NASA CR 65498





# ADAPTIVE COMPRESSIVE TELEMETRY TECHNIQUES





IBM®

Final Report

FINAL REPORT

# ADAPTIVE COMPRESSIVE TELEMETRY TECHNIQUES

Contract No. NAS 9-4618

*31 August 1966*

PREPARED FOR

Manned Spacecraft Center
Houston, Texas

Federal Systems Division
INTERNATIONAL BUSINESS MACHINES CORPORATION
Gaithersburg, Maryland

CONTENTS

ILLUSTRATIONS

## TABLES

# Section 1

# INTRODUCTION

## 1.1 PURPOSE OF STUDY

The objective of the Adaptive Compressive Telemetry Techniques study was the selection and development of an optimum adaptive data compression system for future manned spacecraft. Key interim objectives which supported this goal were:

a. Analysis and classification of existing and proposed compression techniques for selecting the most promising for exhaustive evaluation.

b. Evaluation of the selected techniques by testing their effectiveness in compressing a wide variety of actual spacecraft data signals.

c. Analysis of the queueing problem involved in adapting the randomly occurring compressed data samples to a fixed-rate transmission system, and the development of design procedures for solving this problem in practical compression systems.

d. Development of the complete system design for an adaptive data compression system incorporating the optimum compression methods and including adaptive control of the output queue.

## 1.2 SCOPE OF STUDY

The medical, experimental, and engineering control data used in this study spanned the type of data expected in space probes in the near future. Pulse analog signals (EKG and spacecraft attitude control-pitch data), undulatory signals (respiration data), step function signals (spacecraft roll-rate data), and noiselike signals (vibration data) were used to evaluate a broad spectrum of compression techniques.

Geometric aperture techniques (Zero-Order Prediction, First-Order Prediction, and First-Order Interpolation methods), a purely analytical technique: the Karhunen-Loéve compression method, and quasi-analytical techniques involving fixed and variable reduced sampling rates with such analytical methods as $\frac{\sin x}{x}$, straight-line, optimum linear, and Lagrange interpolation used to recover the reduced data, were simulated on a digital computer with the representative experimental data.

Autocorrelation functions and power spectral densities were computed for the experimental data to obtain insight into possible compression techniques as well as to estimate the amount of compression that could be obtained for each type of data.

The compression techniques simulated were evaluated on the basis of peak and rms error versus compression ratio as well as estimated implementation complexity. In addition to the calculated error performances, overlays of reconstructed and original data were made to permit visual comparison of different compression techniques with the various types of experimental data.

The problem of multiplexing multiple sensor channels into a time-shared compression unit capable of performing more than one compression algorithm was investigated. A system design for such a unit was performed using stored-logic concepts, and requirements were determined for an output buffer to efficiently combine the individual sensor channels, with their associated time gaps arising from data compression, without permitting overflow or underflow.

The problem of transmission errors in the compressed data was also investigated, and the effect of error on the two compression methods recommended for implementation was determined.

## 1.3 CONCLUSIONS OF STUDY

### 1.3.1 No Single Compression Technique "Best" for All Data Tested

Two relatively simple variable sampling rate aperture techniques—the Zero-Order Predictor (ZOP) and the First-Order Interpolator with Two Degrees of Freedom (FOI-2DF)—achieve, for all data tested, performances in terms of

compression ratio for a given rms error which cannot be improved upon significantly by any of the higher-order aperture techniques, fixed sampling rate techniques, or more exotic Transformation Compressors. This conclusion can be drawn even without considering implementation factors which are of paramount importance in the development of a practical, spaceborne data compression system. When implementation is considered, these two techniques have the advantage over the other techniques tested. However, neither technique is "best" for all data tested.

### 1.3.2 ACT System Should be Flexible to Reduce Probability of Obsolescence

The main advantage of an adaptive compressive telemetry system is its ability to increase the bandwidth utilization efficiency for incompletely specified data. This is achieved by reducing the data redundancy which necessarily results from conservatively choosing sample rate-bandwidth combinations for experimental data.

Because this study revealed that no single compression algorithm is "best" for all representative data tested, an ACT system should have the ability to perform two or more algorithms in the same processor. This reduces the probability of obsolescence if as yet unforeseen types of telemetry data are encountered which require different compression algorithms.

### 1.3.3 Flexibility Can be Obtained With Stored-Logic Concept

Flexibility can be realized by implementing sequence logic in a read-only store (memory) which replaces hard-wired decoding logic in identifying the commands to be performed. Thus, the ZOP and FOI-2DF compression methods can be implemented with the same stored-logic digital circuitry. This system approach provides flexibility in that multiplexed sensors with different types of data can be compressed with different algorithms at different aperture levels on a time-domain basis with the same equipment. In addition, other algorithms of the same order of complexity can be implemented by simple changes in the memory containing the logic instructions.

### 1.3.4 Adaptive Buffer Control Recommended

A large reduction in the size of the output buffer required for a compression system can be achieved by adaptive control of the compression aperture of nonpriority data. For example, instead of requiring at least a 100 sample buffer to achieve an overflow probability of 0.0001, the same performance can be achieved with a buffer size of 25 words by doubling the aperture of nonpriority data when overflow is imminent.

Section 2

# CLASSIFICATION OF COMPRESSION TECHNIQUES AND DESCRIPTION OF TECHNIQUES SIMULATED

## 2.1 INTRODUCTION

The first part of this section considers the problem of classification of the various compression techniques and algorithms into categories. The fidelity criteria for evaluation of the compression algorithms is also discussed. The second part of this section describes in detail the algorithms chosen for comparison on the data. The largest algorithms which have received the most attention by engineers in the field and for which there are many variations are the polynomial predictor and interpolators. Rather than test all of the variations of these algorithms, it was decided to select those which have been found to be effective in compression of data and which appeared to be easy to implement. Those chosen were:

a. Zero Order Predictor (floating, fixed, and offset aperture)
b. First-Order Predictor
c. First-Order Interpolator—Two Degrees of Freedom
d. First-Order Interpolator—Four Degrees of Freedom.

In addition to these techniques, the following techniques which do not appear to have received much attention were simulated on the data:

1. Optimum Linear Prediction
2. Fourier Filter
3. Optimum Discrete Filter (Karhunen Loéve)
4. Reduced sampling rate and reconstruction interpolation at the receiver
5. Variable Sampling Rate.

## 2.2 CLASSIFICATION AND EVALUATION OF COMPRESSION TECHNIQUES

The classification and evaluation of compression techniques is not a trivial problem. Because both the classification problem and the evaluation problem are of paramount importance, they will be discussed in some detail. Classification is always an aid to understanding the problem. Unfortunately, the classification of compression techniques does not have a unique solution. Terms such as entropy reducing, information preserving, redundancy reduction, adaptive sampling, encoding, signal reduction and others have been used to classify compression techniques.

To describe the effect the compression technique has on the form of the information or signal being transmitted, it was found that the compression techniques could be divided into four categories.

a. Direct Data Compression
b. Transformation Compression
c. Parameter Extraction Compression
d. Selective Monitoring Compression.

Figure 2-1 shows a chart of data compression techniques by category. The four categories are defined and the best methods of evaluating the compression technique in that group are discussed.

### 2.2.1 Direct Data Compression

A direct data compressor is one which operates on the data in such a way that the outputs of the data compressor are the actual sample values of the input waveform or the actual sample values within a tolerance. Most previous work in data compression falls into this category. A further useful subdivision of Direct Data Compression is made into variable rate and fixed rate. Variable-rate compressors have received the most attention and have been extensively discussed in the literature. A good example of this type of compressor is the Zero Order Predictor Floating Aperture technique originally developed by Jet Propulsion Laboratory and refined by Lockheed Corporation. Other examples

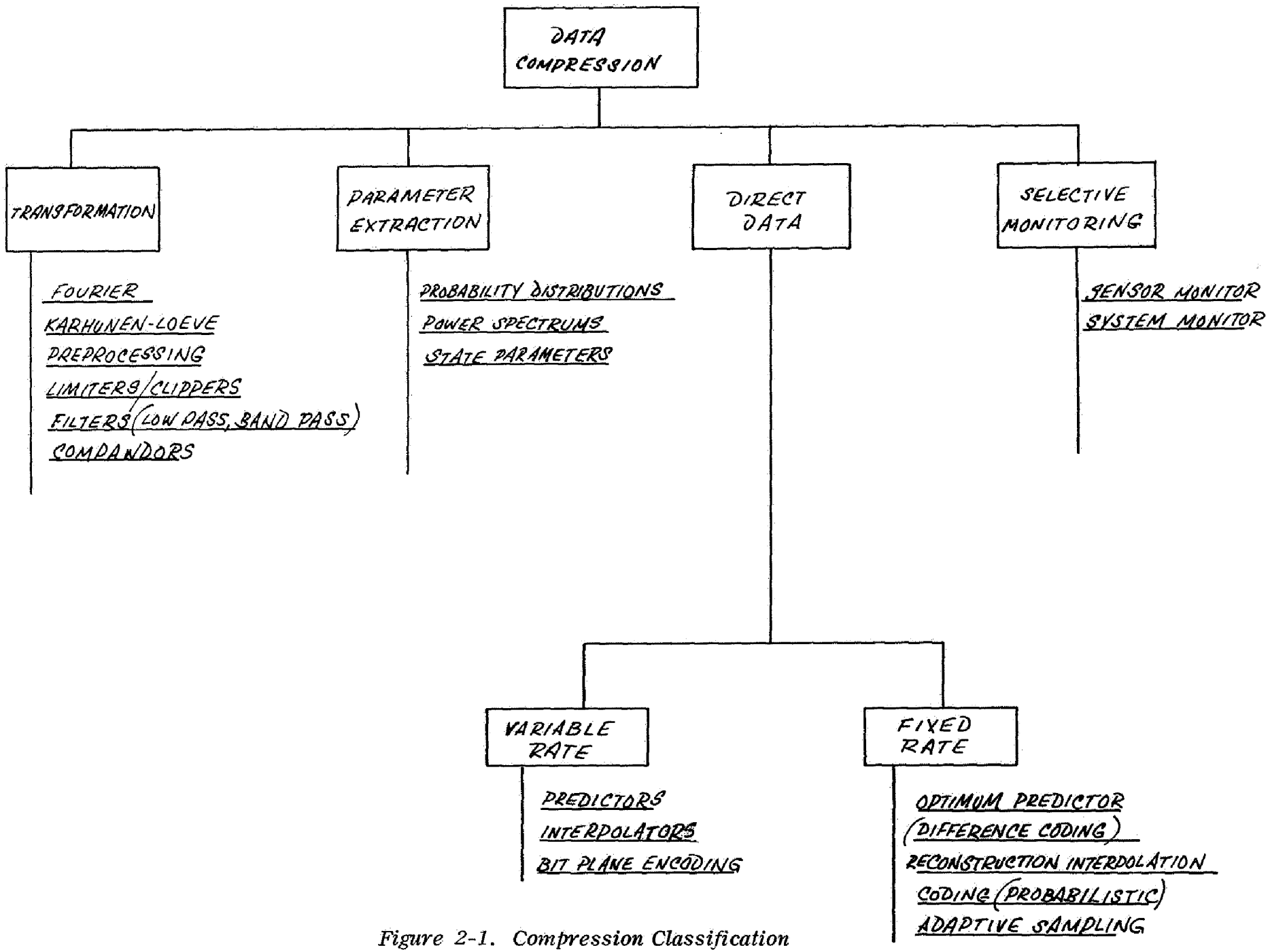


*Figure 2-1. Compression Classification*

of variable-rate compressors include interpolators, polynomial predictors, and bit-plane encoding.

Variable-rate compressors have great potential, however, they require tagging of the data with a time or sensor tag, and in some cases, may result in more data bits being transmitted than was in the original data. Fixed-rate compressors are characterized by the fact that the data is sent at a fixed rate. Examples of fixed-rate compressors are encoding and optimum linear prediction with difference coding. One method of fixed-rate compression which has received little, if any, attention (perhaps because most engineers do not consider this as a compression technique) is to simply sample the data at a rate close to the Nyquist rate and use reconstruction interpolation on the receiving end. Currently, most sensors are sampled for more than the theoretical minimum of twice the highest frequency component (assuming the signal is band limited). As shown from the power spectra of the data given in Section 3, the sampling rate is at least twice the highest significant frequency and more in most cases.

### 2.2.2 Transformation Compressors

A transformation compression is defined as any compression technique which transforms either analog or digital data by nonlinear or linear transformation. The output of the compressor must then go through an inverse transformation to obtain the actual analog waveform on the sampled digital data. Examples of transformation compressors are such preprocessing filters (signal conditioners) as compandors, logrithmic amplifiers, filters (low pass, band pass, high pass), limiters/clippers and compandors. Because the use of preprocessing filters depends on the nature of the signal and the user's requirements, the use of preprocessing is usually tailored to each sensor, therefore, no attention was given in this study to preprocessing filtering. Other types of transformation compressors which did receive attention during this study were Fourier Filtering and Optimum Discrete Filter compression (Karhunen-Loéve compressors) which are described in Section 2.3.4.

### 2.2.3 Parameter Extraction Compressors

Parameter extraction compressors are those which extract a particular characteristic or parameter of the signal. These parameters are then transmitted over the data link. Unlike the direct data and the transformation compressors, the parameter extraction compressor is not an information preserving operation and the original signal cannot be reconstructed from the extracted parameters. An example of this type of compression technique is the measurement of the probability distribution of the signal and the transmission of the quantiles of the distribution. Another example is the extraction of the power spectrum of the signal and the transmission of the amplitude of the spectral components. Still another example would be event recognition. Rather than send all of the relevant signal data to the ground to be monitored for a significant event or critical situation, an on-board event detector would detect the event and send extracted information such as time of event or amplitude. Like the use of preprocessing compressors, a parameter extraction compressor must be designed for the special parameter to be extracted and for the particular sensor characteristics. For this reason, the use of parameter extraction received little attention during this study.

### 2.2.4 Selective Monitoring Compression

Selective monitoring techniques may be defined as processes which monitor either the sensor or the state of the system or subsystem to select the data for transmission. Selective monitoring systems, that use the sensor outputs in establishing the priorities, are more complex than those that have a fixed priority rating. However, these systems would be capable of determining which information was important and then transmit this information. For example, if a temperature sensor indicated overheating in a particular element, this temperature and all other sensors, which might yield helpful information in diagnosing the difficulty, would be considered as highest priority sensors for a period of time. It would be possible to combine the selective monitoring in this case with parameter extraction and transmit the extracted information rather than the sensor signal itself.

An example of a selective monitoring system which monitors the state of the subsystem to control the data for transmission can be found in the case when a variable rate output of a direct data compression exceeds the rate of transmission for a period of time. If all sensors are treated as high priority then the probability exists of a buffer overflow. The probability of buffer overflow will be dependent on the length of the buffer and the ratio of the average rate at which bits are removed from the buffer to the average bit rate into the buffer. (The buffer problem is discussed in detail in Section 4.3.) The possibility of buffer overflow can be prevented by monitoring the buffer occupancy and taking action in one of two ways. One way is to consider certain sensors as low priority and ignore these sensors whenever the buffer occupancy exceeds a certain level. The other way is to decrease the quantization levels per sample on low priority sensors by increasing aperture size for the compression algorithm being used.

### 2.2.5 Evaluation of Compression Techniques

Previously, compression techniques have been evaluated by a measure known as compression ratio. Compression ratio is the ratio of the samples/sec into the compressor to the average samples/sec out of the compressor, or the ratio of the bits/sec into the compressor to the bit/sec out of the compressor. Unfortunately, compression ratio by itself is a meaningless number. If the input data has artificially introduced redundancy from using a much higher sampling rate than is required, then large compression ratios may be obtained. To effectively compare compression techniques, they must be applied to exactly the same data, and compression ratios for a given fidelity criteria of the reconstructed data should be compared. Ideally, the best measure of fidelity is the data user. Unfortunately, the data user was not available on a day-to-day basis during the study. Also, the fidelity criteria of a user is subjective and two users of the same data may have different criteria. It was necessary therefore to compare the compression ratios for a secondary fidelity measure. The fidelity measures used were the peak error and root mean square (RMS) error of the reconstructed data. The simulation programs written for the IBM 7090 actually provided a plot of the histogram of the error as well as the peak and RMS error.

In addition to comparing the effectiveness of compression techniques in terms of compression ratio for a given fidelity, criteria considerations must also be given to the equipment penalties, system penalties, computation time requirements, and effect of transmission noise on reconstructed data. All of these factors were taken into consideration in the development of the recommended system as discussed in Section 4.4.

## 2.3 DESCRIPTION OF COMPRESSION TECHNIQUES SIMULATED

To choose those techniques that would be used for evaluation of the available data, a review of known techniques as well as possible new techniques was first made. Because such techniques as parameter extraction, and such transformation techniques as preprocessing must be designed for a specific application or tailored to a specific sensor or user requirements, these techniques were not pursued. Selective monitoring techniques must also be designed for a given mission in mind. The compression techniques that will have the most general application to compression of data fall into the class of Direct Compression and Transformation Compression. Extensive amounts of literature exist on data compression techniques and bibliographies are also available[1], therefore, a bibliography will not be repeated here.

In choosing the compression technique, preliminary consideration was given to anticipated compression ratio, equipment complexity and system complexity. The methods chosen for simulation with the data and descriptions of the technique are given in this section.

### 2.3.1 Polynomial Predictors

The predicting equations for polynomial predictors are based on a finite difference technique which permits an n'th order polynomial to be passed through n + 1 data points. The polynomial is extrapolated one unit at a time, which produces a predicted data point. A polynomial of the type

$$y(t) = a_0 + a_1 t + a_2 t^2 + \ldots + a_n t^n \qquad (2\text{-}1)$$

may be fitted to the data points by means of a difference equation

$$\hat{y}_t = y_{t-1} + \Delta y_{t-1} + \Delta^2 y_{t-1} + \ldots + \Delta^n y_{t-1} \qquad (2\text{-}2)$$

where:

$\hat{y}_t$ = predicted value at time t

$y_{t-1}$ = data sample value at one sample period prior to t

$\Delta^{n+1} y_t = \Delta^n y_t - \Delta^n y_{t-1}$

$\Delta y_t = y_t - y_{t-1}$ .

Here, the n + 1 previous values, $y_{t-1}$, $y_{t-2}$, $y_{t-3} \cdots y_{t-(n+1)}$ are known and $y_t$ is to be predicted. The various implementations of this approach will be discussed.

2.3.1.1 Zero-Order Predictor Fixed Aperture

The simplest polynomial predictor described above is the zero-order predictor where n = $\phi$. In this case

$$\hat{y}_t = y_{t-1} \qquad (2\text{-}3)$$

and the predicted value is merely the previous data point. A set of fixed tolerance bands are then set up with a width K each. This is done by truncating the last few bits from a binary data word. If two data words have the same truncated value, they belong to the same tolerance band. A prediction is made by Equation (2-3) that the new data point falls in the same tolerance band as the previous point. If the new data point falls outside the tolerance band of the previous point, then this new point is transmitted and the process is repeated.

#### 2.3.1.2 Zero-Order Predictor Floating Aperture

In the floating aperture algorithm of the zero-order predictor, an aperture of 2K is placed about the last transmitted data point. If each new data point lies within the aperture placed about the last transmitted data point, then the new data points are not transmitted. If a new data point falls outside the aperture placed about the last transmitted data point, then the present data point is transmitted and the process is repeated. The predicted point in this case is the last transmitted data point with a tolerance of $\pm$ K placed about it. The aperture, then, has the effect of "floating" with the last transmitted value.

#### 2.3.1.3 Zero-Order Offset Predictor

The zero-order offset method is a modification of the zero-order, floating aperture technique. This approach takes advantage of data trends by offsetting the predicted point by a predetermined amount. The sign of the offset is determined by the last out of tolerance value. If the last transmitted point was out of tolerance in a positive sense, the offset has a positive sign and vice versa. Thus, the predicted value is the last transmitted value plus an offset. A floating aperture of width 2K ($\pm$ K) is placed about the predicted value. If the new data point falls inside the aperture then that point is not transmitted. If a new data point falls outside the aperture, then that data point is transmitted and the process is repeated.

#### 2.3.1.4 First-Order Predictor

The first-order predictor utilizes Equation (2-2) to obtain a first-order extrapolation polynomial of the form:

$$\hat{y}_t = 2y_{t-1} - y_{t-2} \qquad (2\text{-}4)$$

The extrapolation equation is a straight line drawn between the last two data points. Initially, the first two data points are transmitted and a straight line is drawn through them. An aperture of width 2K is placed about the straight line.

The predicted value of the new data points is the point on the straight line. If the new data point is within $\pm$ K of the predicted value, then that point is not transmitted. If the new data point is outside the aperture, then that point is transmitted and a new straight line for prediction is drawn through the present data point, which was transmitted, and the previous predicted data point.

### 2.3.2 Optimum Linear Predictor

The optimum linear predictor predicts the next sample point by using linear combinations of past samples as given by

$$\hat{y}_t = \sum_{k=1}^{H} a_k \, y_{t-k} \qquad (2\text{-}5)$$

where:

$\hat{y}_t$ = predicted value of present data sample value

$y_{t-k}$ = data sample value at kth period prior to present sample at t

$y_t$ = present data sample value.

The optimum linear predictor uses a set of coefficients which minimizes the mean square error between the predicted and the actual value. Thus, the following expression is minimized:

$$E\{y_t - \hat{y}_t\}^2 \qquad (2\text{-}6)$$

where:

$E\{\quad\}$ denotes expected value.

The optimum set of coefficients are found by solving a set of N linear equations involving the autocorrelation matrix as given by:

$$\sum_{\sigma=1}^{H} a_k R_y\{(\tau-\sigma)T\} = R_y\{(\tau+\sigma)T\} \quad \tau = 1, 2, \ldots, N \tag{2-7}$$

where:

$R_y\{(\tau-\sigma)T\}$ = autocorrelation function of the signal for a lag of $(\tau-\sigma)T$

s = number of sample periods since last sample point, for which the prediction is to be made.

Two methods of estimating the autocorrelation function for use in Equation (2-7) were used and are called the direct and indirect method. The direct method uses the autocorrelation function given by

$$R_y\{(i-j)T\} = \frac{1}{N}\sum_{n=1}^{N} y\{(n-i)T\}\ y\{(n-j)T\} \tag{2-8}$$

where:

N = number of points used.

The indirect method uses an autocorrelation function given by

$$R_y\{(i-j)T\} = \frac{1}{N-(i-j)}\sum_{n=i+j+1}^{N} y(nT)\ y\{n-(i-j)\}T\ . \tag{2-9}$$

Two methods of applying the optimum linear predictor for compression were simulated on the data. In the first method, the predicted value of the present data point was found using N previous actual sample data points. The difference between the predicted value and the actual value was then coded and transmitted. Thus, at each new data point the difference between the actual and predicted data value was transmitted. If the receiver uses the same set of coefficients in the predictor

equation, then the actual sample data value can be obtained at the receiver. In this algorithm, there is no error between the actual sample values on-board the spacecraft and the reconstructed value obtained on the ground.

In the second method, an aperture was placed about the predicted value. In starting off the process, the first N sample values are transmitted. The predicted value of the N + 1 sample is obtained using the past N actual sample values. An aperture is then placed about the present predicted value (N + 1 value). If the actual sample data value lies within the aperture placed about the predicted value, then that point is not transmitted. The present predicted value is then used in the prediction equation to obtain a predicted value for the next sample value. If the next actual sample value is within the aperture, then that point is not transmitted and the predicted value is used in the prediction equation. When an actual sample value falls outside an aperture, then that point is transmitted and the actual sample value is used in the prediction equation for predicting the next data point.

The prediction Equation (2-5) is the optimum linear nonrecursive filter (only past values of input are used to obtain the output). The optimum linear recursive filter can be obtained from solving the Wiener Hopf equations for discrete data and is given in Appendix E. Because the results indicated that the effectiveness of the optimum nonrecursive filter did not increase as the number of coefficients were increased above five, the use of a nonrecursive filter would not increase the effectiveness and was not simulated.

Note that to obtain the optimum predictor, the power spectrum of the data must be known a priori. If, as may be the case, the exact nature of the data from a new experiment is not known then a power spectrum must be assumed, which will result in a suboptimum predictor.

### 2.3.3 Interpolation Compressors

Interpolators differ from predictors because all sample values between the last transmitted value and the present value will effect the interpolation. Two types of first-order interpolators were simulated on the data. These were the First-Order Interpolator—Two Degrees of Freedom and the First-Order Interpolator—Four Degrees of Freedom, which are discussed in the following sections.

#### 2.3.3.1 First-Order Interpolator—Two Degrees of Freedom

The first-order interpolator—two degrees of freedom draws a straight line between the present sample and the last transmitted sample so that all intermediate data points are within a tolerance of the interpolated value on the straight line. In this algorithm, the first point is transmitted. A line is drawn between the transmitted point and the second sampled data value after the transmitted point. If the first point after the transmitted value is within a tolerance K of the interpolated value, then a straight line is drawn between the transmitted point and the third point after the transmitted point. The interpolated value of the first and second points are now checked to see if they are within a tolerance of the actual values. If at the Kth sample value after the last transmitted sample value, a line is drawn and the actual value differs from the interpolated value by a quantity greater than the tolerance, then the (K - 1) sample is transmitted and the process is repeated. A method of implementing the FOI-2DF which does not require the storage of all the actual data points between the last transmitted point and the present point has been described by L. W. Gardenhire.[2]

#### 2.3.3.2 First-Order Interpolator—Four Degrees of Freedom

The first-order interpolator—four degrees of freedom draws a line between the sample points such that the most positive error and the most negative error between the sample value and the interpolated value are equal and within the prescribed tolerance. The computed value of both ends of the line are transmitted. The next straight line is started from the next sample value after the last transmitted point.

### 2.3.4 Transformation Compressors

Two types of transformation compression techniques applied to the data were the Fourier Filter technique and the Karhunen-Loéve technique and are discussed below.

### 2.3.4.1 Fourier Filter

The Fourier Filter compression technique is one in which the Fourier transform of the sampled data is obtained and complex coefficient of the transform are obtained at discrete frequencies. The coefficients of the lower frequency components are then transmitted and the data is reconstructed by finding the Fourier transform of the received coefficients. Let the sampled data be represented by

$$x^*(t) = \sum_{n=1}^{N} x(nT)\ \delta(t-nT) \tag{2-10}$$

where:

$T$ = sample interval

$\delta(t)$ = Dirac delta function.

Taking the Fourier transform of Equation (2-10) we obtain

$$X^*(f) = \sum_{n=1}^{N} x(nT)\, e^{-j2\pi fnT} . \tag{2-11}$$

The complex value of the Fourier transform is then obtained at frequencies given by:

$$f = \frac{k}{NT} , \qquad k = 0, 1, 2, \ldots, N/2. \tag{2-12}$$

Thus, the Fourier transform at these frequencies is given by

$$X^*\left(\frac{k}{NT}\right) = \sum_{n=1}^{N} x(nT)\, e^{-j(2\pi kn/N)} \tag{2-13}$$

$$k = 0, 1, 2, \ldots, N/2$$

Equation (2-13) may be written

$$X^*\left(\frac{k}{NT}\right) = A_k - j\,B_k \tag{2-14}$$

where:

$$A_k = \sum_{n=1}^{N} x(nT)\,\cos(2\pi kn/N) \tag{2-15}$$

$$B_k = \sum_{n=1}^{N} x(nT)\,\sin(2\pi kn/N) \tag{2-16}$$

Note that the Fourier transform need only be found for positive frequencies because the Fourier transform at negative frequencies is given by:

$$X^*\left(\frac{-k}{NT}\right) = A_k + j\,B_k\,. \tag{2-17}$$

In the Fourier Filter compression, a large number of sample points are taken and the Fourier transform is evaluated. If the original signal is oversampled, then only the lower frequency complex values need be transmitted. At the ground station, the inverse process is performed. Assume that only M complex values of the Fourier transform are transmitted, then the received Fourier transform may be written as

$$X^*(f) = \sum_{k=-M}^{M} A_k\,\delta\left(f - \frac{k}{NT}\right) + j \sum_{k=-M}^{M} B_k\,\delta\left(f - \frac{k}{NT}\right). \tag{2-18}$$

Taking the Fourier transform of Equation (2-18) the reconstructed signal can be written as

$$x^*(t) = \sum_{n=1}^{N} C_n \, \delta(t-nT) \tag{2-19}$$

where:

$$C_n = A_0 + \frac{1}{2} \sum_{k=1}^{M} A_k \cos\left(\frac{2\pi}{N} kn\right) + \frac{1}{2} \sum_{k=1}^{M} B_k \sin\left(\frac{2\pi}{N} kn\right). \tag{2-20}$$

From inspection of Equation (2-20), note that for N data points and M transmitted values of the Fourier transform, that the onboard compressor is required to make 2MN multiplications and additions. For a single channel which is sampled at a slow speed, this method of compression may be useful, however, when a multiplexed system is considered this method is not practical.

Equation (2-14) may be used to calculate the power spectrum of the data by what is called the direct method. The power spectrum is given by

$$P(k/NT) = A_k^2 + B_k^2 \, . \tag{2-21}$$

2.3.4.2 Optimum Discrete Filter Compression (Karhunen-Loéve)

The optimum discrete filter compressor (Karhunen-Loéve compressor) is a process similar to the Fourier technique described in Section 2.3.4.1. Whereas the Fourier filter uses sines and cosines as the orthogonal function in the expansion, the optimum discrete filter uses an optimum set of orthonormal basis functions. The orthonormal set is optimum because the least number of orthonormal functions are needed for a given RMS error.

Let the sequence of sample data points be given by

$$x(T), \; X(2T), \; \ldots, \; X(NT) \, . \tag{2-22}$$

We seek a set of functions $\phi$ such that

$$\hat{x}(kT) = \sum_{i=1}^{M} a_i \phi(kT) \tag{2-23}$$

where:

$\hat{x}(kT)$ = the reconstructed value of the data point

$a_i$ = coefficients to be transmitted to the ground.

The $\phi_i$ are eigenvectors of the autocorrelation matrix of the x's.[3] The M eigenvectors chosen to represent the data are those with the largest eigenvalue. The coefficients to be transmitted are obtained by taking the inner product of the data points with the eigenvector. Thus, the coefficient is found from

$$a_i = \sum_{j=1}^{M} x(jT)\, \phi_i(jT). \tag{2-24}$$

Thus, each coefficient requires N multiplications and additions. If M eigenvectors are used, then MN multiplications and additions are required. If prior knowledge of the signal statistics is not available, then the autocorrelation matrix must first be obtained and the matrix diagonalized to obtain the eigenvalues and eigenvectors. The number of eigenvectors used to obtain a given RMS must then be determined. If the M eigenvectors with the largest eigenvalues are used to represent the signal, then the mean square error is given by

$$\epsilon^2 = \sum_{k=M+1}^{N} \lambda_k \tag{2-25}$$

where:

$\lambda_k$ = eigenvalue of autocorrelation matrix.

From the above discussion, the implementation of this method is impractical, however, it was used to obtain a comparison of its compression effectiveness against other techniques.

### 2.3.5 Reconstruction Interpolation

The principle of reconstruction interpolation is to sample the signal at a reduced sampling rate and then reconstruct the signal by interpolation filtering at the receiver. Because most of the signals studied were over-sampled by as much as 20 times the Nyquist rate, it should be possible to use a lower sampling rate onboard and reconstruct the signal on the ground. The reconstruction interpolation technique was evaluated by taking every kth sample (where k is an integer) of the original sampled data and assuming that these samples were transmitted. The reconstruction interpolation was applied to these data points to obtain interpolated values of the nontransmitted data points. The original and reconstructed values were then compared to obtain the errors. Five reconstruction interpolations simulated were straight line, optimum, $\frac{\sin x}{x}$, Lagrange, and Fourier. These methods are described in the following sections. In all cases the interpolation formula can be represented in the same general form.

Let the input signal be sampled values of the signal at intervals of T. Interpolated values are to be obtained at intervals T/L where L is an integer. The interpolated values in an interval of T for which an equal number of input samples are used on each side of the interval is given by

$$g\left(\frac{nT}{L}\right) = \sum_{m=-k+1}^{k} x(mT)\, h_m\left(\frac{nT}{L}\right) \qquad (2\text{-}26)$$

where:

$x(mT)$ = sample value of input data at time mT

$h_m\left(\frac{nT}{L}\right)$ = weighting function of interpolation filter for mth sample at time nT/L

$g\left(\frac{nT}{L}\right)$ = interpolation value at time nT/L

$n$ = 1, 2, . . . , L-1

$k$ = number of input samples used on each side of interpolation interval.

2.3.5.1 Straight Line Reconstruction

The simplest type of reconstruction interpolation is straight-line interpolation. A straight line is drawn between the two input sample values on each side of the interval and interpolated values are obtained along the straight line. The interpolation formula is given by

$$g\left(\frac{nT}{L}\right) = x_0 + \frac{n}{L}\left[x(T) - x_0\right] \qquad L = 1, 2, \ldots, L-1 \tag{2-27}$$

2.3.5.2 Optimum Interpolation Filter

The optimum interpolator seeks an optimum linear combination of the sample values such that the error in interpolation is minimized in the mean square sense. Thus, we seek a linear combination of the sample values of the form:

$$g(\tau) = \sum_{i=N+1}^{N} k_i\, x(iT) \tag{2-28}$$

which will minimize the mean square error. The error is given by

$$\epsilon = g(\tau) - x(\tau) . \tag{2-29}$$

The set of coefficients $k_i$ which minimize the mean square error is given by the following set of linear algebraic equations:

$$\sum_{j=-N+1}^{N} k_j R_x\{(i-j)T\} = R_x(iT - \tau) \tag{2-30}$$

$$\text{for } i = -N + 1, -N, \ldots, 0, 1, 2, \ldots, N$$

where:

$R_x\{(i-j)T\}$ = autocorrelation function of the signal for lag of $(i-j)T$.

Note that in obtaining Equation (2-30), it was assumed that the signal was a wide sense stationary stochastic process. In general, the signals will not fall into this classification. The direct and indirect methods can be used to obtain the estimate of the autocorrelation function. The direct method uses an autocorrelation function given by

$$R_x\{(i-j)T\} = \frac{1}{N} \sum_{n=1}^{N} x\{(n-i)T\}\, x\{(n-j)T\} \tag{2-31}$$

where:

N = total number of data points used.

The indirect method uses an autocorrelation given by:

$$R_x\{(i-j)T\} = \frac{1}{N-(i-j)} \sum_{n=i+j+1}^{N} x(nT)\, x\{n-(i-j)\}T \ . \qquad (2\text{-}32)$$

2.3.5.3 Sin x/x Reconstruction

The $\frac{\sin x}{x}$ reconstruction interpolation used approximates the ideal low-pass filter weighting function by a truncated $\frac{\sin x}{x}$ weighting function. The interpolation formula in this case is given by:

$$g\left(\frac{nT}{L}\right) = \sum_{m=-k+1}^{k} x(mT)\left[\frac{\sin\, mT - \frac{n}{L}T}{mT - \frac{n}{L}T}\right] \qquad (2\text{-}33)$$

$$n = 1, 2, \ldots, L-1 \ .$$

2.3.5.4 Lagrange Reconstruction Interpolation

The standard Lagrange interpolation formulas [4] were used to obtain the interpolated value of the signal in an interval. An equal number of received data points were used on each side of the interpolation interval to obtain the interpolated value. Thus, the Lagrange interpolation with 2N coefficients indicates that five data points on each side of the interval were used.

2.3.5.5 Fourier Reconstruction Interpolation

The Fourier filter reconstruction uses the same mathematical techniques as described in Section 2.3.4.1 on Fourier filter compression. The Fourier transform of the 2N data points is obtained and then the inverse Fourier transform is found. The inverse Fourier transform is then evaluated in the middle interval of the 2N data points to obtain interpolated values.

2.3.6 Variable Sampling Rate

The variable sampling rate technique combines the floating aperture zero order predictor technique with a variable sampling rate technique to compress

the data. Tolerance bands are selected and a sampling rate is associated with each tolerance band. Each sampling rate is a multiple of two of the slowest rates. A particular implementation of this technique which was simulated utilized two sampling rates. The high rate was 160 samples per second and the low rate was 40 samples per second. The sampling is switched from the low rate to the high rate when the first, second, third, or fourth high rate sample exceeds a tolerance band of $\pm$ 4 units from the last low rate sample transmitted. The sampling proceeds at the high rate until four successive samples occur which do not exceed a tolerance band of $\pm$ 1 units. When this occurs, the sampling is switched to the lower rate.

Interpolation techniques as described in the previous section may be used for reconstruction interpolation.

## 2.4 REFERENCES


1. W. R. Bechtold, J. E. Medlin, D. R. Weber, "Final Report PCM Telemetry Data Compression Study, Phase 1" (15 September 1964 to 15 August 1965), Contract NAS 5-9729, Lockheed Missiles and Space Co., Sunnyvale, California for Goddard Space Flight Center, Greenbelt, Maryland.

2. L. W. Gardenhire, "Redundancy Reduction, The Key to Adaptive Telemetry," National Telemetering Conference, Los Angeles, California, June 1964.

3. C. J. Palermo, R. V. Palermo, H. Howwitz, "The Use of Data Omission for Redundancy Removal," Record of the 1965 International Space Electronics Symposium, pp.11-D1 to 11-D16.

4. F. B. Hildebrand, "Introduction to Numerical Analysis," McGraw-Hill Book Co., Inc., 1956.

Section 3

# EVALUATION OF COMPRESSION TECHNIQUES BY SIMULATION

## 3.1 INTRODUCTION

Of the four general classes of compression techniques presented in Section 2, two classes (Direct Data Compressors and Transformation of Variable Compressors) were investigated by digital computer simulation of representative methods.

The remaining classes (Selective Monitoring Compressors and Parameter Extraction Compressors) require detailed knowledge of the information to be extracted from the data and are not amenable to general evaluation. No methods in these two classes were simulated.

The class of Direct Data Compressors represents the area in which the greatest effort has been expended to date by other investigators. In general, the methods in this class are the most economical to implement at the present state-of-the-art, and yield results comparable to the more "exotic" transformation of variable compressors.

The class of Direct Data Compressors can be divided into fixed sampling rate and variable sampling rate methods. The variable sampling rate methods appear to be the most efficient in terms of compression versus distortion, as will be evident from the computer simulation results.

All the methods investigated are compared on the basis of simulated compression ratios versus rms and peak error performance for particular types of data.

## 3.2 TEST DATA

Five different types of data were used in the simulation studies. The data was provided in 8-bit digital form and is considered representative of the medical,

experimental, and engineering-control data encountered in space program telemetry. The five types of data are identified by source and code name below; the sample sizes used in the simulations are also listed:

a. FCEKG1—Flight Commander EKG1, Orbit 2, 1 min. into tape.
b. ENGRES—Engr. Respiration Orbit 2, 1 min. into tapes.
c. FA01—Attitude Control-Pitch, Orbit 2, 1 min. into tape.
d. ASMRV3—Apollo (BP-15) SM Radial Vibration-3
e. EA02—Roll Rate, Orbit 2, 1 min. into tape.

Autocorrelation functions $R(\tau)$ for the five types of data are shown in Figures 3-1, 3-3, 3-5, 3-7, and 3-9. Power spectral density for each type of data was obtained by taking the Fourier transform of the autocorrelation functions described in Section 2. The power spectral densities are shown in Figures 3-2, 3-4, 3-6, 3-8, and 3-10.

The autocorrelation functions were used, in addition to obtaining the power spectral density, to obtain the rms signal power, since $R(\tau = 0)$ is the rms signal power.

The power spectral densities were useful in estimating the amount of compression that could be obtained by reducing the sampling rate for each type of data. For example, the Engineer Respiration data was sampled at 80 samples/sec and Figure 3-4 shows that most of the signal power lies below 2 cps. Therefore, one would expect to obtain compression ratios in the order of 40. However, most compression methods simulated provided compression ratios from 10-40 with 5 percent rms distortions for this data. In actual practice, the data is usually not specified sufficiently to calculate a reasonably accurate power spectral density. If it were, the sampling rate could be specified such that redundancy removal would not be necessary. However, as research tools, the autocorrelation function and power spectral density were invaluable.

## 3.3 COMPRESSION METHODS SIMULATED

The compression methods simulated are described briefly to explain the captions in Figure 11-32.

