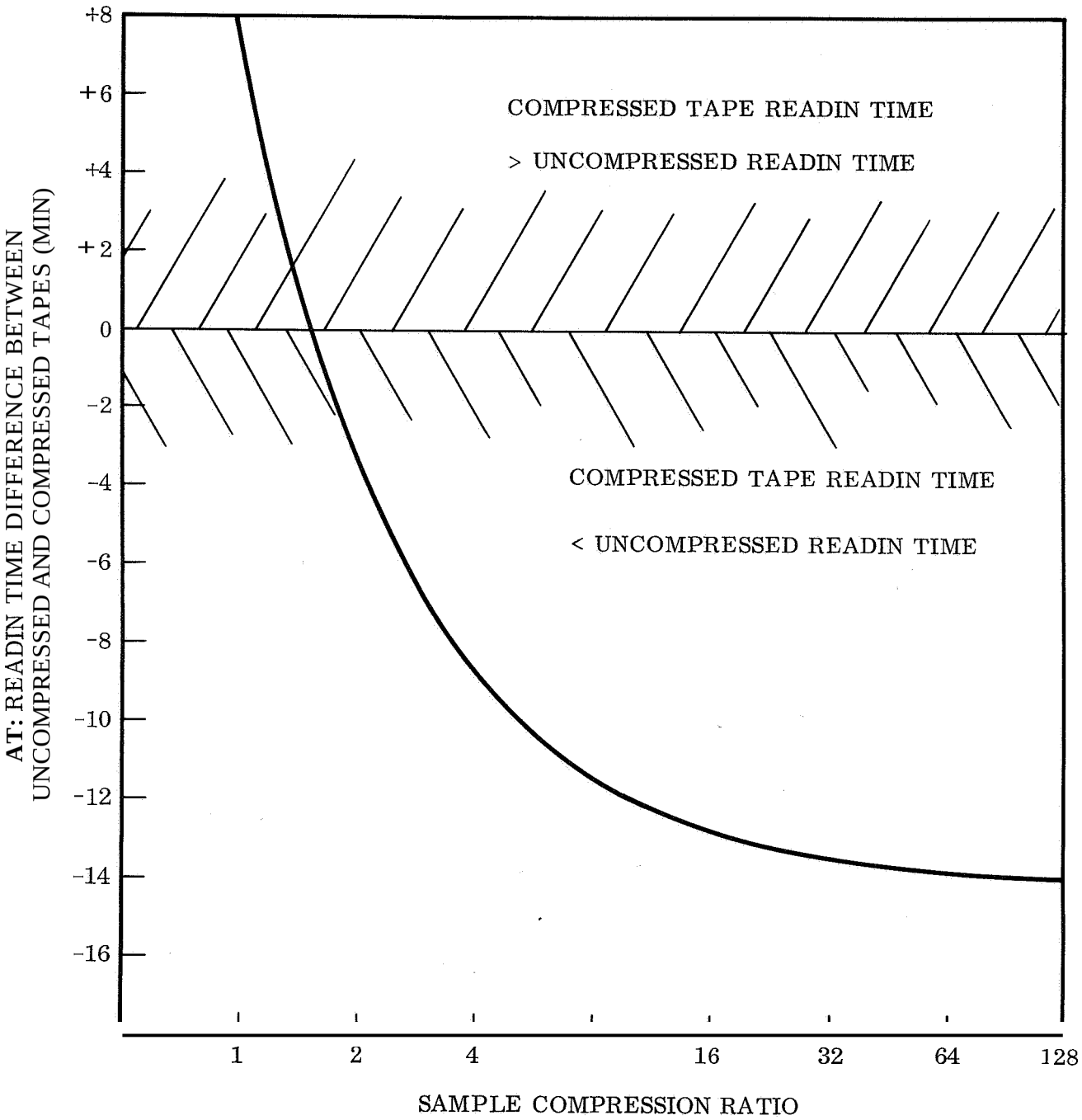


Fig. 7-15 Tape Readin Time Differences

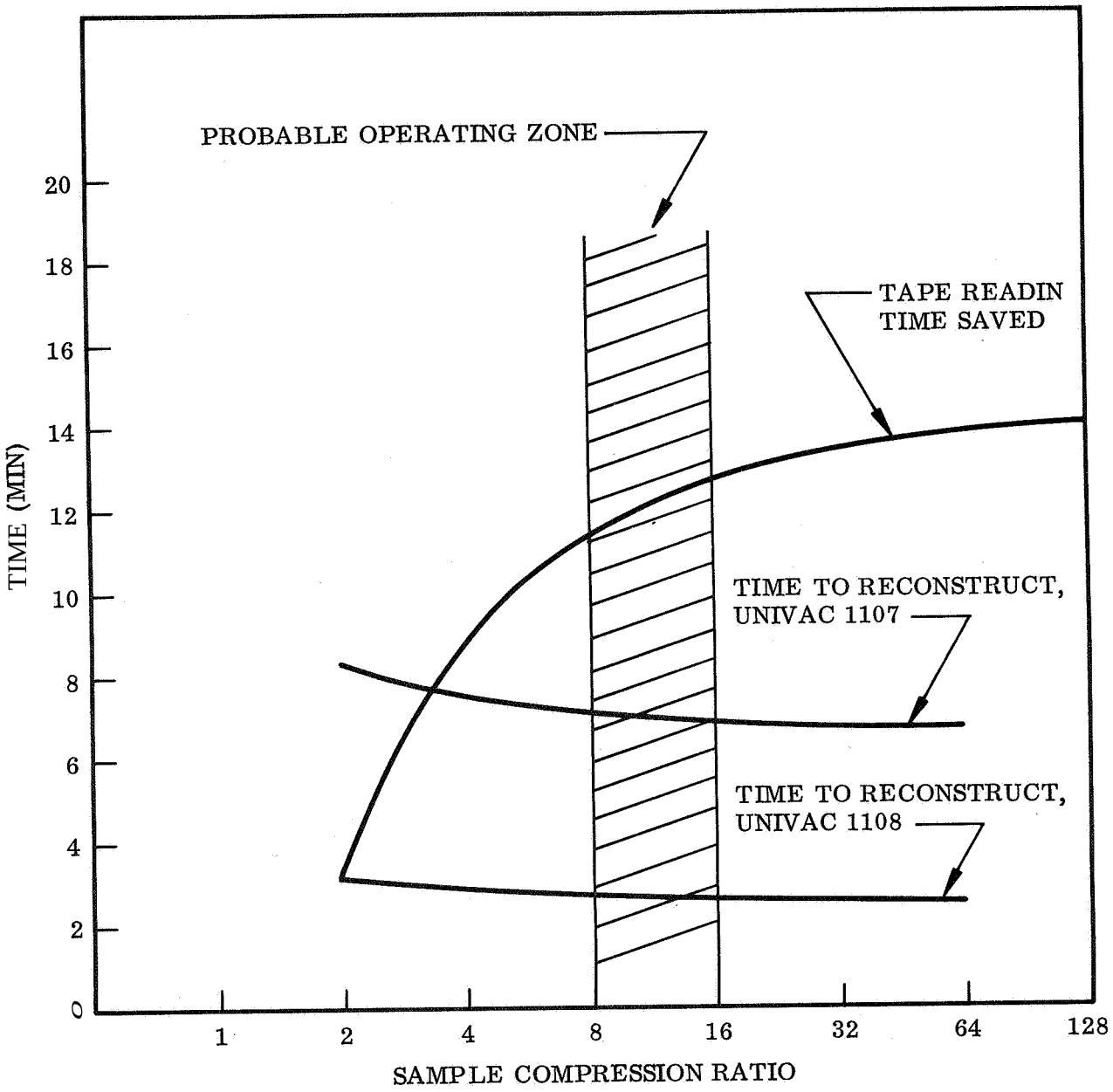


Fig 7-15 Tradeoff Between Tape Read Time and Reconstruction Time

7.3.2 Decommutation Without Data Reconstruction

One of the characteristics of data compression is that sensor identification and time information must be transmitted with each sample (or sample pair, in the case of FOIDIS or FOS4DG selectors) Because of this, decommutation, i e , the identification of each sample in terms of s sor number and time of occurrence, is inherent in the data compression process. If full advantage could be taken of this characteristic, a substantial saving in data processing time, as well as in tape storage requirements might be possible. The following paragraphs explore this possibility

7 3 2 1 Compressed Data Input Tape Format. An input tape format similar to that of the C-Tape, used in the Phase I simulation studies, was selected as the mo l for this analysis. With this format, each nonredundant data sample pair (signifying the end of one line segment and the beginning of the next) is accompanied by the s sor number and the time of occurrence of the pair. Thus, four digital words are required on the tape for each sample pair I was assumed that each word is in the floating-point mode, and consists of six 6-bit characters, for a total of 36 bits. Thus, the data can be accepted by a UNIVAC 1107 or 1108 wi ut need for any unpacking or other modifications. (This tape would be generated by decoding equipment such as that shown in Fig. 8-5)

7.3.2.2 Compressed Data Output (Experimenters') Tape Format. The format of the experimenters' tape was assumed to be exactly the same as that of the input tape, except that only samples from those sensors which are of interest to a particular experimenter would appear on his tape.

7 3.2 3 Effect of C resse Data on C ter O erat Time. Because the compressed data would be entering the computer in format discussed above, and as long as the data did t have to be reconstr ted, there would be requirement for either unpacking or decommutation In fact, the present OGO decommutation process would be reduced with compressed data, to a sorting process. The computer would

simply write each incoming group of four digital words depicting a sample pair onto the proper experimenters' tape, according to the group's sensor identification word.

The first step in the analysis was to determine the time required to read in the four words depicting a sample pair. As in Par. 7.3.1.4, the tape readin rate was assumed to be 0.1 ms/character (Ref. 9). The 24 characters comprising the four words would thus be read into the computer in 2.4 ms. Since the UNIVAC 1107 cycle time is 2.0 μs, 1200 operations can be made by the 1107 during the time it would take to read in the four words depicting a sample pair. Likewise, since the 1108 cycle time is 0.75 μs, 3200 operations could be made by the 1108 during this time. Since the computation to be done by the computer would be a simple sorting process, it was safe to conclude that the input tape reading speed, rather than the computation time, would be the determining factor in computer operating time. (Because exactly the same words would be written out on tape that are read in, it was assumed that the readout time would be comparable to the readin time, with the possible addition of tape unit change-over time, and that readout could essentially be accomplished concurrently with readin)

Because readin time was determined to be the governing factor, it became desirable to estimate the time it would take to read into the computer an amount of compressed data equivalent to 274 records of uncompressed data (the nominal tape length used throughout this study). Assuming a sample compression ratio of 13 (see Par. 7.3.1.4), there would be an average of 16,384/13 = 1260 nonredundant samples, or 630 sample pairs, for every original (uncompressed) record of 128 minor frames. Each pair would require four 6-character words, or 24 characters. Hence, each original 128-minor frame record would contain an average of 15,120 characters, in compressed format. The number of characters in a compressed tape containing the same data as a 274-record uncompressed tape would then be 4,140,000. For a readin period of 0.1 ms per character, it would take 414 seconds $\cong$ 7 minutes to read in the entire tape in compressed format. Allowing a total of 2 additional minutes for possible switching of output tape units, the total estimated operating time per input tape becomes 9 minutes, for either the 1107 or the 1108.

Comparing this time with the estimated values of 144 minutes for the 1107 and 90 minutes for the 1108 with uncompressed data (Fig. 7-12), it is evident that the introduction of data compression, without reconstruction, would reduce the computer operating times for the decommutation process to 6.2 percent of the original value for the 1107, and to 10 percent for the 1108.

7.3.2.4 Effect of Compressed Data on Tape Storage Requirements. If the modified C-Tape format assumed in this analysis were used for the input tape, an estimated storage volume decrease of 34,830 to 15,120, or approximately 2.3 to 1, could be obtained with data compression, for an average sample compression ratio of 13.

It is estimated that the reduction in the total required amount of experimenters' tape would be on the order of the sample compression ratio if compressed rather than reconstructed data were presented to the experimenters. Thus, for the cited case this reduction would be 13:1.

7.4 CONCLUSIONS

If it is assumed that complete reconstruction must be performed during data processing, so that incorporation of data compression would require no change from the present uncompressed experimenters' tape formats, then:

- For 274 records of OGO data, the reconstruction process would require about 7 minutes of added computation time during decommutation on the UNIVAC 1107, and about 3 minutes on the 1108, for sample compression ratios of 8 to 16.
- For a corresponding amount of data, the tape readin time would be reduced by about 12 minutes, assuming sample compression ratios in the range of 8 to 16.
- Because the 1107 is compute-bound in its required OGO decommutation operation, the total run time would be increased by the amount required for data reconstruction. Compared with the total run time of 144 minutes, however the increase would be only about 5 percent.

- Because the 1108 is primarily output bound, the total decommutation run time would not be materially affected by the incorporation of data compression.
- The major advantage of data compression to the data processing operation would lie in the reduction of the amount of tape required to store the input data. For a sample compression ratio of 13, which has been found to be readily achievable with OGO experimental data, this reduction would be about 4 to 1.

If, on the other hand, it is assumed that the experimenters' tapes could be written in a compressed format, so that no reconstruction would be necessary during data processing, then:

- The introduction of data compression would reduce the computer operating times for the decommutation process to an estimated 6.2 percent of the current time for the UNIVAC 1107, and to 10 percent for the 1108.
- The input tape format assumed for this portion of the analysis would allow a reduction in the amount of tape required to store the input data of about 2.3 to 1 for a sample compression ratio of 13.
- The estimated reduction in the amount of tape required to store the experimenters' data is on the order of the compression ratio. Thus, for the cited case, the reduction would be about 13 to 1

It is evident that if the experimenters' tapes were allowed to contain compressed, rather than reconstructed, data, so that full advantage could be taken of data compression during processing, the savings in decommutation time and tape storage requirements would be substantial. However, if the formats of existing, uncompressed, experimenters' tapes could not be changed, the only major advantage in the incorporation of data compression to the data processing operation would be a 4 to 1 reduction in the amount of tape required to store the input data. It would therefore seem more efficient in the long run to write the experimenters' tapes in compressed format, and to modify the experimenters' data analysis computer programs to handle the data in their new [illegible].

Section 8
DATA COMPRESSION SYSTEM CONFIGURATION

The purpose of this study was to determine the configuration of a data compression and reconstruction system for the GSFC OGO ground data transmission network, and the required ground facility modifications for incorporation of this system. Block diagrams are presented, showing the major components of the proposed equipment and where they would be added to the existing system. The study was made for both ground-based and vehicle-borne data compression.