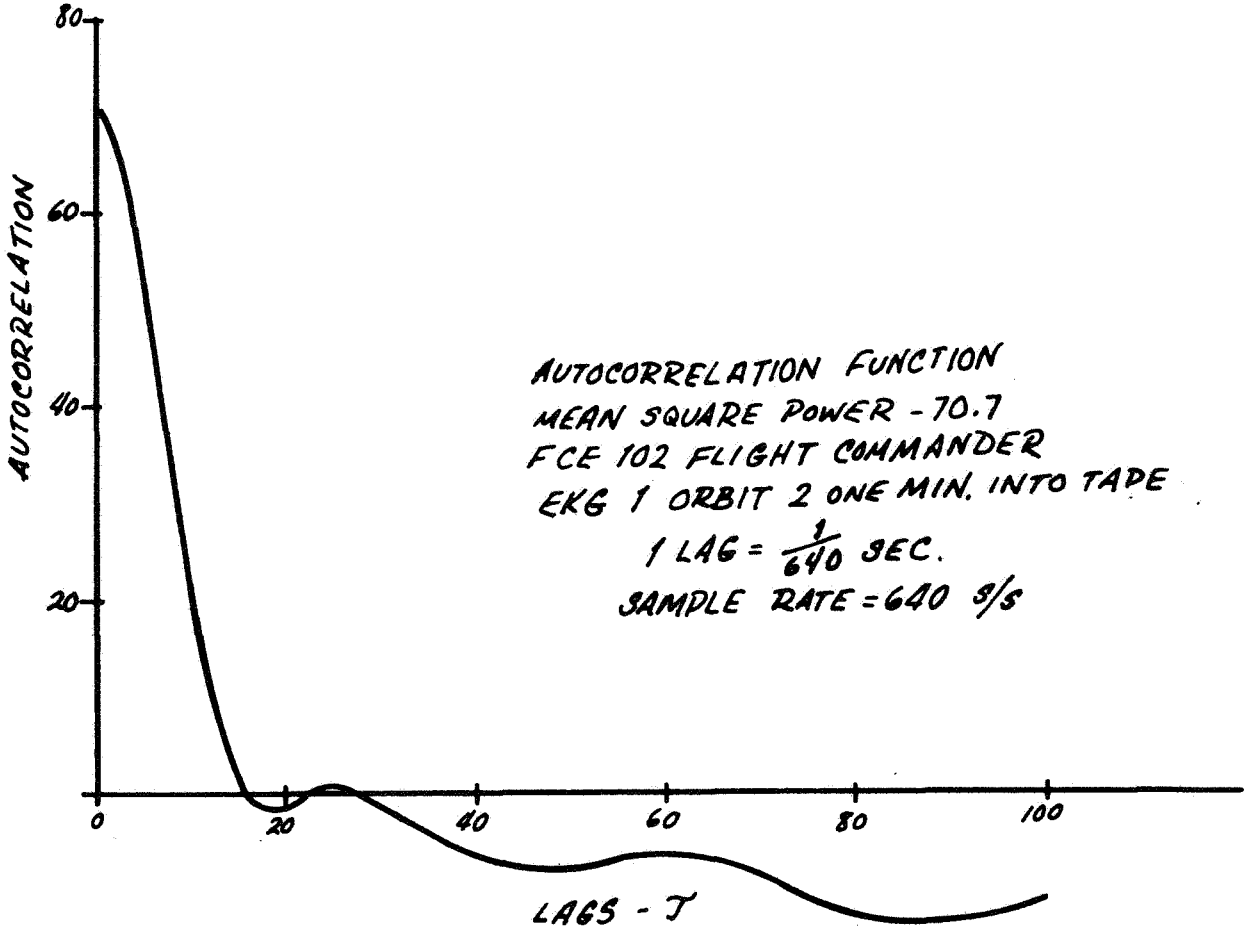


*Figure 3-1. Autocorrelation Function, FCE 102, Flight Commander*

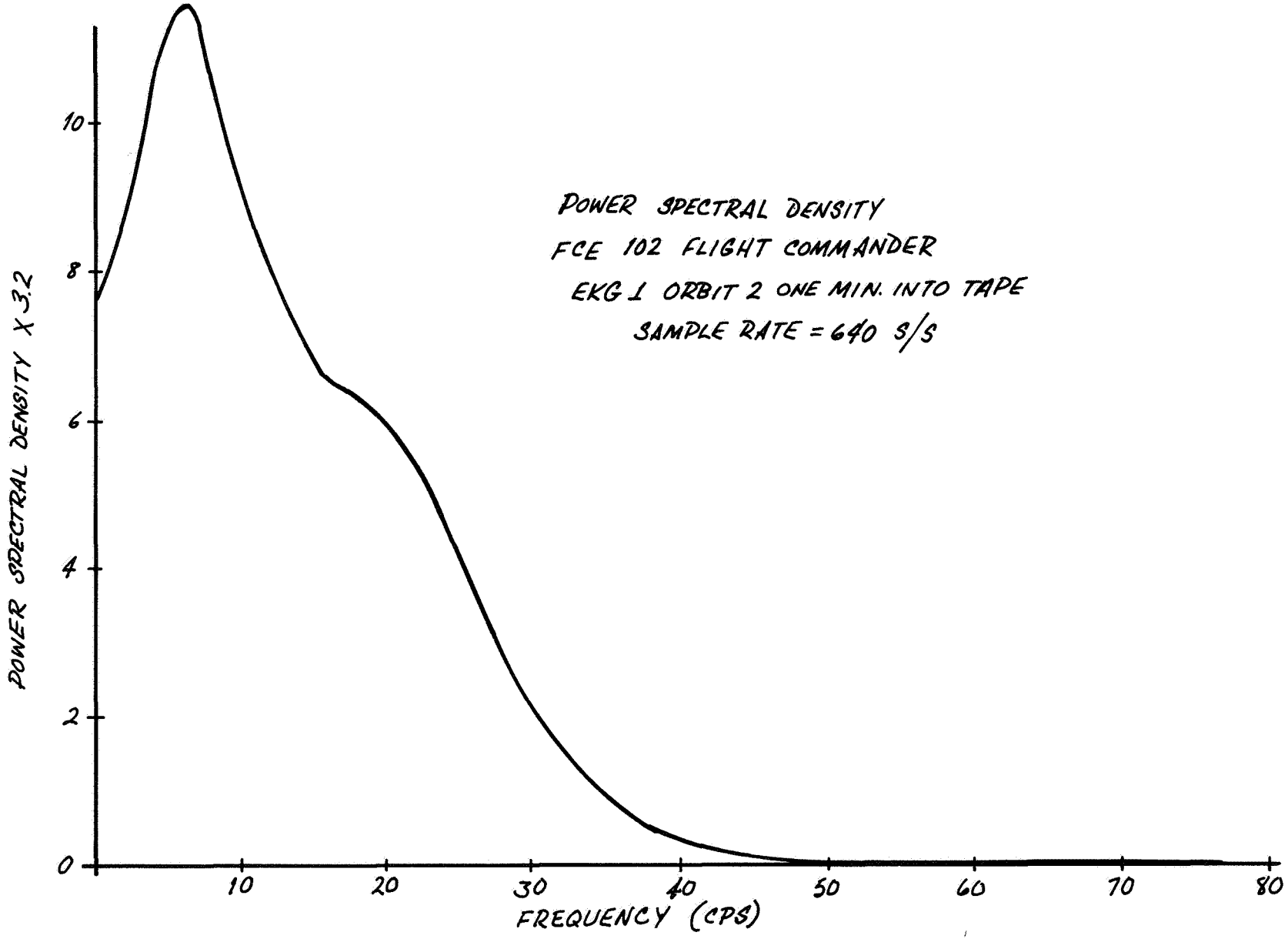


*Figure 3-2. Power Spectral Density, FCE 102, Flight Commander*

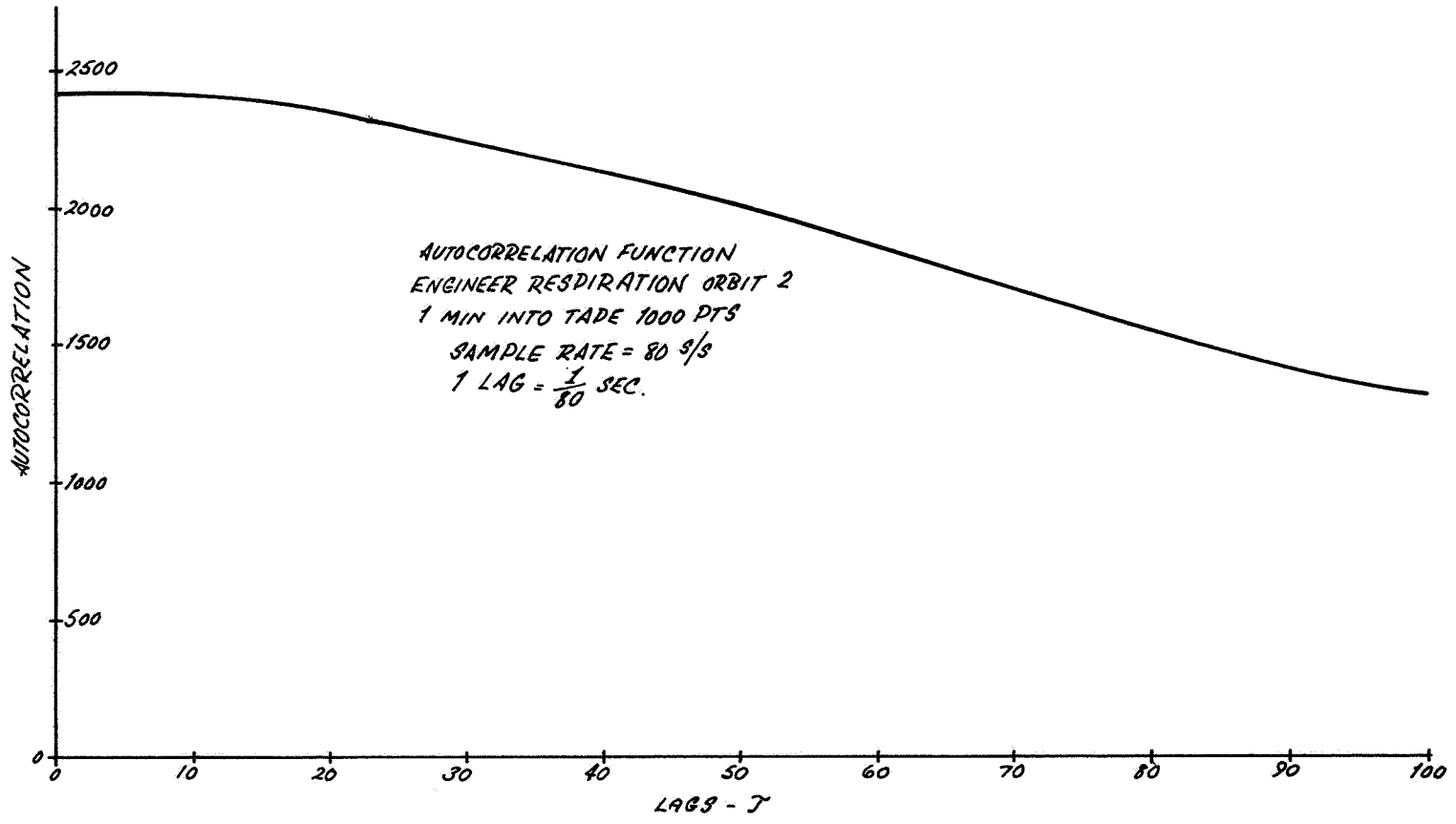


*Figure 3-3. Autocorrelation Function, Engineer Respiration*

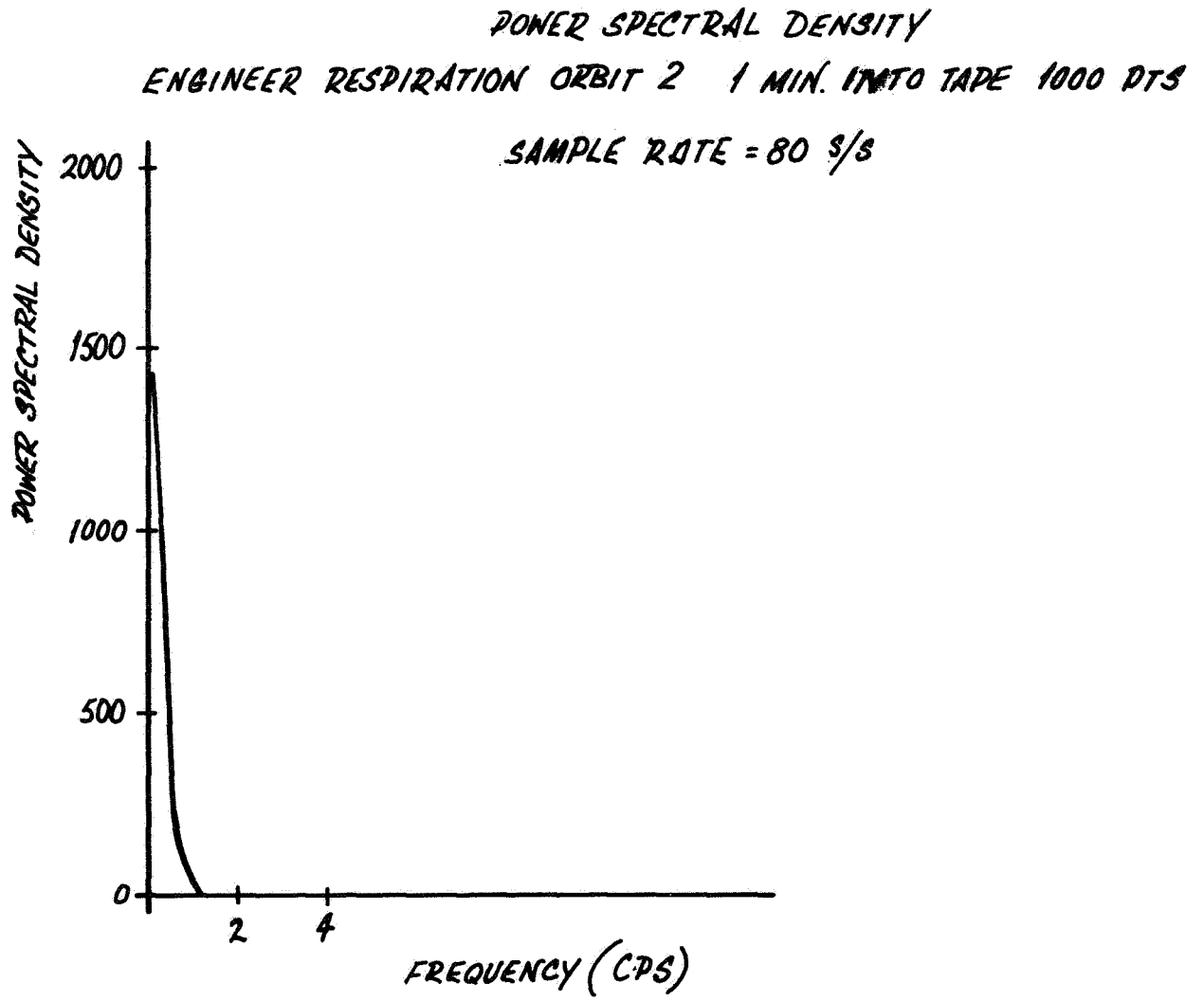


*Figure 3-4. Power Spectral Density, Engineer Respiration*

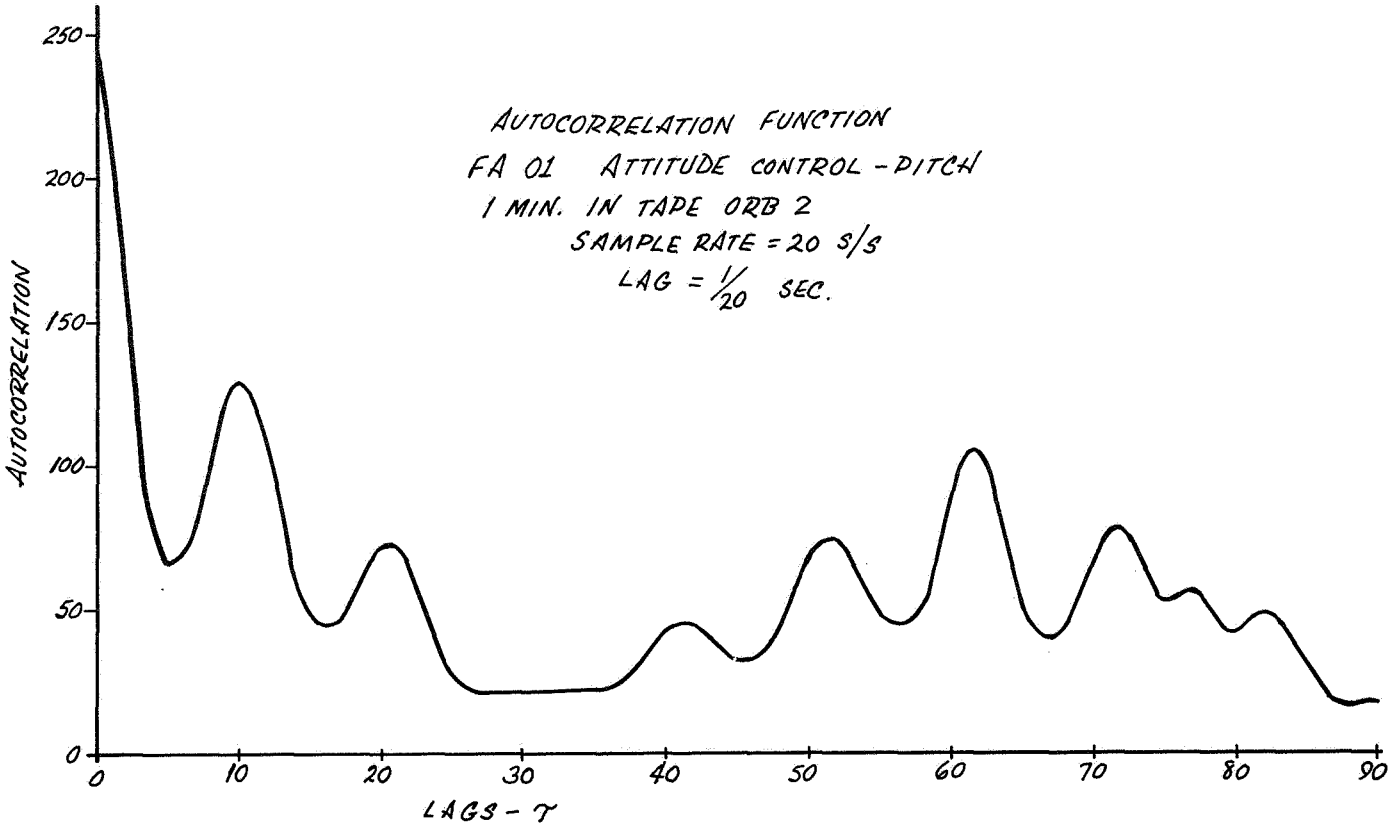


*Figure 3-5. Autocorrelation Function, FA 01, Attitude Control-Pitch*

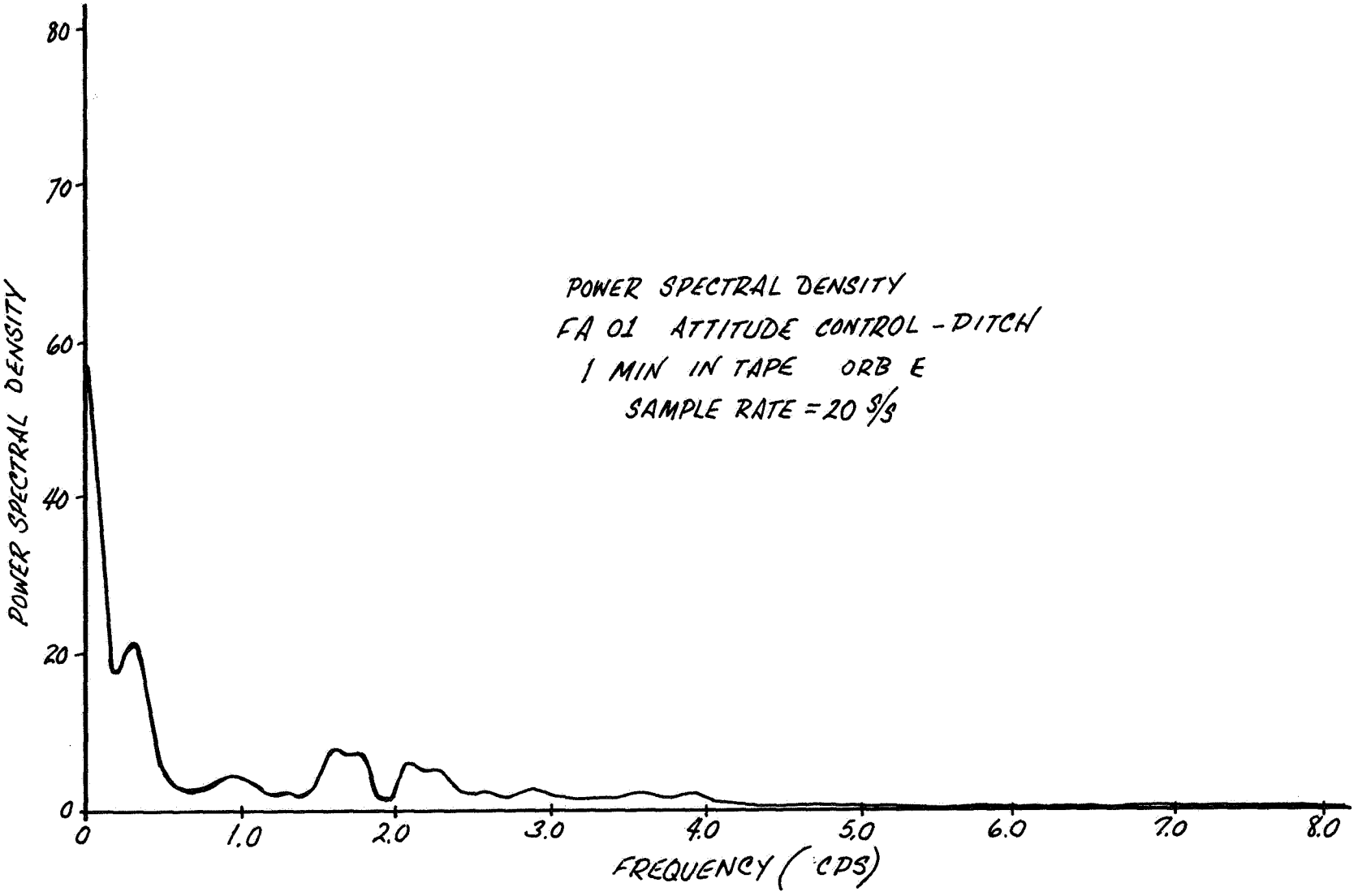


*Figure 3-6. Power Spectral Density, FA 01, Attitude Control-Pitch*

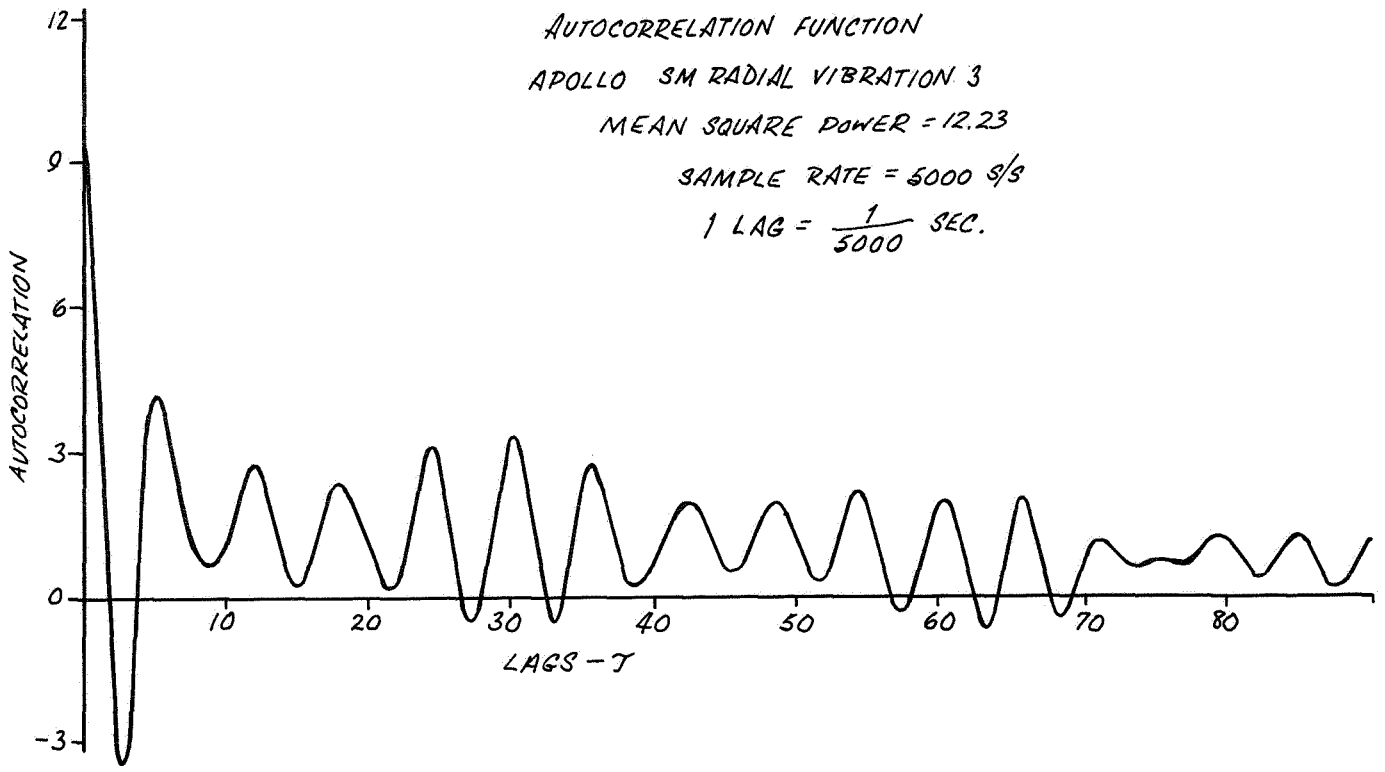


*Figure 3-7. Autocorrelation Function, Apollo SM Radial Vibration 3*

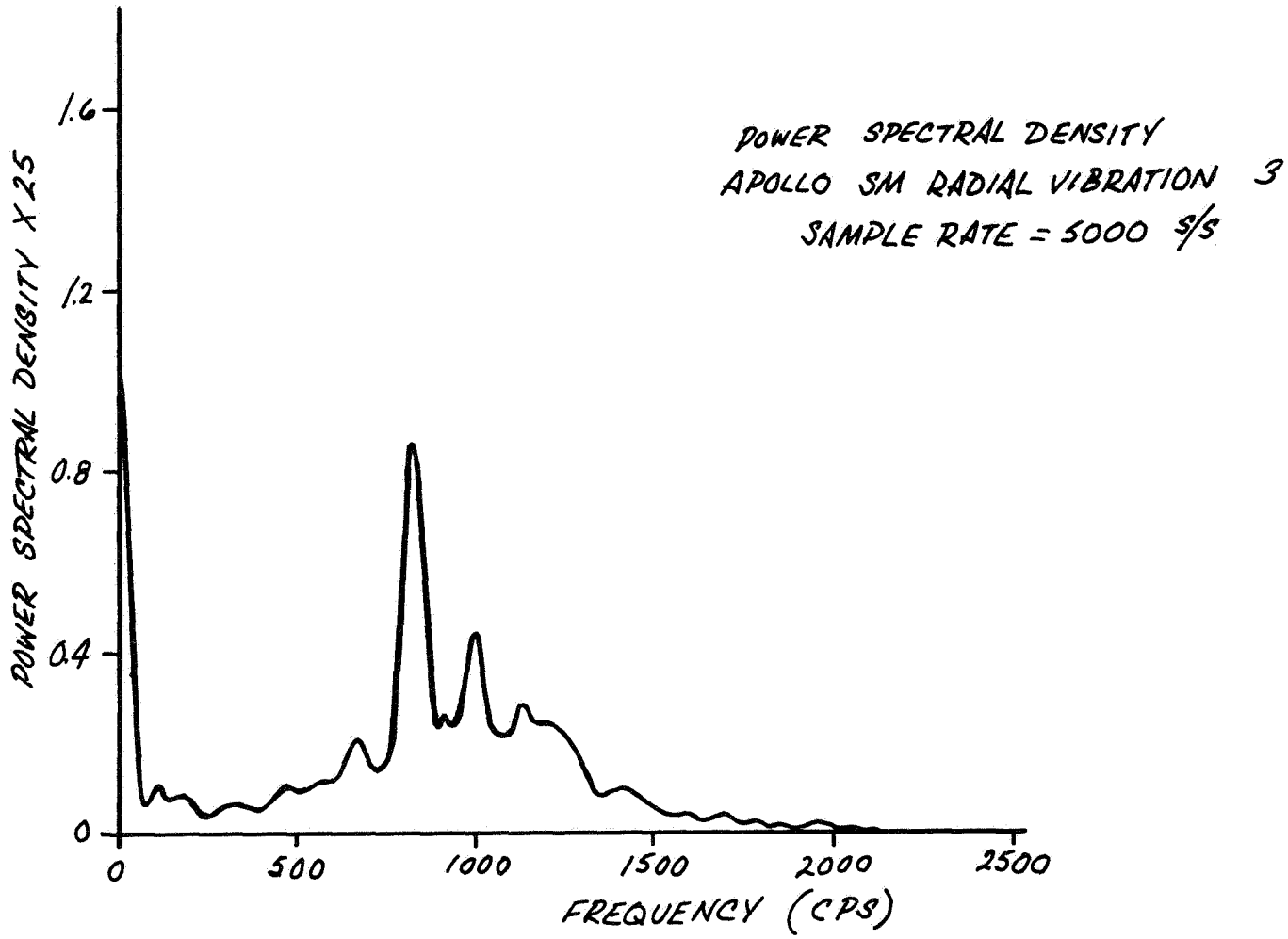


*Figure 3-8. Power Spectral Density, Apollo SM Radial Vibration 3*

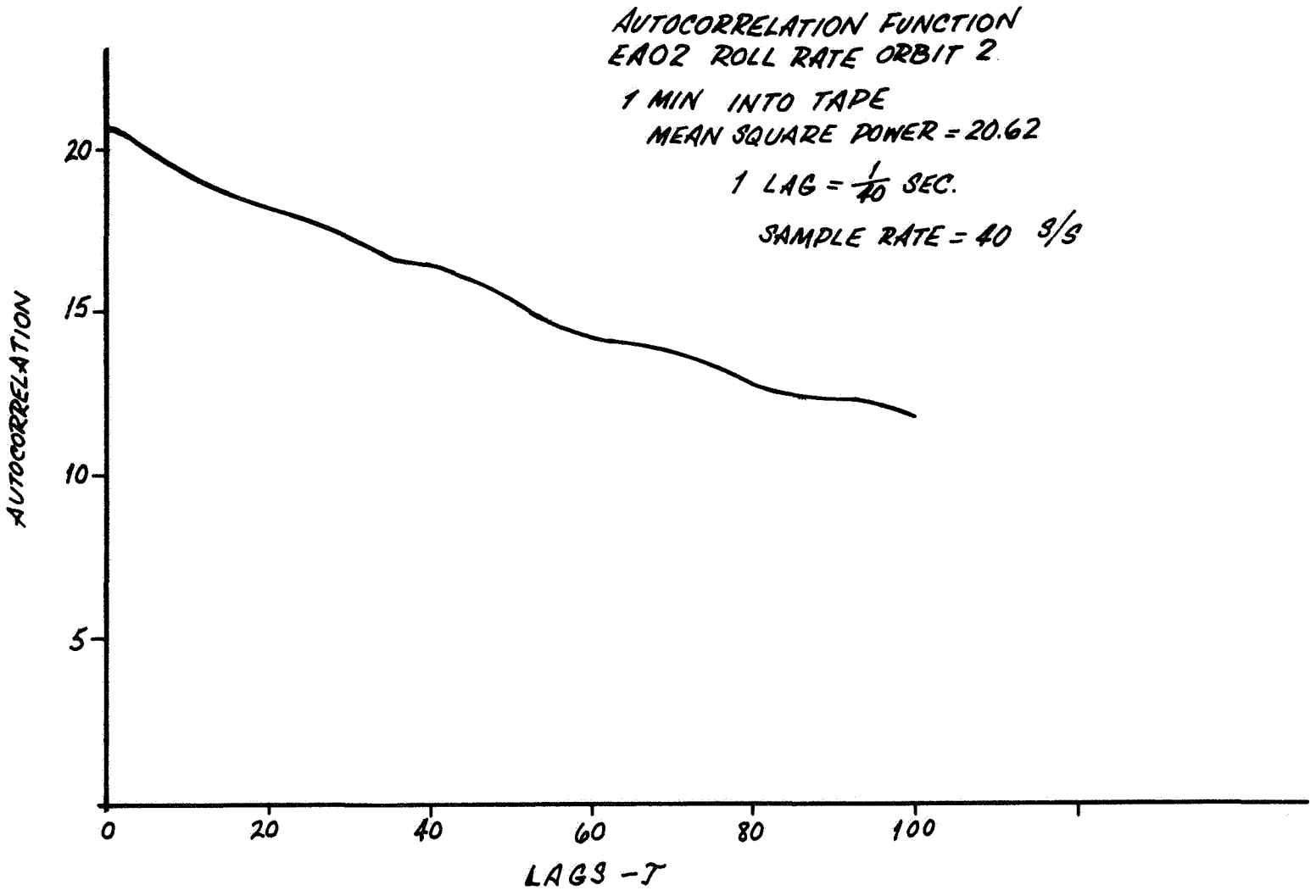


*Figure 3-9. Autocorrelation Function, EA 02, Roll Rate*

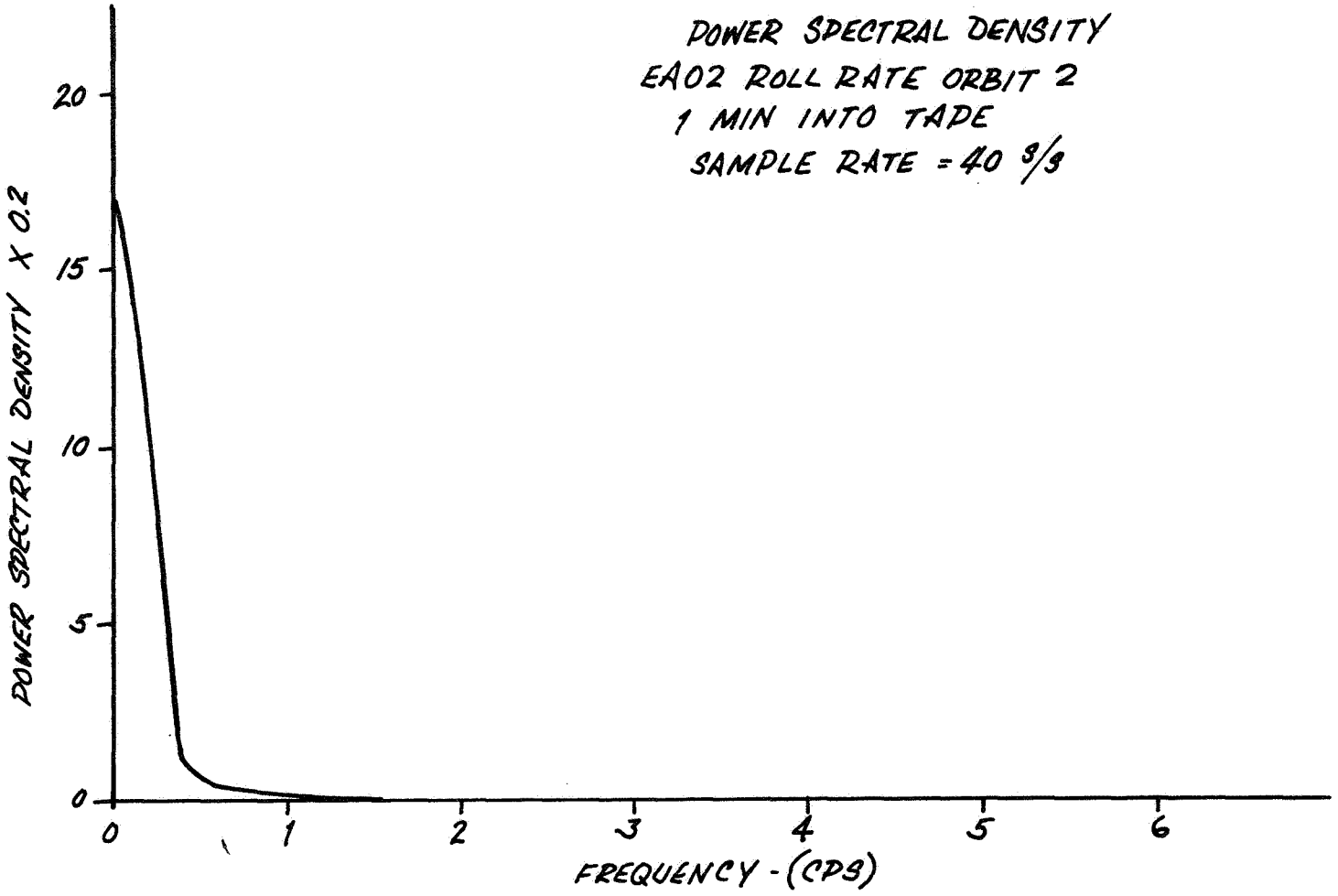


*Figure 3-10. Power Spectral Density, EA 02, Roll Rate*

a. ZOP—The zero order predictor is a variable sampling rate aperture technique which was simulated with fixed aperture, floating aperture, and offset floating aperture. 3000 sample points were used in each simulation.

b. FOI-2DF—A first-order interpolator with two degrees of freedom in which the first and last points of the interpolation line are nonredundant sample points. 3000 sample points were used.

c. FOI-4DF—A first-order interpolator with four degrees of freedom in which the first and last points of the interpolation line are constrained to an aperture about the true data values.

d. LP-i—The optimum linear aperture predictor with i coefficients was simulated with i = 3 and 5. 1000 sample points were used from the first minute of test data.

e. FO—Fixed sampling rate at 1/10 original rate. This method uses optimum linear interpolation with 8 coefficients to reconstruct the samples not transmitted. 3000 sample points were used.

f. FL—This method also uses a fixed sampling rate at 1/10 original rate; however, straight-line interpolation is used for reconstruction. 3000 sample points were used.

g. FS—Fixed sampling rate at 1/10 original rate with $\frac{\sin x}{x}$ interpolation. 3000 sample points were used.

h. FLG—Fixed sampling rate method with the sampling rate at $1/\phi$ original rate using the ten coefficient Lagrange interpolation formula for reconstruction; $\phi$ is the compression ratio. 3000 sample points were used.

i. FF1—A method in which the Fourier transform of the data is taken and only 300 Fourier coefficients are retained. The data is reconstructed from the inverse Fourier transform. 3000 sample points were used.

j. FF2—Fixed sampling rate at 1/10 original rate with reconstruction by Fourier filter as described in Section 2. 3000 sample points were used.

k. VO—Variable sampling rate method with sampling at 1/4 or 1/16 original rate. Tolerances (apertures) of 4 and 1, respectively, determine the sample rate switching. Optimum linear interpolation with four coefficients is used for reconstruction. 3000 sample points were used.

l. VS—Similar to the VO method except straight-line interpolation was used for reconstruction. 3000 sample points were used.

m. Karhunen-Loéve—A transformation of variable technique in which the signal is expressed in terms of a truncated Karhunen-Loéve expansion. Ten- and 50-point expansions were calculated.

n. Linear Prediction With Difference Coding—This is a method in which the next sample point is predicted by optimum linear prediction using previous sample points. The predicted sample is subtracted from the actual sample and the difference is coded and transmitted. As explained in Section 2, 2- and 10-bit coding, and 2-, 5-, and 13-bit coding schemes were simulated.

## 3.4 COMPRESSION OF EKG DATA

All the methods presented above were simulated with FCEKG1 data. The compression ratios obtained vs. rms error are presented in Figure 3-11. Because the data was obtained in 8-bit quantized form, all simulation results are presented in terms of quanta (255 quantum levels for an 8-bit code).

The rms signal power was determined to be 8.4 quanta; thus, the rms compression errors are also presented as percent of rms signal.

Compression ratio is presented as sample/samples sent, which does not allow for sensor and time tags which may be required for some methods. The variable sample rate techniques require sensor tags because a particular time slot in a multiplexed frame cannot be allocated to a particular sensor. Whether or not a sensor transmits a sample point during a frame depends upon whether or not a nonredundant sample is available at the time. The number of bits required per sensor or time tag is dependent upon the particular implementation; therefore, the

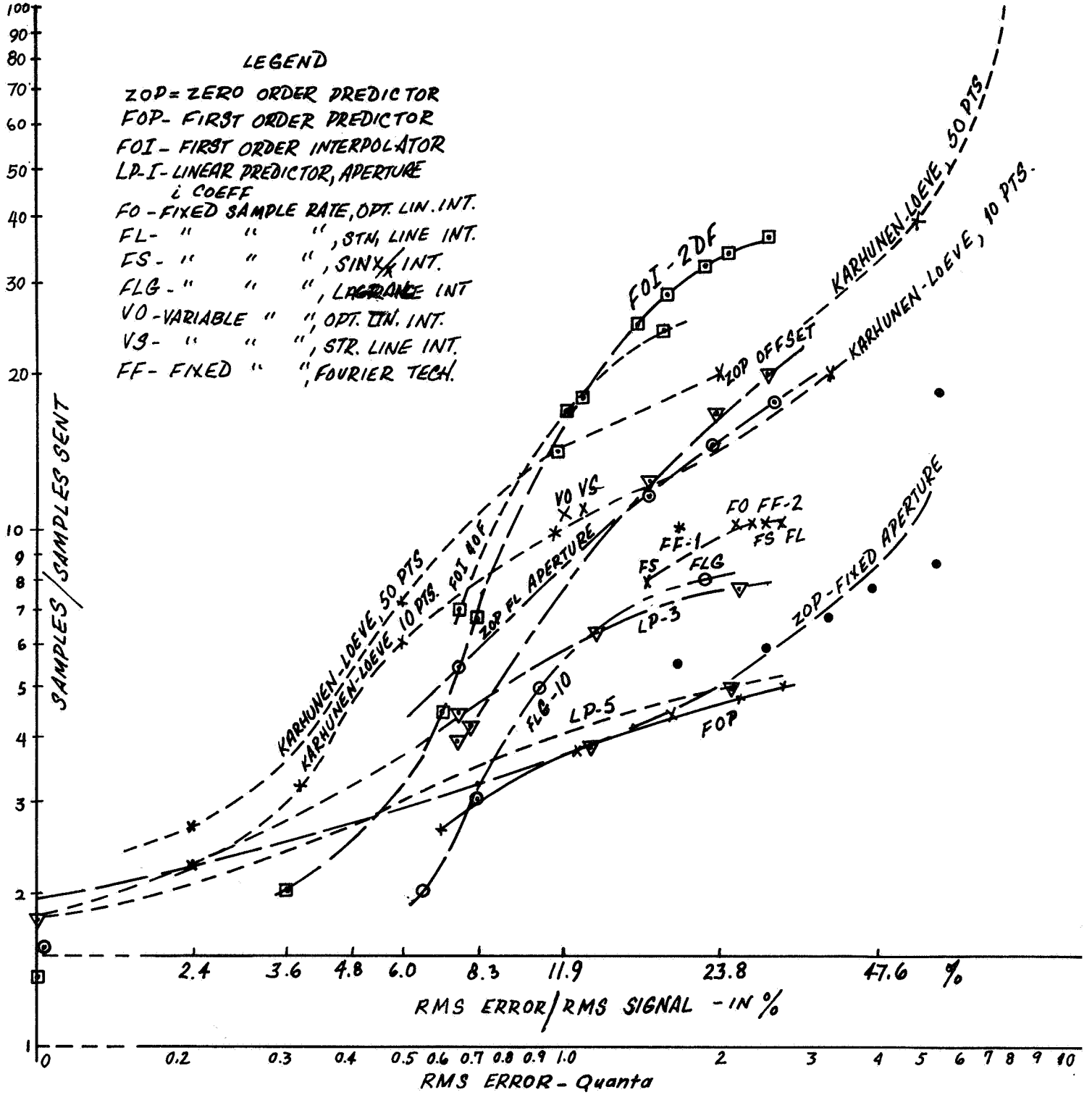


*Figure 3-11. Methods of Compression, RMS Error, Flight Commander, EKG 1*

compression ratios of all the methods simulated except one are plotted as sample/samples sent. The linear prediction with difference coding method achieves compression by sending all samples, but with a reduced number of bits on the average. Thus, for this method the compression ratio is represented as data/data sent.

The linear prediction with difference coding scheme was applied to the data: FCE 102 Flight Commander EKG-1, Orbit-2, 1 min. into tape, 1000 points, with 3, 5, and 10 coefficients using exact and assumed autocorrelation functions. The prediction error is plotted in the form of a histogram which is used for the calculations which follow.

Two coding schemes were postulated for the difference between the predicted points. These are shown in Figure 3-12 for the exact autocorrelation function simulation. The first scheme is called 2- and 10-bit coding and is shown in the Figure 3-12. Two bits give four possibilities. Three of these possibilities are used to code differences of -1, 0, and 1. The fourth is used to indicate that a full 8-bit word will follow, giving a sum of 10 bits.

The second scheme, called 2-, 5-, and 13-bit coding, permits coding of $\pm 2$, $\pm 3$, and $\pm 4$. These six possibilities require 3 bits plus the 2-bit indicator, giving a total of 5 bits. A 5-bit indicator is used to imply that the difference is greater than 4 and a full 8-bit word follows. This gives three types of code words of 2, 5, and 13 bits.

The compression with three prediction coefficients and the first difference coding scheme may be obtained from Figure 3-12. The total number of bits required with the regular 8-bit coding is $997 \times 8 = 7976$. The total number of bits required for the compression methods may be obtained by multiplying the number of occurrences of each difference times the number of bits in the code for that difference. The sum of these products is then the total number of required bits or 3090. The compression in terms of data-over-data sent is then $7976/3090 = 2.58$.

For the second coding scheme, the compression is 3.23. The improvement in coding efficiency is due to a better match between the probability of a difference and the number of bits used to describe it. The desired results would be that the products of the probability and the corresponding number of bits be equal.

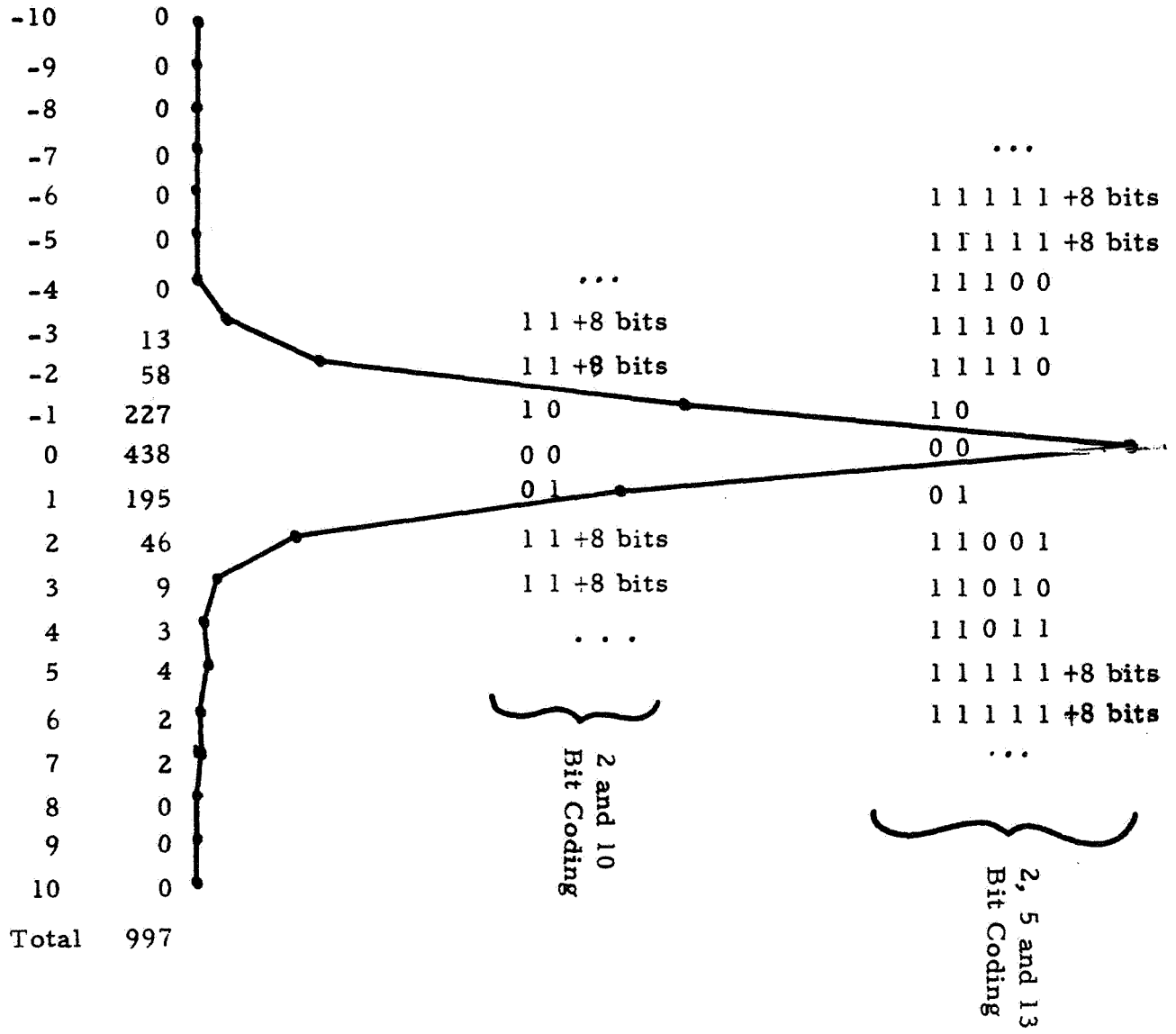


*Figure 3-12. Histogram of Linear Predictor With 3 Coefficients for Difference Coding, FCE 102, Flight Commander, EKG 1*

The 2- and 10-bit coding and the 2-, 5-, and 13-bit coding schemes were tried on the same data with 5-coefficient and 10-coefficient predictors. The former scheme gave compressions of 2.56 and 2.69, respectively. The latter scheme gave compressions of 3.16 and 3.34, respectively. These compressions are obtained with zero error.

From the results of the linear prediction algorithm applied to the FA 01 Attitude Control-Pitch it was found that the peak error was much larger than that obtained on EKG data. There are several reasons for this. First, the FA 01 data is not as oversampled as the EKG data. The FA 01 data was sampled at 20 samples per second and there is some power in the spectrum out to 5 cps whereas the EKG is sampled at 640 samples per second and the spectrum extends to about 40 cps. Secondly, the FA 01 data is characterized by long periods of quiescence with a periodic triangle shape pulse.

It was found that with the 2- and 10-bit coding for the case with 3 coefficients derived from the actual autocorrelation matrix, the compression ratio in terms of data-over-data sent is 2.34. The 2-, 5-, and 13-bit coding scheme results in a compression ratio in terms of data-over-data sent of 2.55. When using the assumed autocorrelation function and a 2-, and 10-bit coding, a compression ratio in terms of data-over-data sent of 0.915 is obtained. When a 2-, 5-, and 13-bit coding scheme is used a compression ratio of 1.63 is obtained. Because this method provided relatively small compression ratios and requires considerable computation time, it was not extended to the other data types. However, where relatively small compression ratios with low distortion (zero for the ideal case) is desired, this method is suitable.