8.1 EXISTING SYSTEM

To better understand the introduction of new components into the OGO data acquisition system, it is first necessary to become familiar with the existing system. Hence, a brief explanation of the Goddard Space Flight Center's data processing system is presented, along with a description of the various data acquisition stations.

8.1.1 Data Handling System for the OGO Satellite

Figure 8-1 shows a simplified block diagram of the OGO control center. The two major real-time data analysis functions of the control center are (1) limit-checking and service and special-purpose (SSP) operations performed by the SDS-920 digital computer, and (2) the display of up to 36 channels of data. The input data for this real-time analysis currently originates from two tracking stations, located in Rosman, North Carolina, and Fairbanks, Alaska. In addition, data word selector outputs from the other tracking stations are analyzed in near real-time. The OGO real-time system is described in Section 6.

In addition to real-time analysis, the data are processed, subsequent to acquisition, for detailed experimental analysis. In general, the data are recorded at the tracking

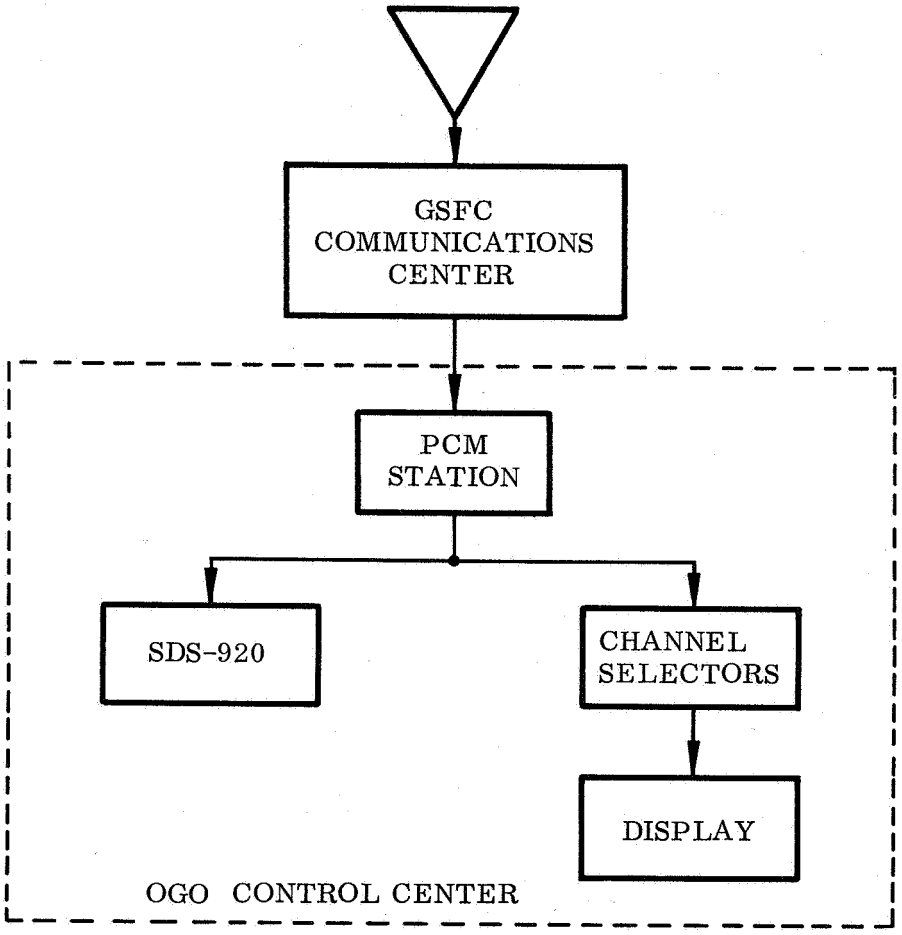


Fig. 8-1 Simplified Diagram of OGO Control Center and Communications Center

stations and transported to GSFC for processing. A description of the data processing system is presented in Section 7.

8.1.2 Existing Data Acquisition System Configurations

There are currently more than 20 NASA tracking stations which are used for data acquisition by GSFC. Table 8-1 presents the locations of the major OGO acquisition stations, the type of antenna of each, and the type of communication with GSFC.

Table 8-1

NASA/GSFC DATA ACQUISITION AND TRACKING STATIONS

Station Location	Type of Antenna	Type of Communication
Fairbanks, Alaska	85 ft parabolic	Microwave/teletype
Rosman, North Carolina	85 ft parabolic	Microwave/teletype
Johannesburg, Union of South Africa	40 ft parabolic	Teletype
Quito, Ecuador, South America	40 ft parabolic	Teletype
Santiago, Chile, South America	40 ft parabolic	Teletype
Winkfield, England	14 ft parabolic	Teletype

As already mentioned, the tracking stations at Rosman and Fairbanks have real-time data links with Goddard. The Rosman station is required to record and supply to the OGO Operations Control Center (OCC), via the microwave data link, real-time PCM, playback PCM, and special-purpose telemetry. The Fairbanks station records all data and supplies the PCM data to OGO-OCC via its microwave data link. The other stations record the data and transport these tapes to GSFC. In addition to this data acquisition procedure, each station generates a "pass summary" in conjunction with every data-reading operation. Each time one of these stations acquires data, two summaries are transmitted to OGO-OCC. First, within 10 minutes of the end of operations, a message is transmitted indicating a normal or abnormal pass. If the pass is abnormal, the abnormalities are noted. Second, within 30 minutes of the end

of operations, a more detailed summary is transmitted. This second summary contains the following:

(1) Beginning time of scheduled operations

(2) Beginning and end times of each data recording period, with the type of data (real-time PCM, playback or special purpose) recorded during that period

(3) Each command sent, time of transmission, and whether verification was received

(4) Data word selector outputs as scheduled, giving the word number, frame number, description of readout in volts or octal count and the time of readout

(5) All special or unusual events deemed important

Items (3) and (4) are omitted from the pass summary for Rosman operations. Additional requests for information may be placed on the station by the network controller. Figure 8-2 shows the data acquisition stations in simplified block diagram form. The specific capabilities for the Rosman and Fairbanks stations are indicated.

8 2 DATA COMPRESSION SYSTEM REQUIREMENTS

A data compression system may be divided into two basic elements a compressor and a reconstructor A typical compressor, shown in Fig. 8-3a consists of a data compression selector a control (to code the data into a suitable transmission bit stream) a buffer and a buffer queuing control system. A typical reconstructor, shown in Fig. 8-3b consists of a decoder (to transform the data from the bit stream into a "compressed data' format*) an organizer/buffer (to prepare the data for reconstruction into the original multiplexer format), and an interpolator (to actually reconstruct missing data points). When an interpolator is used to compress the data, the reconstruction process is delayed by the forced-readout time periods of the data sensors. This delay is necessary in order to ensure that all data can be reconstructed and displayed in time-synchronism, in the same manner as it is currently displayed. The

*The compressed data format contains the sensor identification with each nonredundant data sample or pair, along with information from which the time of each sample can be easily derived.

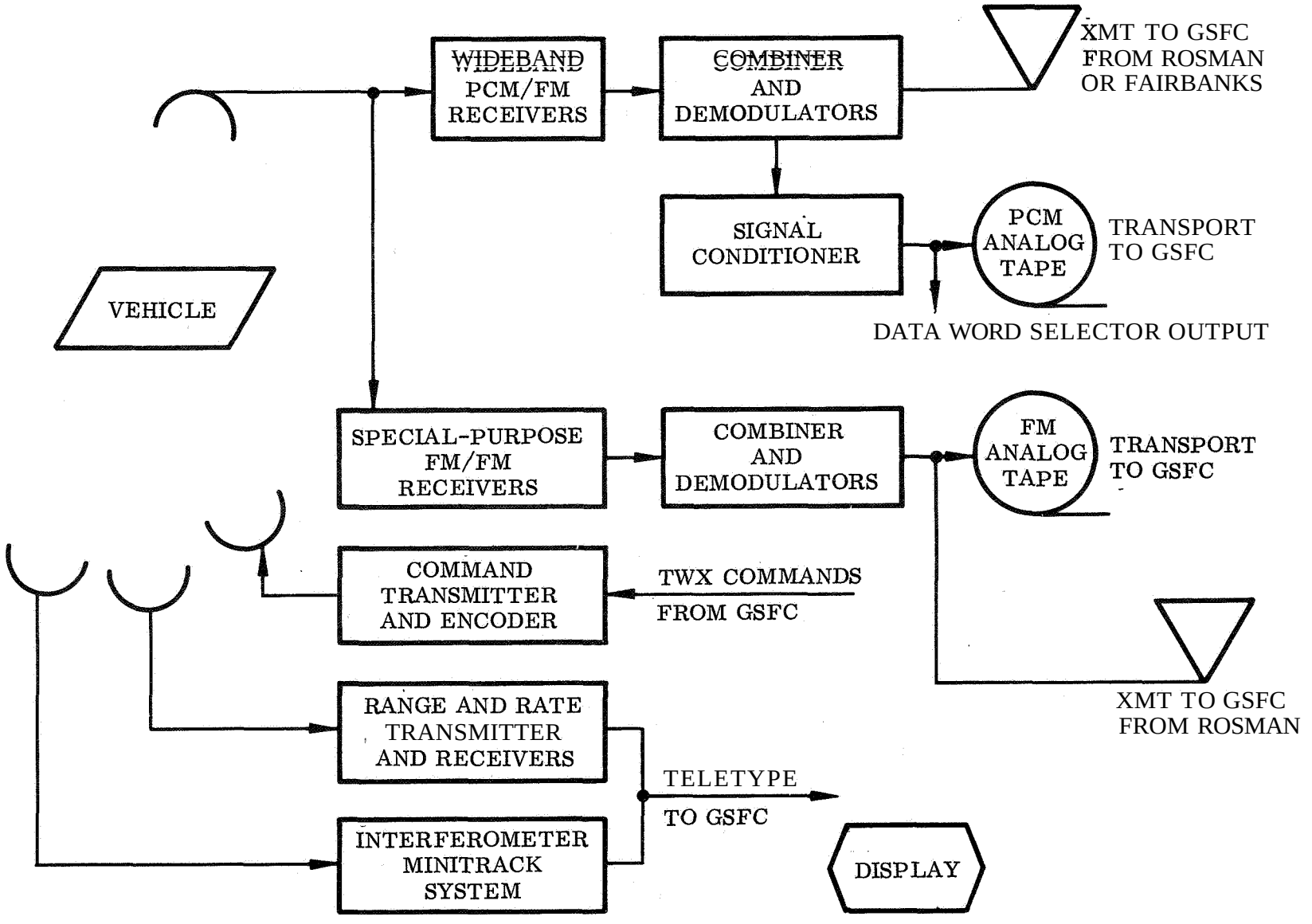


Fig. 8-2 Data Acquisition Station

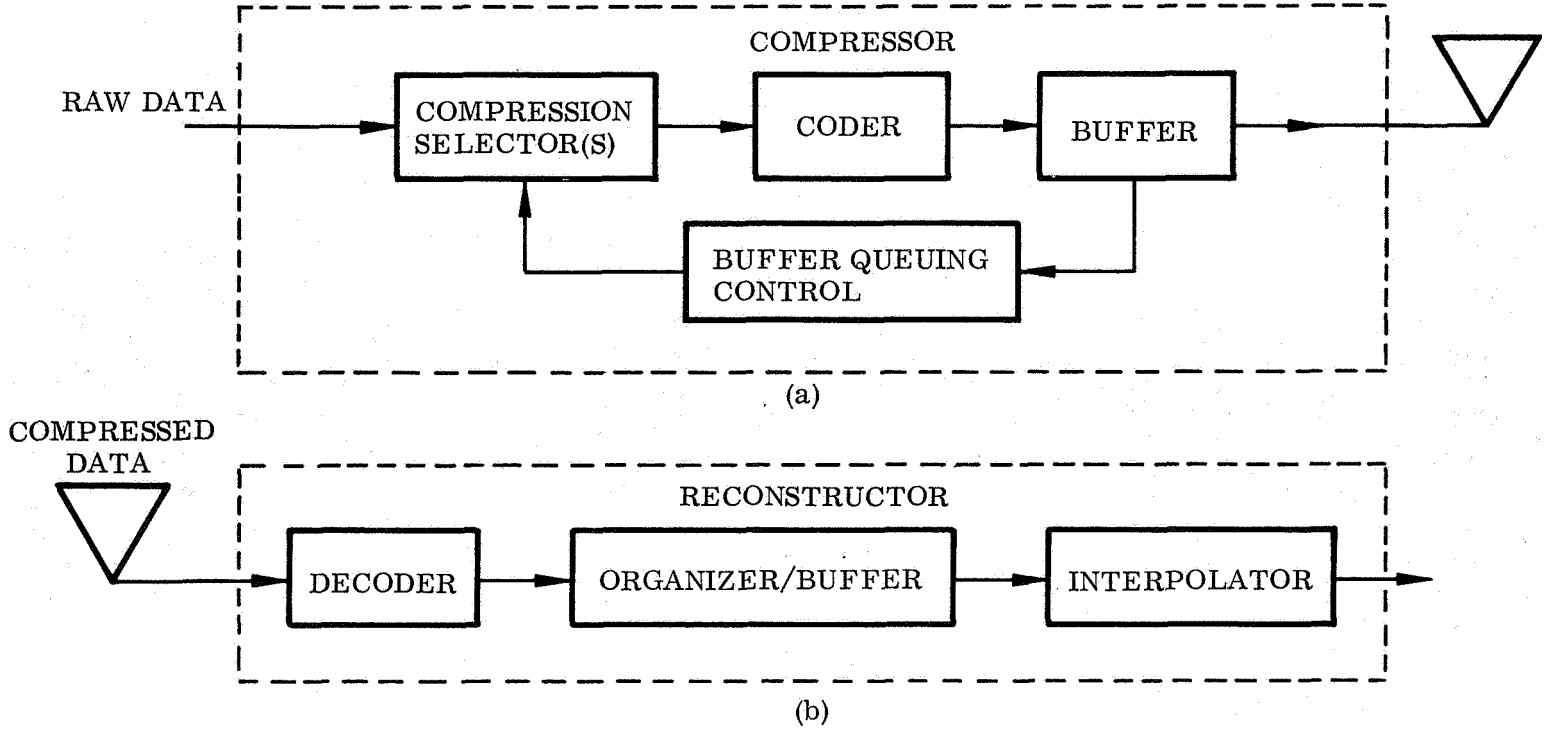


Fig. **8-3** Data Compression System

incorporation of the data compression and reconstruction system components shown in Fig. 8-3 into the GSFC OGO system is discussed in the following paragraph.