The simulation results indicate that when sensor tagging is considered (the samples/samples sent values are reduced by a factor of approximately 2 for the aperture techniques), the Karhunen-Loéve method provided maximum compression. However, the advantage lessens rapidly as the rms distortion approaches 3 percent. The excessive computation required for the Karhunen-Loéve method limits its usefulness to theoretical considerations.

However, the LP-3, and FOI-2DF, ZOP methods are quite practical with the ZOP method being about the simplest of all the variable sample rate techniques.

The FOI-4DF method provides greater compression than the FOI-2DF method up to approximately 12 percent rms distortion. However, as discussed in Section 4, the computation complexity of the FOI-4DF is much greater than the 2DF method. The small improvement in compression does not warrant further consideration of the FOI-4DF method.

The FOI-2DF method is second only to the Karhunen-Loéve method above approximately 7 percent rms distortion (not counting the 4DF method). However, for practical analog data, the rms distortion is usually limited to 5 percent. Therefore, on the basis of rms distortion the ZOP-floating aperture method is the most efficient of the "practical" compression techniques and (ignoring sensor tags) can provide compression ratios of nearly 4 with 5 percent rms distortion. Because of this efficiency compared with other ZOP techniques, the ZOP-floating aperture method will be referred to as the ZOP method for simplicity purposes.

Figure 3-13 shows the relationship between rms error and peak error for the methods simulated with FCEKG1 data. Note the excessive peak errors obtained for the fixed sampling rate with reconstruction interpolation methods and the variable sample rate straight-line interpolator. The similarity of the relationship for the other methods (except the ZOP-fixed aperture method) is somewhat surprising and indicates that percent peak error increases more rapidly than percent rms error with increased compression ratios.

Figure 3-14 shows the relationship between compression ratio and peak error for the ZOP and the two FOI methods. The advantage of both FOI methods reduces rapidly as the peak error approaches aperture values which yield practical rms error values of less than 5 percent (peak error $\leq$ 1 quantum).

## 3.5 COMPRESSION OF RESPIRATION DATA

Figure 3-15 presents simulation results with the ENGRES data for several of the more promising methods as determined by the FCEKG1 data simulation results. The fixed sampling rate and variable sample rate with interpolation methods were not applied because of the poor peak error performance with the FCEKG1 data.

Figure 3-16 shows the relationship between rms and peak errors for the methods simulated with the ENGRES data. The results are very similar to those

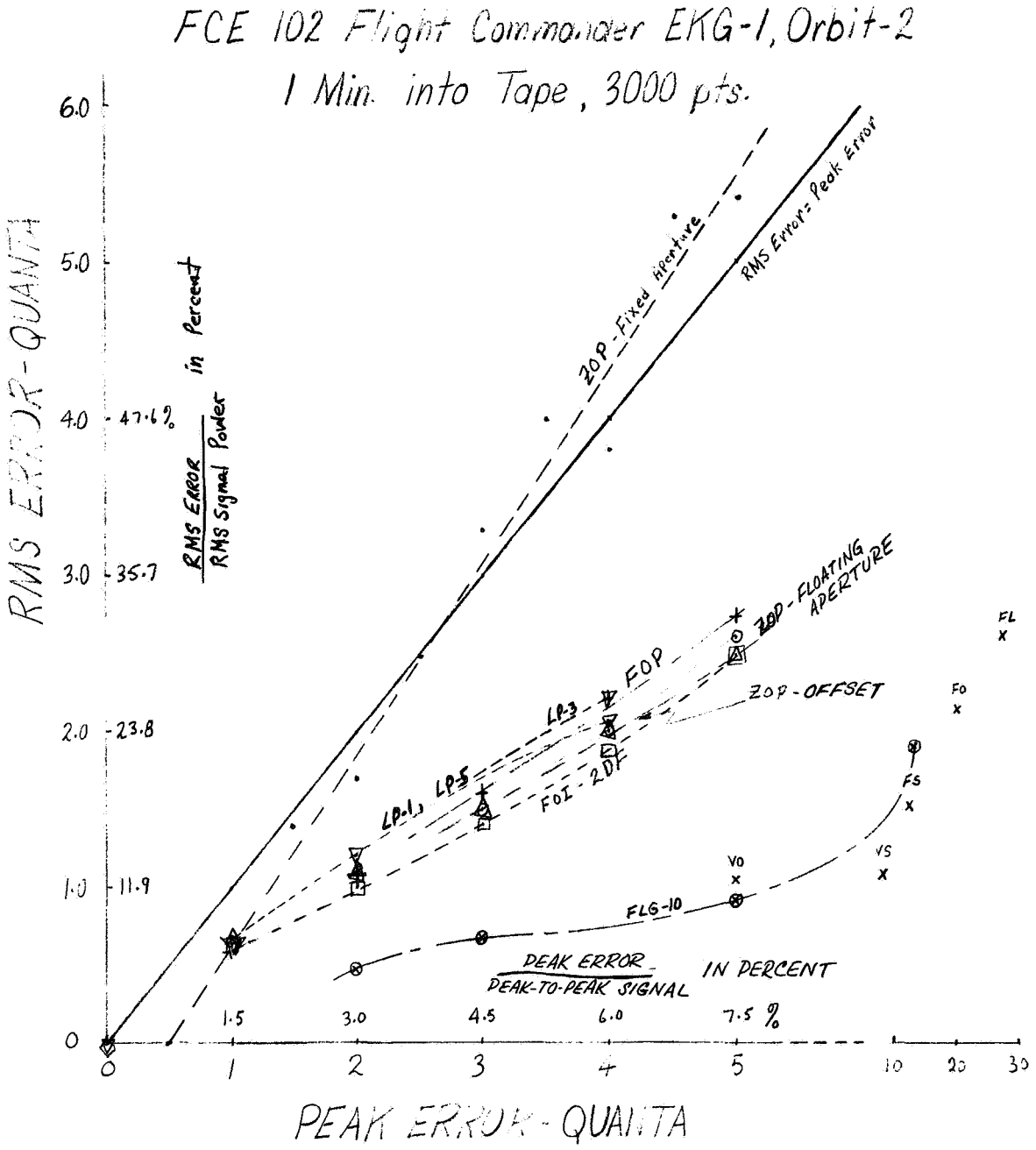


*Figure 3-13. RMS Error vs Peak Error for Aperture Techniques, FCE 102, Flight Commander, EKG 1*

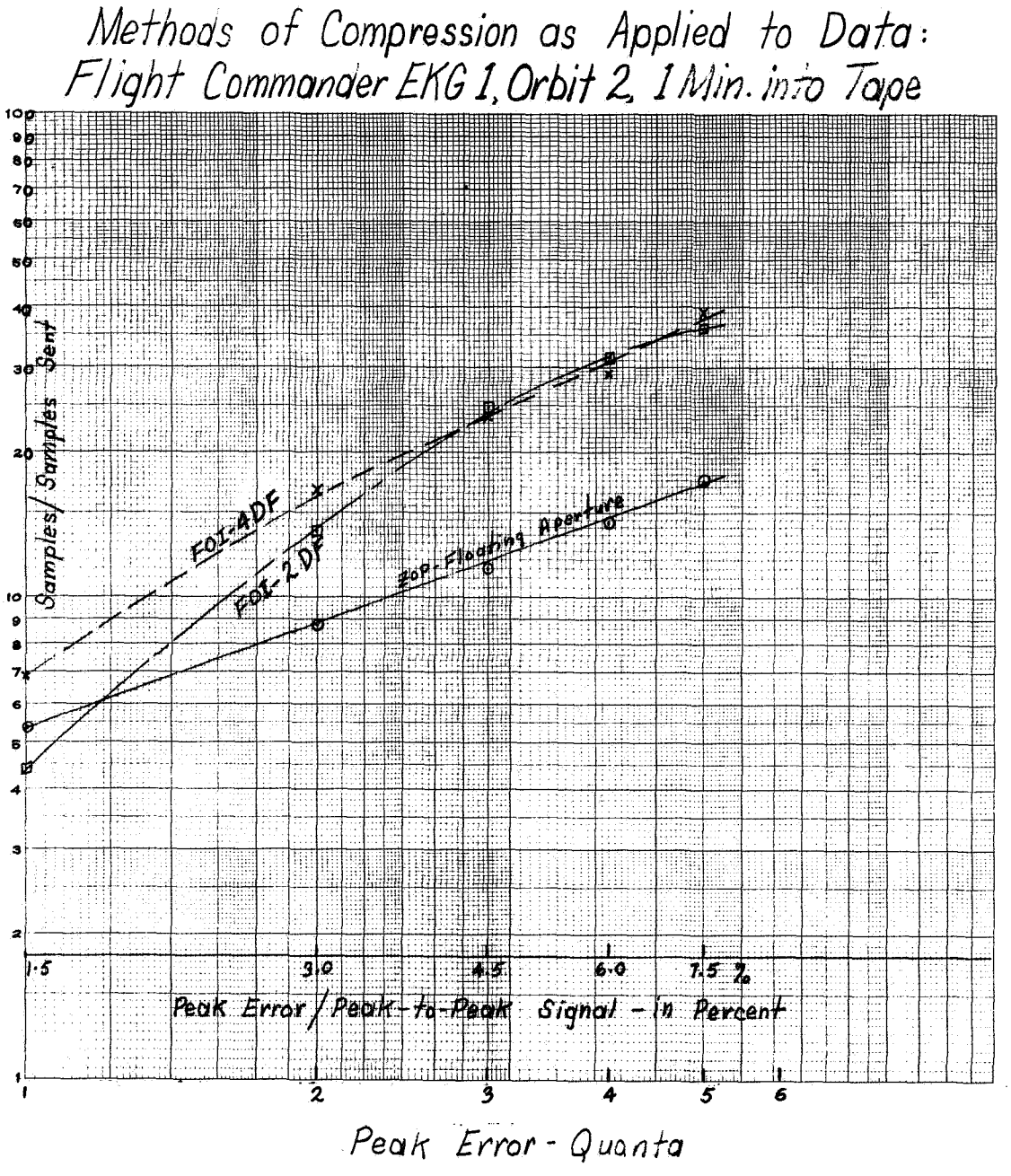


*Figure 3-14. Methods of Compression, Peak Error, Flight Commander, EKG 1*

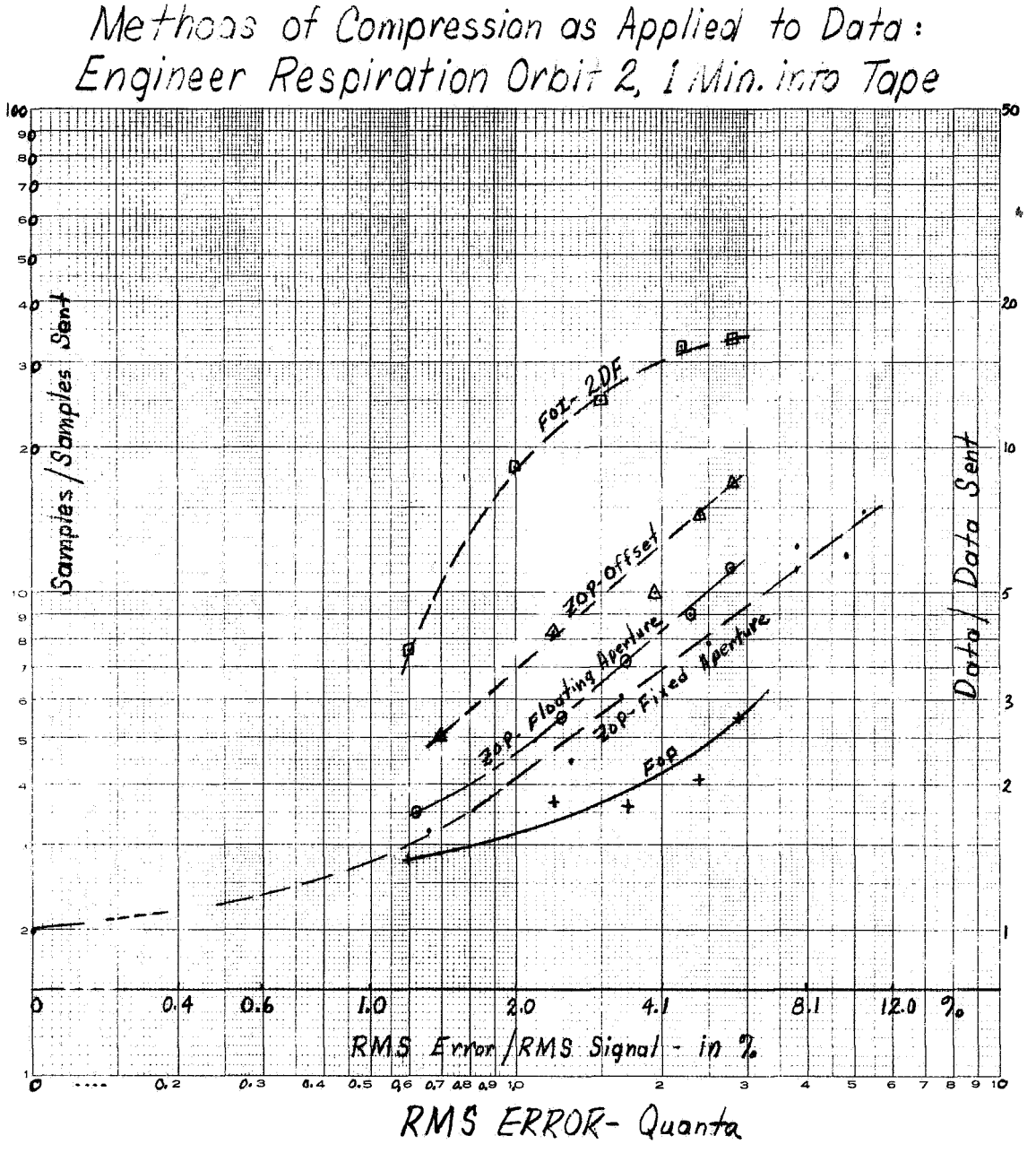


*Figure 3-15. Methods of Compression, Engineer Respiration*

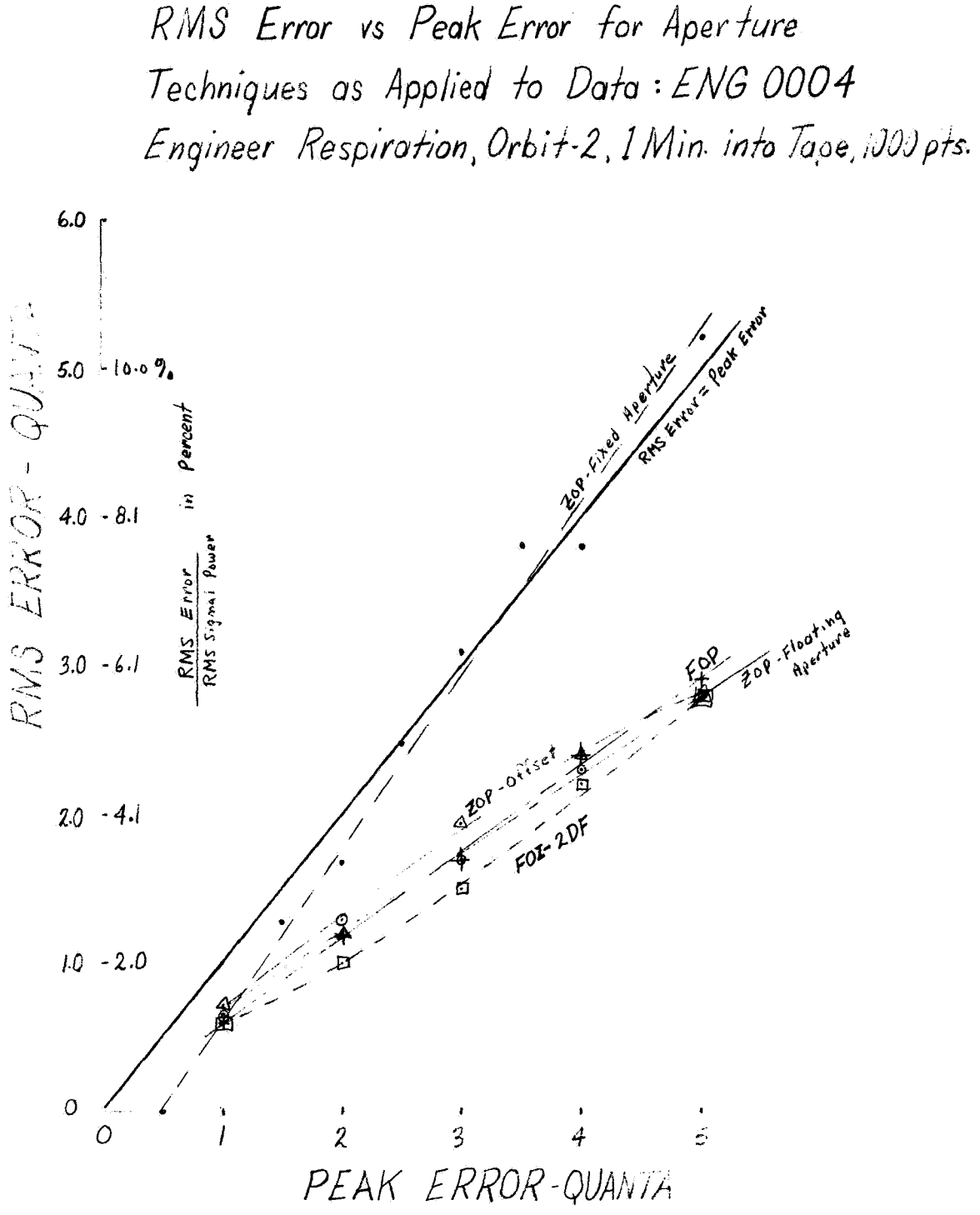


*Figure 3-16. RMS Error vs Peak Error for Aperture Techniques, ENG 0004, Engineer Respiration*

obtained with the FCEKG1 data as presented in Figure 3-13. Peak error is not presented as a percentage of peak-to-peak signal because the respiration data has nonstationary peak-to-peak values.

## 3.6 COMPRESSION OF ATTITUDE CONTROL AND VIBRATION DATA

Simulation results for the ZOP and FOI-2DF methods obtained with the FA 01 Attitude Control-Pitch and ASMRV3-Apollo SM Radial Vibration data samples are presented in Figure 3-17. The results are obvious, the vibration data is essentially non-compressible for reasonable rms distortion.

The Attitude Control data, however, is highly compressible with the ZOP method having a clear advantage over the FOI-2DF method when implementation complexity is considered and the results are extended to practical rms error values.

Figure 3-18 presents the Attitude Control data simulation results in terms of compression ratio versus peak error for the ZOP, FOI-2DF, and FOI-4DF methods. Again, the ZOP method has a clear advantage.

## 3.7 COMPRESSION OF ROLL RATE DATA

The ZOP and FOI-2DF methods were simulated with EA 02 Roll Rate data. The compression ratio results are presented in Figure 3-19 versus rms error. For this data, the FOI-2DF method shows a clear advantage for practical percent rms distortion (less than 10 percent), unless the ZOP trend of sharp roll-off at lower percent distortion changes drastically.

Figure 3-20 and 3-21 present the relationships between rms error and peak error for the ZOP and FOI-2DF methods, respectively, for all five data samples used in the simulation studies. The results are significant in that whereas compression ratios obtained varied significantly for different data types and for different compressive methods, the rms error is a relatively constant function of peak error (aperture, for the two methods considered) for all give five types of data for both compression methods.

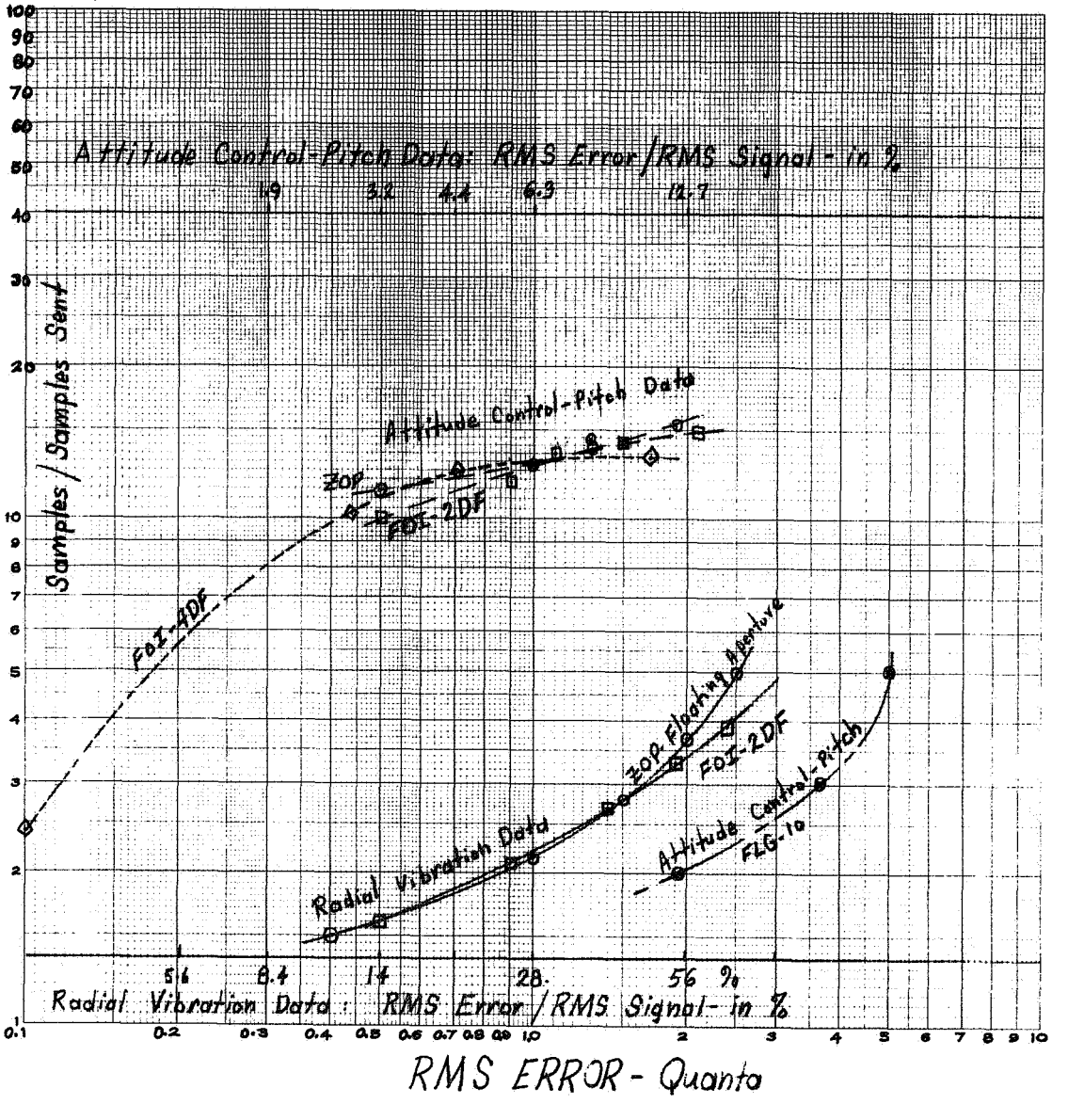


*Figure 3-17. Methods of Compression, RMS Error, FA 01, Attitude Control-Pitch*

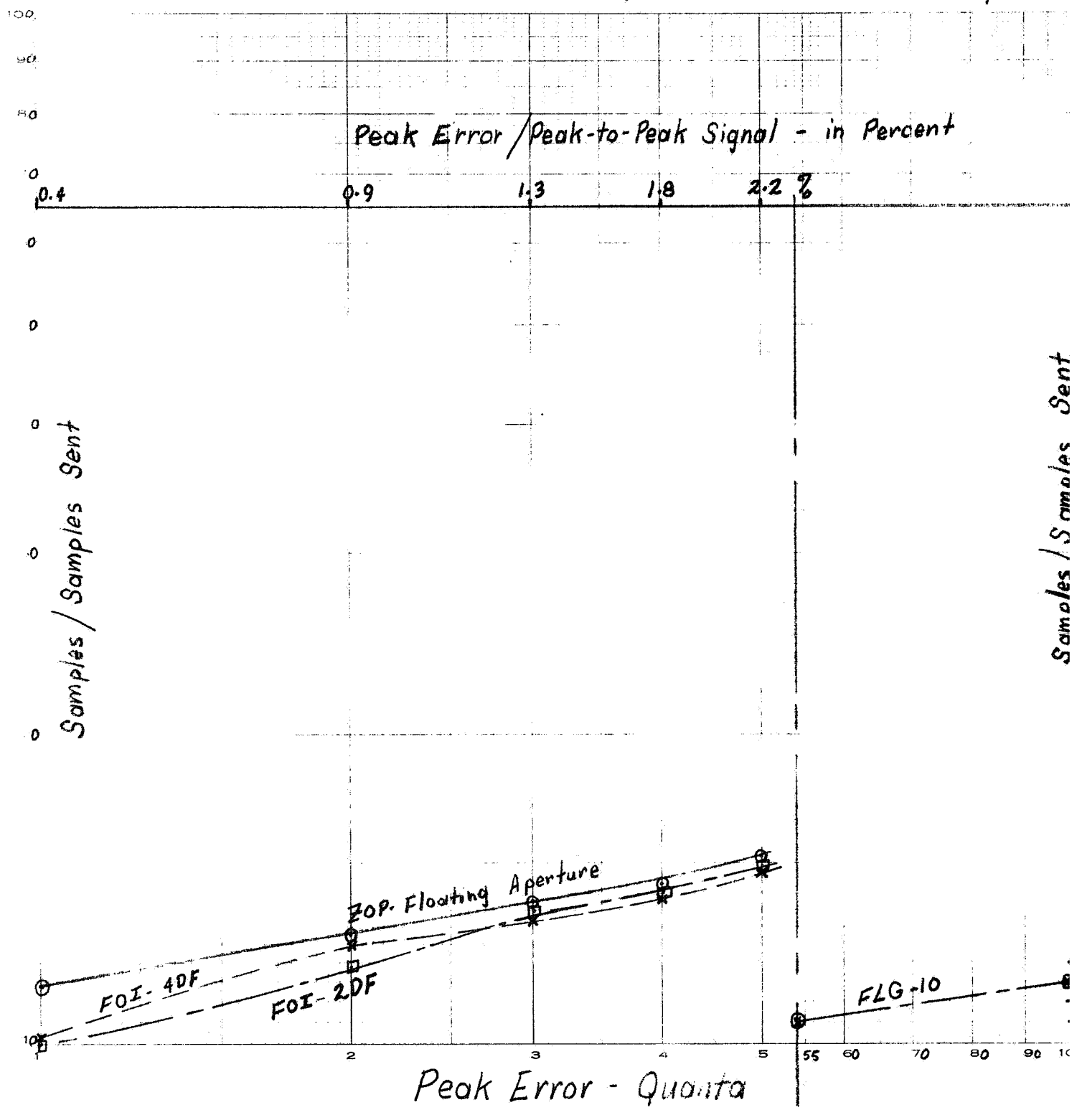


*Figure 3-18. Methods of Compression, Peak Error, FA 01, Attitude Control-Pitch*

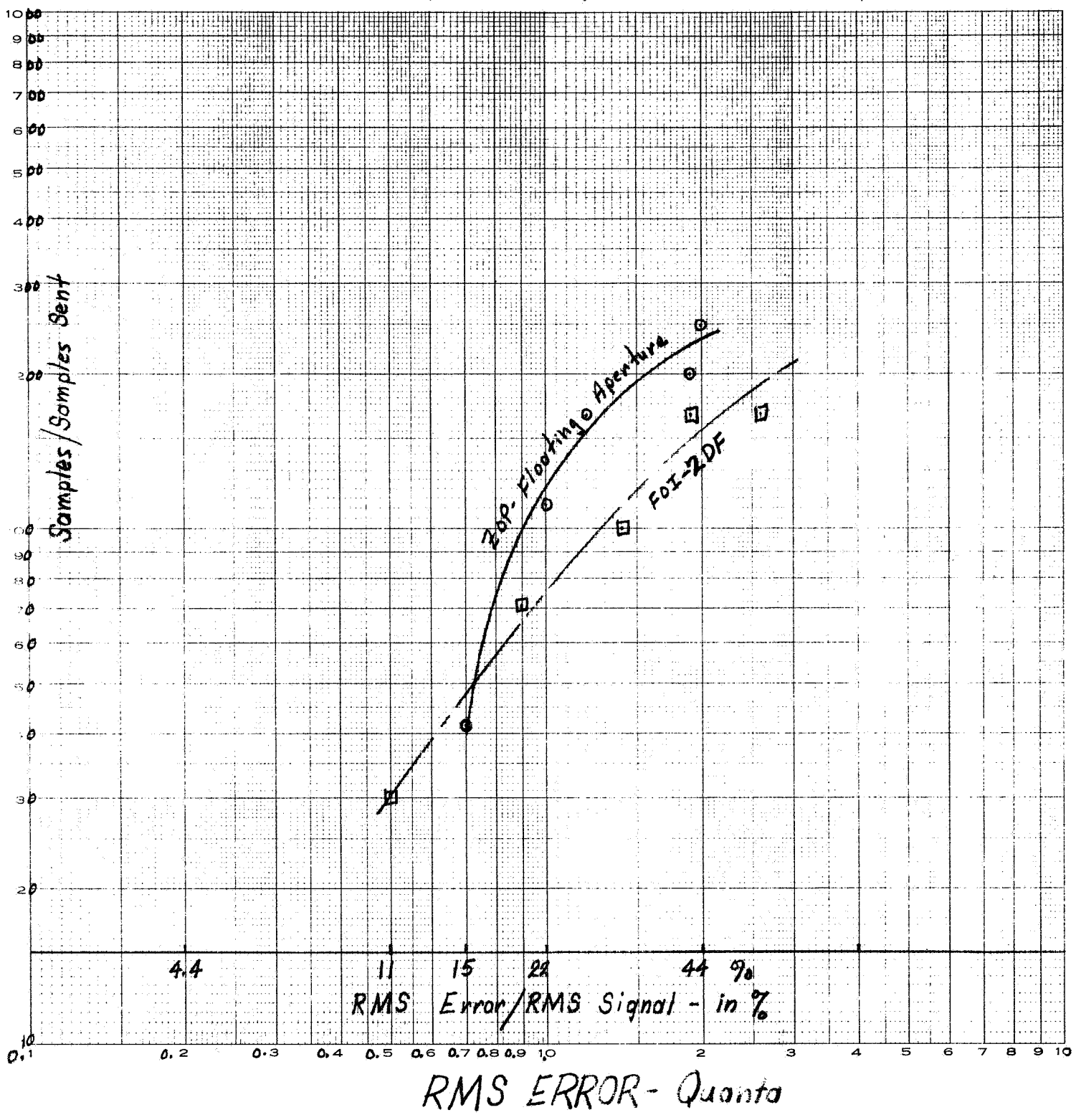


*Figure 3-19. Methods of Compression, EA 02, Roll Rate*

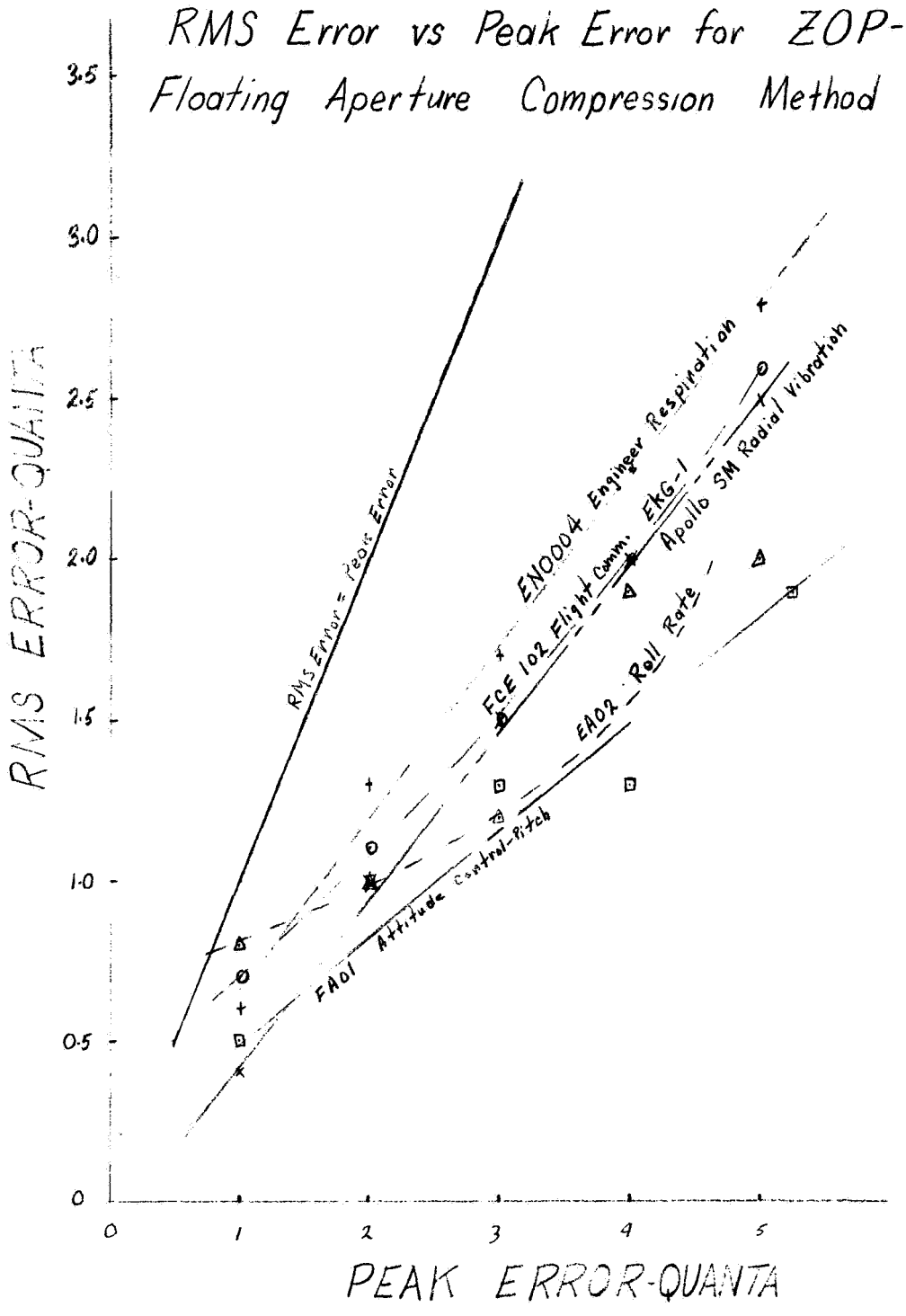


*Figure 3-20. RMS Error vs Peak Error for ZOP-Floating Aperture Compression Method*

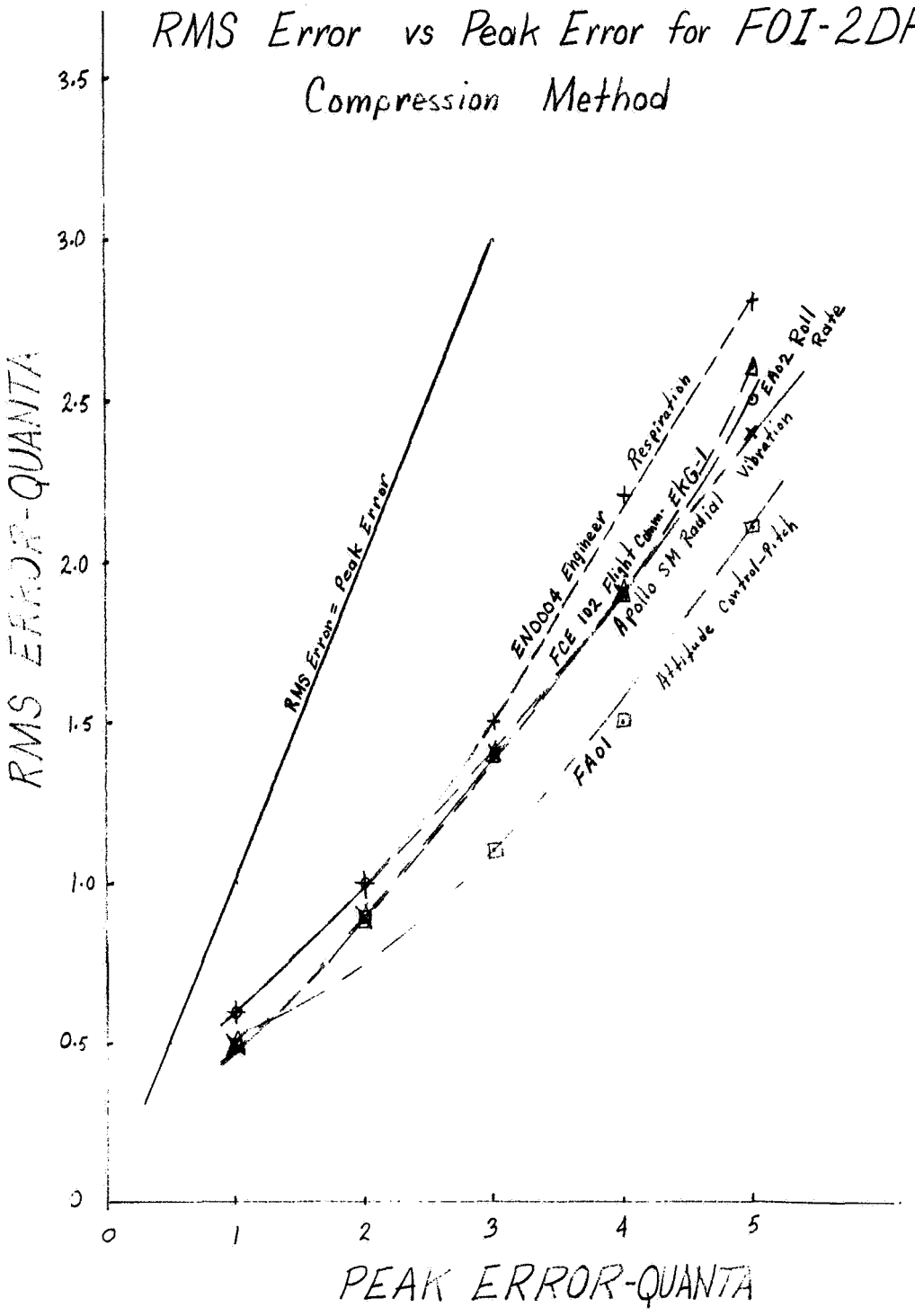


*Figure 3-21. RMS Error vs Peak Error for FOI-2DF Compression Method*

## 3.8 EVALUATION OF ZOP AND FOI-2DF METHODS

The previous results have indicated how various compression methods fared with the different types of test data. However, the problem remains of determining a method for systematic evaluation of any compression method for a wide range of data types.

It is proposed that a rating factor similar to the well known gain-bandwidth amplifier factor be employed. Consider the factor

$$R(k) = \frac{\alpha\phi(k)}{D^2(k)}$$

where:

$\phi(k)$ = the compression ratio in sample/samples sent as a function of peak error or aperture k

$D(k)$ = the percent rms distortion as a function of k

$\alpha$ = a normalizing constant.

The second power of D(k) (mean square distortion) is used to stress the importance of low distortion in the reconstructed signal. It is, desirable to choose the method with maximum R(k) for a specified k.

Now normalize the factor R(k) such that for $\phi(k) = 2$, and $D(k) = 5$ percent, $R(k) = 1$. R(k) could be normalized to other values of $\phi$ and D, but these values appear reasonable because a compression ratio of 2 is the minimum integral compression ratio greater than 1 (which is no compression); and 5 percent is a conservative upper bound on allowable rms distortion. Therefore,

$$R(k) = 12.5\frac{\phi(k)}{D^2(k)} .$$

The effect of sensor tags can be accounted for by multiplying R(k) by the ratio:

$$\frac{\Sigma \text{ bits/sample}}{\Sigma \text{ bits sent/sample sent}} .$$

Apply this rating factor to the ZOP and FOI-2DF simulation results with the five types of test data and take the average R(k) with equal weighting for each data type. We obtain the results shown in Figure 3-22.

Figure 3-22 indicates that the FOI-2DF method has an average advantage over the ZOP method. However, for most of the data tested, the range of practical rms distortion was obtained with peak errors of approximately one quantum. Thus, Figure 3-22 emphasizes that for low distortion, the ZOP method is competitive with the more complex FOI-2DF method. The results are based upon the assumed values for the normalization of R(k).

In summary, the two aperture techniques ZOP and FOI-2DF yielded better compression performance than the other techniques studied (when implementation is considered). However, when averaged over all five types of test data for practical values of distortion, the two methods give comparable performance. A reasonable conclusion appears to be the implementation of both methods; each method to be used on the data types that yielded the best simulated compression ratios.

## 3.9 COMPARISON OF ORIGINAL AND RECONSTRUCTED SIGNALS

The previous sections have considered the results of the simulated compression techniques in terms of such calculable parameters as compression ratio, peak error, rms error, and percent rms distortion. These are valid performance criteria; however, one more is needed in evaluating compression methods for various types of data. This additional criterion is: "how does it look?"

Certain data such as EKG signals are evaluated by visual study. Thus, peak and rms error performance may be similar for two different compression methods, but one method may be totally unacceptable because of some peculiar perturbations introduced in the reconstructed signal.

Therefore, to obtain a measure of reconstructed signal fidelity, original sample data and reconstructed sample data were plotted with a California Computer Products, Inc., plotter. Because the ZOP and FOI-2DF methods gave the best performance of all the practical compression techniques, these two methods were compared with large and small apertures (k = 5, 1) for all data tested. The results for all data types except the radial vibration data, which did not compress well, are shown in Figures 3-23 through 3-32.

In addition to the ZOP and FOI-2DF methods, several fixed sample rate interpolation methods were evaluated with the EKG data to determine the distribution of peak errors because only the aperture compression techniques are peak limited with the peak error known before reconstruction. Figures 3-25 and 3-26 show the reconstructed EKG signal obtained with Lagrange, $\frac{\sin x}{x}$, and Fourier filter interpolation. The results were similarly deficient in reproducing the relatively sharp pulse peaks; whereas, the variable sample rate aperture methods (ZOP and FOI-2DF) reproduced the peaks very well because, probably, all sample points occurring during the pulses were transmitted without deletion.

This limitation of peak error is the very feature of the aperture techniques which makes them so well suited to compressing data which is to be evaluated on a point-by-point basis upon reconstruction rather than upon a statistical, or averaged error basis. Because most medical and scientific data is still evaluated by the human eye, the ZOP and FOI-2DF aperture techniques appear to have great advantage over the other methods which are not peak error limited.

Figures 3-27 through 3-32 present comparisons between the ZOP and FOI-2DF methods for the other types of data tested. Both methods provided good reconstruction fidelity for the aperture k = 1 on all data shown except the EA 02 Roll Rate data. However, this data was too low in signal power for the 8-bit quantizing used in the analog-digital conversion. The signal should have been amplified before quantizing, or the quantum levels should have been normalized to the signal level. Thus, we find that the Roll Rate data provided compression ratios of 41 and 30 for the ZOP and FOI-2DF methods with rms distortion levels of 15 percent and 11 percent, respectively, with an aperture of one quantum level. Surely, a finer grained quantizing would have yielded less but still acceptable compression ratios at acceptable levels of distortion. This could be done without increasing the number of bits per sample by employing a gain factor code word before transmission of the sampled data.