8 3 DATA COMPRESSION AND THE GODDARD SYSTEM

8 3.1 Vehicle-Borne Data Compression

With the data compression system on the vehicle, data acquisition stations without real-time data links would have a complete reconstruction capability, as shown in Fig. 8-4. The reconstructed data would be necessary so that the data word selector outputs needed in the pass summary could be obtained. The data tapes would be recorded in a compressed data format (after decoding). If the necessity for the data word selector outputs could be eliminated, or, due to the reduction in bit rate caused by compression, a real-time link suitable for transmission of compressed PCM data could be established between GSFC and these stations, there would be no need for a data reconstructor at these remote sites and the coded compressed PCM could be transmitted directly to GSFC. The data acquisition stations which already have real-time data links (Fairbanks and Rosman) need no reconstruction facility because the coded compressed PCM can be transmitted to GSFC for reconstruction.

Figure 8-5 shows how the reconstructor would be incorporated into the OGO Control Center shown in Fig. 8-1. The compressed PCM bit stream decoder would output compressed-formatted data to the SDS-920 computer for analysis, to a tape recorder for future data processing, and to an organizer/buffer for subsequent interpolation and display.

Two methods of introducing compressed data to the SDS-920 were discussed in Par. 6.4.1.1. The first method (the one used as the study model) requires the use of a buffer along with the decoder, and inputs the data in such a way that limit checking and SSP operations are time-shared on the basis of an 18-ms minor-frame period. The second method inputs the compressed data samples at a constant rate, and the

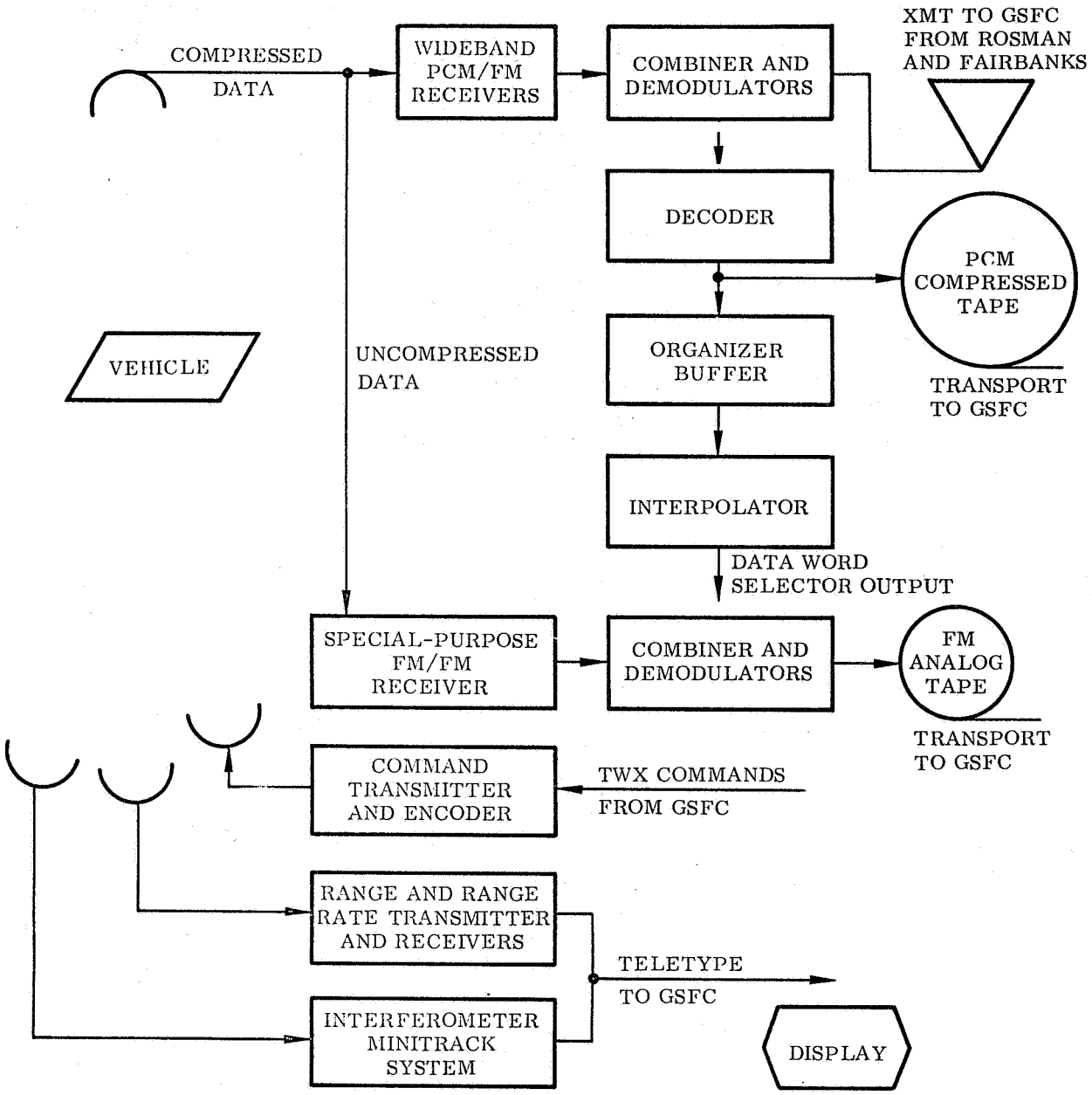


Fig. 8-4 Remote Station With Vehicle Compression

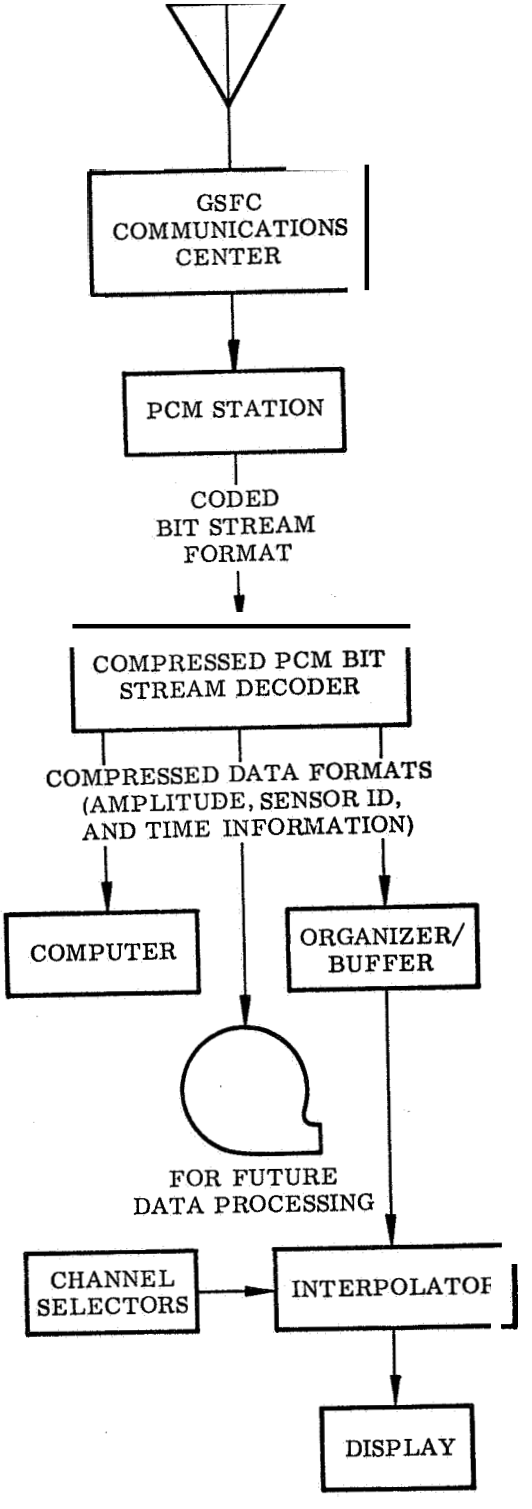


Fig. 8-5 GSFC Compressed Data Handling System

b ering action is incorporated within the computer The second method was chosen in this study for the sake of overall system simplicity.