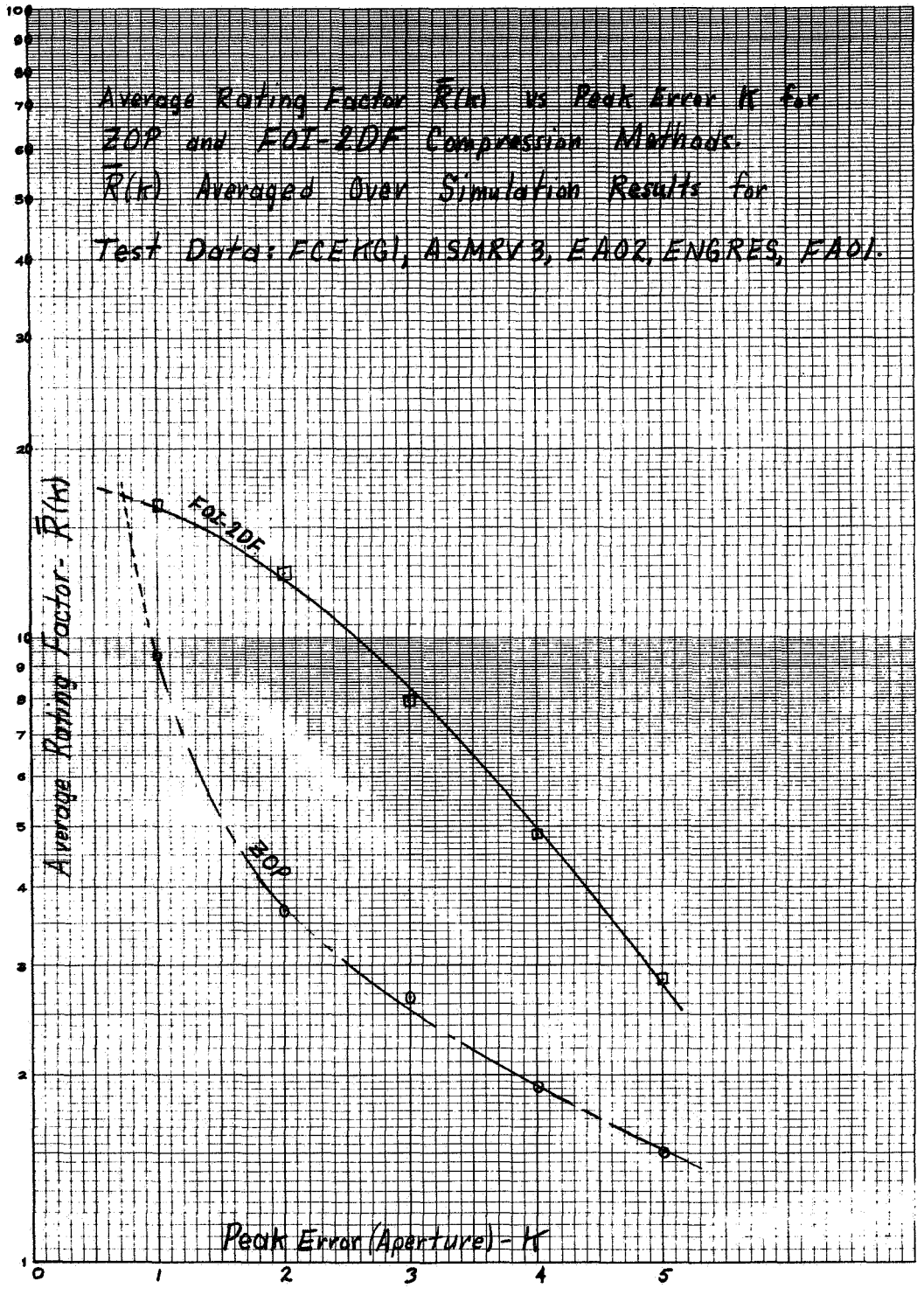


*Figure 3-22. Average Rating Factor $\bar{R}(K)$ vs Peak Error K for ZOP and FOI-2DF Compression Methods*

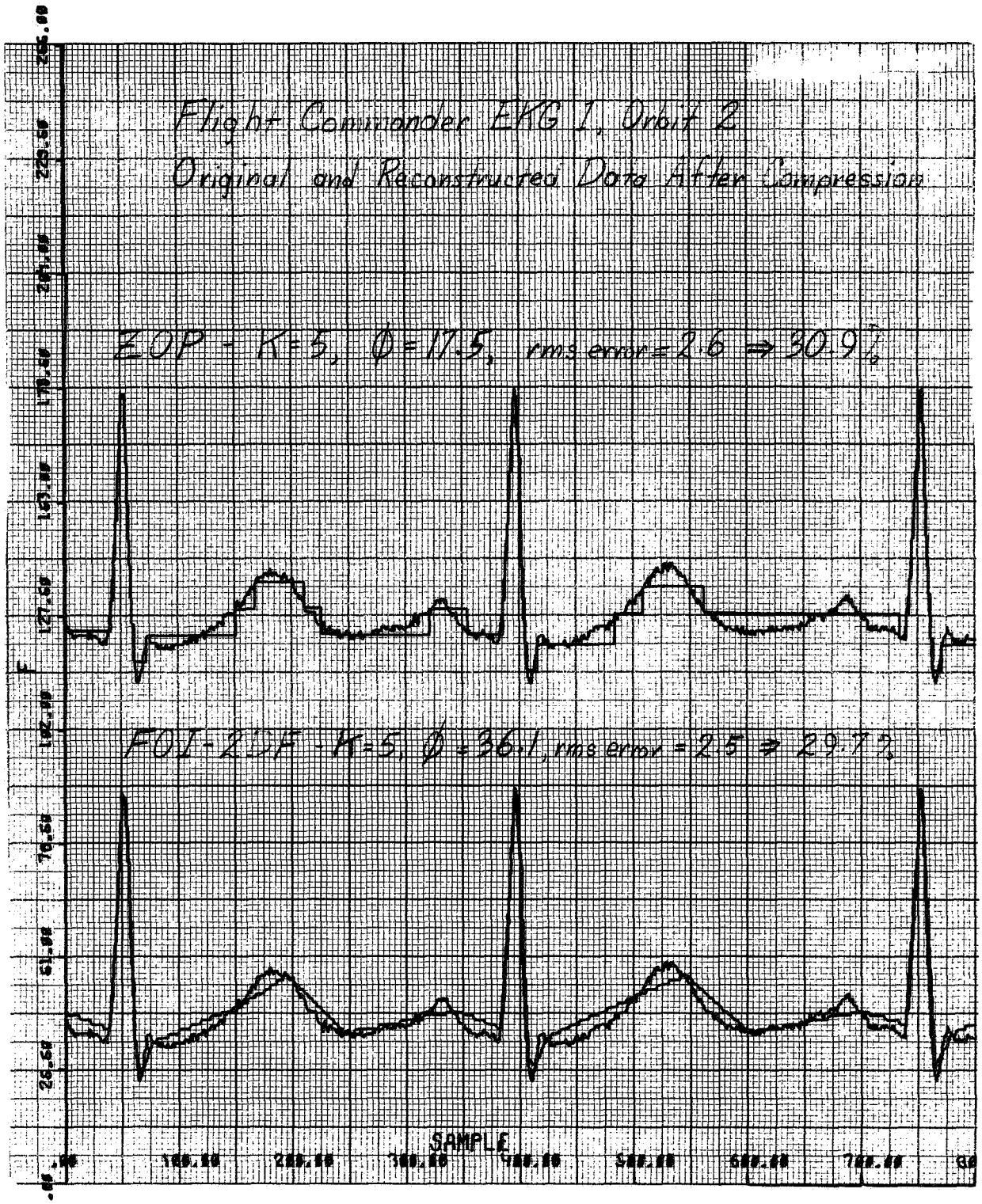


*Figure 3-23. Original and Reconstructed Data After Compression, Flight Commander, EKG 1*

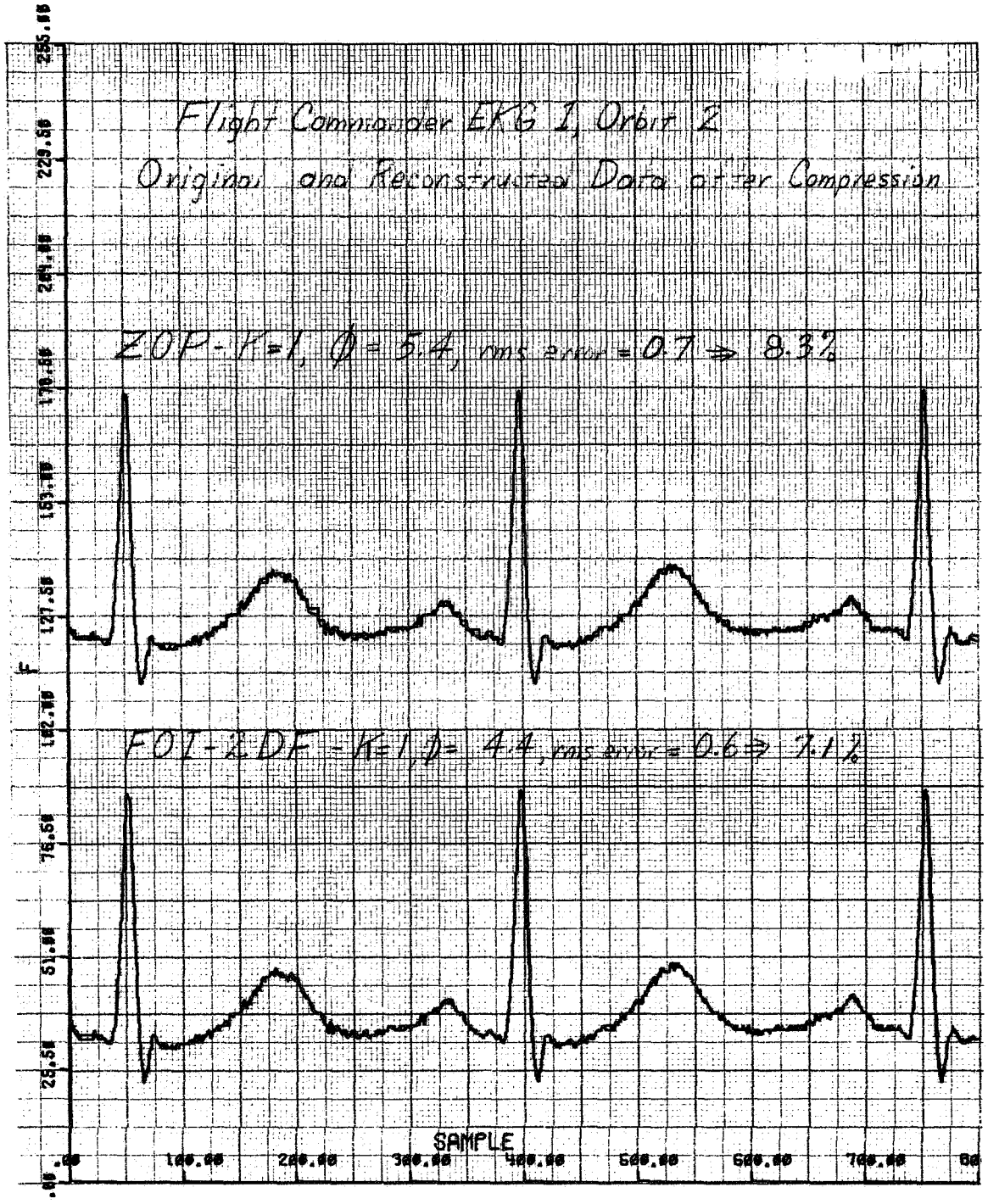


*Figure 3-24. Original and Reconstructed Data After Compression, Flight Commander, EKG 1*

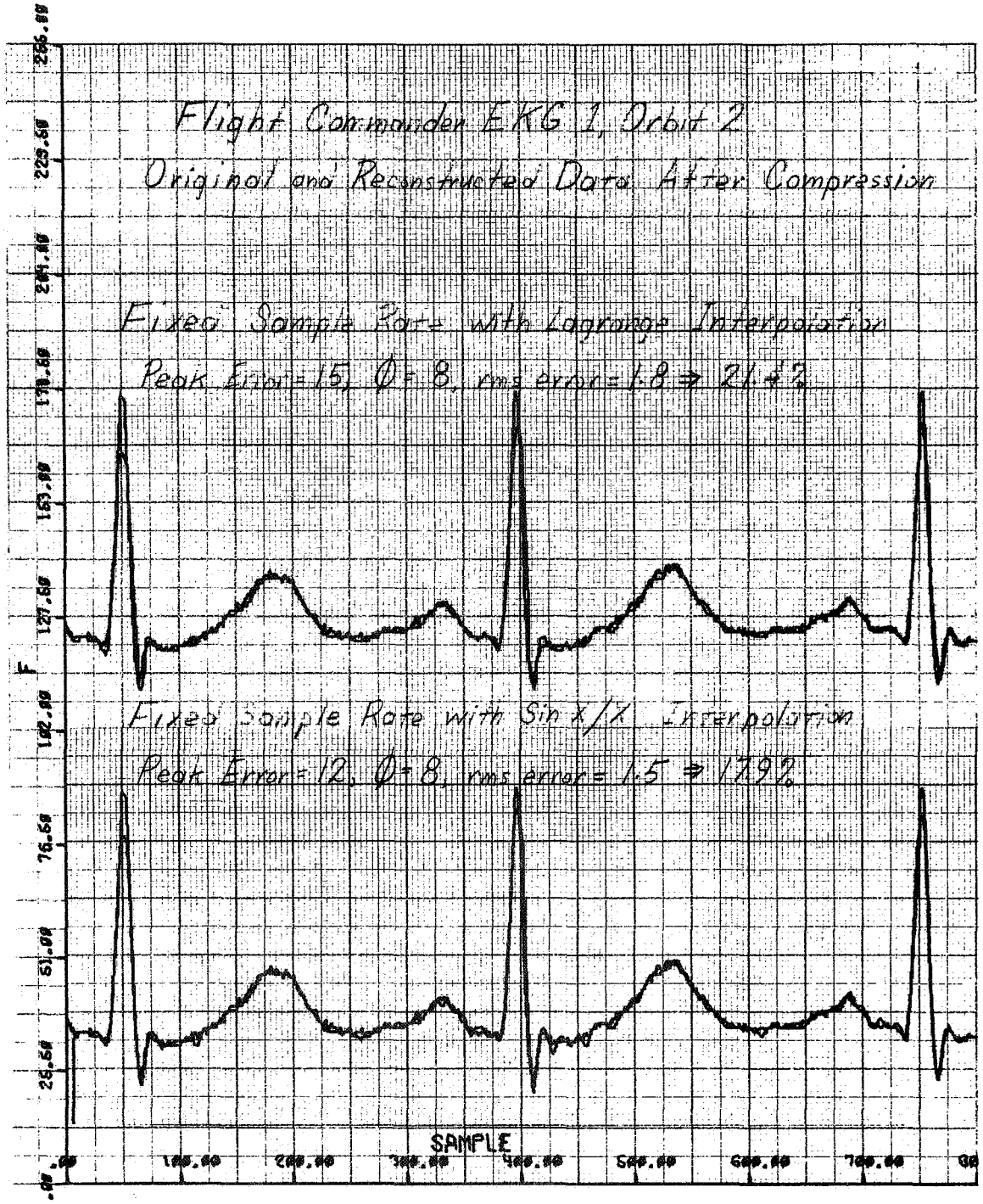


*Figure 3-25. Original and Reconstructed Data After Compression, Flight Commander, EKG 1*

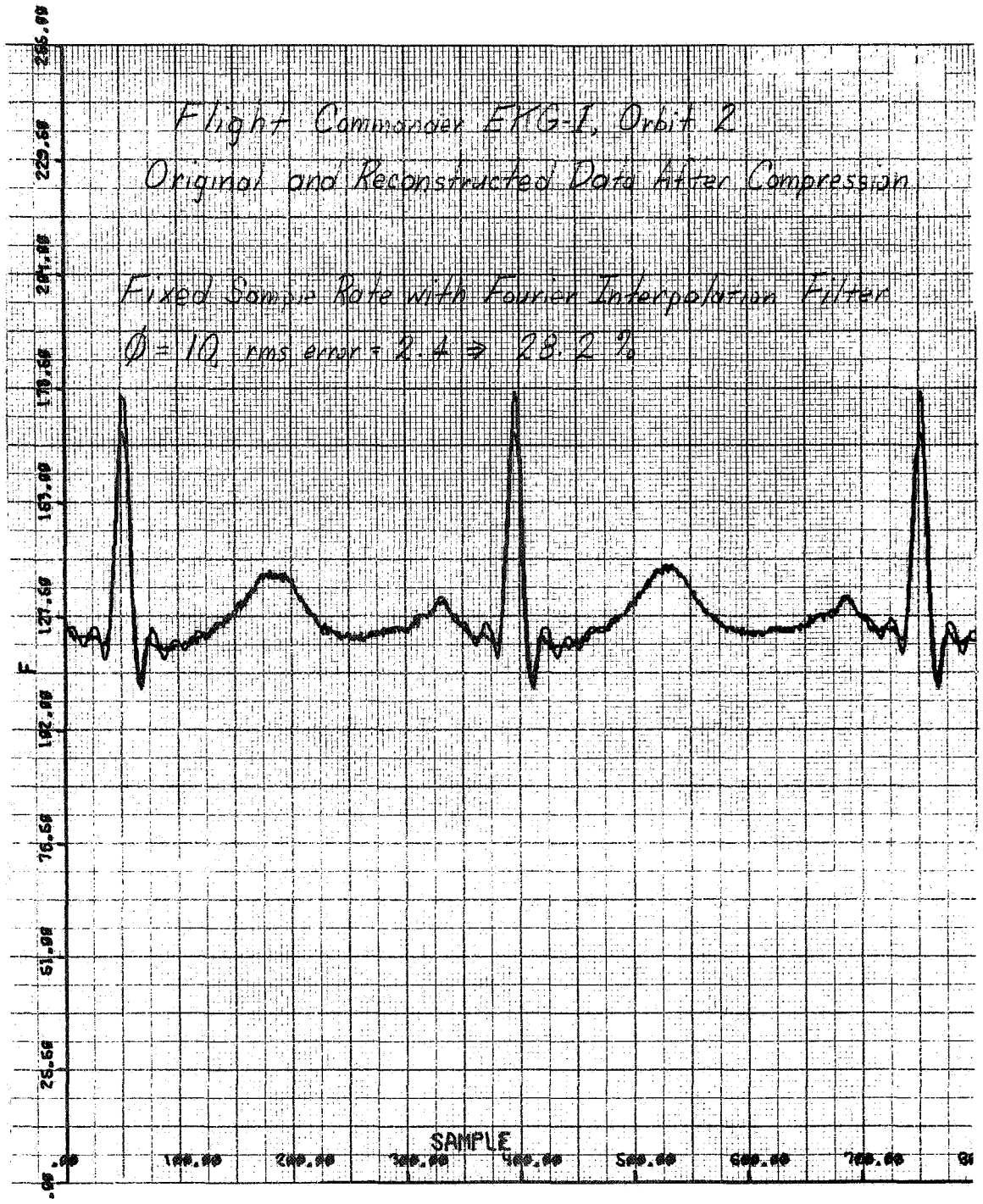


*Figure 3-26. Original and Reconstructed Data After Compression, Flight Commander, EKG 1*

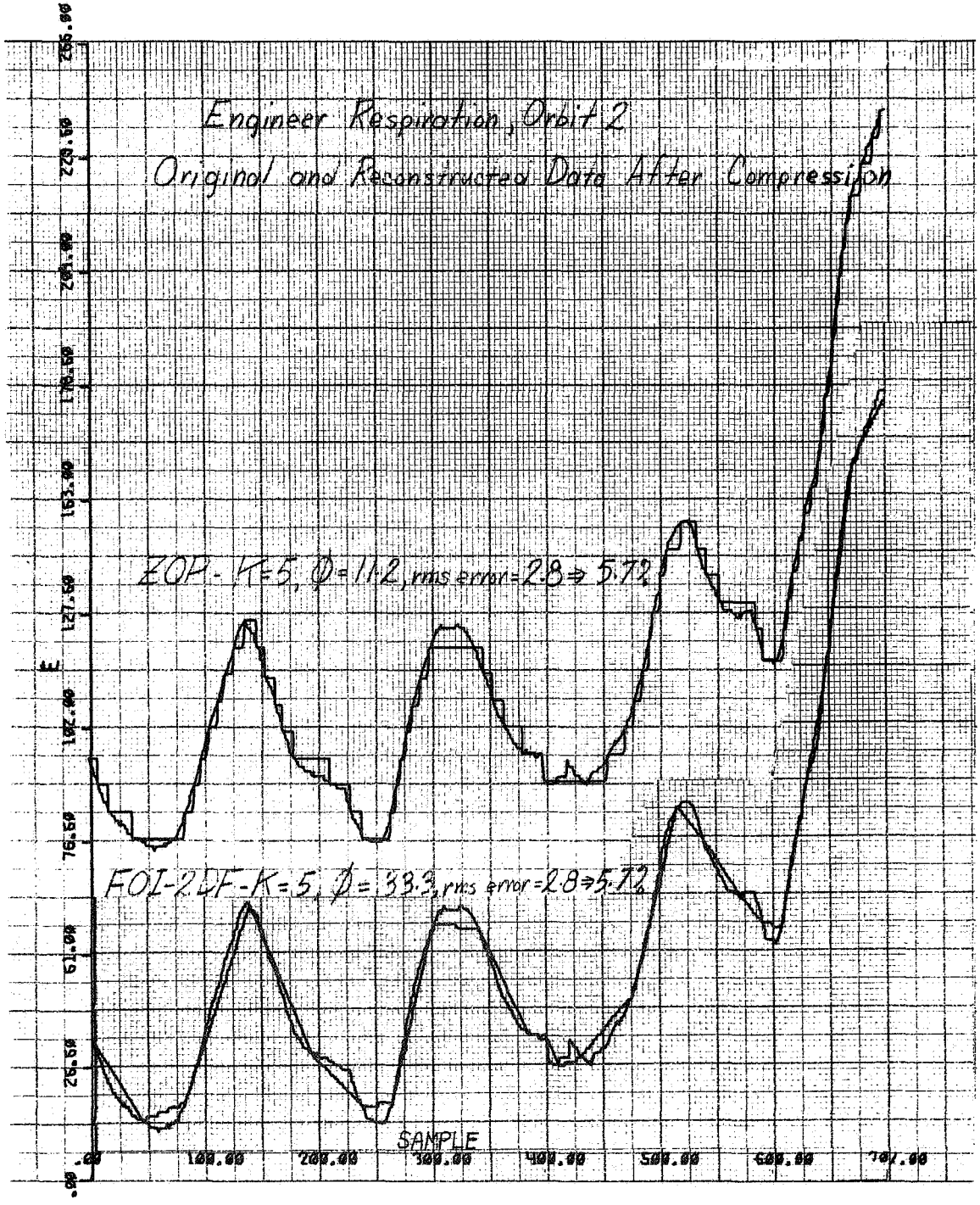


*Figure 3-27. Original and Reconstructed Data After Compression, Engineer Respiration*

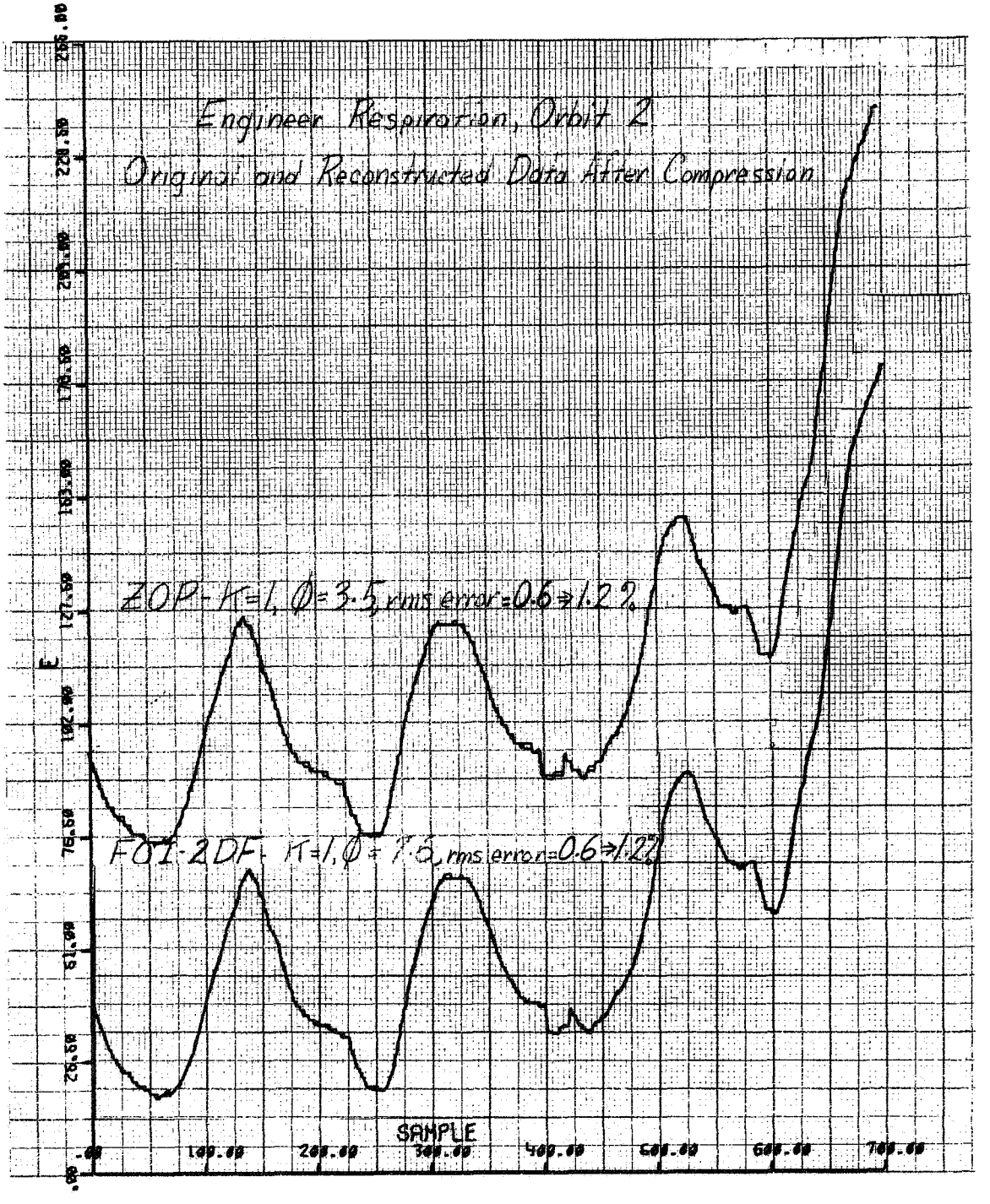


*Figure 3-28. Original and Reconstructed Data After Compression, Engineer Respiration*

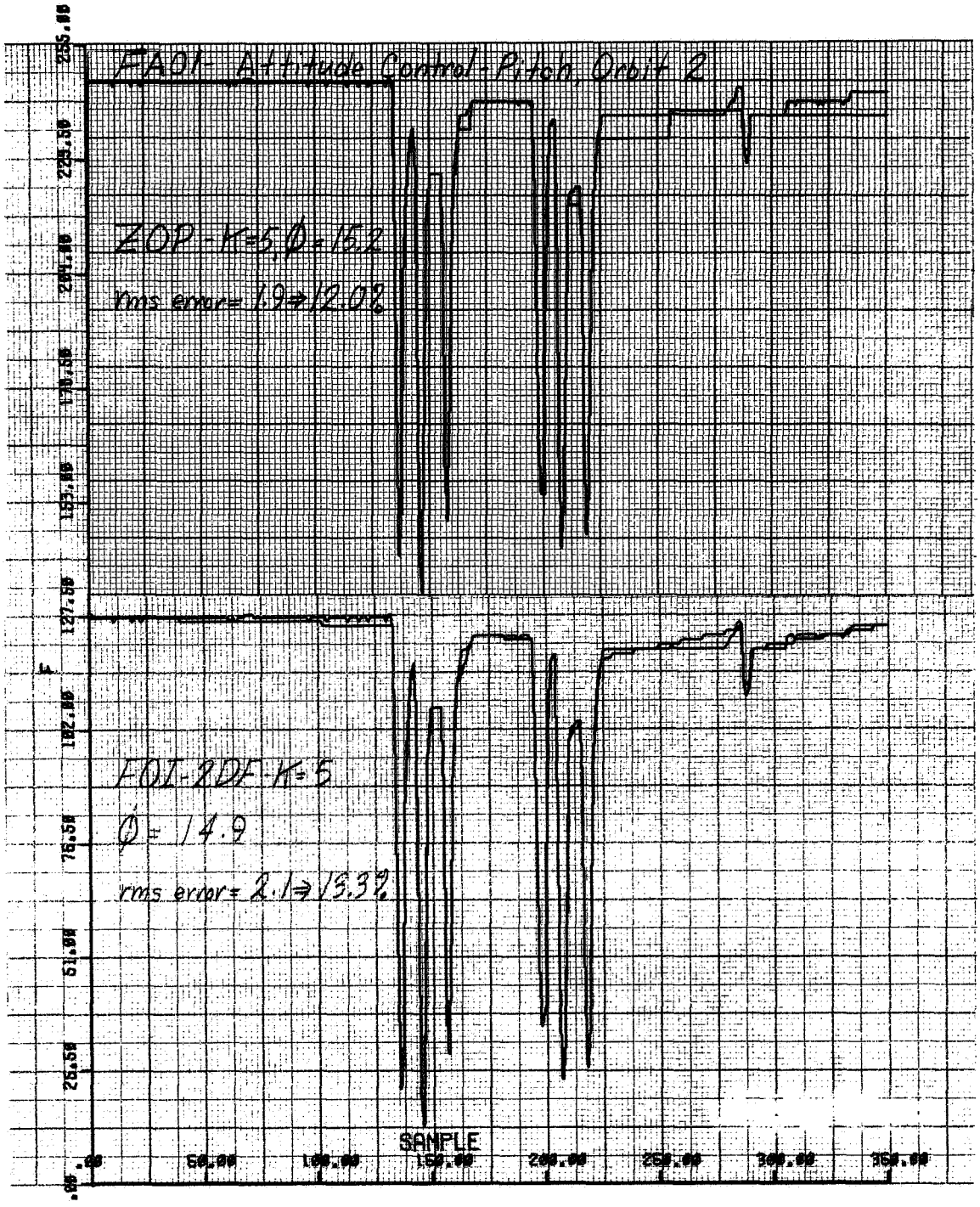


*Figure 3-29. FA 01, Attitude Control-Pitch*

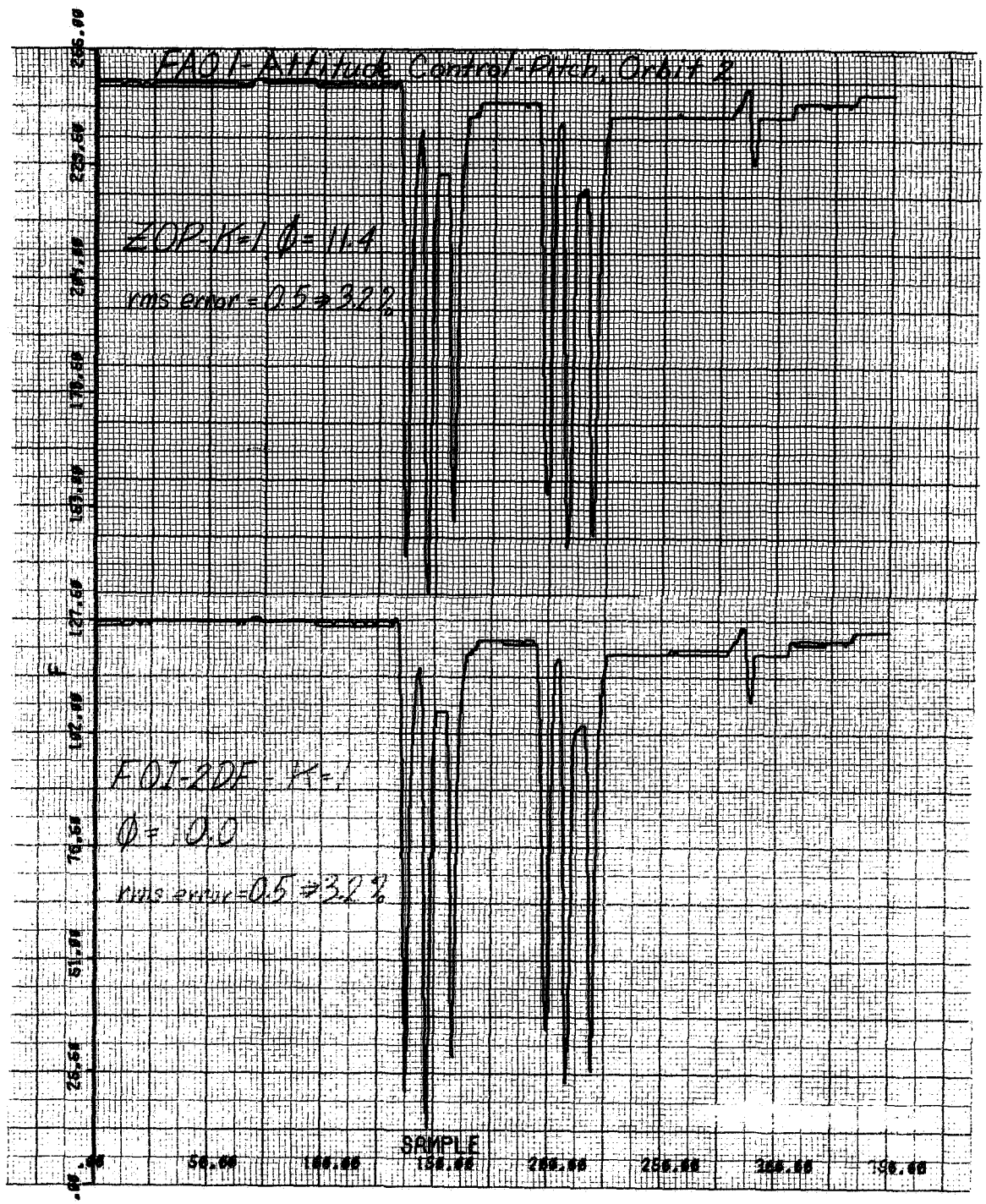


*Figure 3-30. FA 01, Attitude Control-Pitch*

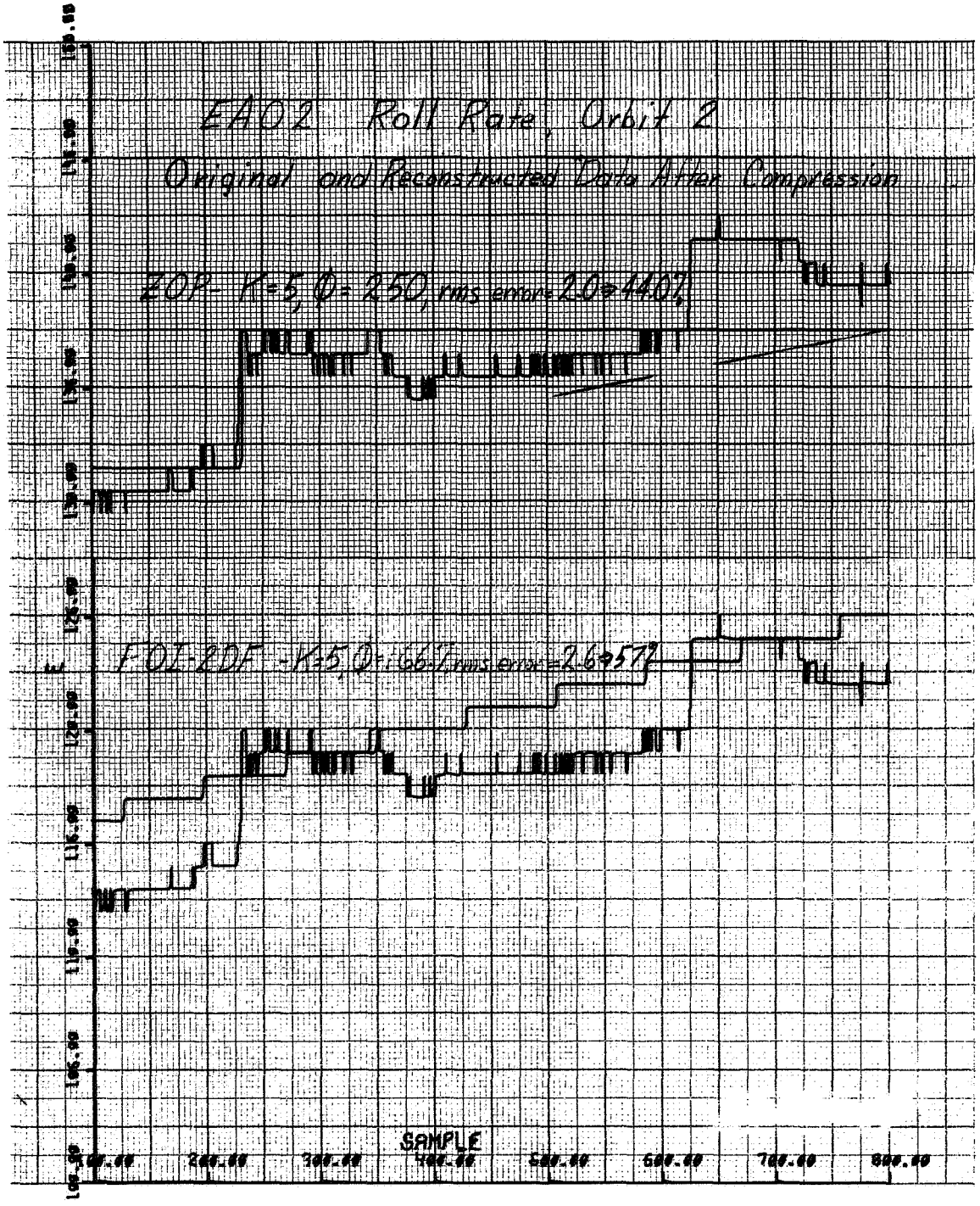


*Figure 3-31. Original and Reconsiructed Data After Compression, EA 02, Roll Rate*

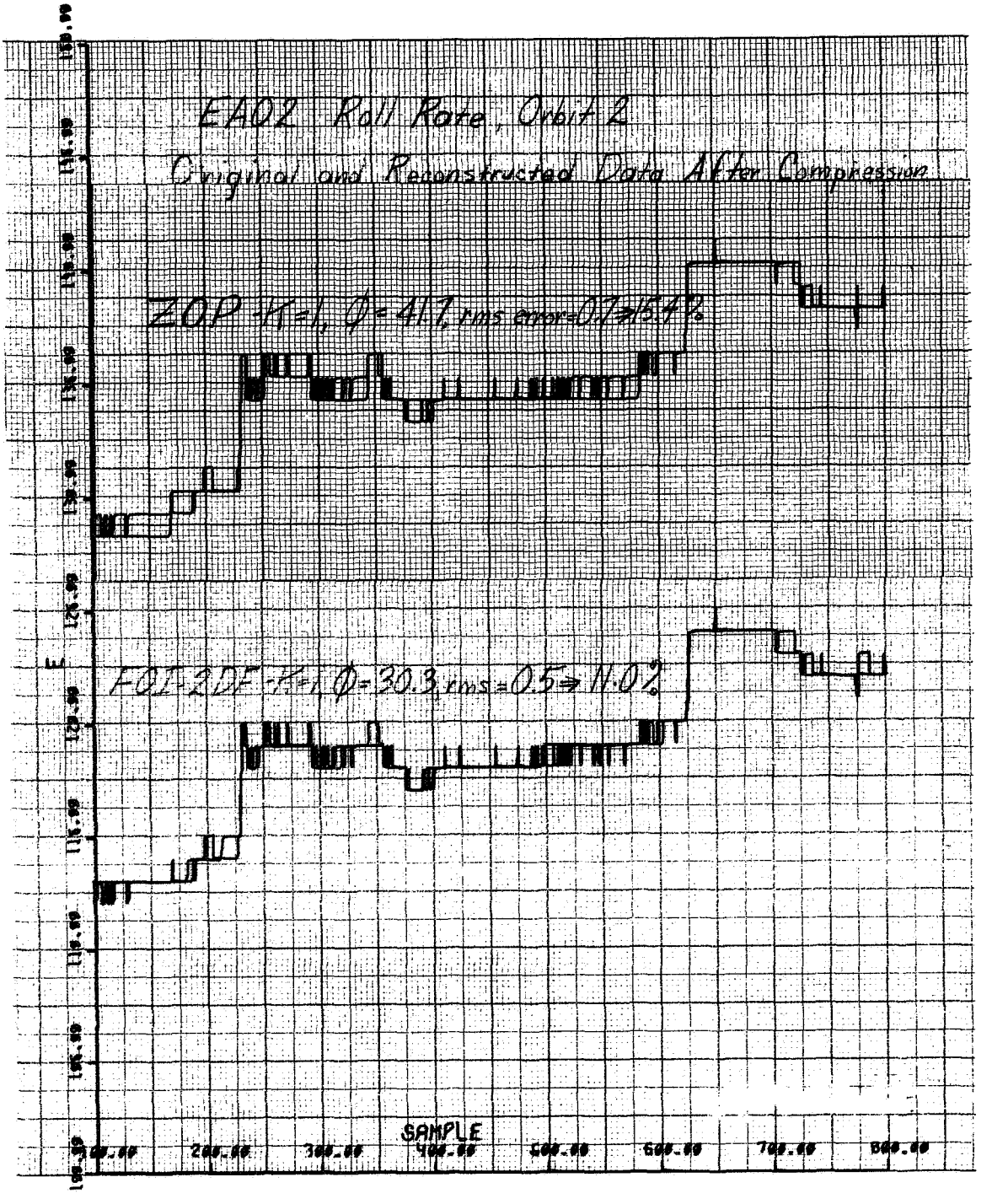


*Figure 3-32. Original and Reconstructed Data After Compression, EA 02, Roll Rate*

Section 4

# SYSTEM DESIGN

## 4.1 INTRODUCTION

The discussion of the design of a complete, adaptive, multipurpose data compression system is divided into three sections. Section 4.2 discusses a tradeoff study which compares significant aspects of a variety of data compressors. Section 4.3 discusses the problems of adaptive buffering and control of compression parameters, while Section 4.4 presents the basic design of a complete, adaptive data compression system which is directed particularly toward data systems of the type planned for manned spacecraft.

## 4.2 TRADEOFF ANALYSIS

Section 3 evaluated the relative effectiveness of a variety of data compression methods by observing the compression ratios they achieved as a function of both the rms and peak errors in the reconstruction for several representative types of manned spacecraft telemetry data. This section will extend the evaluation work by considering the relative hardware penalty that might be incurred by the implementation of each method, and by relating various aspects of this penalty factor to the relative effectiveness, will develop a comprehensive tradeoff analysis.

In using the term "relative effectiveness," it is recognized that this is a difficult judgement to make. The data presented in Section 3 makes it clear that there is no method which is superior for all data types and under all conditions of allowable rms and/or peak error. For example, although Figure 3-15 shows the FOI-2DF technique to have a distinct advantage over the others tested on this particular data type, this advantage does not appear in Figure 3-11 except at large

rms error levels, and at low rms error levels, the FOI-2DF rapidly becomes the least effective method of those tested. If only one technique is to be used in a given data system, it must therefore be carefully selected, and for a wide variety of data sources, there is a strong case for having more than one technique at the disposal of the system.

The following section will discuss each technique tested in sufficient detail to permit clear comparisons of various features to be made. In discussing the implementation of each technique, it will be assumed that the compressor is to be used in a manned spacecraft data system such as that found in the Gemini spacecraft. The input to the compressor will be a multiplexed PCM data stream derived from analog and digital sensors.

In developing a critical comparison of the various proposed compression methods, the following general criteria will be considered:

a. Effectiveness of performance as measured in terms of rms and peak errors as functions of compression ratio for the various data types (reference is made to Section 3).
b. Implementation cost in terms of arithmetic capability required, program complexity, and storage requirements.
c. Other factors which may be significant for particular compression methods, such as impact on ground equipment, excessive coding, and transmission difficulty.

### 4.2.1 Aperture Techniques

This section considers the implementation of those aperture techniques that were evaluated in Section 3, with the exception of the First-Order Predictor and the Zero-Order Predictor, Fixed Aperture; their poor performance on the EKG data in comparison with the other techniques precludes their consideration as a generally useful compression method. Each of the other aperture algorithms is discussed below.

#### 4.2.1.1 Zero-Order Predictor (ZOP)

There are several forms of aperture compression algorithms which employ zero-order prediction; the one which has been most extensively evaluated, and which has consistently given the best results is the floating aperture type. A flow diagram for this computation is shown in Figure 4-1. The computations required per data point in the ZOP are very simple, consisting of only two subtractions and a zero-comparison together with several transfer operations and two memory cycles. The storage requirements for this method also are minimal, because only the previously transmitted sample, the tolerance values and the number of samples, n, since the last transmission need be stored. Assuming 8-bit data samples, five 3-bit alternate tolerance values, and 8 bits for n, a total of 31 bits of storage per channel is required.

The computational simplicity of the ZOP technique, coupled with its minimal hardware complexity and its generally effective performance give it a very advantageous position in a tradeoff study of practical compression techniques, as has been recognized by previous investigators. Its only disadvantage is that for most data types a somewhat higher compression ratio can be achieved by more sophisticated techniques. Note that it is entirely feasible to implement the ZOP calculation by analog techniques resulting in even greater hardware savings, whereas the general complexity of the computation program required by all the other techniques dictates the use of a digital processor for their implementation.

#### 4.2.1.2 First-Order Interpolator—Two Degrees of Freedom Method (FOI-2DF)

Of several first-order interpolation techniques, the FOI-2DF method requires significantly simpler implementation and produces results which are comparable to or improve upon, the others. A flow diagram showing the required computations for each data point is given in Figure 4-2. The computation by the longest flow diagram path requires seven additions or subtractions and two divisions. The storage requirements for this method are greater than those of the ZOP, because it is necessary to store the two slope limits, $L_{max}$ and $U_{min}$, and the sample value immediately preceding the present value. With the same assumptions as for the ZOP, the total storage is 53 bits per channel.

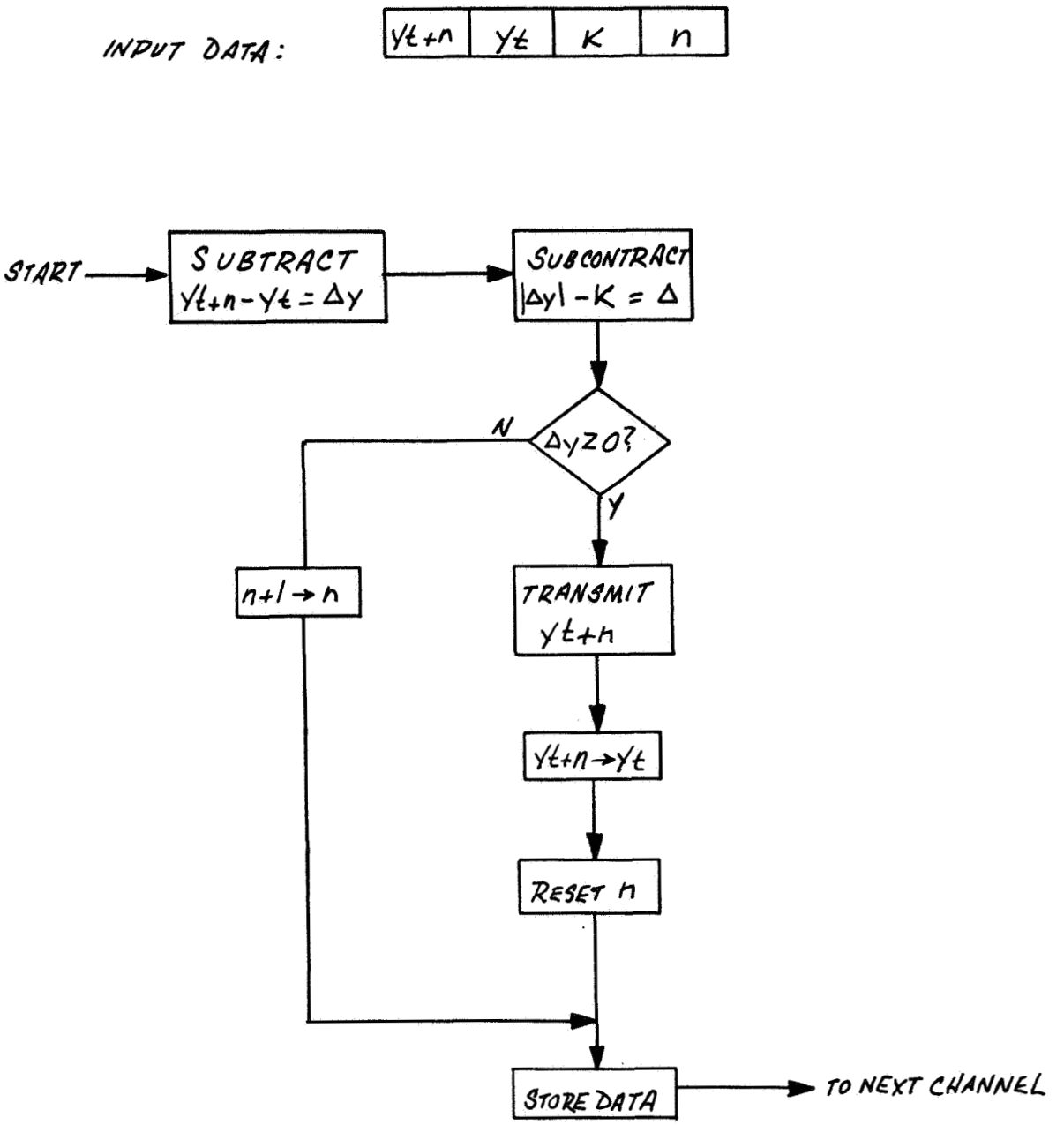


*Figure 4-1. Flow Diagram—ZOP Algorithm*

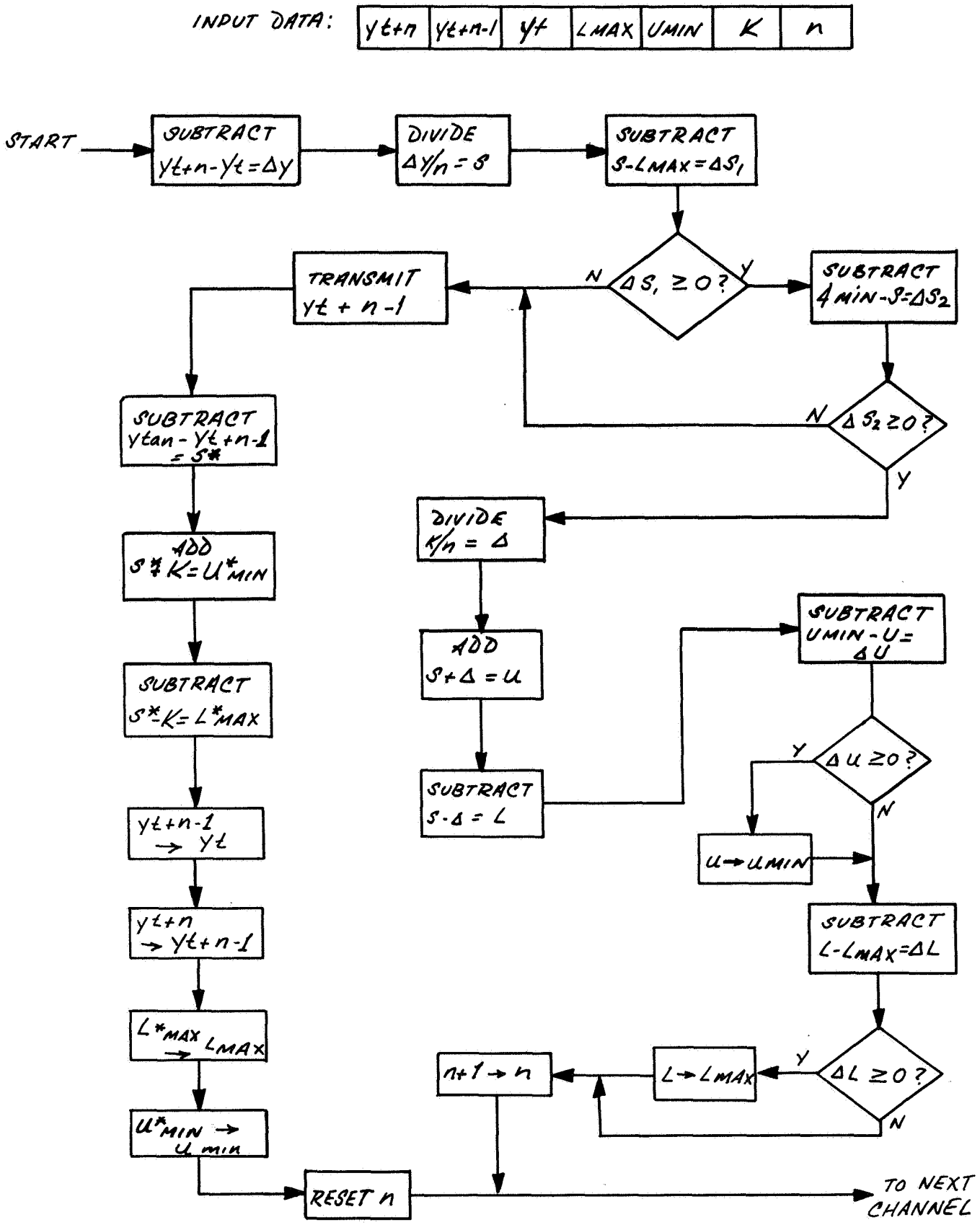


*Figure 4-2. Flow Diagram—FOI-2DF Algorithm*

The implementation of the FOI-2DF method requires a divider (or multiplier) and an adder-subtractor. The multiply function can be accomplished in the processor by repeated additions and shifts, therefore, the arithmetic capability required is not appreciably more than that of the digital implementation of the ZOP; the chief penalty is the approximate doubling of memory size with its attendant weight increase, and a longer computation time.

Note that the results presented in Section 3 show that the FOI-2DF and ZOP techniques are in a sense complementary, with the FOI-2DF exhibiting good compression performance at larger peak error (aperture) values, while the ZOP performance is good at smaller aperture values where the effectiveness of the FOI-2DF decreases sharply. Figure 4-3 shows the general shape of the performance curves for both methods. Note that the performance curves for most other techniques (except, notably, the Karhunen-Loéve method) fall to the lower right of a composite curve made up of the best sections of the FOI-2DF and ZOP curves.

#### 4.2.1.3 First-Order Interpolator—Four Degrees of Freedom Method (FOI-4DF)

This technique was investigated principally because of its academic interest as the optimum first-order interpolation method. By constraining neither end of the approximating line to lie on a data point, the longest straight line within a fixed peak error of all points lying along the line is achieved. Its performance, however, does not appear to be markedly better than the FOI-2DF method for the cases studied, primarily because of the necessity of transmitting two data points (i.e., the beginning and end) for each line segment. In the FOI-2DF method, only one point per line segment is required, thus although more line segments are needed to approximate the signal, the FOI-2DF is not necessarily inferior to the FOI-2DF method.

Implementation of the FOI-4DF method presents several difficult problems. The most significant of these is that the storage requirements increase as the number of redundant points since the last transmission increases, since each intermediate point must be tested during every computation. Thus, the storage capacity required is theoretically unlimited, although in practice a limit could be set by

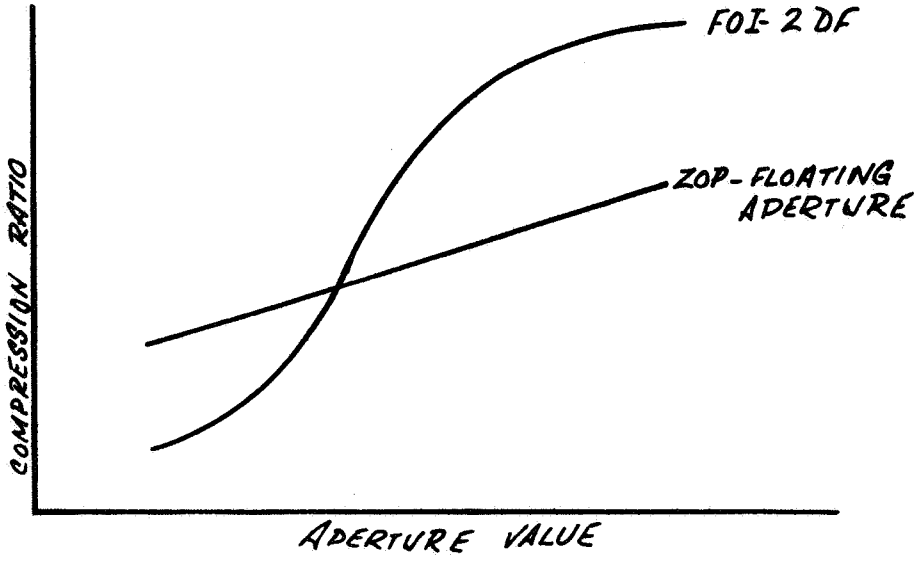


*Figure 4-3. ZOP and FOI-2DF Performance Outline*

requiring a sample to be sent at certain intervals, regardless of the compressor, and restarting the computation. Even with this relief, the storage requirement rapidly becomes impractical. In addition, the computations associated with the FOI-4DF method are lengthy, and not only vary with the number of redundant points, but also are dependent upon the shape of the signal being compressed. Because of these difficulties, the FOI-4DF interpolation method was not considered as a serious candidate for implementation.

#### 4.2.1.4 Linear Predictor Aperture Method

This method operates in principle in a similar manner to the ZOP method, except that the prediction of the next data point is made by a linear combination of N previous points according to the relation:

$$y_t = \sum_{n=1}^{N} a_n y_{t-n} .$$

For N = 1, and $a_1$ approximately equal to unity, this is the exact equivalent of the ZOP technique. The coefficients used in the prediction are functions of the statistics of the signal, and are constant only for stationary signals.

For a small number of fixed coefficients, the implementation of the linear predictor is not difficult, and can be accomplished with N + 2 multiplications and additions in a digital processor. The results obtained for this method were consistently poorer than either the ZOP or the FOI-2DF techniques, and these results deteriorated as more terms were used in the prediction. Also, for optimum results the prediction coefficients should vary adaptively as the autocorrelation function of the signal changes. This variation requires computational capacity far beyond that used in the compression calculations.

#### 4.2.2.1 Reconstruction Interpolation Techniques

It was noted early in the study that a large proportion of the data signals were sampled at a rate considerably in excess of the minimum indicated by the

calculation of their power spectra. Therefore, the possibility of reducing the data sampling rate by improving the interpolation technique during the data reconstruction was investigated. By this approach, the implementation penalty in the spacecraft is held to a minimum, because no additional hardware is required; the data system is designed to provide close to Nyquist-rate sampling on all analog channels, which in some of the cases studied reduces the sampling rate by a much as a factor of ten. This reduction in sampling rate requires more sophistication in the data interpolation performed on the ground. Several interpolation methods were investigated, and are listed below.

a. Optimum linear interpolation—8 coefficients
b. Straight-line interpolation
c. Lagrange interpolation—8 coefficients
d. Sin x/x interpolation
c. Fourier filtering

The chief drawback in the fixed sampling rate methods is that, unlike the aperture techniques, the peak error in the recovered data is not limited, because the interpolation algorithms are derived by minimizing the rms error. The test results show that although the rms error may be within acceptable levels, the peak error can be quite large (see Figure 3-13). This lack of control of peak error is a serious shortcoming because the experimenter can never know what confidence to place in his data, and it is probably a sufficient argument for eliminating the fixed-rate compression in spite of its hardware advantages. In addition to the peak error problem, the overall performance curves of the fixed-rate compression methods are consistently poorer than those of the simpler aperture techniques for all cases tested.

#### 4.2.2.2 Linear Prediction With Difference Coding

A very different approach to fixed sample rate compression is represented by the linear prediction and difference coding technique. In this method, a linear combination of points is used to predict the next data point. The difference between the predicted value and the actual value is coded and transmitted.

Although this method permits exact reconstruction of the data on the ground, the coding used to represent the successive differences requires variable length words. This variation in word length introduces serious problems into the PCM transmission and ground recovery equipment, which operate on a fixed word length basis. Although the computations per point are not excessive, the problem of handling the resulting random-length word data stream was considered sufficiently difficult to eliminate this method from practical consideration.

### 4.2.3 Variable Sampling Rate

The variable sample rate compression method, as it was conceived and tested in this program, is in effect a combination of the aperture techniques and the fixed sample rate techniques. The data sample rate is switched between two submultiples of the basic rate in response to a simple measure of data activity implemented by observing when the difference between successive points exceeds a given aperture.

The effectiveness of the variable sampling rate is, like that of the fixed rate, strongly dependent upon the accuracy of the interpolation used on the ground. Two methods were tested, one using optimum linear interpolation with four coefficients and one using straight-line interpolation. The optimum linear method gave results measured against rms error that were quite good, however, its peak error performance was poor. The straight-line interpolation gave comparable rms error performance, but the peak error was very poor.

The implementation of the variable sample rate method is not different in concept from that of the aperture methods. A digital processor is used to test each sample against the algorithm and determine whether or not it should be transmitted, and also whether the sample rate should be changed. The compressor output is random, and a buffer is required ahead of the transmitter. Sensor identification and current sampling rate must be transmitted along with each data sample. The general implementation complexity is roughly equivalent to that of the FOI-2DF, and while its rms error performance for a given compression ratio is only slightly worse, its poor peak error performance and its requirement for sophisticated ground interpolation equipment cannot be ignored.

### 4.2.4 Transformation Compressors

Data was obtained on two transformation compressors: the Karhunen-Loéve expansion and the Fourier transform. Both of these techniques are described in some detail in Section 2. The Karhunen-Loéve expansion is a highly effective compression technique in terms of performance, and some work was done during the study toward using Karhunen-Loéve coefficients as a measure of an upper limit of data compressibility. The Fourier filter was also a reasonably effective interpolation technique. Both methods, however, are only of academic interest as far as implementation is concerned, because the computations involved in determining the coefficients of the expansions are far beyond the capability of a spaceborne processor. For example, a 50-point Karhunen-Loéve expansion requires the inversion of a 50 x 50 matrix. Similarly, the computation of Fourier coefficients in real time on a multiplexed data stream is impractical.

### 4.2.5 Summary and Conclusions

This section has presented the salient points both in favor of and against each of the methods tested. These points are summarized in Table 4-1 where the algorithms are compared according to three general criteria: effectiveness of compression, implementation cost, and impact on the data system design. From this presentation, there are two techniques, the ZOP and the FOI-2DF, which have favorable comments in all three categories. All of the other methods have a disadvantage of more or less seriousness in at least one area. Because the ZOP and the FOI-2DF tend to complement each other in performance, implementation should at least have the capability of executing either of these algorithms on selected data channels. If a compression system is designed to use these two algorithms, then it may be desirable to extend its capability to include other algorithms of comparable complexity to increase the ability of the system to meet the requirements of a data system having a wide variety of sensors. This extra capability would also permit critical evaluations to be made in a real system environment.

This section has discussed the comparative aspects of several data compressors. Section 4.3 will consider in some detail the problems associated with the

Table 4-1. Summary of Tradeoff Factors

| Algorithm | Performance | Implementation | Impact on Data System |
|---|---|---|---|
| ZOP | Good on all data types | Moderate | Small - ground reconstruction is straightforward |
| FOI-2DF | Good, particularly for large apertures | Moderate | Small |
| FOI-4DF | Good | Requires large and variable number of calculations | Small |
| Lin. Pr. Ap. | Poor | Moderate (if coefficients are constant) | Small |
| Interpolation Fixed Sample | Large peak errors | Minimal | Complex ground interpolation |
| Lin. Pr. Diff. coding | Zero error | Coding implementation is difficult | Variable-word length recovery problem |
| Variable Sample | Large peak errors | Moderate | Complex ground interpolation |
| K-L | Very good | Impractical for spacecraft | Complex recovery computation |
| Fourier | Good | Impractical for spacecraft | Complex recovery computation |

second major subsystem—that of adaptive buffering of a compressed, multiplexed data stream. Section 4.3 will describe briefly an approach to the design of a complete adaptive data compression system having considerable flexibility and including adaptive buffer control.

## 4.3 BUFFER CONSIDERATIONS

### 4.3.1 Introduction

Design of the output buffer is one of the most important tasks to be faced in implementing ACT. Upon proper design of the buffer, including such parameters as size, input-output data rates, and occupancy control, rests the overall compression efficiency and error performance of ACT.

The buffer permits the efficient merging of nonredundant samples from several sensor channels into one constant rate data stream for transmission. The samples from different sensors could have been operated upon by different compression methods.

The problems of efficient buffer design are manifold. Even for a single stationary compressed sample stream, the variance in individual runs of deleted (redundant) samples produces problems of overflow and emptying of the buffer.

Overflow causes data samples to be lost, all the more undesirable because the redundancy of the data has been reduced. Underflow, or buffer emptying, leaves information gaps in the transmitted signal that could have been used to improve received error performance by either lengthening the transmitted sample period, or reducing the compression ratio, or both.

It is desirable, therefore, that the buffer be designed to neither overflow nor underflow. This, as will be shown, is difficult to achieve without using a buffer with greater capacity than necessary unless a system using adaptive control is considered.

For most of the compression methods considered, including the two methods which yielded the best overall results (ZOP and FOI-2DF), the compressor operation time per sample point is constant, or very nearly constant. Because the input to the compressor consists of a synchronous multiplexed stream of sampled data,

and the buffer output is also synchronous for transmission purposes, the buffer implementation would be synchronous. Thus, the queueing analysis required for efficient buffer design is based upon the binomial distribution rather than the usual queue model Poisson distribution.

The details of the analysis, including the derivation of equations for the probabilities of overflow and no readout, the required buffer length, expected fill and variance as functions of average compression ratio $\phi$, and input-output transmission rates C are presented in Appendix C. Because most of the derived relationships are recursive, digital computer solutions were necessary. In this section we will use the results of the buffer analysis to outline methods for efficient buffer design for ACT.

### 4.3.2 Computer Solutions of Buffer Equations

There are three independent buffer parameters which must be specified in a buffer design. They are: $\phi$, the average compression ratio; C, the input-output transmission ratio; and R, the probability of overflow. As shown in Appendix C, the ratio $C/\phi$ is usually called $\rho$, the buffer activity factor, and must be equal to or less than unity for equilibrium to occur. That is, if $\rho > 1$, the buffer will have on the average more incoming data than outgoing data and will eventually overflow with the resultant loss of data.

For the Poisson distribution model of input data, which is a reasonable model for a high-speed asynchronous buffer, the dependent parameters such as average buffer fill, and required buffer length for a specified overflow probability R are a function of $\rho$. However, for the synchronous buffer with binomial input distribution, the parameters are a function of $\phi$ and C; thus, the design parameters cannot be expressed solely as a function of $\rho = C/\phi$.

Note that $\phi$, the average compression ratio, when applied to the buffer analysis, is the average ratio: $\frac{\Sigma \text{ data bits into ACT}}{\Sigma \text{ data bits out of ACT}}$ for all multiplexed sensor data. The simple relationship between individual sensor compression ratios and the average compression ratio needed for buffer design is derived in Appendix C.

It is assumed that all sensor data is statistically regular; i.e., we assume that short duration sensor compression ratios do not deviate greatly from the long

term average compression ratios. If this is not the case, detailed transient buffer studies must be undertaken. This would probably be most tractable with computer simulation. However, even for the stationary case, it will be shown that a transient analysis is desirable to aid in designing an adaptive controlled buffer.

Results of the computer solutions to the buffer equations are shown in Figures 4-4 through 4-7. Figure 4-4 shows the relationship between buffer length -L, average buffer fill -E(n), and average percent fill -E(n)/L as functions of the average compression ratio $\phi$ for several values of R; the probability that an input event will overflow the buffer.

The results agree with intuition in that as $\rho = C/\phi \rightarrow 1$, the required buffer length for a particular R increases rapidly. The expected buffer fill E(n) behaves, as would be expected, similarly to L. Note that as $\rho \rightarrow 1$, the percent buffer fill E(n)/L increases rapidly. However, even for $\rho \approx 1$, E(n)/L is quite small for R = 0.0001. Thus, we have our first indication that restricting overflow by increasing buffer size is relatively inefficient.

Figure 4-5 presents buffer length L as a function of C, the input-output transmission ratio. Here again, the extreme sensitivity of buffer size to $\rho$ as $\rho \rightarrow 1$ is apparent. Figures 4-6 and 4-7 present the probability of overflow R, and the probability of no readout $1 - \rho(1 - R)$, which is discussed in detail in Appendix C, as functions of buffer size and average compression ratio respectively.

### 4.3.3 Adaptive Buffer Control

The point of interest in Figures 4-6 and 4-7 is the difference between the probabilities of no readout and overflow. The two approach equality as $\rho \rightarrow 1$.

Thus, for maximum efficiency in both buffer utilization and compression (the larger C can be made, the greater the transmission bandwidth reduction) the buffer should be designed to operate with $\rho \approx 1$. However, as shown in Figure 4-4, this would require a large capacity buffer ($L > 100$ for $\phi > 4$, with R = 0.0001). The probability of overflow can never be made exactly zero for all sensor data no matter how large L is made (it can be made arbitrarily small, but the required buffer size increases rapidly); therefore, the following system is recommended.

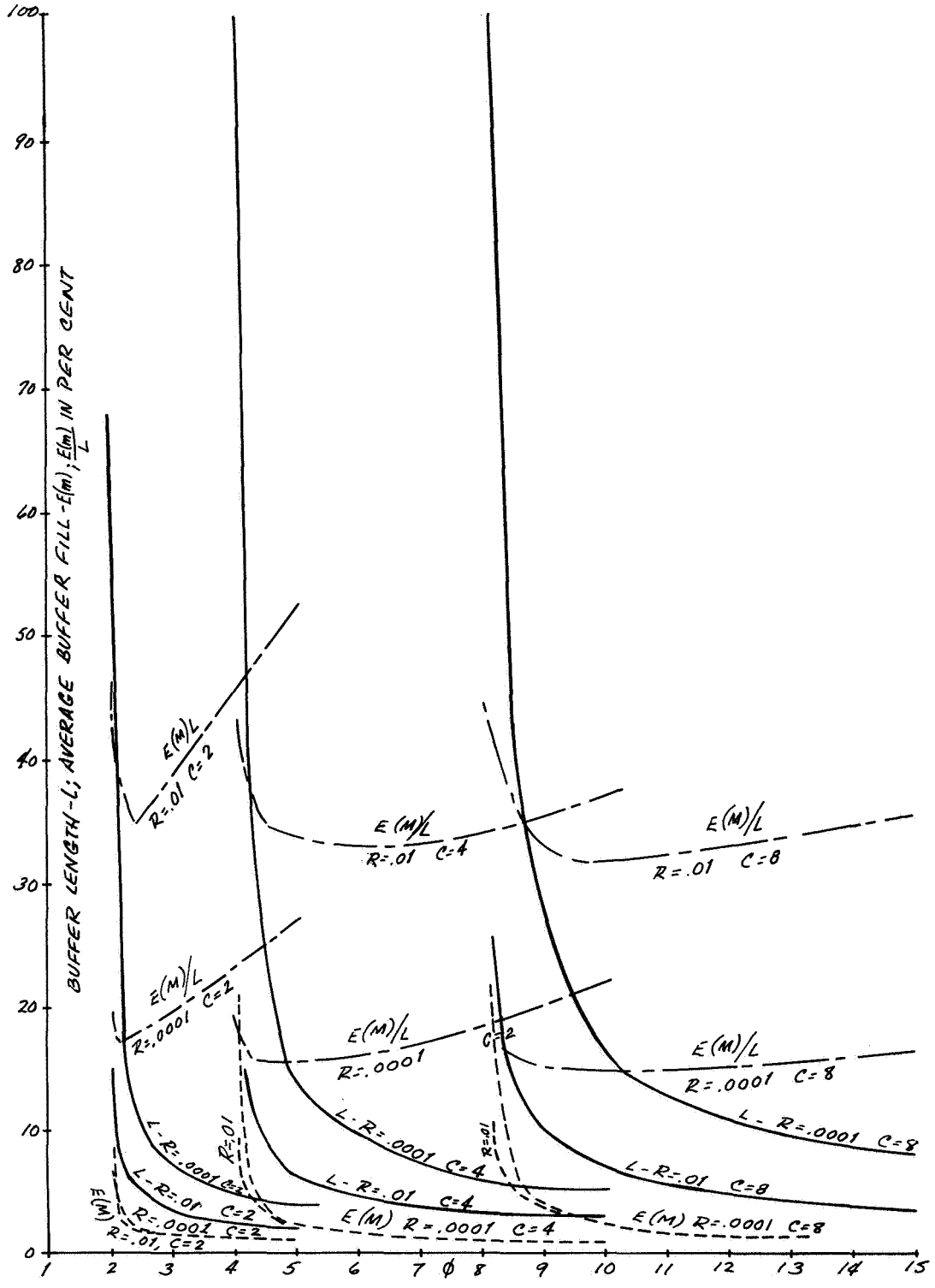


*Figure 4-4. Buffer Length-L, Average Buffer Fill-E(n), and E(n)/L vs Compression Ratio φ for Constant Probability of Overflow-R*

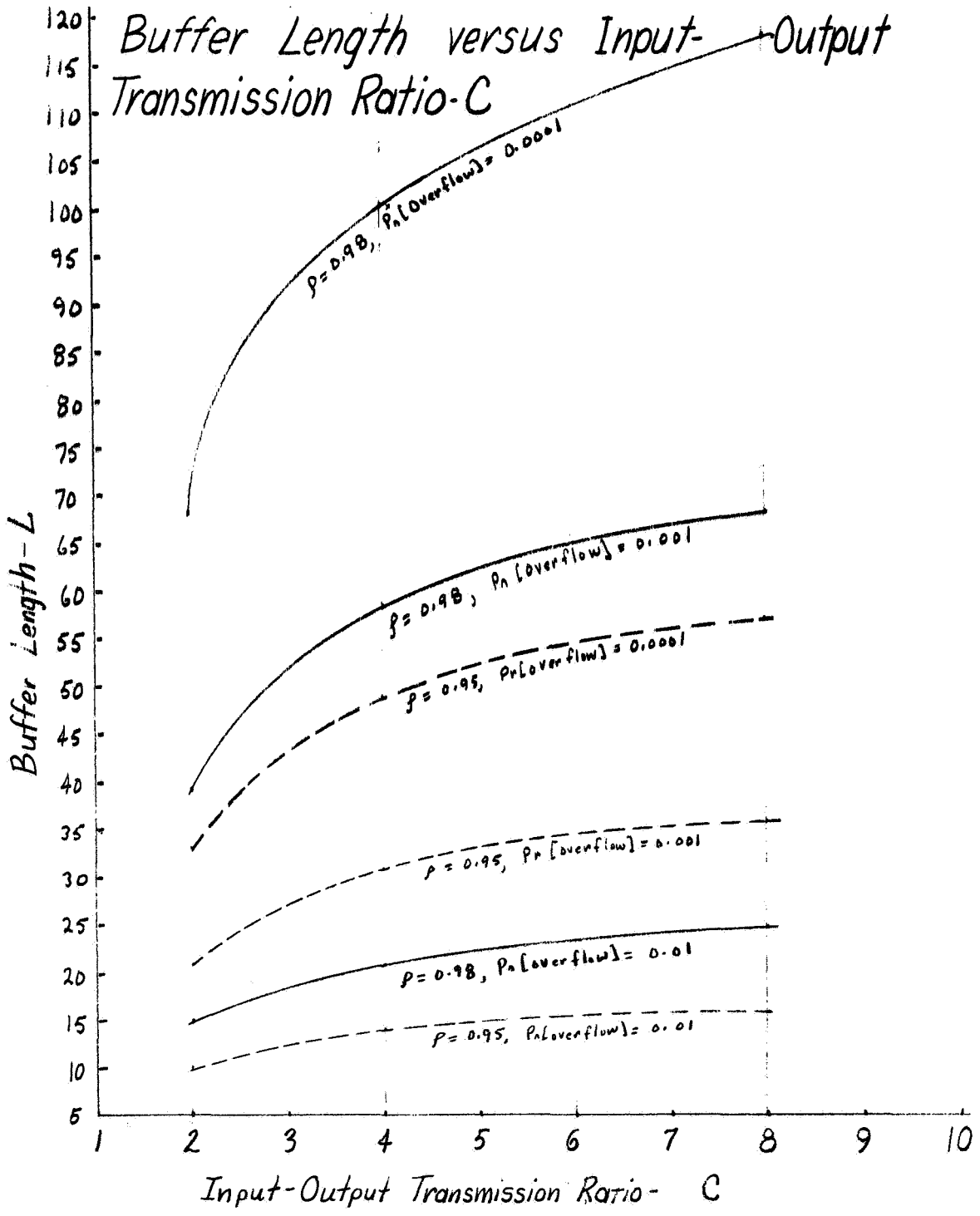


*Figure 4-5. Buffer Length vs Input-Output Transmission Ratio-C*

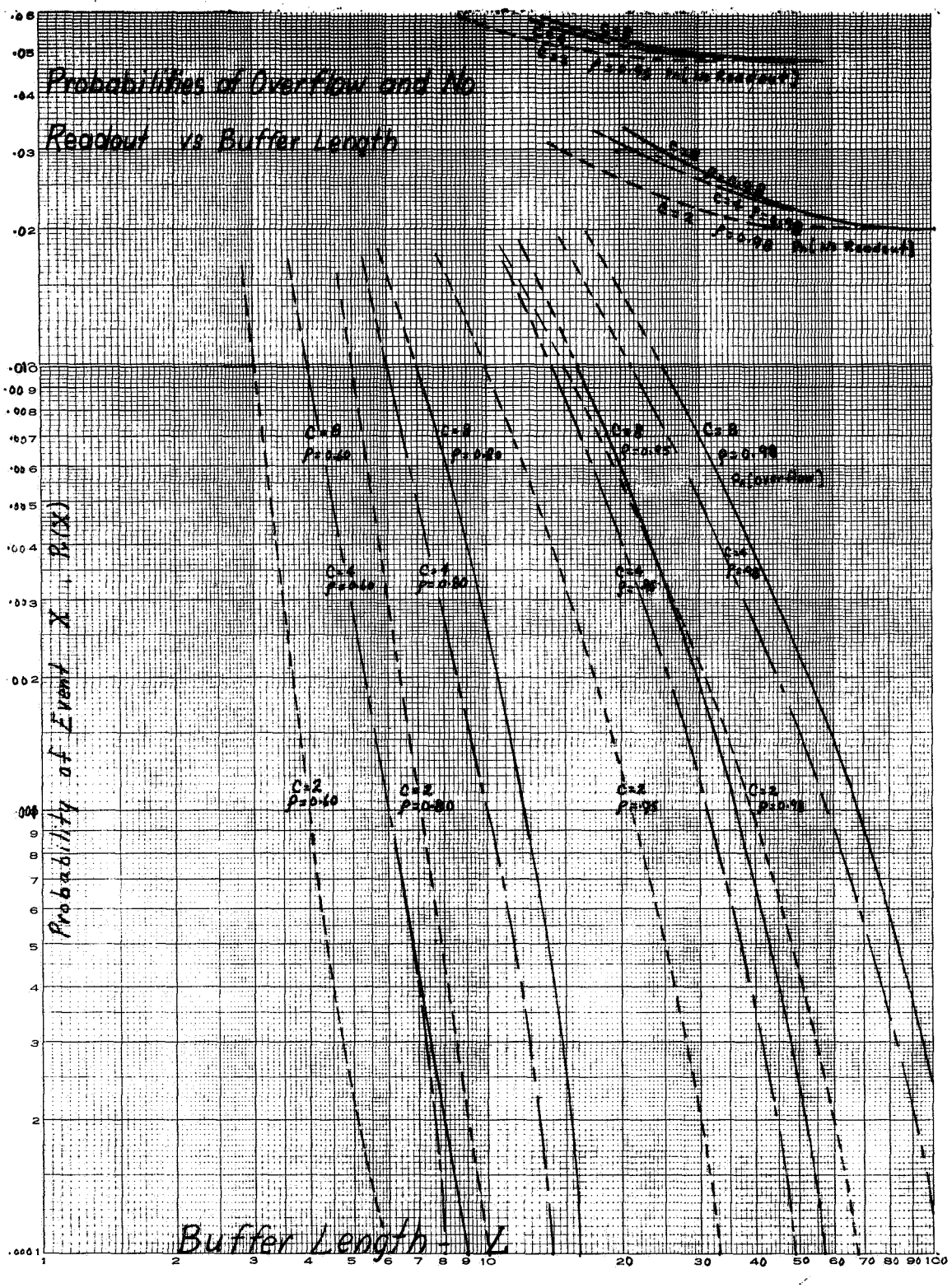


*Figure 4-6. Probabilities of Overflow and No Readout vs Buffer Length*

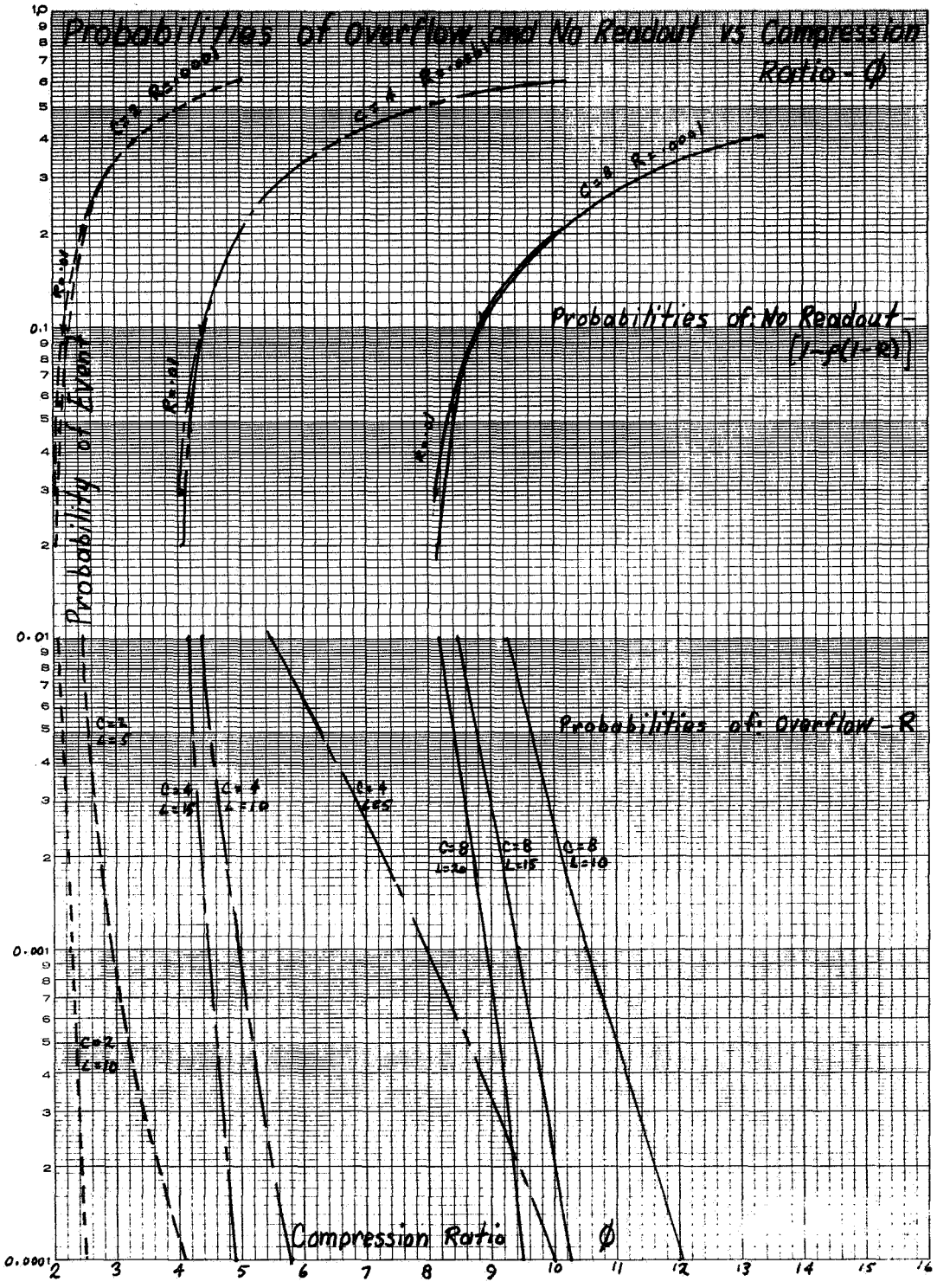


*Figure 4-7. Probabilities of Overflow and No Readout vs Compression Ratio $\phi$*

The buffer should be designed with a fairly good estimate of the average compression ratio $\phi$ that will be achieved for all the sensor data (the composite $\phi$ is discussed in Appendix C). Any data that is considered too important to risk a small but finite probability of loss due to buffer overflow should be designated priority sensor data. The total bandwidth of the uncompressed priority sensor channels should be no greater than the total transmission bandwidth. Thus, the low priority data is compressed and transmitted in the transmission bandwidth obtained by compression of the priority sensor data. When buffer overflow becomes imminent, the low priority data is heavily compressed, or excluded from transmission altogether. Thus, priority data is always guaranteed transmission bandwidth. The statistics of such a priority system are considered in more detail in Appendix C.

Assuming the system average compression ratio is known (computer simulation may be necessary on typical sensor data), the input-output transmission ratio C should be chosen such that $\rho = C/\phi \approx 0.98$. This will provide the probability of overflow approximately equal to the probability of no readout in the vicinity of R = 0.01, and requires a buffer size less than 1/4 of that required for R = 0.0001 (see Figure 4-5).

Overflow and no readout probabilities of approximately 0.01 would probably not be acceptable for most applications. Therefore, an adaptive control is required to reduce R without increasing the buffer size. Figure 4-7 is of assistance in estimating the change in $\phi$ required to reduce R to a desirable range for a fixed L.

Consider an average compression ratio $\phi \cong 8$, such that with C = 8, $\rho = 0.98$. From Figure 4-5, L is found to be 25. Using the curve for L = 20, C = 8 in Figure 4-7, we find that $\phi$ must be increased from approximately 8.2 by 9.5 to reduce R from 0.01 to 0.0001; thus $\phi$ must be increased by a factor of 1.16. A buffer load register such as an up-down counter, could be used to trigger a change in compression ratio when the buffer appeared in danger of overflowing or becoming empty.

Appendix C shows that for all five types of data tested, with the aperture compression techniques simulated, an increase in aperture by a factor of two

yields a new compression ratio $\phi_2$ which lies in the range: $1.2\,\phi_1 \leq \phi_2 \leq 2.3\,\phi_1$. Thus, for our example, a doubling of the apertures in the compression algorithms would more than provide the required increase in compression to decrease R from 0.01 to 0.0001. If a priority system was involved, the nonpriority data compression would have to be increased by more than a factor of 1.16 to offset the constant compression probably required for the priority data.

The previous example considered the buffer control problem from a quasi-stationary standpoint; i.e., the time required to reach equilibrium after the change in $\phi$ was neglected. Whereas, in fact, the rate of change would be of utmost importance to determine the probability of data loss before equilibrium is attained. As stressed in Appendix C, the buffer equations derived (from which Figures 4-4 through 4-7 evolved) are valid for the system in equilibrium.

It will be necessary to analyze the transient conditions before a detailed buffer control design can be undertaken.

## 4.4 SYSTEM DESCRIPTION

### 4.4.1 Introduction

The algorithm performance study of Section 3 and in the preceding trade-off analysis show that a compression system capable of performing both the FOI-2DF and the ZOP compression methods on any selected combination of data channels would be a highly effective compressor. In addition, it is desirable, particularly in the evaluation phase of system development, to incorporate the maximum flexibility that can economically be implemented. This section describes a design approach to such a data compression system which uses the concepts of microprogramming to achieve the capability of executing both of the recommended algorithms, as well as being able to be reprogrammed for other algorithms of comparable complexity. The particular algorithm to be applied to each data channel is selectable by ground-station command. In addition, flexible control over the tolerances used in the compression calculations is provided to control the data rate out of the compressor and maintain the output buffer occupancy at a

nominal level. To place these ideas in their proper context, the discussion of the compression subsystem will be preceded by a brief description at a general level of a typical complete data compression system from data source to transmitter.

### 4.4.2 General Data Compression System

A broad-level block diagram of a typical data compression system is shown in Figure 4-8. Data is derived from both analog and digital sensors and combined by a digital multiplexer at sampling rates appropriate to the individual data sources by super and subcommutation. The output of the digital multiplexer is a serial-by-word data stream conforming to a predetermined format. Because the compression subsystem must be able to identify the source of any given data sample, sufficient information must be available to permit this determination to be made. The necessary identity information can be derived from the data stream by a frame sync recognition and word counting process similar to that performed at the ground receiving station, or it may be made directly available by the multiplexer. In either case, each data sample appearing at the input to the compressor is accompanied by a sensor identification word.

Upon receipt of an input sample, the compressor uses the sensor identification word to address its storage. This storage carries, for each sensor, all the reference data (previous points, slope limits, tolerances, etc.) required to perform the necessary calculations, upon the basis of which the sample is either transmitted or eliminated. After the calculation, the new reference data is stored and the nonredundant sample is sent to the buffer storage, accompanied by sensor identification and any other data that may be required, such as tolerance limits, compression algorithm used, etc. The buffer storage is provided as a rate buffer between the random rate output of the compressor and the fixed rate of the transmitting subsystem. To prevent either data loss from buffer overflow during periods of high data activity or inefficient use of the link resulting from buffer emptying during quiescent periods, a feedback control path is provided from the buffer to the compressor. This control path varies the tolerance limits applied in the compression process according to a fixed priority in response to detection of the level of buffer occupancy. This variation of tolerance limits adjusts the

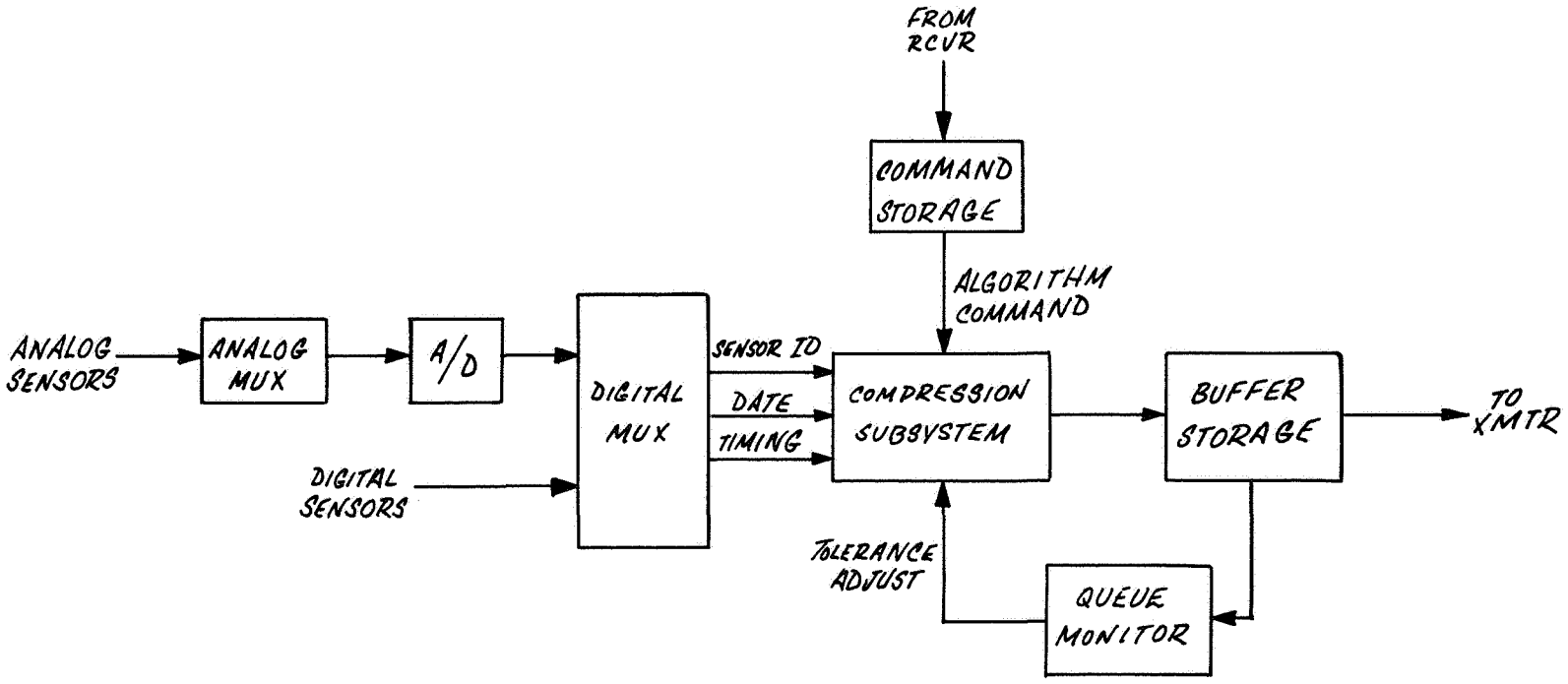


*Figure 4-8. Block Diagram—Data Compression System*

average output rate of the compressor to match the transmission rate. The foregoing discussion lays the groundwork for a detailed description of the design of the compression subsystem.

### 4.4.3 Compression Subsystem

The essential design requirements which must be met by the compression subsystem are summarized below:

a. Accept a fixed-rate, commutated data stream in digital form with associated sensor identification data and word timing.
b. Perform upon each channel the appropriate compression algorithm.
c. Transmit nonredundant data to a buffer storage for transmission.
d. Accept and implement ground commands as to the compression algorithm to be applied to each channel. This includes the capability to command that selected channels be transmitted without compression or eliminated entirely.
e. Respond to a control signal which is a function of buffer occupancy by adjusting the tolerance limits applied in the compression calculation to control the data rate out of the compressor.