The compressed format was chosen for the data processing tapes instead of a reconstructed data format. This was done in spite of a possible need for reconstruction during decommutation, because it is believed that a 5-percent increase in UNIVAC 1107 running time (assuming reconstruction in the 1107 is necessary) would be more than offset by the 4:1 reduction in the amount of tape required to store the input data.

If vehicle compression is used and the entire data acquisition system remains unaltered, it would be necessary to have one compressor (on the vehicle) and five reconstructors (one at GSFC, and four at the non-real-time data acquisition stations). However if real time data links could be added to the other four remote stations because of the bandwidth compression, then only one compressor (on the vehicle) and one reconstructor (at GSFC) would be needed.

8 3.2 Ground-Based Data Compression

If the compression of the data could allow all the remote stations to be real time links to GSFC, then a compressor would be put at every remote station, as seen in Fig. 8-6. Note that the compressed data are also recorded for transporting to GSFC. This guards against a data link failure. If the present system is maintained, then only Fairbanks and Rosman would have compressors. If no real-time transmission link were available at the other stations, data compression would still conserve magnetic tape. However, it is believed that this saving alone would not justify the installation of data compressors. In either case, a reconstructor, as shown in Fig. 8-5, would be needed at GSFC. Thus, one reconstructor and either two or six compressors would be required to implement this design of a ground-based data compression system.

8 4 DATA COMPRESSION MODEL

After consideration of a number of weighted factors Section 5 concluded that the zero-order predictor (ZOP) is the most desirable sampling vector for both spaceborne

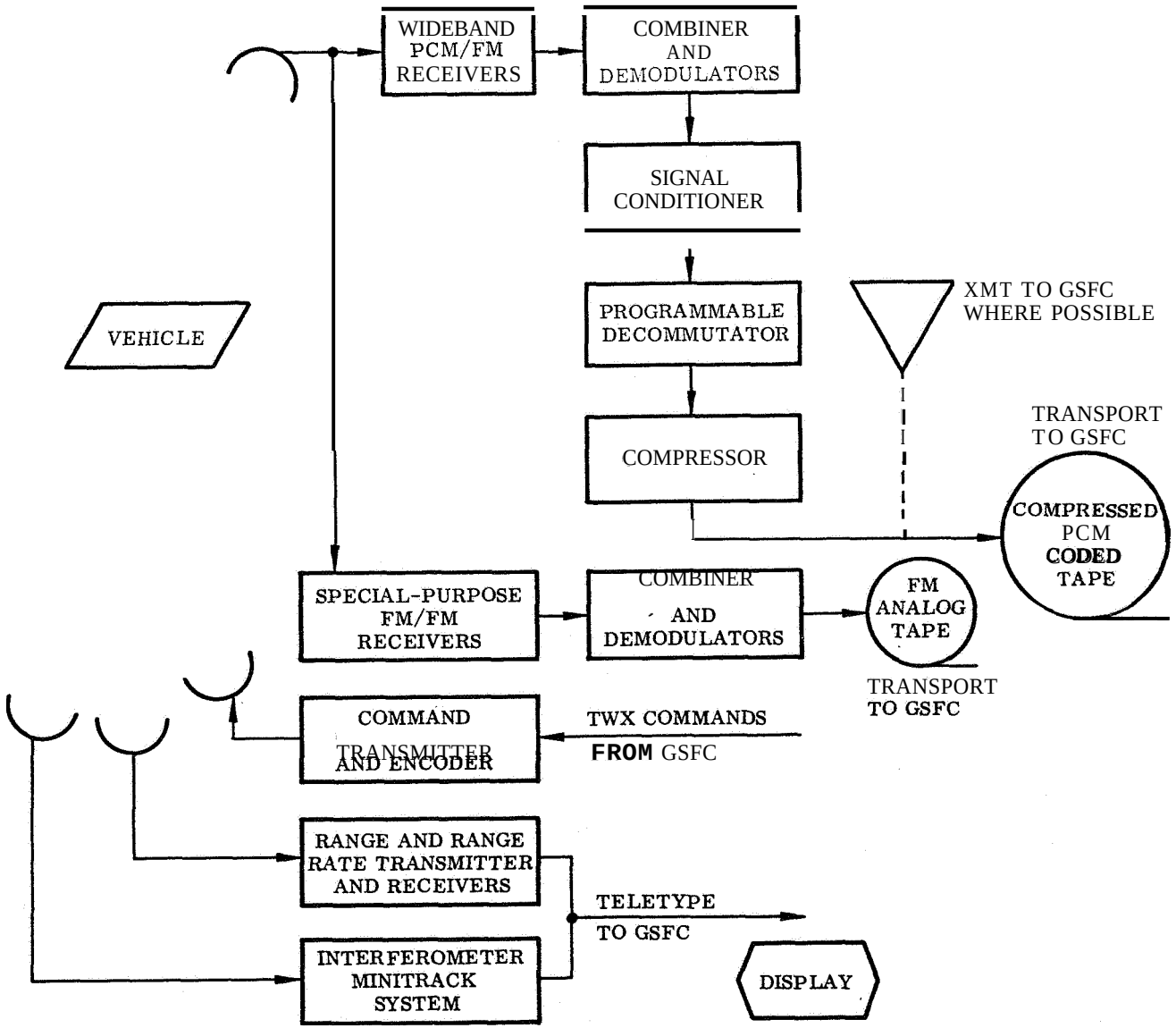


Fig. 8-6 Remote Station With Ground Compression

and ground-based applications. This compression model is thus recommended for the system described above. Paragraph 7.3.1 states that only 2.3 percent of the data are contained in the subcom channels, and because compression and reconstruction of the subcom channels (using interpolator selectors) would require storage capacities many times greater than those needed for the minor-frame channels, it may not be advisable to compress the subcom channels in an OGO data compression system. However, the use of the ZOP selector on the subcom channels would permit relatively high compression with little increase in complexity, both in compression and reconstruction.

It is assumed that any degradation of the OGO digital data would render them essentially useless to the data analyst. Hence, even though the same compression algorithm would be used for both the analog and the digital sensors, the tolerance would be set so low on the digital sensors that even a change in the least significant bit position would be noted. Possibly bit plane encoding (BPE) with no Class C planes would perform better on these sensors than do the polynomial selectors with low values of tolerance. It is also possible that use of BPE would alleviate the queue-buildup problems described in Section 3, since the buildup was observed to be largely attributable to the digital sensors.

If the same philosophy regarding digital data degradation is applied to queuing control, then the digital sensors should not be involved in the control function. Because of the large proportion of digital channels on the OGO satellites, the queuing control study found that application of control to only the analog sensors was very ineffective, even with high analog waveform degradation. A good way to circumvent this problem would be to decrease buffer input rate during high data activity only by completely eliminating sensors both digital and analog, on a priority basis. Then the experimenter would at least be assured that the data he obtains have acceptable accuracies.

The method of channel and time coding (compressed PCM bit stream format) that was recommended in Par. 7.1 of Ref. 4 was used for the noise study of Phase II. This

method provides a minor frame channel identification word along with each transmitted sample. A flag must be used to identify the first word of each minor frame, and a forced readout must be made during each empty minor frame. This general format is recommended for the system described above, but it would be advisable to transmit the flag bits redundantly in order to reduce the reconstructed data degradation caused by flag bit errors. (See Section 4.)

8.5 CONCLUSIONS AND RECOMMENDATIONS

This section has presented the basic configuration of a data compression and reconstruction system for the GSFC OGO ground data transmission network. The placement of data compression system components has been indicated for both vehicle and ground compression, and for the existing and improved data acquisition stations. With average S-49 bit compression ratios on the order of 10, the 64-kilobit transmitted bit rate would be reduced to about 6.4 kilobits. If data links could be established between GSFC and the four secondary data acquisition stations which could handle this bit rate, data compression could allow real-time analysis at GSFC from data acquired at any of these stations.

Paragraph 8.4 stated that a good way to solve the problem of queuing control with the OGO data, which contains a large proportion of digital channels, would be to eliminate temporarily both digital and analog sensors on a priority basis as the buffer tended to fill. This method of queuing control is recommended for consideration in this system.

It is recommended that a detailed data compression system hardware design be carried out for the OGO network, based on the system configuration described.

Section 9
CONCLUSIONS AND RECOMMENDATIONS

This section presents important conclusions derived from the several facets of the data compression study, and offers recommendations for continued study and analysis, with special emphasis on problems associated with NASA satellite telemetry systems.

9.1 CONCLUSIONS

9.1.1 Compression Model Study

- After weighing the performance of the nine compression models tested during Phase II, with respect to seven performance criteria, it is concluded that the zero-order predictor (ZOP) is the most desirable selector for both ground-based and spaceborne applications. The main reason for the high overall rating of ZOP is its relative simplicity and the ease of reconstructing ZOP-compressed data.
- The use of a higher order polynomial algorithm to further compress data which have been initially compressed with a low-order selector would increase compression performance only in those cases where the high-order selector could not have been used originally (e.g., because of excessive complexity in a spaceborne application).

9.1.2 Queuing Control Evaluation

9.1.2.1 Evaluation With S-6 Data.

- Although the adaptive sample deletion method is a more effective means of controlling the buffer queue length than any of the adaptive filtering techniques, it does not perform as well as the technique using adaptive tolerance. Excessive distortion of the reconstructed data waveform is the main reason for the poor performance of the sample deletion system.

- In an adaptive tolerance control system, the addition of integral feedback [i.e., $\sigma_I \neq 0$ in Eq. (3.1)] definitely enhances overall control system performance.
- For the set of parameter values and performance criteria used in this investigation, maximum adaptive tolerance control system performance is obtained for values of the queue length feedback constant β_Q [compare Eq. (3.1)], in the range of 0.1 to 0.2, combined with values of the integral feedback constant σ_I in the range of 0.00005 to 0.0001.

9.1.2.2 Evaluation With S-49 Data.

- If control is applied only to the analog sensors, queuing control is virtually ineffective. This is because (1) a relatively large percentage of the S-49 sensors were digital, and (2) the digital sensors were the sensors that were causing the queue buildup. If the queue length is to be effectively controlled it is necessary also to apply some method of control to the digital sensors. Since it is assumed that a partial degradation of the digital outputs would be unacceptable to the user of the data it is suggested that sensor outputs, both analog and digital, be completely eliminated on a priority basis as the queue tends to build up. With this method, all the data the user obtained would be acceptable, and the queue length could probably be effectively controlled.

9.1.3 Compressed Data Transmission Error Analysis

- Visual examination of extensive amounts of reconstructed data from contaminated bit streams has shown, for the coding and selector models used in this study, that contaminated reconstructed waveforms are characterized by few (compared with the total number) erroneous line segments, which deviate markedly from their true positions. This characteristic generally accounts for the relatively high rms errors observed.

- At normal operating signal-to-noise ratios, which would produce bit error rates on the order of 10^{-6} errors per bit, ground transmission link noise will not degrade compressed data to any appreciable extent, if compressed PCM formats and bit stream decoding procedures are designed to minimize the effect of link noise.
- In an empirical analysis of this type, rms error alone by no means presents a complete picture of the extent of data waveform degradation caused by link noise; visual observation of the reconstructed waveform is also necessary. However, the calculation of theoretical rms error is worthwhile, so that comparisons can be made among compression methods, compressed data formats, and decoding techniques.
- For the compressed data bit stream format examined, it was found that flag bit errors accounted for an exceptionally high percentage of the reconstructed data rms error. Therefore, flag bits should be transmitted redundantly.

9.1.4 Real-Time Computer Analysis

- For a combined sample compression ratio of 13 (which is a very conservative estimate), the average MONITOR time required by the SDS-920 computer to handle OGO data in the accelerated subcom mode would be approximately 7.0 ms in each 18-ms minor frame period. Thus 11.0 ms would be available, on the average, for service and special-purpose (SSP) calculations. This is compared with the 6.0 ms per minor frame period which is now available for SSP calculations, without compression.
- Under conditions similar to those described above, the SDS-930 computer would require an average MONITOR time of 1.5 ms per minor frame period, leaving 16.5 ms available for SSP calculations. This is compared with the 15.4 ms which would be available on the SDS-930 without compression.
- The MONITOR time required for the accelerated subcom mode was found to be the only operating time which would be materially affected by the introduction of data compression.

9.1.5 Data Processing Computer Analysis

If it is assumed that complete reconstruction must be performed during data processing, so that the incorporation of data compression would require no change from the present uncompressed experimenters' tape formats, then

- For 2094 records of OGO data the reconstruction process would require about 9 minutes of added computation time during decommutation on the UNIVAC 1109, and about 3 minutes on the 1108, for sample compression ratios of 8 to 16
- For a corresponding amount of data, the tape readin time would be reduced by about 12 minutes, assuming sample compression ratios in the range of 8 to 16
- Because the 1107 is commutated in its current OGO decommutation operation, the total run time would be increased by the amount required for data reconstruction. Compared with the total run time of 144 minutes, however, the increase would only be about 5 percent.
- Because the 1108 is primarily output bound, the total decommutation run time would not be materially affected by the incorporation of data compression.
- The major advantage of data compression to the data processing operation would lie in the reduction of the amount of tape required to store the input data. For a sample compression ratio of 13, which has been found to be readily achievable with OGO experimental data this reduction would be about 4 to 1

If, on the other hand, it is assumed that the experimenters' tapes could be written in a compressed format, so that no reconstruction would be necessary during data processing, then

- The introduction of data compression would reduce the computer operating time s for the decommutation process to an estimated 6 2 percent of current time for the UNIVAC 1107, and to 10 percent for the 1108.
- The input tape format assumed for this portion of the analysis would allow a reduction in amount of tape required to store the input data of about 2 3 to 1 for a sample compression ratio of 13.

- The estimated reduction in the amount of tape required to store the experimenters' data is on the order of the compression ratio. Thus, for the cited case, the reduction would be about 13 to 1.