#### 4.4.3.1 Compression Algorithms

The evaluation of the effectiveness of various compression algorithms given in this section has shown that the ZOP and FOI-2DF methods of compression are both effective in redundancy removal, in addition to which they both require only relatively simple calculations. Figures 4-1 and 4-2 show detailed flow diagrams in which the computations are expressed entirely in terms of addition, subtraction, division and comparison with zero. Section 4.4.3.2 describes the development of a central processing unit that will perform the necessary arithmetic operations, and shows how this central processor can be incorporated into a compression subsystem which fills all the requirements listed above.

#### 4.4.3.2 Central Processing Unit (CPU)

In discussing the implementation of the central processing unit, the FOI-2DF algorithm will be used as a specific example wherever necessary; it will become evident that the implementation of the ZOP will present no additional problems.

The general philosophy in developing the CPU is to provide a general-purpose arithmetic unit which performs all the necessary calculations on a sequential basis, and to develop a programming subsystem which performs the necessary routing and sequencing functions.

##### 4.4.3.2.1 Arithmetic Unit

Figures 4-1 and 4-2 show that the FOI-2DF and ZOP algorithms, and indeed, most other algorithms, can be performed by the operations of addition, subtraction, division, and comparison with zero (combined with a subtraction, this permits comparison of any two numbers). An arithmetic unit is required, therefore, which will add, subtract, or divide a given pair of numbers upon command and present the results in bi-polar form using a sign bit in the answer to indicate polarity. Because the detailed design of such a device is straightforward, it will not be discussed here; the Arithmetic Unit will be considered as a functional building block in the development of the CPU. The relationship of the Arithmetic Unit to the other elements in the CPU is shown in Figure 4-9. Upon receipt of a signal upon one of the three input command lines, it forms the sum, difference, or quotient of the contents of registers A and B, and stores the result in Register $R_1$.

##### 4.4.3.2.2 Storage Registers and Selector Gating

Figure 4-4 shows the configuration of the CPU. The reference data for a particular calculation (shown for the FOI-2DF algorithm) are brought from storage to a set of registers, along with the new data point. Registers A and B are the input registers to the Arithmetic Unit, Register $R_1$ is for the Arithmetic Unit result, and Registers $R_2$, $R_3$, and $R_4$ are for temporary storage of intermediate results.

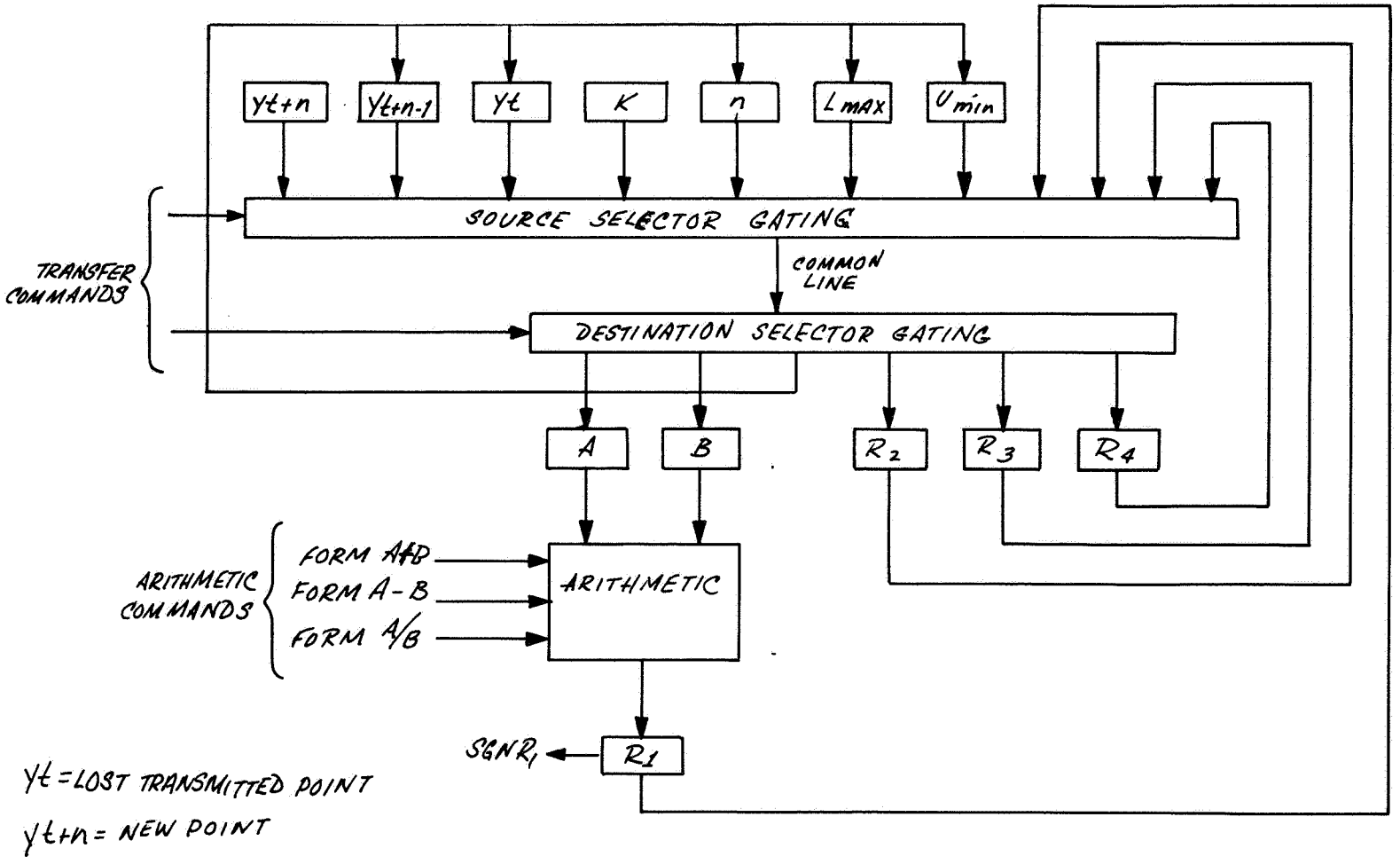


*Figure 4-9. Block Diagram—Central Processing Unit*

The actual execution of the algorithm consists of sequentially executing the instructions shown in Figure 4-5. To do this, each instruction is broken down into elementary commands. For example, the instruction: Subtract $Y_{t+n} - Y_t = Y$ would be executed by the following sequence of commands:

a. Transfer $Y_{t+n}$ to A

b. Transfer $Y_t$ to B

c. Subtract

This sequence leaves the result, $\Delta Y$, in $R_1$. The actual transfer operations are accomplished by control signals which are applied to the source and destination gating. A transfer consists of gating the selected source register onto the common data line, and gating the data line into the selected destination register. Arithmetic operations are accomplished by placing a signal on the appropriate input to the Arithmetic Unit. The actual sequence of commands which performs a complete algorithm calculation is generated in a Program Control subsystem.

4.4.3.3 Program Control Subsystem

A block diagram of the Program Control Subsystem is shown in Figure 4-10. It consists essentially of a Read-Only Storage (ROS), and ROS Address Generator, and a Command Decoder. The ROS contains as many words as there are possible commands to be executed by the CPU. The extraction of a particular ROS word causes the CPU to execute that particular command. The command words are read out of the ROS in the correct sequence by the ROS Address Generator, and decoded into actual control signals by the Command Decoder. A typical ROS word format is shown below.

| SOURCE REGISTER IDENTITY | DESTINATION IDENTITY | ARITHMETIC COMMAND | MEMORY ROS ADDRESS PER | OTHER COMMANDS |
|---|---|---|---|---|
| 4 bits | 4 bits | 2 bits | Sgn $R_1$ 2 bits | 3 bits |

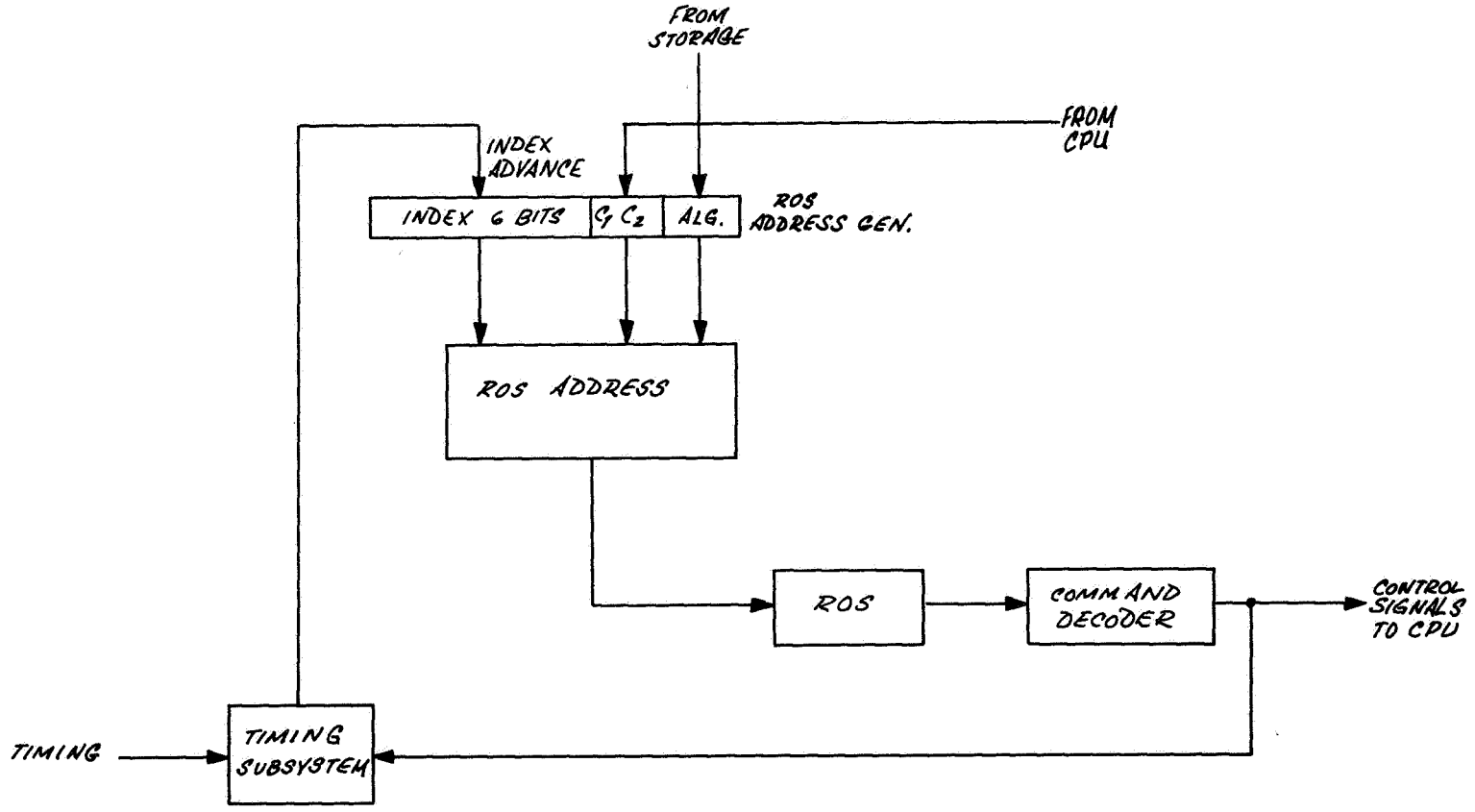


*Figure 4-10. Block Diagram—Program Control*

The command sequence required to execute a particular compression algorithm can be written from the detailed flow diagram. A section of a typical sequence, complete with ROS addresses, is given in Table 4-2. Each command in the sequence is identified by the address, which is made up of several parts:

a. The algorithm being executed—In the system being considered there are four possibilities:
   1. FOI-2DF
   2. ZOP
   3. No compression
   4. Eliminate

   Two bits are required for this part of the address, which comes from the main storage.

b. The state of control bits, $C_1$ and $C_2$, which are used to record the result of branch point decisions in the program.

c. The actual step of the program being executed—This is simply an index register which advances one count after each command. The allocation of six bits to this section permits sequences having up to 64 commands. Thus, 10 bits are required for the ROS address in the system under consideration.

The sequence of events in a typical calculation is initiated by a timing signal from the external system. This signal indicates that the required data has been placed in the registers, and the part of the ROS address which identifies the algorithm to be executed has been supplied from main storage. The index register is set to step one, and the first command in the sequence is read out of the ROS. This command is executed, and after its completion the index register is advanced and the next command is read out and executed. This continues through the entire calculation, until the last command in the sequence, which is a STOP command, indicates that the calculation is complete.

It is evident from the foregoing description that this approach to system implementation offers considerable flexibility in the programs that can be executed. A single set of commands is developed which operate upon a simple CPU. The

Table 4-2. Program Sequence—FOI-2DF Algorithm (Sheet 1 of 2)

| ROS Adress Index | | | | | | $C_1C_2$ | Algorithm | | Command |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | | 0 | 1 | Transfer $Y_{t+n}$ to A |
| 0 | 0 | 0 | 0 | 1 | 0 | | 0 | 1 | Transfer $Y_t$ to B |
| 0 | 0 | 0 | 0 | 1 | 1 | | 0 | 1 | Subtract ($R_1$ = y) |
| 0 | 0 | 0 | 1 | 0 | 0 | | 0 | 1 | Transfer $R_1$ to A |
| 0 | 0 | 0 | 1 | 0 | 1 | | 0 | 1 | Transfer n to B |
| 0 | 0 | 0 | 1 | 1 | 0 | | 0 | 1 | Divide ($R_1$ = S) |
| 0 | 0 | 0 | 1 | 1 | 1 | | 0 | 1 | Transfer $R_1$ to A and $R_2$ |
| 0 | 0 | 1 | 0 | 0 | 0 | | 0 | 1 | Transfer $L_{max}$ to B |
| 0 | 0 | 1 | 0 | 0 | 1 | | 0 | 1 | Subtract ($R_1$ = $S_1$) |
| 0 | 0 | 1 | 0 | 1 | 0 | | 0 | 1 | Set $C_1$ per sign $R_1$ |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | Do not decode |
| 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | Transfer $U_{min}$ to A |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | Do not decode |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | Transfer $R_2$ to B |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | Do not decode |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | Subtract ($R_1$ = $S_2$) |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | Do not decode |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | Set $C_2$ per sgn $R_1$ |

Table 4-2. Program Sequence—FOI-2DF Algorithm (Sheet 2 of 2)

Note: At this point in the program the state of $C_2$ defines which main branch of the flow diagram is to be followed, and $C_1$ is no longer needed. Two complete sets of instructions are needed from this point on, depending on $C_2$.

| ROS Address Index | $C_1C_2$ | Algorithm | Command |
|---|---|---|---|
| 0 0 1 1 1 1 | | 0 0 1 | Transmit $y_{t+n}$ |
| 0 0 1 1 1 1 | | 1 0 1 | Transfer k to A |
| 0 1 0 0 0 0 | | 0 0 1 | Transfer $y_{t+n}$ to A |
| 0 1 0 0 0 0 | | 1 0 1 | Transfer n to B |
| 0 1 0 0 0 1 | | 0 0 1 | Transfer $y_{t+n-1}$ to B |
| 0 1 0 0 0 1 | | 1 0 1 | Divide ($R_1$ = ) |
| 0 1 0 0 1 0 | | 0 0 1 | Subtract ($R_1$ = S*) |
| | | etc., to end of program | |

sequence in which the commands are executed determines the nature of the calculation performed, and this sequence is determined by the ROS and its Address Generator.

#### 4.4.3.4 Compression Subsystem Description

A functional diagram of the compression subsystem is shown in Figure 4-11. The principal features yet to be discussed are the Main Storage and its characteristics, the incorporation of the ground command subsystem, the Buffer Status control path, and the overall system operation.

##### 4.4.3.4.1 Main Storage

The Main Storage contains for each sensor a data word which includes all required previous data, tolerances to be used under different buffer conditions, number of samples since the last transmitted sample, and compression algorithm presently being used for that sensor. The storage is addressed by the sensor identification word received from the external system. A typical word format for a FOI-2DF algorithm is described below. All the required data is contained in a 55-bit word.

| | | | |
|---|---|---|---|
| Bits | 1-8 | $Y_t$ | last transmitted sample |
| Bits | 9-16 | $Y_{t+n-1}$ | previous sample |
| Bits | 17-24 | $L_{max}$ | lower slope limit |
| Bits | 25-32 | $U_{min}$ | upper slope limit |
| Bits | 33-35 | $K_1$ | tolerance limit |
| Bits | 36-38 | $K_2$ | tolerance limit |
| Bits | 39-41 | $K_3$ | tolerance limit |
| Bits | 42-44 | $K_4$ | tolerance limit |
| Bits | 45-47 | $K_5$ | tolerance limit |
| Bits | 48-53 | n | number of samples since $Y_t$ |
| Bits | 54, 55 | Algorithm identification | |

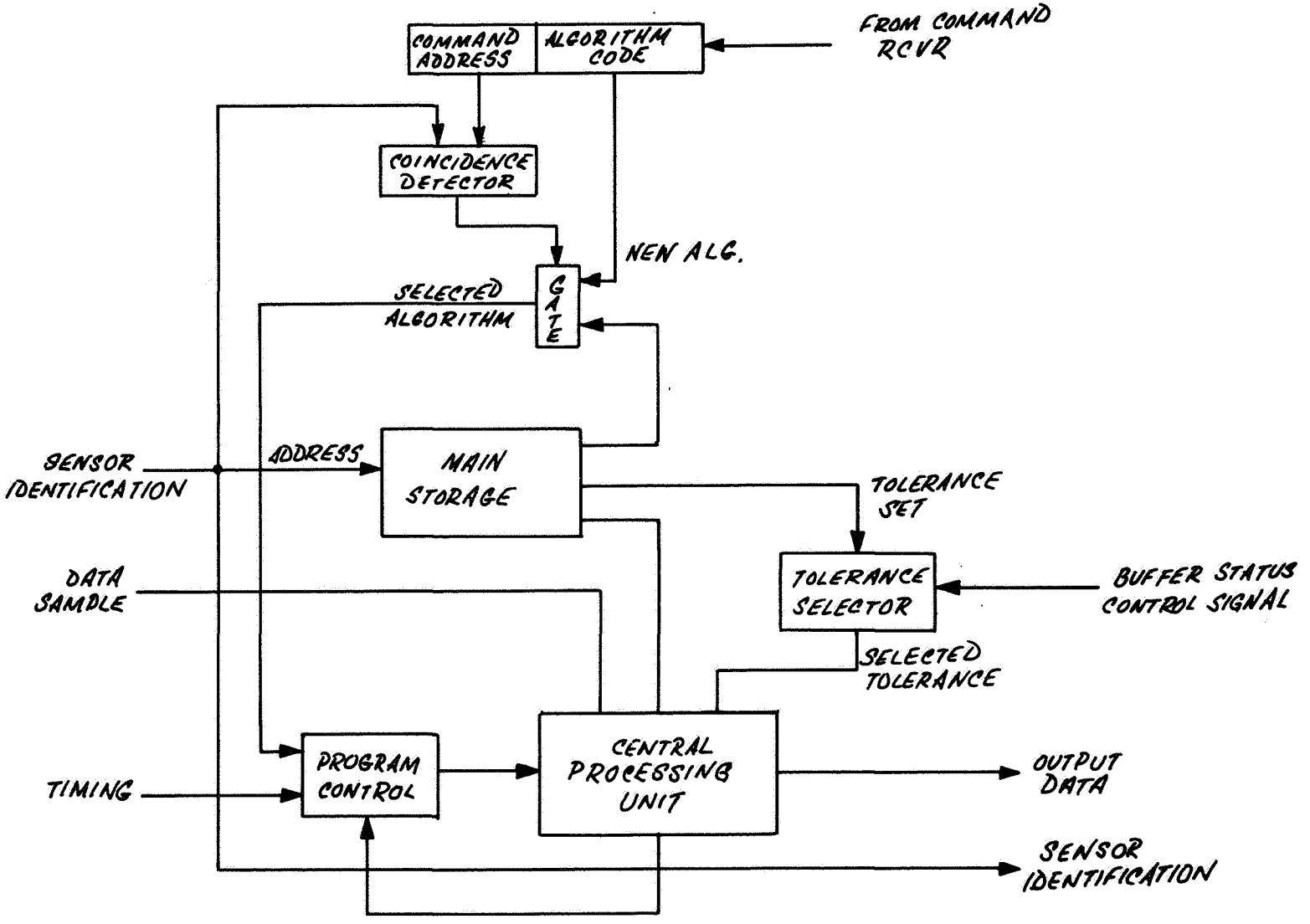


*Figure 4-11. Block Diagram--Compression Subsystem*

#### 4.4.3.4.2 Adaptive Control

An essential feature of an adaptive compression system such as the one being described is the ability to regulate the rate of data flow through the compressor in response to variations in data activity as reflected in the occupancy of the queueing buffer which stores the data for transmission. This regulation is accomplished in the proposed system by providing several different tolerance limits for each data source. During periods of data activity, the tolerances will be increased, thus reducing the data flow, while during quiescent periods a decrease in tolerances will increase the data flow. This tolerance selection is accomplished by a multi-level control signal derived from the buffer queue length monitor. This control signal is essentially a direct control over system compression ratio, and it acts by selecting one of the tolerance values stored for each channel for actual use in the CPU calculation.

#### 4.4.3.4.3 Ground Control

Provision is made for the ground monitoring station to select the compression algorithms to be used in each channel, and also to command that a particular channel either be transmitted without compression or eliminated entirely. The ground command is stored in the compressor along with its sensor address. Each input address to the main storage is compared with the command address, and when coincidence is detected, the commanded algorithm code is gated into the program control rather than the code in the storage. After the computation is complete, the new code is returned to storage, and the new algorithm is used until a change is commanded. Thus, the ground station has considerable control over the data compression system.

#### 4.4.3.4.4 Output Data Format and Sensor Identification

The actual data stream being transmitted to the ground must contain certain information that is not required in a system which has no compression. Because the regular format has been destroyed, sensor identification must be transmitted with each data sample. Because the tolerance level being used is variable, it

must also be transmitted. To permit accurate time placement of the data, it is desirable that the main frame sync pattern be transmitted each time it occurs, with appropriate identification. If real time is transmitted as a data sample at a very low subcommutated sampling rate, then the time occurrence of any sample can be computed by counting the number of main frames from the last time data word.

### 4.4.5 Summary

This section has described a hardware implementation of an adaptive data compression system capable of executing the ZOP and FOI-2DF algorithms upon a multiplexed data stream of the type normally encountered in PCM telemetry systems. The system is flexible in its ability to apply different algorithms to different channels, and in responding to data flow, increases or decreases as reflected in the buffer activity. A ground command may alter the algorithm applied to a particular channel, or may cause a channel to be transmitted without compression or eliminated entirely.

These features provide the system with the versatility that is needed in a compression system which is to be used with a variety of sensors and data types, and which can be reprogrammed to fit the requirements of different missions.

## 4.5 PROGRAM FOR IMPLEMENTATION OF RECOMMENDED SYSTEMS

### 4.5.1 Phasing

The program will be broadly divided into three phases:

Phase I—System Design and Subsystem Specification
Phase II—Hardware Design, Fabrication, and Checkout
Phase III—System Evaluation

Phase I will be concerned with establishing the requirements for the overall compression system—the input data characteristics, the sensor types and statistical properties, sampling rates and algorithms to be implemented; also

the transmitter characteristics, the output format design, and the specification of interfaces between the data compressor and the adjoining system elements. This system definition will lead naturally into the preparation of specifications for the compression subsystem, and for the buffer and adaptive control characteristics. The end result of Phase I will be a detailed set of specifications for hardware development.

Phase II will consist of the detailed design, fabrication, and checkout of a model of the adaptive data compressor. The end result of this effort will be a working model for verification of the concepts of adaptive compression. Also during Phase II, a detailed and comprehensive evaluation plan will be prepared for use in Phase III.

Phase III will consist of an evaluation program for testing the effectiveness of the compression system in a live-data system. Key parameters to be studied will include: error rate in the reconstructed data; overall compression ratio achievable; effects of transmission channel errors, optimum coding methods, transmission efficiency, etc. This phase will result in a final report which will include the final design of a spacecraft compression system.

### 4.5.2 Task Breakdown

This section will consider each of the phases described above in some detail, and will discuss the tasks which must be accomplished to support the overall objectives of the program.

#### 4.5.2.1 System Design and Subsystem Specification

This task is concerned with detailed technical specification of the system to be implemented, and the effort will be directed toward the achievement of three objectives.

a. Overall System Specification—The compression system must be designed around the characteristics of the spacecraft data system type to which it will be applied. The specification of the compressor will require a detailed study of the data source—the probable number and

type of sensors, multiplexing and formatting characteristics, the statistical parameters of the individual data channels, required accuracies, and compression algorithms to be applied. In addition, the compressor output format must be carefully specified to be compatible with existing or proposed data reconstruction facilities. The desirability of command links with either the astronaut or the ground to permit control of the compressor characteristics must be considered. The entire system design must be guided by the requirement for maximum generality in the resulting equipment.

b. Compression Subsystem Specification—Specification of the compression subsystem must include: main store characteristics, size, and cycle time; arithmetic unit requirements in terms of operation times, word lengths, and functions to be performed; local store and data flow characteristics; specification of the read-only store and its ancillary addressing and command decoding functions; microprogramming; and input/output interfaces.

c. Buffer Requirements and Adaptive Compressor Control Loop Definition—The effort supporting this task will interact strongly with the source data analysis undertaken in the overall system design. The system buffer requirements will be largely a function of the statistical characteristics of the data, as will the adaptive control loop parameters. The buffer and control analysis is a crucial item in the development of an effective compressor. The result of this task will be a buffer design and specification of the control to be applied by the compressor in terms of channel priorities, accuracies, etc.

#### 4.5.2.2 Hardware Model Development and Test

The objective of the hardware development program will be the construction of a laboratory model of a practical compressive telemetry system. Its functional operating characteristics will be stressed rather than its physical characteristics, however, where economically feasible, use will be made of components which will

be most easily adapted to a spacecraft environment. The system will be designed to accept "live" data and to operate in a functional environment which simulates as closely as possible that of an actual spacecraft data system. Its purpose will be verification of the feasibility of an adaptive compression system. The tasks supporting the main objective are outlined below.

#### 4.5.2.2.1 Compression Subsystem

The development effort for the compression subsystem can be divided into four parallel efforts which can be pursued somewhat independently.

a. Central processor design, fabrication, and test
b. Main store design, fabrication, and test
c. ROS and microprogram design, development, and test
d. Mechanical packaging, system integration, and testing

#### 4.5.2.2.2 Buffer and Control System, System Integration

The hardware development effort involved in the buffer and control element design is minimal compared to that required for the remainder of the system. This overall task also includes the design and development of all system elements which do not fall into the categories of compression computer, buffer, or control system. This will include any interface equipment for input/output compatibility, also command interface elements. Also included is the integration of the system components.

### 4.5.3 System Evaluation

During the fabrication of the hardware model, effort will be directed toward preparation of a comprehensive evaluation plan which will permit the effectiveness of such a compressor to be measured by operation upon "real" data in a realistically simulated operational environment.

The parameters for evaluation will include: compression ratio as a function of peak and rms errors in the reconstructed data, for various compression schemes; effectiveness to buffer and control in minimizing both data loss and blank transmission; optimum formatting schemes; and evaluation of the system effectiveness for various data types.

Section 5

# RECOMMENDATIONS FOR FUTURE WORK

This study demonstrated that no single compression method is "best" for all experimental data likely to be encountered in space telemetry systems. Two relatively simple aperture techniques (ZOP and FOI-2DF) provided the maximum compression of all the "practical" methods simulated for all data considered. Because neither of the two aperture methods has a lcear advantage, it is recommended that a full-scale working model using both compression algorithms be implemented with stored-logic circuitry as presented in Section 4.

The advantage accruing from a stored-logic system is the flexibility in applying different compression algorithms to different data on a time-division basis. In addition, other algorithms of similar complexity can be implemented by simply changing the stored-logic memory.

The output buffer is a very important unit in an adaptive compressive telemetry system. Equations were derived for a steady-state analysis of the buffer design parameter. Also, it was shown how a large reduction in required buffer capacity could be obtained by employing adaptive control of the compression algorithm aperture for nonpriority sensor data.

To complete the buffer design, a transient analysis is necessary to determine the necessary rate of change of aperture for specified buffer size. This information is also necessary to estimate the resultant increase in rms distortion of nonpriority data. The buffer study should be continued as part of an overall system implementation.

The simulation of the recommended aperture compression methods revealed that rms distortion was approximately a linear function of aperture for the data tested. However, this does not assure all data would behave in the same manner. In addition, one of the main limitations of the interpolation compression method

was the "unpredictable" peak errors in the reconstructed data. Therefore, a monitoring system should be studied for possible inclusion in an adaptive compression system. The stored-logic system concept would appear to provide the capability of sampled peak and/or rms distortion calculations without added complexity. The sampled distortion calculations could be done on a Monte Carlo basis to reduce the required number of samples, and the calculations could be used to control changes in compression algorithms or in aperture values.

The effect of transmission noise on the fidelity of the received data with and without compression was analyzed. This analysis was done for two compression methods: the Zero-Order Predictor, and First Order Interpolator—Two Degrees of Freedom. Expressions for the rms error in the reconstructed signal as a function of the bit-error probability were developed for errors in either the sensor word or data word. The noisy transmission channel should be simulated to determine the effect of noise on the reconstructed signal for various compression algorithms. The effect on compression ratio and reconstruction fidelity should also be studied when error correcting coding is used on the transmitted data.

# Appendix A

# ANALOG AND HYBRID IMPLEMENTATIONS

This appendix presents a discussion of a number of analog and hybrid techniques for implementation of compression algorithms. Paragraph A.1 discusses possible analog compression configurations in which the compressor is implemented in each sensor line, ahead of the multiplexer and digitizer, and mentions the problem areas in designing such a system. A description of a hybrid arithmetic unit which can replace part of a digital compressor for certain algorithms, is contained in A.2.

## A.1 ANALOG COMPRESSION TECHNIQUES

### A.1.1 System Aspects of Analog Compression

The main elements in a digital compressive data system are a multiplexer, a digitizer, a data compressor, and a data buffer and transmitter. The general configuration is shown in Figure A-1. The compressor in this system follows the digitizing unit, which is constrained to be common to a number of data channels by size and weight limitations, and thus must follow the multiplexer. An analog compression unit must, however, precede the digitizer and will, in general, be implemented in each channel separately (although there may be cases where the compressor could be common to a number of channels). The essential difference between analog and digital compression techniques is that while the digital approach seeks to eliminate data points already digitized and stored, the analog approach is to prevent an unnecessary or redundant sample from being stored at all. The decision process as to whether data is redundant or not is made ahead of the digitizer, by operating on an analog signal. The important aspects of an analog compression scheme are, therefore, clearly:

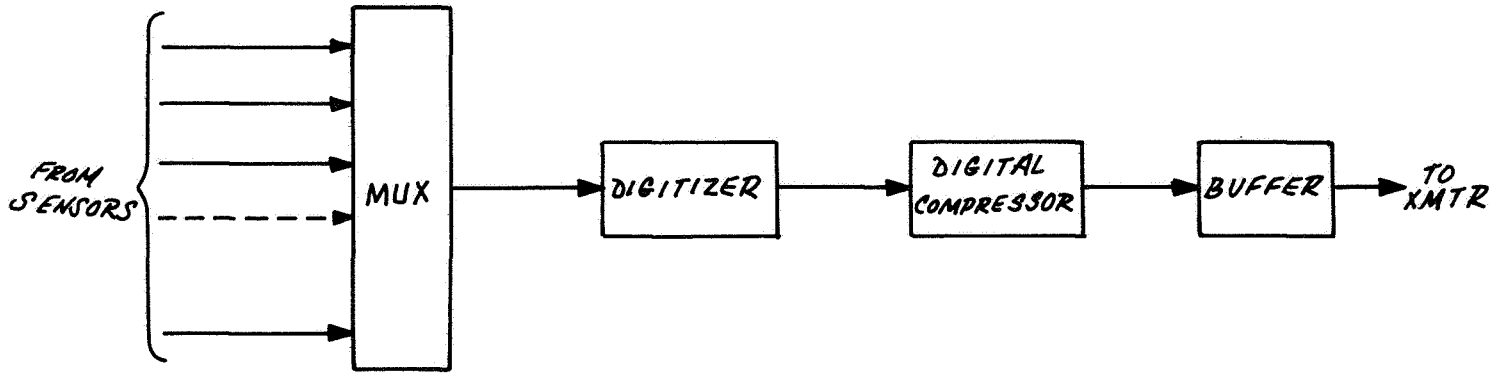


*Figure A-1. Digital Compression System*

1. The development of an analog decision-making process which will, upon quasi-instantaneous evaluation of a signal, determine whether or not a data point need be transmitted or, alternately, establish a sampling rate.

2. The development of a multiplexing and digitizing system that will be responsive to varying sampling rate requirements in each data channel. This will include provision for sensor identification and time data to be inserted as required in the transmitted data stream. These two aspects of the problem are discussed below.

### A.1.2 Possible Multiplexing Approaches

In an analog compression system where the compressor is implemented in each channel separately, the transmission rate for each channel is established by that channel's compressor, independently of the activity of the other channels. The multiplexer must, therefore, be capable of varying the sampling rate in each channel independently. There are two general ways of accomplishing this, depending primarily upon the technique of compression that is used. In either approach there must be a sample rate control signal of some type sent from each channel to the multiplexer. The system configuration is shown in Figure A-2. The two approaches differ in the nature of the sample rate control signal which is transmitted to the multiplexer. In one method, each channel may be sampled at any one of a discrete number of sample rates; each a multiple of the basic multiplexer rate. The sample rate control signal in each channel, which may be either analog or digital, is capable of taking on as many values as there are discrete sampling rates.

The multiplexer proceeds as follows. Each frame begins by sampling all channels calling for maximum-rate sampling. Every second frame continues by sampling all channels calling for half-rate sampling. Every fourth frame includes those channels calling for one-fourth rate sampling, and this process continues through as many frames as there are sampling rates. After the completion of the last frame, the cycle begins again. Every channel is scanned during every frame, but only those channels calling for the appropriate sampling

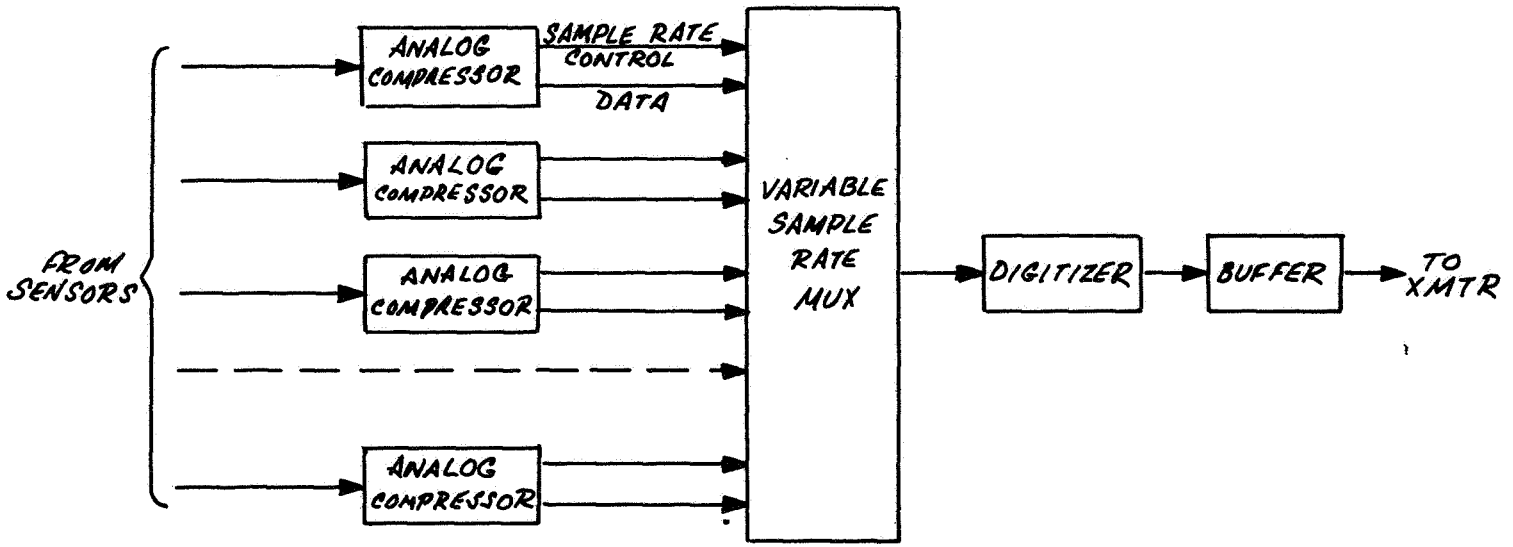


*Figure A-2. Analog Compression System*

rates are sampled during any particular frame. The second sampling method is used when the compression criterion is such that each channel simply indicates whether or not it should be sampled at the next opportunity. In this approach the multiplexer simply scans each channel and samples those which indicate through the sample rate control signal (in this case a binary indication) that non-redundant data is available. Both of these techniques result in a variable frame length, and it is understood that sensor identification and time data must be added to each message. An alternate approach to adapting the sample rate in the multiplexer proper is to allow the multiplexer and digitizer to operate at a fixed rate, thus sampling and converting every channel during every frame as would be the case in a non-compressive system. The sample rate control signal generated by the compressor in each channel would by-pass the multiplexer and operate directly on the pre-transmission buffer, either permitting or inhibiting buffer loading during that particular channel time. This approach may be simpler, however, a study of the hardware factors involved will be required before any decision can be reached.

### A.1.3 Analog Compression Criteria

The basic restriction that is imposed upon any analog compression technique is the same as that for digital approaches; namely, that any decision to either change sample rates or eliminate samples must be done such that the receiving terminal is aware of the change or can supply the deleted sample on the basis of data it already has. Since the receiver data is in discrete form, compressor redundancy determinations must be made using the same discrete values, hence a memory capacity of the sample-and-hold variety is required for sample elimination techniques. For a variable sample rate compression scheme, the current sample rate can be digitized and transmitted as part of the data word for each channel, or this information may be included only when a change in sample rate occurs.

### A.1.3.1 Variable Sample Rate Compression

In this compression technique, the sample rate is varied discretely in response to a measurement of the difference between the present value of the parameter and the last transmitted value. When the difference exceeds a threshold associated with the sample rate, the rate is increased and a larger threshold established. The analog implementation of this technique is shown in Figure A-3. Each time a sample is taken by the multiplexer, the sample-and-hold memory stores the transmitted value. This value is continuously subtracted from the current value, and the difference is compared with a threshold value which is determined by the present sample rate. The sample rate is increased whenever the threshold is exceeded, and is reduced when the threshold is not exceeded for two successive samples. The memory is cleared and re-sampled each time the channel is sampled by the multiplexer, therefore, the threshold computation is always made by comparing the present value with the last transmitted value. This compressor makes use of the first multiplexing approach, where the sample rate control signal is capable of taking on any of a discrete number of states, each one corresponding to a particular sampling rate.

### A.1.3.2 Linear Prediction and Difference Coding

Figure A-4 shows a possible analog implementation of the prediction and difference computation. Two memory stages are required to implement the prediction, which is simply a first-order extrapolation of the form:

$$y(t) = y(t-1) + \int_{t-1}^{t} [e(t-2) - e(t-1)]\, dt$$

More accurate predictions could be used, with corresponding increase in equipment complexity. This approach requires somewhat more circuitry that the variable sample approach or the aperture techniques, but it makes use of a fixed-rate multiplexer.

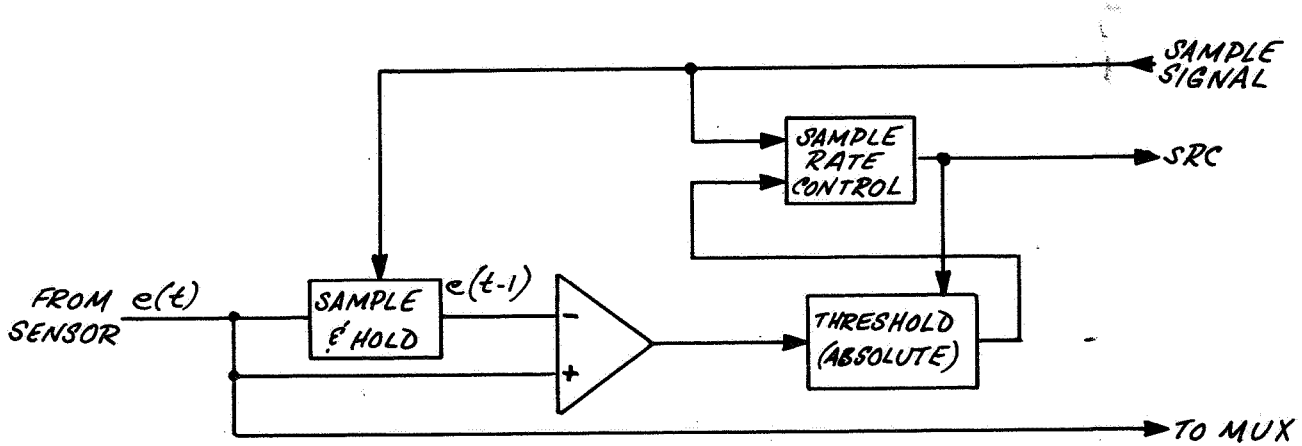


*Figure A-3. Analog Implementation of Variable Sample Rate Compressor*

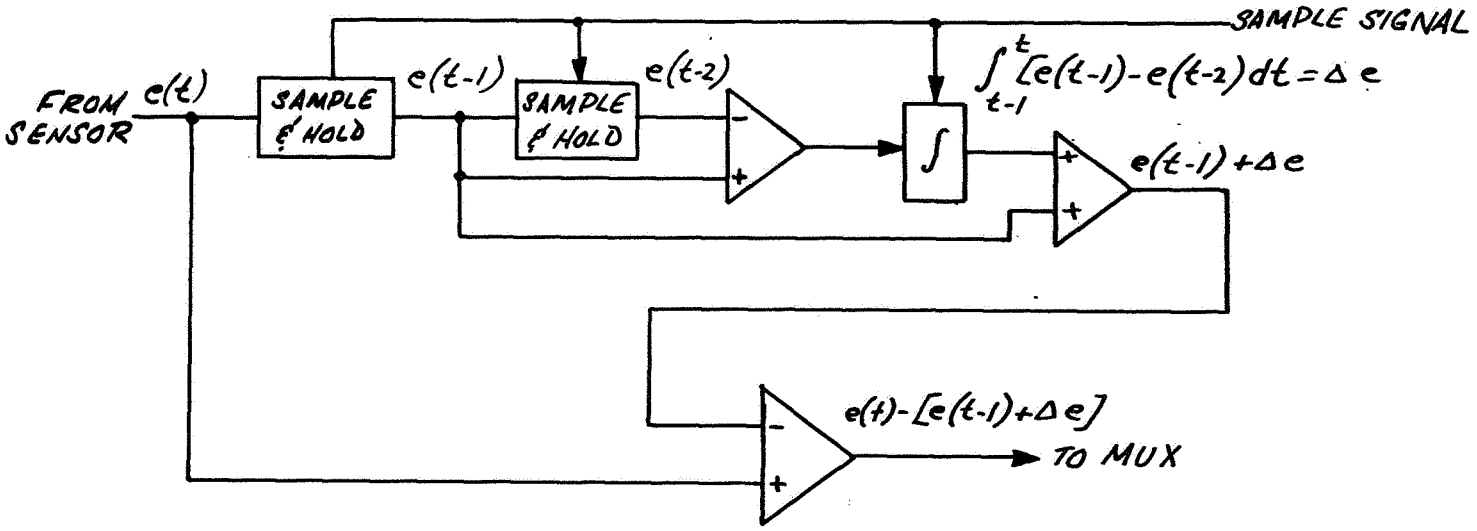


*Figure A-4. Analog Implementation of Prediction and Difference Coding Compressor*

### A.1.3.3 Aperture Techniques—Prediction

The general approach of the aperture techniques is to predict the present data point and to compare the predicted point with the actual point. If they agree within a predetermined error tolerance, the point is judged redundant, and is deleted. The prediction must always be made on the basis of data available to the ground terminal, so that uncontrollable errors do not occur in data recovery.

An analog version of the zero-order, floating tolerance band aperture compressor (ZOP) is shown in Figure A-5. A single memory circuit retains the last value transmitted, and subtracts it from the present value. The difference is compared with a predetermined error threshold. If the difference exceeds the threshold, the data is sampled and transmitted during the next multiplexer frame in response to the sample rate control signal which is sent to the multiplexer by the threshold circuit. The second, and simpler, multiplexer approach is used in this compression technique. The relative simplicity of this implementation is quite attractive, particularly when it is considered that this was found to be the most generally effective technique of all the aperture schemes tested.

A first-order floating tolerance band aperture technique is shown in Figure A-6. The prediction used is a first-order polynomial approximation identical to that used in the difference coding compressor. The tests made on actual data indicate that this technique is not as effective as the zero order method, possibly due to the inherent difficulty in making a prediction in a noise environment.

## A.2 A HYBRID COMPRESSOR

This section gives a brief description of an analog-digital hybrid computing technique which may have hardware advantages over completely digital implementations for certain types of compression algorithms.

### A.2.1. Signal Conversion

The principles and hardware techniques of analog-to-digital and digital-to-analog signal conversion are well known, and will not be discussed in detail here.

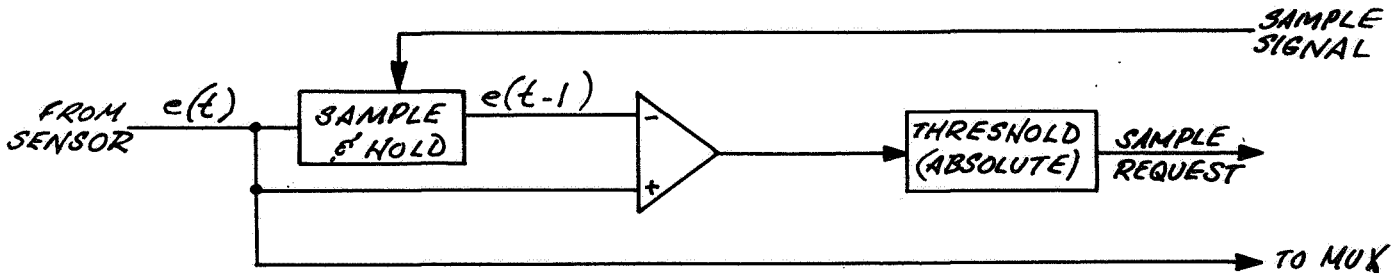


*Figure A-5. Analog Implementation of Zero Order Predictor Floating Aperture Compressor*

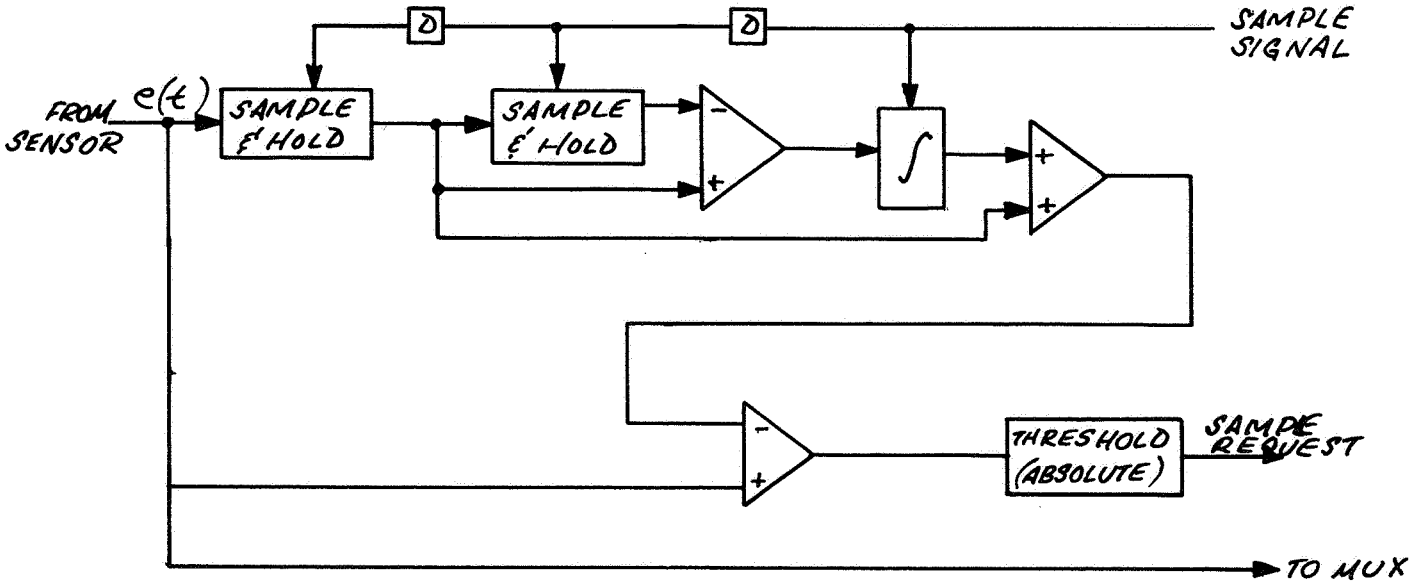


*Figure A-6 Analog Implementation of First-Order Predictor Floating Aperture Compressor*

Some general remarks must be made, however, in order to establish a framework upon which a discussion of the possible advantages of hybrid implementation can be based. Certain inherent characteristics of each type of computation are generally recognized. Digital computation can be made arbitrarily precise, and digital memory implementations are less expensive and more reliable than their analog counterparts. While sacrificing a certain amount in precision, analog computation techniques generally have a speed advantage over digital, being capable of providing solutions to complex problems essentially in real time. Conversion from digital to analog is quite easily implemented in hardware, while analog to digital signal conversion is generally complex and expensive. Since the sampled data in a telemetry system is normally digitized and stored, and is thus available to the compressor in digital form, the type of computation which would benefit from hybrid techniques would be one in which the digital contents of certain storage locations are converted to analog, a computation performed, and the results of the computation expressed as a logical decision rather than a number. If the result of the computation were a number to be used in future computations, the necessity would arise for either an analog storage device or a reconversion to digital, both of which are undesirable. Most compression techniques fall into this category, and therefore are not conveniently implemented in hybrid form. All predictions which are made on the basis of previously predicted points (as required by the receiving terminal), such as the optimum linear predictor, and all higher-than-zero order polynomial predictors, are of this type, as are all interpolation methods. For two of the compression methods which have been studied, i.e., the variable sample rate and the zero-order predictor, the necessary computation consists of taking the difference between the present data point and the last transmitted point, and making a logical decision as to a course of action based upon whether this difference does or does not exceed a predetermined threshold level. There is no requirement in either of these two algorithms for either reconversion to digital or for analog storage, hence they are particularly suited to being implemented in hybrid form.

The hybrid implementation of the compression algorithms has the same general configuration as does a digital compressor described in Section 4.4 of this report.

The hybrid section replaces the Central Processing Unit in performing the arithmetic operations. Figure A-7 shows a hardware implementation of the ZOP algorithm computation and the threshold determination using hybrid circuitry. The contents of the $y_t$, $y_{t+n}$, and K registers are converted to analog form in the conventional manner, by use of weighted resistive summing networks. The aperture value, K, is also stored, since it is adjusted as a function of output buffer activity. The analog difference between the $y_t$ and $y_{t+n}$ registers is compared by threshold circuits with both the positive and negative of K. The threshold circuits are such that the output is a binary one if input 1 exceeds input 2, and a binary zero, otherwise. The gated output is one if $| y_t - y_{t+n} | < K$, and zero, otherwise. Thus, this hybrid circuit accepts digital inputs, and provides a binary decision as its output, while the internal computations are in analog form. This approach makes maximum use of the desirable features of both types of computation, and results in an efficient hardware implementation, although it is quite limited in its capabilities.

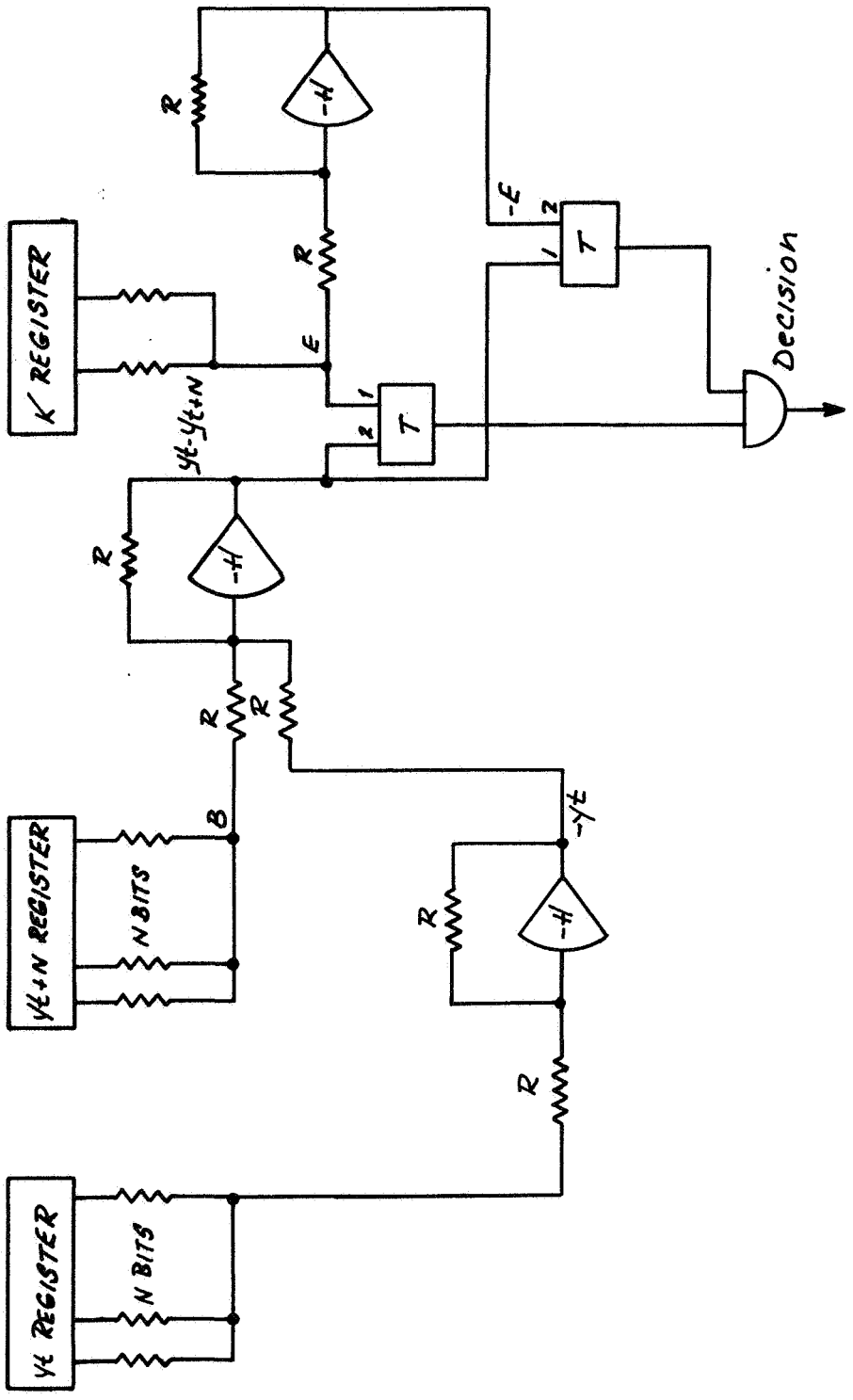


*Figure A-7. Hybrid Implementation—ZOP Algorithm*

Appendix B

# ADAPTIVE ANALOG LOW-PASS FILTERS

This appendix documents the work which was done during the study on the subject of analog low-pass filters. This investigation is an outgrowth of the study of fixed sampling rate compression methods. In fixed sampling near the Nyquist rate it is sometimes desirable to pre-filter the signal before sampling to eliminate aliasing errors. For a signal which has a varying bandwidth, it may also be desirable to vary the cutoff frequency of the pre-filter in an adaptive manner. This appendix is intended to amplify on low pass filter theory, and to discuss possible realization methods which could be used for adaptive filtering.

## B.1 APPROXIMATION OF LOW-PASS CHARACTERISTICS

The two essential steps in the network synthesis process are approximation and realization. The approximation procedure consists of developing the realizable transfer function that most closely approximates the desired transfer function. The desired transfer function for a filter network can be expressed in any of a number of ways. The most common are given in terms of voltage transfer ratio, which is generally specified in terms of amplitude, phase, or both. In the case of the data signal prefiltering requirement, the basic objectives are to realize as closely as possible the amplitude characteristics of the ideal low pass filter. The filter phase response is not particularly important for this application, as long as it does not become excessive in the passband. There are two types of functions which are commonly used in low pass filter design. Each is derived using a different approximating criterion. The first to be considered is the Butterworth function. The basic steps in the derivation of the Butterworth

approximation are first, to express the approximating amplitude function as a Taylor series expansion, with coefficients $B_i$, about the point $\omega = 0$. The passband error between the approximating function and the desired function (which is normalized to unity amplitude) is then written as a Taylor series in terms of the $B_i$. If the approximating amplitude-squared function is:

$$|F(j\omega)|^2 = G(\omega^2) = \frac{1}{P(\omega^2)} = \frac{1}{1 + B_1\omega^2 + B_2\omega^4 + \cdots + B_n\omega^{2n}}$$

Then the error function in the passband is:

$$1 - G(\omega^2) = \frac{P(\omega^2) - 1}{P(\omega^2)} = \frac{B_1\omega^2 + B_2\omega^4 + \cdots + B_n\omega^{2n}}{1 + B_1\omega^2 + B_2\omega^4 + \cdots + B_n\omega^{2n}}$$

In order for $G(\omega^2)$ to approximate the ideal characteristic as closely as possible, the maximum possible number of error derivatives at $\omega = 0$ must vanish. It can be shown that in order for the k'th derivative of the error to be zero, it must be true that $B_k = 0$ for an n'th order approximation of the low pass characteristic, therefore:

$$G(\omega^2) = \frac{1}{1 + B_n\omega^{2n}}$$

for Bn = 1, this becomes:

$$G(\omega^2) = \frac{1}{1 + \omega^{2n}}$$

which is the familiar Butterworth low pass amplitude characteristic. In order to realize this amplitude function, it must be expressed as a rational function of s such that the pole locations can be determined. It can be shown that the n poles of this function lie equally spaced on a unit semicircle in the left half s-plane, and are symmetrically placed with respect to the real axis. The general expression for this function is:

$$F(s) = \frac{1}{1 + b_1 s + b_2 s^2 + \cdots + b_n s^n}$$

where the denominator is known as the n'th order Butterworth polynominal. The characteristic feature of the Butterworth approximation, which arises directly from the nature of the Taylor expansion, is that the error is minimum at the point about which the expansion is made ($\omega = 0$ for a low pass filter), and the approximation is less and less accurate at greater distances from this point.

A second type of approximation is the Tchebyscheff approximation. This technique seeks to distribute the error evenly across the passband rather than favoring one end as the Butterworth approach does. In the Tchebyscheff approximation, the derived function oscillates about the desired flat response with equal peaks both above and below. The resulting amplitude response which minimizes the error peaks is:

$$G(\omega^2) = \frac{1}{1 + e^2 T_n^{\ 2}(\omega)}$$

where e is the peak-to-peak error deviation, and where $Tn(\omega)$ is known as the n'th order Tchebyscheff polynominal, and is defined as:

$$T_n(\omega) = \cos(n \cos^{-1} \omega)$$

It is found that the pole locations of the function of s corresponding to this amplitude function are distributed along a semi-ellipse in the left half s-plane, whose major axis is the imaginary axis and whose minor axis is the real axis. For given values of n and e, the pole locations are found by a simple geometrical construction.

The Butterworth and Tchebyscheff low pass approximations each have features which may be applicable to particular data types. One criterion of choice between the two for prefiltering applications is the ratio between average spectrum bandwidth and maximum bandwidth. For a data source where most of

the energy is concentrated near the low end, a Butterworth filter will give the best results, whereas for a source with a more uniform spectrum, the distributed error features of the Tchebyscheff design may be more desirable. The fact that the Butterworth filter has somewhat less oscillatory transient response, however, makes it probable that it would be more likely to be chosen for general low pass applications.

While this discussion has mentioned only two approximation criteria, it should be noted that there are many others; for example, the amplitude approximation could be made on a least squares error basis, or a new class of criteria could be derived involving phase response (delay characteristics) or transient response shape. Since this application is concerned primarily with amplitude response, these subjects will not be discussed except to point out their existence for specialized applications.

## B.2 REALIZATION OF LOW-PASS CHARACTERISTICS

The preceding discussion of the Butterworth and Tchebyscheff approximations to a low-pass filter revealed that the poles of both filter types of order 2n or 2n+1 occur in n conjugate pairs or n conjugate pairs plus one pole on the real axis. This discussion has assumed no finite zeros in the response characteristic, although it could be extended to cover finite zeros. One approach to the realization of a transfer function consisting of a number of conjugate pole-pairs and at most one negative, real, pole is to realize a single "building block" having one conjugate pole-pair, and to place these blocks in series, with appropriate component values to correctly place the pole-pairs. This technique cannot be used directly if only passive RLC components are permitted in the realization because of the impedance matching problem between "building blocks," and classical synthesis techniques must be employed. The use of active RC circuits for realizing the building block makes it possible to approach the required high input impedance and low output impedance that is mandatory if the building block method is to be successfully used. The use of RC circuits also eliminates the necessity for large, heavy inductors for low frequency filters.

The transfer function corresponding to one conjugate pole-pair and no finite zero is:

$$Z(s) = \frac{1}{(s-s_k)(s-s_k^*)}$$

$$= \frac{1}{s^2 + 2\sigma_k s + \sigma_k^2 + \omega_k^2} \qquad \text{where } S_k = -\sigma_k + j\omega_k$$

This can be written in the form of the transfer function of a damped, second-order system:

$$Z(s) = \frac{\omega_c^2}{s^2 + 2\zeta\omega_c s + \omega_c^2}$$

where $\omega_c$ is the cutoff frequency and $\zeta$ is the damping factor. The familiarity of this function in the synthesis and simulation of dynamical systems makes it natural to seek a realization in terms of analog simulation networks. Several have been investigated, and the most attractive both in terms of component economy and in ease of parameter variation is the operational amplifier network shown in Figure B-1.

$$-\frac{e_o}{e_1} = \frac{\frac{1}{R^2C^2}}{s^2 + \frac{2\zeta}{RC}s + \frac{1}{R^2C^2}}$$

The realization of a Butterworth low-pass circuit of order 2n requires the cascading of n circuits of this type with appropriately placed pole positions.

## B.3 REALIZATION OF ADAPTIVE LOW-PASS FILTERS

The realization problem for an adaptive filter - that is, one whose characteristics can be varied in a predictable manner by means of an external control signal - may be divided into two parts. The first requirement is to realize a circuit configuration which gives the desired filter characteristics and which lends itself to characteristic

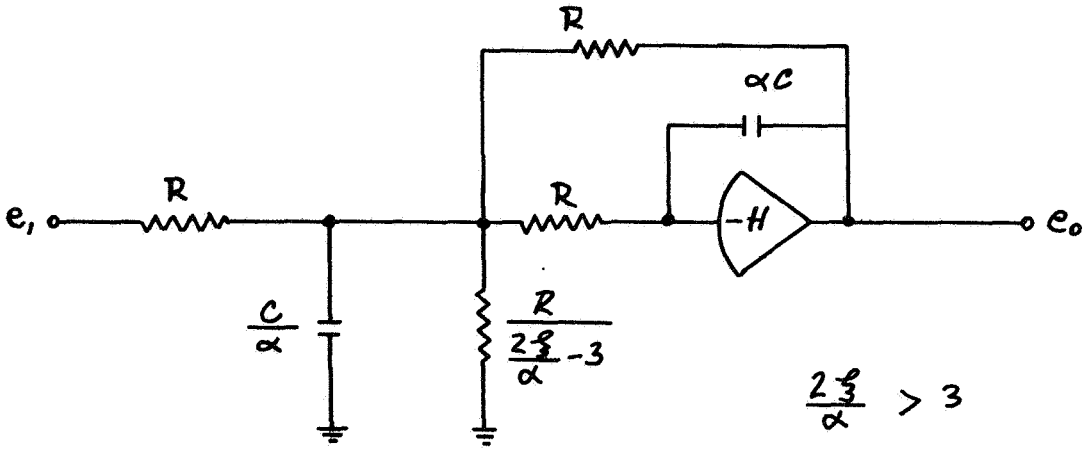


*Figure B-1. Adaptive Analog Filter "Building Block"*

variations by means of a number of parameter changes. The second part of the problem is to develop a hardware technique that will achieve the desired component variation in the most economical and reliable manner.

There are two general approaches to the variation of filter parameters in a given data channel. One is to switch the parameters between discrete values, and the second is to vary them continuously in linear response to an applied control signal. A brief consideration of the necessity of transmitting not only the data itself, but also sufficient information concerning the filter bandwidth (and sample rate, if applicable) to permit reproduction of the original signal leads to the conclusion that the switched characteristics are preferable. With the filter capable of assuming only discrete variable, predetermined characteristics, the only information which must be transmitted is the time of occurrence of a change. Alternatively, two, or at the most, three bits in the data word could be reserved for transmission of the filter bandwidth. Two bits would permit up to four discrete bandwidths to be used, which is probably sufficient for most data types.

The implementation of the discretely variable filter can also follow one of two possible approaches. The different filter characteristics can be assumed by physical switching of discrete components, either by solid state or mechanical means, or it may be more attractive and more versatile to use a filter whose parameter effective values are varied continuously by means of gain-controlled elements. In the latter approach the control voltage would be permitted to assume only discrete values, thus the filter characteristics would in effect be discretely switched. The realization of a sixth order Butterworth filter, for example, would require three adaptive building blocks. It is feasible to consider a filter of this type being capable of changing its overall response nature from Butterworth to Tchebyscheff by means of external control signals.

The foregoing conclusion of the probable desirability of switched characteristics is based upon the assumption that the entire adaptation process takes place in the spacecraft, and that for the data to be successfully recovered complete information on the current status of the adaptive system must be transmitted to the ground station. A further possibility is that the adaptation of filter characteristics might be done for various experimental purposes on command

from the ground. Since knowledge of the filter characteristics is under these circumstances already available at the receiving terminal, the restrictions under filter adaptation imposed by the desire to conserve down-link capacity are removed, and continuous control of the filter characteristics becomes quite practicable.

## B.4 REFERENCES

N. Balabanian, "Network Synthesis," Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1958.

Appendix C

# BUFFER ANALYSIS

## C.1 INFINITE LENGTH SYNCHRONOUS BUFFER WITH BINOMIAL INPUT DISTRIBUTION

This analysis begins with the single channel, infinite length, buffer with constant service time and binomial input distribution. This is a good model for a large synchronous buffer with stationary input data. After considering this model, and investigating the effects of parameter variations, consideration will be given to finite length buffers and methods of controlling them.

Most papers on buffer design have considered the data inputs to be Poisson distributed. This yields a conservative buffer design since the Poisson is the "most random" of possible discrete input distributions. However, in any practical attempt at optimization, one must know the cost incurred in using the easier to analyze conservative model. The Poisson distribution allows discrete inputs to appear anywhere over a time interval; whereas, for synchronous buffers the data may or may not appear at discrete time instants (neglecting clock jitter). Therefore, the binomial distribution is used to describe the probability of input events, assuming that the buffer output clock rate is commensurate with the buffer input clock rate.

The expected queue length (average fill for an infinite length buffer) for arbitrary input and holding-time (buffer output word period) distributions is derived in Reference 1, p. 336. We will therefore begin with Equation (23-4) from Goode and Machol which assumes that buffer fill transients have ended and that the system is in a state of equilibrium with a stationary input process.

$$E(n) = \rho + \frac{E(r^2) - \rho}{2(1 - \rho)} , \qquad \rho \neq 1 \tag{1}$$

where:

n = number of entries in buffer immediately after a buffer output

$\rho$ = m/M ($\rho < 1$ for equilibrium to be attained)

m = mean number of inputs per unit time

M = buffer output clock rate (events per unit time)

r = number of input events in time $T = \frac{1}{M}$

Now, define compression ratio: $\phi = \frac{\Sigma \text{ data bits into ACT}}{\Sigma \text{ data bits out of ACT}}$ , and consider the probability P of a nonredundant data sample point appearing at a time mark at the output of the compressor and feeding into the buffer. Over the time span for which $\phi$ is measured, $P = \frac{1}{\phi}$ .

Define $C = \frac{T}{\Delta t}$ (C is an integer), where T is the buffer output events period and $\Delta t$ is the buffer input event period. Note that C is the ratio of input to output transmission rates. The term "event" is used rather than "word" because different compression methods require different numbers of bits per nonredundant samples, and the buffer analysis is intended to be sufficiently general to include all methods. This will be of importance when parallel and serial transfer between compressor and buffer are considered.

Therefore, the probability of r inputs in time T becomes:

$$P_r(r) = \binom{C}{r} P^r (1 - P)^{C-r} \tag{2}$$

The expected value for the binomial distribution is: $E(r) = CP = \frac{C}{\phi}$ ; and the variance $\sigma^2(r) = CP(1 - P)$.

The mean number of inputs per unit time: $m = \frac{E(r)}{C\Delta t} = \frac{1}{\phi \Delta t}$ ; therefore: $\rho = m/M = \frac{C}{\phi} = E(r)$. Since: $\sigma^2(r) = E(r^2) - E^2(r)$, we have:

$$E(r^2) = CP(1 - P) - (CP)^2 . \tag{3}$$

Substituting Eq. (3) into Eq. (1), and letting $P = \frac{1}{\phi}$, we obtain:

$$E(n) = \bar{n} = \rho + \frac{\rho^2 - \rho/\phi}{2(1 - \rho)} . \tag{4}$$

This function is plotted in Figure C-1 for several values of C.

It is of interest to compare this result for the binomial input distribution with the result Goode and Machol obtain for the Poisson input distribution with constant holding-time (fixed output clock rate).
From Eq. (23.5) of Reference 1:

$$E_p(n) = \rho + \frac{\rho^2}{2(1 - \rho)} \tag{5}$$

Thus, for large compression ratios the results coincide.

Consider the reduction in expected buffer fill between the binomial and Poisson cases. The difference is: $\frac{1}{2\phi(1/\rho - 1)}$. Remember that $\rho = \frac{C}{\phi} < 1$ for stability. Now compute the percent difference in the expected buffer fill:

$$\Delta E(n) = \frac{E_p(n) - E(n)}{E_p(n)} \% = \frac{\frac{\rho/\phi}{2(1 - \rho)}}{\rho + \rho^2/2(1 - \rho)} \times 100 = \frac{100}{\phi(2 - \rho)} \% . \tag{6}$$

In the limit as $\rho \to 1$, $\Delta E \to \frac{100\%}{\phi}$ which for $\phi < 10$ yields an error greater than 10 percent. It is later shown that it is desirable to operate the buffer with $\rho$ as close to 1 as is possible. From the computer simulation of compression methods it is known that $\phi < 10$ for most of the test data compressed with resulting peak error of 1 quantum level. Therefore, in designing an optimum buffer, the binomial input distribution should be used rather than the Poisson, even though the Poisson is more tractable.

Having found the expected buffer fill, consider the rate of change of E(n) with $\phi$. Substituting $\rho = \frac{C}{\phi}$ into Equation (4) and taking the derivative with respect to $\phi$, we obtain:

$$\frac{dE(n)}{d\phi} = \frac{2C\phi[C + 1 - \phi] - C^2(C + 1)}{2\phi^2(\phi - C)^2} . \tag{7}$$

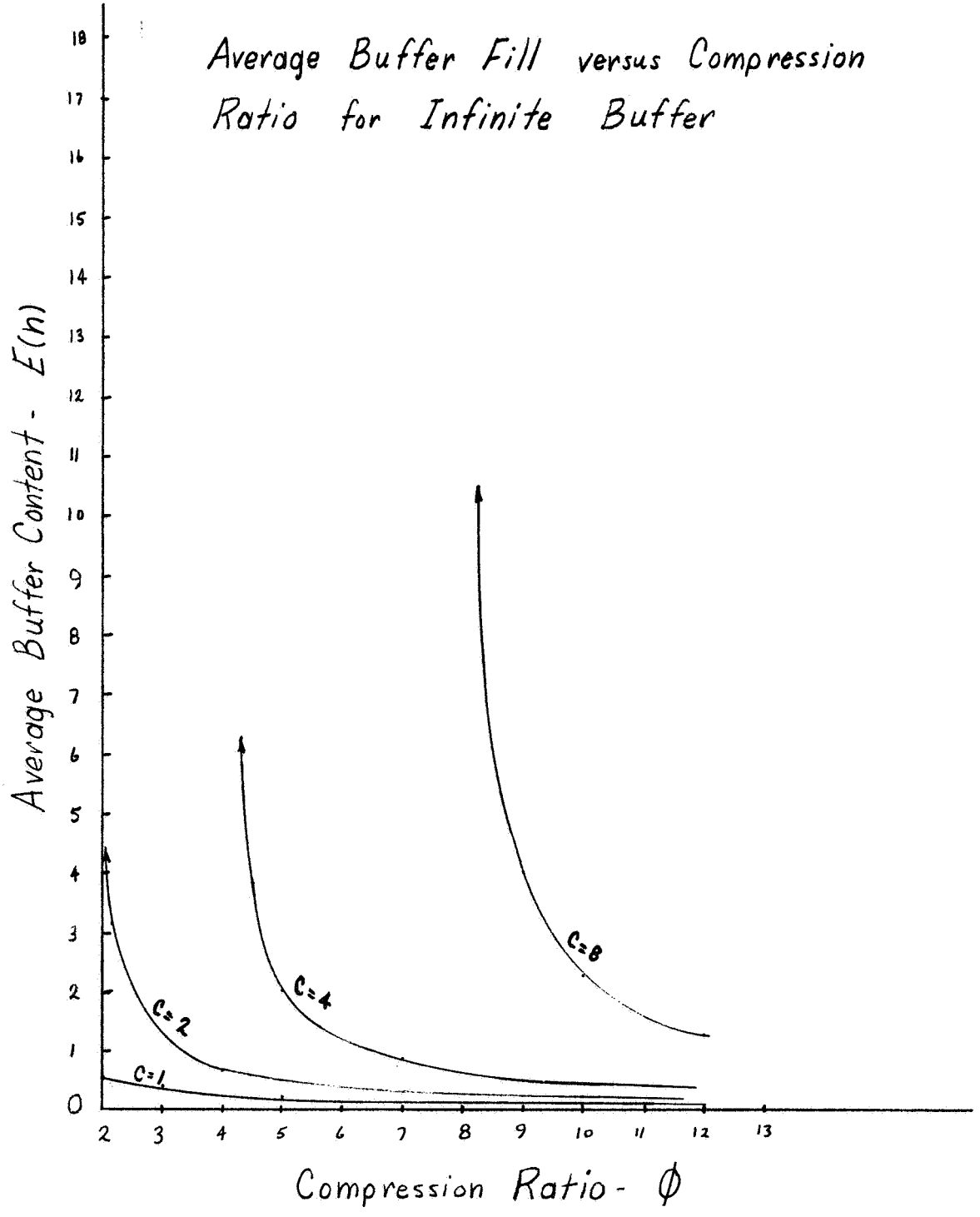


*Figure C-1. Average Buffer Fill vs Compression Ratio for Infinite Buffer*

Then, multiplying by $\phi/E(n)$, we obtain a sensitivity function $S(\bar{n})$ of E(n) for small changes in $\phi$. That is, $S(\bar{n})$ is the percent change in E(n) per percent change in $\phi$.

$$S(\bar{n}) = \frac{\Delta\bar{n}/\bar{n}}{\Delta\phi/\phi} \approx \left[ \frac{1 - C}{2\phi - 3C - 1 + \frac{C(C+1)}{\phi}} - 1 \right] . \tag{8}$$

$S(\bar{n})$ is plotted versus $\phi$ for several values of C in Figure C-2. The important thing to consider is the sharp break in the sensitivity curves near $\phi = C + 1$. Since C is the ratio of input to output transmission rates it would be desirable for any average compression ratio to operate with as high a C as possible. This would minimize the probability of transmission error (more time per bit) and reduce transmission bandwidth. However, it is evident that as $\frac{C}{\phi} = \rho \rightarrow 1$, the expected buffer fill increases rapidly. Not only does the fill increase, but it increases at a rapidly increasing rate. Thus, for C = 4, and $\phi = 4.2$, a 1% change in $\phi$ results in a 20 percent change in the expected buffer content after equilibrium is regained.

Only the quasi-stationary condition of small changes in compression ratio has been considered and the effect after equilibrium has been attained. For control design considerations the transient conditions should be examined to determine the rate that ACT parameters must be modified to prevent overflow and underflow. This will be considered in more detail for the finite length buffer.

The important conclusions obtained from the infinite length buffer are the need to use the binomial input distribution for optimum buffer design, and the sensitivity of average buffer fill to the input-output transmission ratio C. The analysis will now be extended to the practical case of the finite length buffer.

## C.2 FINITE LENGTH SYNCHRONOUS BUFFER WITH BINOMIAL INPUT DISTRIBUTION

Beginning with Eq. (11) of Reference 2, let P(n) be the probability that n events are stored in the buffer immediately after an attempt to remove an entry. $P_r(r \geq x)$ is the probability that more than x events are fed into the buffer during output event period T. Then:

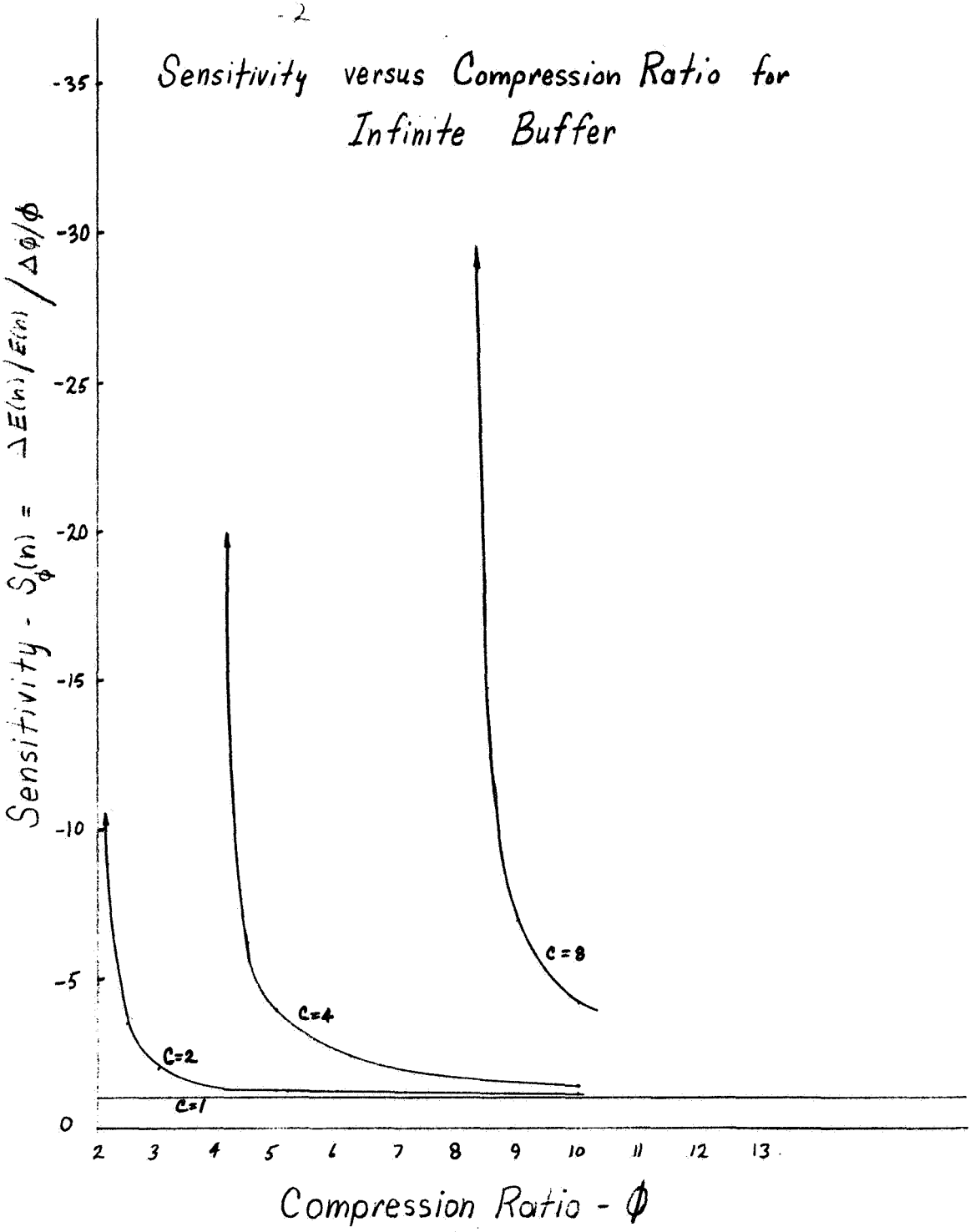


*Figure C-2. Sensitivity vs Compression Ratio for Infinite Buffer*

$$P(n) = \frac{1}{P_r(r=0)} \left[ P(n-1)\, P_r(r \geq 2) + P(n-2)\, P_r(r \geq 3) + \ldots + P(0)\, P_r(r \geq n+1) \right], \qquad (9)$$

for $1 \leq n \leq L - 1$, where L is the buffer length in "events."

Now, for the binomial distribution,

$$P_r(r \geq x) = \sum_{r=x}^{C} \binom{C}{r} P^r (1-P)^{C-r},$$

and $P_r(r=0) = (1-P)^C$; therefore,

$$P(n) = \sum_{j=n-C+1}^{n-1} P(j) \sum_{r=n-j+1}^{C} \binom{C}{r} \left(\phi - 1\right)^{-r} \qquad (10)$$

This, of course, is a recursive relationship in P(n), and holds for infinite and finite length buffers. To solve for P(n), we must first determine P(n=0).

For an infinite length buffer, the probability of removing an entry is simply $\rho$. For the finite length buffer the probability is reduced by the amount of overflow. Calling the fractional event loss R for the finite buffer, P(Removing an event) = $\rho(1-R)$, and P(Not removing an event) = $1 - \rho(1-R)$. Now, an event will not be removed only if the buffer became empty (or was empty) at the last removal time, and no arrivals occurred before the removal attempt being considered. Therefore:

$$P(n=0) = \frac{1 - \rho(1-R)}{P_r(r=0)} = \frac{1 - \rho(1-R)}{\left(1 - \frac{1}{\phi}\right)^C}, \qquad (11)$$

for the binomial distribution.

It still remains to express the fractional event loss R in terms of buffer length L. We shall use the result derived in Reference 2 and substitute P(n) for the binomial input distribution. From the reference:

$$R = \frac{(1-\rho)}{\rho}\left[\frac{1-\sum_{n=0}^{L-1} P_I(n)}{\sum_{n=0}^{L-1} P_I(n)}\right] \tag{12}$$

where $P_I(n)$ is the probability of queue length n in the infinite length buffer. It can be shown that $P(n) = \left[1+\frac{\rho R}{1-\rho}\right]P_I(n)$.

$P_I(n)$ for $n \neq 0$ is expressed in Equation (10). For $n = 0$, Equation (11) with $R = 0$ is used. Finally, after substitution and simplification, we obtain the following set of equations for the finite buffer of length L in terms of $P_I(n)$:

$$P_I(n=0) = \frac{1 - C/\phi}{\left(1-\frac{1}{\phi}\right)^C}, \tag{13}$$

$$P_I(n) = \sum_{j=n-C+1}^{n-1} P_I(j) \sum_{r=n-j+1}^{C} \binom{C}{r}\left(\phi-1\right)^{-r}, \tag{14}$$

$$1 \leq n \leq L-1$$

$$\sum_{n=0}^{L-1} P_I(n) = \frac{1}{\frac{R}{\phi/C-1}+1}. \tag{15}$$

Expected value:

$$E(n) = \left[1 + \frac{R}{\phi/C - 1}\right] \sum_{n=0}^{L-1} n P_I(n) . \tag{16}$$

Variance:

$$\sigma^2(n) = \left[1 + \frac{R}{\phi/C - 1}\right] \left[\sum_{n=0}^{L-1} n^2 P_I(n)\right] - E^2(n) . \tag{17}$$

These relationships were programmed for digital computer solution for representative values of R, C, and $\phi$. The results were used in Figures 1 to 4 in Section IV-3.

Before considering data transfer between the compressor and the buffer and methods of controlling the buffer, consider the following problem of buffer design. It would appear desirable to design the buffer such that the probability of overflow equals the probability of underflow (no entry available during a readout interval). The probability of no readout is P (Not removing an event) = $1 - \rho(1-R)$; the probability of an event overflowing the finite buffer is P (overflow) = R.

Let: $1 - \rho(1-R) = KR$, K some constant. (18)

Then, $\rho = \frac{1 - KR}{1 - R} < 1$ for equilibrium to occur. (19)

Therefore, $K > 1$, which indicates that without added control the probability of blank output periods must be greater than the probability of overflow. K can be made as close to one as desired, however, the cost is a larger buffer for the same compression ratio $\phi$. This can be seen in Figure C-1 for the infinite buffer. The expected buffer content increases rapidly as $\rho = C/\phi \rightarrow 1$.

The above results, of course, are intuitively obvious from the definition of

$$\rho = \frac{\text{mean number of input events per unit time}}{\text{buffer output rate}} .$$

## C.3 DATA TRANSFER FROM COMPRESSOR TO BUFFER

The transfer of data from the compressor to the output buffer is considered here. The previous analysis has been in terms of compression ratios and input-output transmission ratios. It will be shown that the analysis holds for both serial and parallel data transfer to the buffer.

Consider an $\alpha$ bit sample into the Compressor as an event occurring in time $\Delta t$. After compression, all nonredundant samples appear as $\beta$ bit events occurring in time $\Delta t$. $\beta$ will usually be greater than $\alpha$ because time and sensor tags may be required at the output of the ACT. In any case, the time per event must be the same at the input and output; otherwise a buffer would be required (in addition to the buffer we are considering).

The actual bit rate of transmission between the compressor and buffer may be greater or less than that into the compressor depending upon whether serial or parallel transfer is used. For both cases, the compression ratio $\phi = \frac{\Sigma \text{data bits into ACT}}{\Sigma \text{data bits out of ACT}} = \frac{\alpha}{\beta}\left[\frac{\Sigma \text{samples in}}{\Sigma \text{samples out}}\right]$. The probability of an event or sample appearing at the input to the buffer in a $\Delta t$ period is: $P = \frac{\Sigma \text{samples out}}{\Sigma \text{samples in}} = \frac{\alpha}{\beta\phi}$. Thus, in the previous analysis, the compression ratio would have to be modified by the factor $\beta/\alpha$ since it makes its appearance through the binomial probability of occurrence P.

Note also that the ratio $\beta/\alpha$ appears in the ratio of transmission rates with and without compression:

$$\frac{\text{Bits per sec. with compression}}{\text{Bits per sec. without compression}} = \frac{\beta/T}{\alpha/\Delta t} = \frac{\beta}{\alpha}\left(\frac{1}{C}\right).$$

## C.4 BUFFER CONTROL

There are several parameters at the designer's command in specifying the "optimum" ACT for a given mission. Some are more readily controlled than

others. For example, $\rho = C/\phi$ so that changing C and/or $\phi$ will vary $\rho$. However, if C is changed by modifying the transmission rate out of the buffer there results a difficult control problem at the receiver station.

Changing C by modifying the input event period $\Delta t$ essentially changes the data sampling rate. Adaptive sampling has been considered as a possible data compression method but has been shown to be inferior to the zero and first order interpolation methods. Thus, C will be considered a fixed design parameter; a parameter to be optimized and then considered constant.

Compression ratio $\phi$ and buffer length L are the remaining buffer independent parameters. Probability of overflow and underflow will be considered dependent parameters.

Obviously one does not change buffer length during a mission to prevent data loss as the data changes its characteristics. However, the buffer designer needs a trade-off function between cost of buffer length for a maximum allowable R (fractional event loss) and cost of adapting $\phi$ (in resulting data error as well as circuit costs).

Therefore, expressions are required for the change in overflow probability as a function of L and $\phi$. From Equation (12) we have:

$$P[\text{overflow}] = R = \frac{(1-\rho)}{\rho}\left[\frac{1}{\sum_{n=0}^{L-1} P_I(n)} - 1\right]. \tag{20}$$

Since buffer length change is constrained to integral values, we have:

$$\frac{\Delta P(\text{overflow})}{\Delta L} = \frac{\Delta R(L)}{\Delta L} = \frac{[R(L+\Delta L) - R(L)]}{\Delta L} \tag{21}$$

$$= \frac{1-\rho}{\rho\Delta L}\left[\left(\frac{1}{\sum_{n=0}^{L+\Delta L-1} P_I(n)} - 1\right) - \left(\frac{1}{\sum_{n=0}^{L-1} P_I(n)} - 1\right)\right] = \frac{1-\rho}{\rho\Delta L}\left[\frac{\sum_{n=0}^{L+\Delta L-1} P_I(n)}{\left(\sum_{n=0}^{L+\Delta L-1} P_I(n)\right)\left(\sum_{n=0}^{L-1} P_I(n)\right)}\right]$$

For incremental change in L($\Delta L = 1$), we obtain, after some simplication:

$$\left.\frac{\Delta P(\text{overflow})}{\Delta L}\right|_{\Delta L=1} = -\left[\frac{(R-1) + 1/\rho}{\frac{1}{P_I(L)}\left(\frac{1-\rho}{\rho(R-1)+1}\right) + 1}\right], \tag{22}$$

where $P_I(L)$ is obtained from Equation (14) with $n = L$.

For $R << 1$, Equation (22) reduces to:

$$\left.\frac{\Delta P(\text{overflow})}{\Delta L}\right|_{L=1} \cong \frac{1 - (1/\rho)}{1 + \frac{1}{P_I(L)}} . \tag{23}$$

Now consider the change in the probability of overflow as a function of compression ratio $\phi$. We have:

$$\frac{\partial P(\text{overflow})}{\partial \phi} = \frac{\partial R(\phi)}{\partial \phi} = \frac{\partial R(\phi)}{\partial \phi} .$$

From Equations (14) and (15):

$$\frac{\partial R(\phi)}{\partial \phi} = \frac{\partial}{\partial \phi}\left[\left(\frac{\phi}{C} - 1\right)\left(\frac{1}{\sum_{n=0}^{L-1}\sum_{j=n-C+1}^{n-1} P(j) \sum_{r=n-j+1}^{C} \frac{C!}{r!\,(C-r)!\,(\phi-1)^r}} - 1\right)\right]. \tag{24}$$

Equations (21) and (24) can be solved graphically from the computer solutions of Equations (13) through (17).

Application of Equation (24) is made easier by the observation that compression ratio $\phi$ is approximately a linear log function of peak error (aperture) K for the ZOP and FOI-2DF compression methods simulated on a digital computer with test data.

In addition, both methods have approximately the same slope for a particular type of data. Thus, for the methods considered an adaptive discrete change in aperture K will result in a known change in $\phi$.

Therefore, for a particular method with known $\phi_1$, and $K_1$, we have:

$$\text{Log}_{10}\,\phi_2 - \text{Log}_{10}\,\phi_1 = m(\text{Log}_{10}\,K_2 - \text{Log}_{10}\,K_1)\,. \qquad (25)$$

Thus:

$$\phi_2 = \phi_1\left(\frac{K_2}{K_1}\right)^m ,$$

where the slope m depends upon the data being compressed.