These results show that the introduction of data compression into the OGO telemetry system would reduce the data processing time and tape storage requirements by a substantial margin if the data were delivered to the experimenters in compressed form. If this were done, the entire OGO data handling system could operate on compressed data, and the full benefit of data compression would be realized.

Note that the conclusions from both the Real-Time Computer Analysis (Par. 9.1.4) and the Data Processing Computer Analysis (Par. 9.1.5) are based on the use of a first-order interpolator as the compression model, which requires a complex reconstruction operation. If the zero-order predictor were used, as is recommended for the OGO data compression system (Par. 8.4), the times required for reconstruction would be considerably reduced.

9.2 RECOMMENDATIONS FOR FUTURE WORK

This study has shown that the incorporation of data compression into the Orbiting Geophysical Observatory (OGO) ground telemetry network could:

- Reduce the data processing decommutation time from the present requirements by at least a factor of 10
- Reduce the experimenters' tape storage requirements by about a factor of 13, and those of the decommutation input tape by more than a factor of 2
- Double the average available service and special-purpose (SSP) calculation-time by the SDS-920 computer in the accelerated subcom mode, during real-time data analysis

It was further shown that, with proper system design, ground transmission link noise will not appreciably degrade the compressed data. The placement of data compression

into the OGO system, and major design features, such as sample selection, buffer overflow prevention, and channel and time coding techniques, have been presented.

The incorporation of data compression into the OGO ground telemetry system, in the configuration outlined in this report, is now recommended. To attain this goal, the following steps are considered desirable.

- Demonstration: The potential of data compression should be demonstrated by compressing selected channels of OGO data at a remote location from GSFC, transmitting the compressed PCM data to GSFC over a land line, and reconstructing and displaying the data at GSFC. The entire operation should be accomplished in simulated real time, using existing equipment at both the transmitting and receiving locations.
- Hardware Design. A detailed hardware design, based on the system configuration presented in Section 8, should be performed on a data compression and reconstruction system for the OGO ground telemetry network.
- Breadboard Construction and Testing. The appropriate breadboard equipment should then be built, based on the detailed design. This equipment should be demonstrated in the OGO system, on a parallel basis with uncompressed data, so that the two methods of data handling and transmission can be compared.
- Data Compression System Installation. The final version of the data compression and reconstruction system should be constructed and installed into the OGO telemetry network.

Because data compression research and development have advanced the state-of-the-art sufficiently that an effective system can now be profitably constructed, the steps outlined above can and should be immediately begun. However, this activity should be supplemented by continued study of a more general nature, to improve the compression efficiency and performance in a number of different areas. At present, additional study is most badly needed on the following specific items:

- Redundancy Removal From Periodic Data. The study of broadside and other techniques to remove redundancy from sampled periodic data should be continued and intensified. Because so much of the NASA/GSFC satellite data are periodic, the importance of this effort cannot be overemphasized.

- Wild Point Rejection. By employing techniques which are similar in principle to some redundancy removal algorithms, it is possible to reject, before compression, most wild sample points caused by transmission link noise. The elimination of these points will increase the compression ratio and should also enhance the utility of the reconstructed data. A complete study of wild point rejection techniques and their relationship to the compression of GSFC telemetry data should be made.
- Parameter Extraction. A comprehensive study of parameter extraction techniques (called "signal reduction" in Ref. 4), and of their integration with redundancy-removal methods, should be made.

Section 10
BIBLIOGRAPHY

1. Lockheed Missiles & Space Company, First Quarterly Report for PCM Telemetry Data Compression Study, Phase II, by Bechtold, Bjorn, and Medlin, M-62-66-1, Sunnyvale, Calif., Jan 1966

2. -----, Second Quarterly Report for PCM Telemetry Data Compression Study, Phase II, by Bechtold, Bjorn, and Medlin, M-62-66-2, Sunnyvale, Calif., Apr 1966

3. -----, Third Quarterly Report for PCM Telemetry Data Compression Study, Phase II, by Bjorn, Friccero, and Medlin, M-62-66-3, Sunnyvale, Calif., Jul 1966

4. -----, Final Report for PCM Telemetry Data Compression Study, Phase I, by Bechtold, Medlin, and Weber, M-62-65-2, Sunnyvale, Calif., Oct 1965

5. IEEE Trans. on Space Electronics and Telemetry (Correspondence), by Karp, Vol. SET-10, Sep 1964, p. 124

6. ADCOM, Inc., Study, Evaluation, and Analysis of Unified S-Band System for Apollo Ground Network, Fourth Quarterly Progress Report: 1 April 1965 – 30 June 1965, Contract No. NAS 5-9702, Cambridge, Mass., 1965

7. Scientific Data Systems, SDS 920 Computer Reference Manual, SDS 900009A, 1963

8. -----, SDS 930 Computer Reference Manual, 90 00 64D, Feb 1966

9. International Business Machines, Inc., IBM 7090 Principles of Operation, File No. 7090-01, Form A22-6528-5, Revised Aug 1963

Appendix I

COMPARISON OF COMPRESSION ERROR WITH TOTAL RMS ERROR

This appendix describes an empirical comparison made between the total rms error encountered in reconstructed data that were contaminated with transmission link noise with the rms error caused by compression alone. The comparison was made on S-6 Sensor 9, compressed with the FOIDIS selector. The rms error for Sensor 9 due to compression only was 1.66 counts. The rms error due to both compression and bit errors was 24.49 counts. Since the noise study deals with errors in compressed-reconstructed data caused by noise-imposed bit errors, and not errors caused by compression, it would be desirable to eliminate the portion of the total reconstructed data error caused by compression alone, if that portion proved to be a significant part of the total. Since the errors from the two sources are independent, and thus additive, the rms error caused by noise alone can be found in the following manner:

Let Σ_o = a magnitude error due to compression

Σ_b = a magnitude error caused by a bit error due to noise

N = number of samples processed (14,400 for the data processed here)

E_{rms} = total reconstructed data rms error (=24.49 for Sensor 9)

$(E_c)_{rms}$ = rms error due to compression alone (=1.66 for Sensor 9)

$(E_b)_{rms}$ = rms error caused by bit errors alone

Then,

$$\Sigma_{rms} = \frac{\qquad}{\qquad} = \frac{\qquad}{\qquad}$$

Because Σ_o is symmetrically distributed about zero the term $2\sum E_c \Sigma_b \rightarrow 0$

Therefore,

$$E_{rms} = \sqrt{\frac{\sum E_c^2 + \sum E_b^2}{N}} \qquad \text{(I. 1)}$$

Since

$$(E_c)_{rms} = \sqrt{\frac{\sum E_c^2}{N}}$$

$$\sum E_c^2 = N(E_c)_{rms}^2$$

For Sensor 3

$$\sum E_c^2 = 39,680$$

From Eq. (I. 1),

$$Z4\ 4\ 3 = \sqrt{\frac{39,680 + \sum E_b^2}{14,400}}$$

$$600\ \ 14\ 400 = 39,680 + \sum E_b^2$$

$$E_b^2 = 8,600,320$$

Therefore

$$(E_b)_{rms} = \sqrt{\frac{8,600,320}{14,400}}$$

$$(E_b)_{rms} = 24.44$$

It is seen that $(E_b)_{rms} \cong E_{rms}$, since the contribution of the error due to compression only is exceedingly small. Although the value of $(E_c)_{rms}$ is easily obtainable, the cost in computer time needed for its evaluation did not justify the increase in accuracy from using it.