For the five types of data considered in this report, the slope m ranged from 0.3 to 1.2. This means that for an increase in aperture by a factor of two (which yields doubled peak error) the new compression ratio lies somewhere in the range: $\left[1.2\,\phi_1 \leq \phi_2 \leq 2.3\phi_1\right]$.

## C.5 AVERAGE COMPRESSION RATIO $\overline{\phi}$ FOR MULTIPLEXED SENSOR DATA

In determining buffer requirements for multiplexed sensor data, it becomes necessary to consider the system average compression ratio $\overline{\phi}$. In examining only stationary systems, the expression for $\overline{\phi}$ in terms of individual sensor compression ratios $\phi_i$ is quite simple.

Let each sensor compression ratio over message time T be:

$$\phi_i = \frac{\Sigma \text{ samples}}{\Sigma \text{ samples sent}} = \frac{\alpha_i N_i}{N_i} = \frac{N_i}{\beta_i N_i}\,. \qquad (26)$$

Then, for S sensors, the average compression ratio $\overline{\phi}$ over time T becomes:

$$\overline{\phi} = \frac{\sum_{i=1}^{S} \sum \text{samples}}{\sum_{i=1}^{S} \sum \text{samples sent}}\,. \qquad (27)$$

If we let: $N_i = m_i N$, where $m_i$ is the number of appearances sensor i makes in the basic multiplexed frame, and N is the number of frames in the time T, we obtain:

$$\overline{\phi} = \frac{\sum_{i=1}^{S} m_i N}{\sum_{i=1}^{S} \beta_i m_i N} = \frac{\sum_{i=1}^{S} m_i}{\sum_{i=1}^{S} \frac{m_i}{\alpha_i}} . \tag{28}$$

Therefore, for the case in which all sensors are sampled at the same rate,

$$\overline{\phi} = \frac{S}{\sum_{i=1}^{S} 1/\alpha_i} .$$

In addition, if all sensors have the same value compression ratio $\alpha$, $\overline{\phi} = \frac{S}{S/\alpha} = \alpha$.

In the buffer analysis presented in this appendix it is assumed that the system average compression ratio $\overline{\phi}$ is known or can be calculated. Actually, in designing a compression system with buffer control, $\overline{\phi}$ could only be estimated and the probability of underflow and overflow would depend upon the accuracy of the estimate. This in itself is a major reason for buffer controls since the stationarity and value of $\overline{\phi}$ can never be known exactly in advance.

## C.6 PRIORITY SYSTEM TO GUARANTEE AGAINST OVERFLOW OF SENSOR DATA

The probability of buffer overflow can be made as small as one desires by several different methods such as increasing the buffer size, increasing compression ratios, increasing the output-input transmission rate, or combinations of all three. However, with these methods, the probability of overflow remains finite for all sensors; although, with added expense it could be made as small as desired.

A relatively simple method to guarantee zero overflow for certain sensor channels is to design the system such that the total system transmission bandwidth equals the sum of the high priority sensor channel bandwidths without compression. The low priority channel data is compressed and transmitted in the bandwidth obtained by compression of the high priority data. If buffer overflow is imminent, non-priority sensor data is either further compressed or eliminated completely from the buffer input until the buffer content is stabilized at an acceptable level.

For the general case of P priority channels each with bandwidth $W_p$ and compression ratio $\phi_p$, and N non-priority channels each with bandwidth $W_n$ and compression ratio $\phi_n$, the total bandwidth available for non-priority channels is:

$$W_n = \sum_{p=1}^{P} W_p(1 - 1/\phi_p) .$$

The bandwidth available to a particular non-priority sensor $\alpha$ is:

$$W_\alpha = \phi_\alpha \left[ \sum_p W_p(1 - 1/\phi_p) - \sum_{n \neq \alpha} \frac{W_n}{\phi_n} \right] . \qquad (29)$$

In designing such a system, $\phi_p$ will usually be a random variable. Assume that the mean $\bar{\phi}_p$ and variance $\sigma^2(\phi_p)$ are known, and the P priority channels have bandwidth W. Then it can be shown that the expected available bandwidth for the non-priority sensor channels is: $\overline{W}_n = WP\left[1 - (\overline{1/\phi_p})\right]$, and the variance: $\sigma^2(W_n) = W^2 P\sigma^2(1/\phi_p)$.

The mean and variance of $1/\phi_p$ can, of course, be obtained from the distribution of $\phi_p$.

## C.7 REFERENCES


1. H. H. Goode and R. E. Machol, "System Engineering," McGraw-Hill, 1957, chap. 23.

2. J. E. Medlin, "Buffer Length Requirements for a Telemetry Data Compressor," 1962 National Telemetering Conference.

3. T. L. Saaty, "Elements of Queuing Theory," McGraw-Hill, 1961.

# Appendix D

## TRANSMISSION ERROR ANALYSIS

### D.1 INTRODUCTION

To evaluate the utility of a compression technique for a telemetry system, it is necessary to determine the effects of channel (transmission) noise on the data with and without compression. To study the effects of transmission noise on compressed data, the channel error(s) must be expressed in a form which is readily combined with the error in the expanded data due to disturbances in the analog-to-digital conversion and the compression processes. For the digital channel, the bit error probability (p) is normally used to characterize the effects of transmission noise. Using a bit error probability (p), which represents the sum total of all errors in the transmission chain (nonlinearities of the various stages, rf leakage, intermodulation cross products, etc.) and not just the channel per se, the problem is to transform p such that it may be easily combined with the compression error. Because the RMS error interior (due primarily to its mathematical convenience) is commonly used in the evaluation of compression techniques, it is desirable to convert the bit error probability to an equivalent RMS error.

If a formatted message structure (similar to the structure employed in the Gemini and planned for the Apollo telemetry systems) is used, the resultant compressed data word consists of two separate parts. The first part contains the actual samples of sensor outputs and the second is the sensor tag or the word location number as it is referred to in the Gemini format. The conversion of the bit error probability (p) to an equivalent rms error must then be performed on each part, and combined to obtain the complete expression for the transmission error for operation in the compressed mode. A fundamental assumption in the

conversion of p is that the error in any data bit is independent of that occurring in any other bit. Because the bit errors are independent, it is logical to assume the rms error resulting from each of the foregoing two types of data are likewise independent. Thus, the conversion for each type of data may be done separately and then summed to obtain the total equivalent rms error.

## D.2 CONVERSION OF BIT ERROR PROBABILITY (p) TO AN EQUIVALENT RMS ERROR FOR AN ACTUAL DATA SAMPLE

For this analysis, the transmission system is characterized as a binary symmetric channel. The transmitted and received symbols are $x_i$ and $y_j$, respectively. The symbols occur with probabilities $p(x_i)$ and $p(y_j)$.

$$p = P(y_j = 1/x_i = 0) = P(y_j = 0/x_i = 1) \tag{1}$$

$$q = P(y_j = 1/x_i = 1) = P(y_j = 0/x_i = 0) \tag{2}$$

$$\begin{aligned} \text{Probability of error} &= P(y_j = 1)\, P(x_i = 0/y_j = 1) \\ &\quad + P(y_j = 0)\, P(x_i = 1/y_j = 0) \\ &= P(y_j = 1)p + P(y_j = 0)\, p. \end{aligned} \tag{3}$$

$$\text{Probability of error} = p \text{ since } p(y_j = 1) + P(y_j = 0) = 1.$$

Let U represent the amplitude of the bit error, which is illustrated in Figure D-1. Whenever an error occurs, the amplitude is unity and otherwise it is zero.

The bit error causes an amplitude error which may be considered as a binomial random variable with

$$P(U = 1) = p \quad \text{and} \quad P(U = 0) = 1 - p = q.$$

The variance of the error U is given by

$$\sigma_U^2 = pq\,. \tag{4}$$

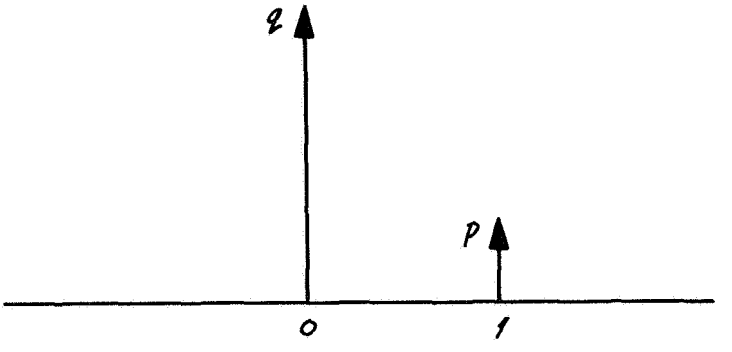


*Figure D-1. Bit Error Probability Amplitude (U)*

The above error is for a single bit error which causes an amplitude error of 1. If we consider a binary word of length m, the amplitude error in the word must be weighted by the bit position in the binary word. Let the total amplitude error in y be designated by the random variable R, then

$$R = \sum_{n=1}^{m} 2^{m-n} R_{m-n} \tag{5}$$

where:

$R_{m-n}$ = bit error with amplitude of one.

Because the bit errors are independent the variance in the amplitude error of an m bit binary word is given by

$$\sigma_R^2 = \sum_{n=1}^{m} (2^{m-n})^2 \sigma_U^2 \tag{6}$$

where:

$\sigma_R^2$ = Variance in word amplitude error.

Substituting from Equation (4) and simplifying Equation (6) we obtain

$$\sigma_R^2 = \left(\frac{2^{2m} - 1}{3}\right) pq . \tag{7}$$

The rms word error in a noncompressed word is then given by:

$$\text{rms word error}_{NC} = \left[\left(\frac{2^{2m} - 1}{3}\right) pq\right]^{1/2} . \tag{8}$$

Equation ( 8 ) is the equivalent rms word error in quanta for noncompressed data. If the data is compressed then each received data word represents essentially C data words in the expanded data stream, where C is the net compression defined as the ratio of samples to samples sent. In other words, the effects of noise in the transmission system are amplified in proportion to the amount of net compression. A compressed word or a word representing C words at the output of the data expander would have its variance multiplied by C (the net compression factor). Using then Equation ( 7 ), the variance for the compressed word would be as follows:

$$\sigma^2_{R_C} = C\sigma^2_R \quad . \tag{9}$$

Therefore, the rms word error for the compressed word is

$$\text{rms word error}_C = \sqrt{C}\,[\text{rms word error}_{NC}] \tag{10}$$

$$= \left[C\left(\frac{2^{2m} - 1}{3}\right)pq\right]^{1/2} \quad . \tag{11}$$

## D.3 CONVERSION OF BIT ERROR PROBABILITY (p) TO AN EQUIVALENT RMS ERROR FOR DATA CONTAINING ONLY SENSOR TAGGING INFORMATION—ZOP ALGORITHM

One method of deriving the expression for the transmission error in a sensor tag word or word location number is to consider the effect of the production of wild points in the transmitted data of any of the operating sensors. In addition to the two assumptions: (1) that the channel is binary symmetric; and (2) that the data bit errors are independent; the following conditions are necessary to obtain a useful expression for the sensor tag error:

a. Channels are independent.
b. All channels are statistically the same.
c. Net compression (C) defined as the ratio of samples to samples sent is the same for each operating sensor.

d. The bit error amplitude distribution is a binomial random variable (see conversion of p for actual data sample).

e. A sensor tag word (data group) is lost when one or more bits in the word are in error.

f. Compression technique employed for all sensors is zero-order prediction.

g. Number of bits in a sensor tag word is m.

h. Number of spacecraft sensors in operation is k.

Considering the sensor tag word to be lost when only one bit is in error eliminates the complex problem of weighting as to the degree of error that results as the bit error occurs in the different bit position throughout the sensor tag word. If p is the bit error probability, then the probability of correct bit reception = $1 - p = q$, and the probability of correct reception of an m bit sensor word = $(q)^m$. Then the probability of a sensor tag word being lost ($P_{sw}$) = $(1 - q)^m$.
Let:

$S_{in}$ = number of samples per second into the compressor

$S_{out}$ = number of samples per second out of the compressor

$S_{out} = S_{in}/C$ where C is the net compression ratio.

Assuming a sensor tag word is transmitted for each sample out of the compressor, the number of sensor tag words ($N_s$) in error per second is

$$N_s = P_{sw} S_{out} = P_{sw} \frac{S_{in}}{C}. \tag{12}$$

$N_s$, the total number of sensor tags in error per second, is considered to be uniformly distributed across k - 1 (where k is the number of active sensors in the spacecraft) operating channels or sensors, in terms of producing wild points. Therefore, the number of sample points in error per second per channel is

$$N_{sc} = \frac{P_{sw} S_{in}}{(k-1) C}. \tag{13}$$

Because there are k - 1 operational sensors, each with its sensor tag error uniformly distributed across the remaining number of sensors, the total number of sample points in error (wild points) in any channel per second is $(k - 1)N_{sc}$, which is equal to $N_s$.

In Figure D-2, the data amplitude versus time (sample points) for one of the k sensors is illustrated to show the occurrence of a wild point caused by sensor tagging error(s). The time scale of Figure D-2 (between points A and B) is expanded in Figure D-3 to see resultant error in the received data caused by the wild point. The data is compressed by zero-order prediction. Considering the location of the wild point, $y_w(nt)$, to have a uniform probability distribution throughout the nontransmitted sample points, such as between points A and B in Figure D-3, then on the average, one-half of the nontransmitted samples will be in error because of the received wild point. The number of nontransmitted sample points between two transmitted points is determined by the net compression, C. Therefore, a wild point resulting from sensor tag word errors will, on the average, cause an error in C/2 sample points in the reconstructed data. Referring to Figures D-2 and D-3, the mean square error (MSE) in the expanded data because of the wild point $y_w(nt)$ is given by

$$MSE = \frac{1}{T_e S_{in}} E\left\{\sum_{n=1}^{C/2} [y_w(nt) - x(nt)]^2\right\} \qquad (14)$$

where:

$E[y_w(nt)] = E[x(nt)] = 0$

$T_e$ = time for the occurrence of one sensor tag word error = $1/N_s$

$T_e S_{in}$ = number of sample points over which the sensor tag error must be averaged.

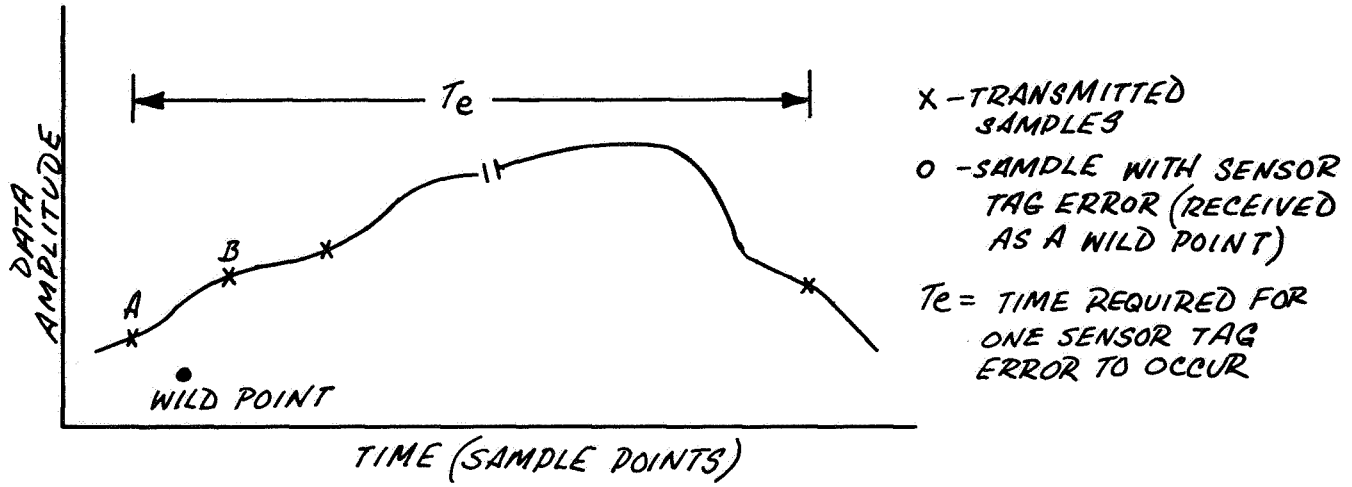


*Figure D-2. Data Sampling and Selection of a Sensor with a Wild Point Caused by a Sensor Tag Error*

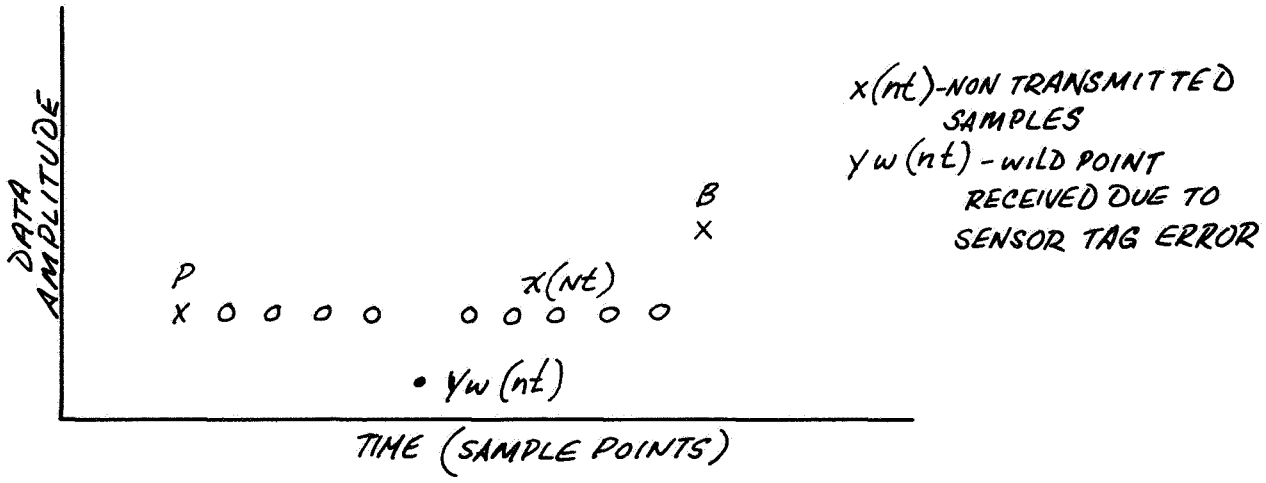


*Figure D-3. Sensor Tag Error is Received as a Wild Point for One of the k Onboard Sensors*

Expanding Equation (14):

$$MSE = \frac{C}{2 T_e S_{in}} E\left\{[y_w(nt)]^2\right\} - \frac{C}{T_e S_{in}} E\left\{[x(nt)\, y_w(nt)]\right\} + \frac{C}{2 T_e S_{in}} E\left\{[x(nt)]^2\right\}. \quad (15)$$

Because channels are independent, the correlation between x(nt) and $y_w(nt)$ is zero. Thus,

$$E\{[x(nt)\, y_w(nt)]\} = 0 .$$

Then,

$$MSE = \frac{C}{2 T_e S_{in}} \left[ E\left\{[y_w(nt)]^2\right\} + E\left\{[x(nt)]^2\right\} \right] . \quad (16)$$

Let:

$\sigma_y^2 = E\left\{[y_w(nt)]^2\right\}$ = mean signal power in wild point, and

$\sigma_x^2 = E\left\{[x(nt)]^2\right\}$ = mean signal power in sample point.

Then,

$$MSE = \frac{C}{2 T_e S_{in}} \left[\sigma_y^2 + \sigma_x^2\right] \quad (17)$$

The mean powers in each channel are equal because the channel statistics are identical. Thus,

$$\sigma_y^2 = \sigma_x^2 = \sigma^2$$

$$MSE = \frac{C\sigma^2}{T_e S_{in}} = \frac{\sigma^2}{T_e S_{out}} \quad (18)$$

and because

$$T_e = 1/N_s = \frac{1}{P_{sw} S_{out}},$$

then,

$$MSE = P_{sw} \sigma^2 \tag{19}$$

$$\text{rms sensor tag error}_1 = \sqrt{P_{sw} \sigma^2} \tag{20}$$

The resultant partial expression for the rms error in compressed data due to transmission error in a sensor tag word indicates that the tagging error is proportional to the signal power and the probability of a sensor tag word error. Thus, the expression for the first part of the sensor tag error appears to be reasonable.

The foregoing expression for the rms sensor tag error has a one subscript because this expression constitutes only part of the total rms error in compressed data due to sensor tagging error. A second part of the error occurs in the channel in which the wild point originated. In Figure D-4, the data sampling and selection of the sensor which lost the transmitted sample point is illustrated. Because this channel is operating at the same rates and with an identical probability of sensor tag word error, the mean square error for this sensor is

$$MSE = \frac{1}{T_e S_{in}} E\left\{ \sum_{n=1}^{C} [x_2(nt) - x_1(nt)]^2 \right\} . \tag{21}$$

Because of the lost sample point, on the average, C sample points will be in error in the reconstructed data. Expanding the foregoing expression, the mean square error is given by

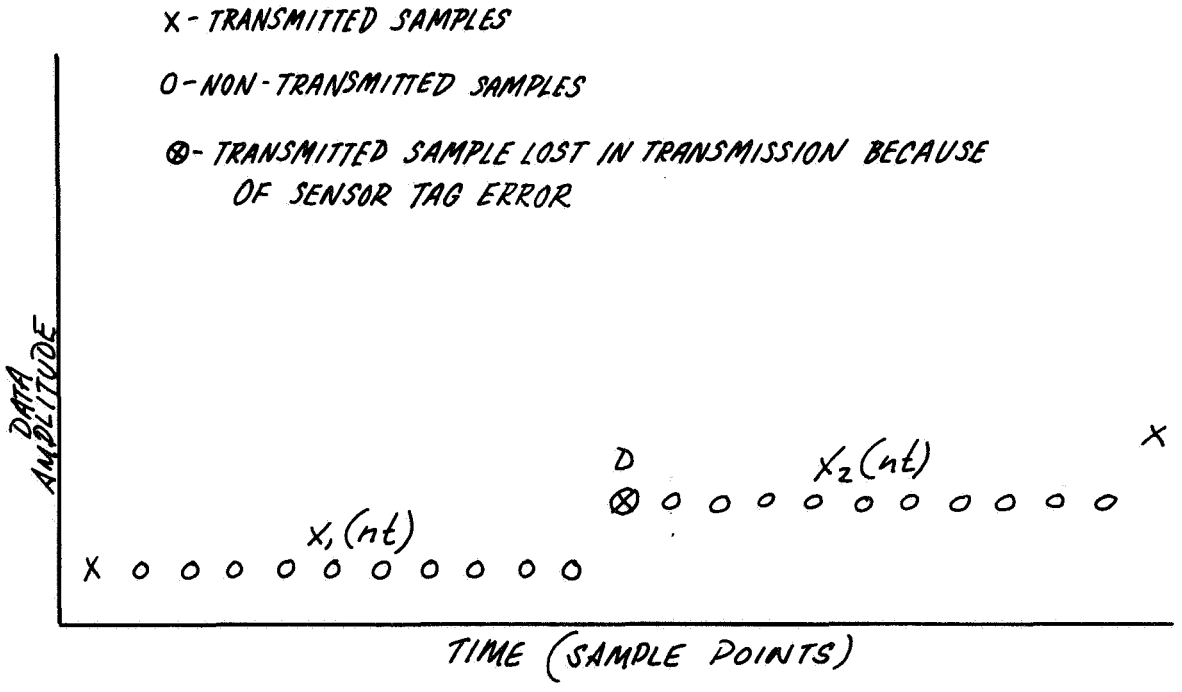


*Figure D-4. Data Sampling and Selection of Sensor that Lost Sample Point at D Because of Sensor Tag Error*

$$MSE = \frac{C}{T_e S_{in}} \left[ E\left\{[x_2(nt)]^2\right\} - 2E[x_2(nt)\, x_1(nt)] + E\left\{[x_1(nt)]^2\right\} \right] . \quad (22)$$

Assuming the compression has reduced the natural redundancy to the extent that adjacent sample points are independent, then their autocorrelation is zero. Then,

$$MSE = \frac{C}{T_e S_{in}} \left[ E\left\{[x_2(nt)]^2\right\} + E\left\{[x_1(nt)]^2\right\} \right] \quad (23)$$

$$MSE = \frac{2C\sigma^2}{T_e S_{in}} \quad (24)$$

$$MSE = \frac{2\sigma^2}{T_e S_{out}} \quad (25)$$

where:

$$T_e = \frac{1}{P_{sw} S_{out}} .$$

Therefore,

$$\text{rms sensor tag error}_2 = \sqrt{2P_{sw}\sigma^2} . \quad (26)$$

$$\text{rms sensor tag error}_2 = \sqrt{2}\ \text{rms sensor tag word error}_1 .$$

The rms error in the second channel due to the distortion of one of its sensor tag words because of transmission noise is identical to that for the first channel except for a constant multiplier. The multiplier $\sqrt{2}$, results because the number of sample points in error is twice that in the first channel. Combining the two expressions, the total rms error in the compressed data because of sensor tagging error is

$$\text{Total rms sensor tag error} = 2.4\sqrt{P_{sw}\,\sigma^2} \tag{27}$$

$$= 2.4\sqrt{\left[1-(1-p)^m\right]\sigma^2}\ . \tag{28}$$

Notice the expression for the total rms sensor tag error is independent of the net compression ratio, C, defined as the ratio of samples-to-samples sent. The offsetting factors here, making the sensor tag error insensitive to C, are (1) the probability of a sensor tag error occurring in any given channel, and (2) the number of actual sample points represented by the transmitted point in error. If for any given channel C is small, then a large number of sample points are actually transmitted. The probability of a sensor tag word error occurring in that channel is then increased as C decreases because more samples must be transmitted. However, when the error occurs, the number of reconstructed sample points which are in error is small. As C is increased, the number of samples transmitted per channel is less, and therefore, the probability of a sensor tag word error occurring in a given channel is reduced. This reduction in word error probability per channel is directly offset when the error occurs because it affects an increased number of actual sample points in the reconstructed data.

## D.4 SENSOR TAG ERROR—FOI-2DF ALGORITHM

Suppose the compression technique employed for all the sensors is the FOI-2DF technique rather than the zero-order predictor originally assumed. In Figure D-5 the data amplitude versus time (sample points) is illustrated to show the resultant error in the reconstructed data caused by a wild point. Using the FOI-2DF technique, the sampled data is reconstructed by connecting adjacent transmitted sample points with a straight line between them. This line connecting the samples is such that the redundant points do not deviate more than some preestablished tolerance band (this actually defines the peak error). With no wild point occurring due to a sensor tag error, the data is reconstructed along line AC with a slope, $m_t$. When the wild point occurs at

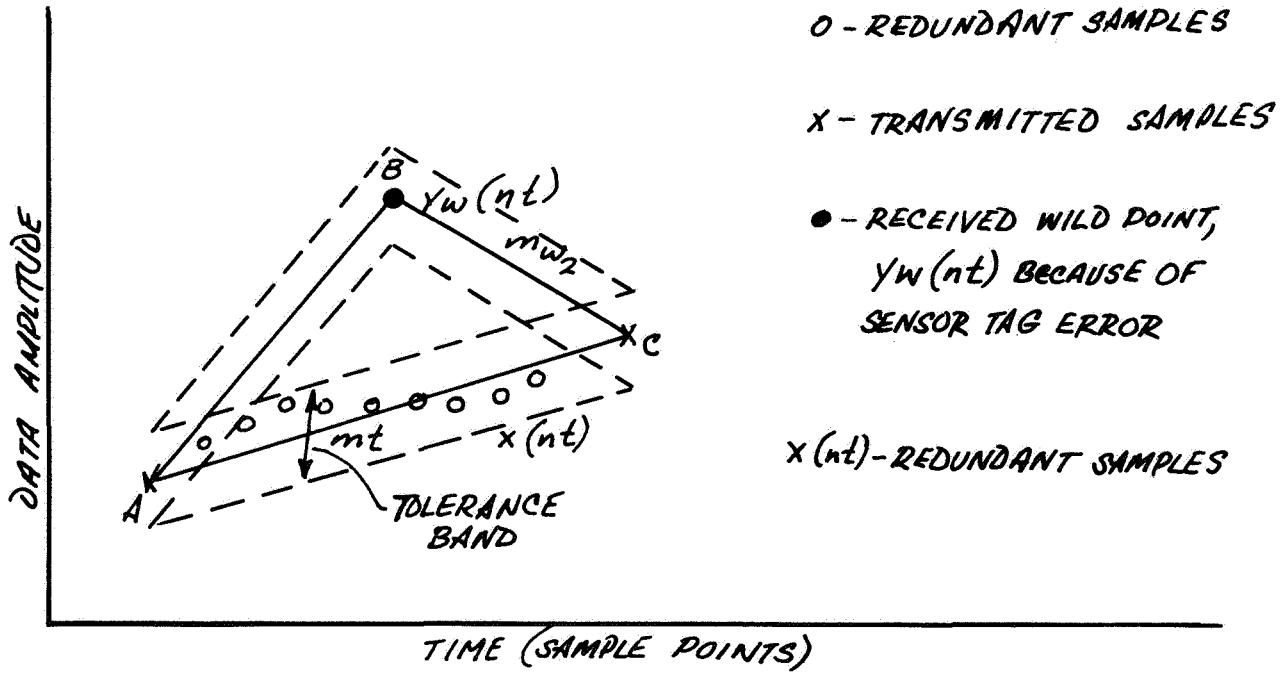


*Figure D-5. Sensor Tag Error Received as Wild Point*

some point such as B, the received data is reconstructed using line segments AB and BC which have slopes $m_{w1}$ and $m_{w2}$, respectively. Considering the location of the wild point, $y_w(nt)$, to be uniformly distributed throughout the redundant sample points, then on the average, the wild point will occur half way between the transmitted samples. As shown in Figure D-2, the wild point, $y_w(nt)$, resulting from the sensor tag error, causes an error in C (net compression) sample points in the reconstruction data. However, because of the method of reconstruction of the data using the fan technique, the resultant error varies with each redundant sample between the transmitted points, A and C. Therefore, the error must be weighted in accordance with the difference slopes between $m_t$ and $m_{w1}$ and $m_{w2}$. The mean square error ($MSE_1$) in the expanded data because of the wild point, $y_w(nt)$ is

$$MSE_1 = \frac{1}{T_e S_{in}} \left[ E\left\{ \sum_{n=1}^{C/2} w_{1n} [y_w(nt) - x(nt)]^2 \right\} + E\left\{ \sum_{n=C/2}^{C} w_{2n} [y_w(nt) - x(nt)]^2 \right\} \right] \quad (29)$$

where:

$$w_{1n} = |(m_{w1} - m_t)| \cdot nt$$

$$w_{2n} = |(m_{w2} - m_t)| \cdot nt \ .$$

$$MSE_1 = \frac{1}{T_e S_{in}} \left[ E\left\{ \sum_{n=1}^{C/2} w_{1n} [y_w(nt)]^2 \right\} - 2E\left\{ \sum_{n=1}^{C/2} w_{1n}\, y_w(nt)\, x(nt) \right\} \right.$$
$$+ E\left\{ \sum_{n=1}^{C/2} w_{1n} [x(nt)]^2 \right\} + E\left\{ \sum_{n=C/2}^{C} w_{2n} [y_w(nt)]^2 \right\}$$
$$- 2E\left\{ \sum_{n=C/2}^{C} w_{2n}\, y_w(nt)\, x(nt) \right\} \qquad (30)$$
$$\left. + E\left\{ \sum_{n=C/2}^{C} w_{2n} [x(nt)]^2 \right\} \right] .$$

Because the channels are independent, the correlation between $y_w(nt)$ and $x(nt)$ is zero, and thus,

$$E\left\{ \sum_{n=1}^{C/2} w_{1n}\, y_w(nt)\, x(nt) \right\} = E\left\{ \sum_{n=C/2}^{C} w_{2n}\, y_w(nt)\, x(nt) \right\} = 0 .$$

Therefore:

$$MSE_1 = \frac{1}{T_e S_{in}} \left[ E\left\{ \sum_{n=1}^{C/2} w_{1n} [y_w(nt)]^2 \right\} + E\left\{ \sum_{n=1}^{C/2} w_{1n} [x(nt)]^2 \right\} \right.$$
$$\left. + E\left\{ \sum_{n=C/2}^{C} w_{2n} [y_w(nt)]^2 \right\} + E\left\{ \sum_{n=C/2}^{C} w_{2n} [x(nt)] \right\} \right] . \qquad (31)$$

The variance of a weighted sum of random variables is equal to the weighted sum of their variances. Therefore since

$$E\{[y_w(nt)]^2\} = \text{mean signal power in wild point} = \sigma_y^2$$

$$E\{[x(nt)]^2\} = \text{mean signal power in sample point} = \sigma_x^2$$

Then,

$$MSE_1 = \frac{1}{T_e S_{in}} \left[ \sum_{n=1}^{C/2} w_{1n}\left(\sigma_y^2 + \sigma_x^2\right) + \sum_{n=C/2}^{C} w_{2n}\left(\sigma_y^2 + \sigma_x^2\right) \right]$$

$$MSE_1 = \frac{\sigma_y^2 + \sigma_x^2}{T_e S_{in}} \left[ \frac{C}{2} w_{1av} + \frac{C}{2} w_{2av} \right] . \tag{32}$$

Since, $T_e = \frac{1}{P_{sw} S_{out}}$, and $S_{in} = C S_{out}$;

$$MSE_1 = \frac{P_{sw}(w_{1av} + w_{2av})}{2} \left[ \left(\sigma_y^2 + \sigma_x^2\right) \right] . \tag{33}$$

Therefore,

$$\text{rms sensor tag error}_1 = \left[ \frac{P_{sw}(w_{1av} + w_{2av})}{2} \left(\sigma_y^2 + \sigma_x^2\right) \right]^{1/2} . \tag{34}$$

The second portion of the total sensor tagging error occurs in the channel in which the wild point originated. Figure D-6 illustrates the data sampling and selection of the sensor losing the transmitted sample point. Using the fan technique, the data would be reconstructed with lines DE and EF which have slopes $m_{11}$ and $m_{12}$, respectively. Because the transmitted sample at E is lost

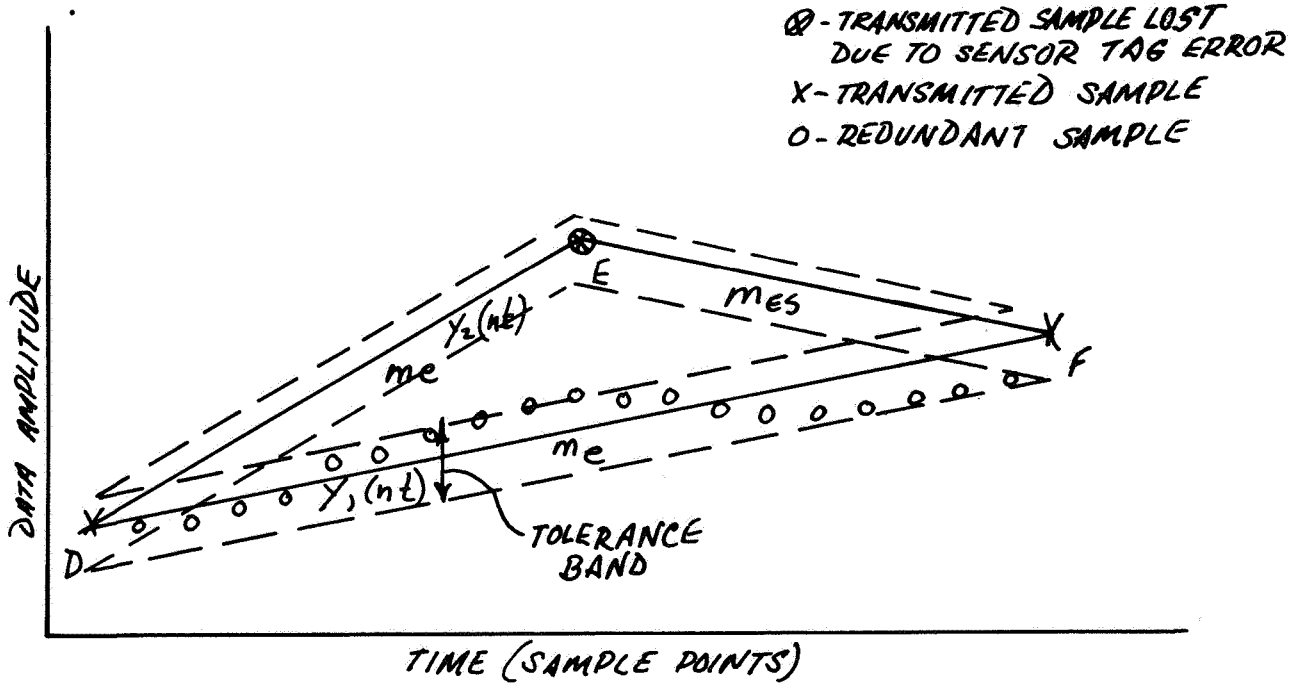


*Figure D-6. Data Sampling and Selection of Sensor Losing a Sample Point at E Because of Sensor Tag Error*

because of sensor tag error, the data segment between transmitted samples D and F is reconstructed using a straight line between the points with a slope $m_e$. Because of the lost sample point, there are 2C sample points in error in the reconstructed data. The mean square error for the sensor losing a sample is given by

$$MSE_2 = \frac{1}{T_e S_{in}} \left[ E\left\{ \sum_{n=1}^{C} L_{1n} [y_2(nt) - y_1(nt)]^2 \right\} + E\left\{ \sum_{n=C}^{2C} L_{2n} [y_2(nt) - y_1(nt)]^2 \right\} \right] \tag{35}$$

where:

$$L_{1n} = |m_{11} - m_e| \cdot nt$$

$$L_{2n} = |m_{12} - m_e| \cdot nt .$$

Expanding the foregoing expression,

$$MSE_2 = \frac{1}{T_e S_{in}} \left[ E\left\{ \sum_{n=1}^{C} L_{1n} [y_2(nt)]^2 \right\} - 2E\left\{ \sum_{n=1}^{C} L_{1n}\, y_2(nt)\, y_1(nt) \right\} + E\left\{ \sum_{n=1}^{C} L_{1n} [y_1(nt)]^2 \right\} + E\left\{ \sum_{n=C}^{2C} L_{2n} [y_2(nt)]^2 \right\} - 2E\left\{ \sum_{n=C}^{2C} L_{2n}\, y_2(nt)\, y_1(nt) \right\} + E\left\{ \sum_{n=C}^{2C} L_{2n} [y_1(nt)]^2 \right\} \right] . \tag{36}$$

Assuming the compression has reduced the natural redundancy to the extent that adjacent sample points are independent, then correlation is zero. Then,

$$MSE_2 = \frac{1}{T_e S_{in}} \left[ E\left\{ \sum_{n=1}^{C} L_{1n} [y_2(nt)]^2 \right\} + E\left\{ \sum_{n=1}^{C} L_{1n} [y_1(nt)]^2 \right\} + E\left\{ \sum_{n=C}^{2C} L_{2n} [y_2(nt)]^2 \right\} + E\left\{ \sum_{n=C}^{2C} L_{2n} [y_1(nt)]^2 \right\} \right] \tag{37}$$

$$= \frac{2\sigma_y^2}{T_e S_{in}} \left[ \sum_{n=1}^{C} L_{1n} + \sum_{n=C}^{2C} L_{2n} \right] \tag{38}$$

$$= \frac{2C\sigma_y^2}{T_e S_{in}} \left[ L_{1av} + L_{2av} \right] \tag{39}$$

$$= 2P_{sw}\sigma_y^2 \left[ L_{1av} + L_{2av} \right] \tag{40}$$

$$\text{rms sensor tag error}_2 = \left[ 2P_{sw}\sigma_y^2 \left( L_{1av} + L_{2av} \right) \right]^{1/2} \tag{41}$$

Combining the expressions in Equations (34) and (41), the total rms error in the expanded data caused by transmission noise disturbing the sensor tagging is

$$\text{Total rms sensor tag error} = \left[ 2P_{sw} \frac{\left( w_{1av} + w_{2av} \right)}{2} \left( \sigma_x^2 + \sigma_y^2 \right) \right]^{1/2} + \left[ 2P_{sw}\sigma_y^2 \left( L_{1av} + L_{av} \right) \right]^{1/2} \tag{42}$$

$$\text{Total rms sensor tag error} = \left[1 - (1-p)^m\right]^{1/2} \left\{ \left[ \left( \frac{w_{1av} + w_{2av}}{2} \right) \left( \sigma_x^2 + \sigma_y^2 \right) \right]^{1/2} + \left[ 2\sigma_y^2 \left( L_{1av} + l_{2av} \right) \right]^{1/2} \right\} \tag{43}$$

If all channels were of equal power, then

$$\text{Total rms sensor tag error} = \left\{ \left[ 1 - (1-p)^m \right] \sigma^2 \right\}^{1/2} \left\{ \left[ w_{1av} + w_{2av} \right]^{1/2} + \left[ 2\left( L_{1av} + L_{2av} \right) \right]^{1/2} \right\} \tag{44}$$

## D.5 TOTAL TRANSMISSION ERROR

For a formatted message structure like that employed in the Gemini and planned for the Apollo telemetry systems, the total transmission error is the combined errors which occur in the actual sample data and the sensor tag or the word location number. The data format framework eliminates the need for individual sample time tags. The timing information is transmitted on only a frame and subframe basis or as required by the ground synchronizing circuitry. Because a formatted message structure is assumed in designing the ACT System, the total error due to transmission noise is the sum of the two foregoing errors. Because the bit errors are independent, the rms error in an actual data sample is likewise independent of the sensor tag error. Therefore, the expression for the total rms transmission error for operation in a compressed mode using the ZOP algorithm is as follows:

$$\text{Total rms transmission error} = \text{rms word error}_C + \text{total rms sensor tag error}$$

$$\text{Total rms transmission error} = \left[ C \frac{(2^{2m} - 1)}{3} pq \right]^{1/2} + 2.4 \left[ \left\{ 1 - (1-p)^m \right\} \sigma^2 \right]^{1/2} \quad (45)$$

Using the FOI-2DF algorithm,

$$\text{Total rms transmission error} = \left[ C \frac{(2^{2m} - 1)}{3} pq \right]^{1/2} + \left\{ \left[ 1 - (1-p)^m \right] \sigma^2 \right\}^{1/2} \left\{ \left[ w_{1av} + w_{2av} \right]^{1/2} + \left[ 2 \left( L_{1av} + L_{2av} \right) \right]^{1/2} \right\} . \quad (46)$$

If channels have different powers and the compression technique is zero-order prediction then,

$$\text{Total rms transmission error} = \left[ C \frac{(2^{2m} - 1)}{3} pq \right]^{1/2} + \left[ 1 - (1-p)^m \right]^{1/2} \left[ \left( \frac{\sigma_1^2 + \sigma_2^2}{2} \right)^{1/2} + \left( 2 \sigma_2^2 \right)^{1/2} \right] . \quad (47)$$

## D.6 REFERENCES


1. Proceedings of the Apollo Unified S-Band Technical Conference, Goddard Space Flight Center (NASA SP-87, 14-15 July 1965.

2. G. Hondros, J. H. Painter, "Unified S-Band Telecommunications Techniques for Apollo, "Volume 1, Functional Description (NASA TN D-2208), March 1965.

3. Secretariat, Range Commanders Council, White Sands Missile Range, New Mexico, "Telemetry Standards" (Revised July 1965). Document 106-65, November 1965.

Appendix E

# SOLUTION OF THE WIENER-HOPF EQUATION FOR SAMPLED DATA FILTERS

The problem of designing a filter which is optimum in a least square sense has been solved by Wiener (Ref. 1 p. 392). Wiener also gives a solution to the optimum sampled data predicting filter (Ref. 2 p. 79), which is in error. This note is an attempt to formulate a more useful expression for the optimum least square sampled data filter. The z-transform was found to be advantageous in this formulation.

Let h(t) be the impulse response of a filter which has input $f_i(t)$ and output $f_0(t)$. These functions are related by the convolution integral

$$f_0(t) = \int_{-\infty}^{\infty} f_i(t - \tau)\, h(\tau)\, d\tau. \tag{1}$$

Let the desired output be $f_d(t)$. One wishes to find a filter $h_{opt}(t)$ which minimizes the mean square error between $f_d(t)$, the desired output, and $f_0(t)$, the actual output.

This minimization leads to the Wiener-Hopf equation (Ref. 1 p. 369).

$$\phi_d(\tau) = \int_{-\infty}^{\infty} \phi(\tau - \sigma)\, h_{opt}(\sigma)\, d\sigma, \qquad \tau \geq 0, \tag{2}$$

where $\phi(\tau)$ and $\phi_d(\tau)$ are correlation functions given by

$$\phi(\tau) = \lim_{T_1 \to \infty} \frac{1}{2T_1} \int_{-T_1}^{T_1} f_i(t + \tau)\, f_i(t)\, dt \tag{3}$$

and

$$\phi_d(\tau) = \lim_{T_1 \to \infty} \frac{1}{2T_1} \int_{-T_1}^{T_1} f_d(t+\tau)\, f_i(t)\, dt. \tag{4}$$

The symbol $T_1$ appears here, rather than T, because T will be used later to indicate sampling interval.

The solution of Equation (2) is (Ref. 1 p. 392)

$$H_{opt}(\omega) = \frac{1}{2\pi\Psi(\omega)} \int_0^\infty e^{-j\omega t}\, dt \int_{-\infty}^{\infty} \frac{\Phi_d(u)}{\overline{\Psi}(u)} e^{jut}\, du\,, \tag{5}$$

where $\Phi_d(\omega)$ is the Fourier transform of $\phi_d(\tau)$ and $\overline{\Psi}(\omega)$ is the complex conjugate of $\Psi(\omega)$.

The transform $\Psi(\omega)$ is found from $\Phi(\omega)$, the Fourier transform of $\phi(\tau)$, by a procedure called factorization (Ref. 1 p. 376). Factorization permits one to find $\Psi(\omega)$ such that

$$\Phi(\omega) = \Psi(\omega)\, \overline{\Psi}(\omega) \tag{6}$$

and

$$\psi(t) = 0, \quad t < 0, \tag{7}$$

where $\psi(t)$ is the inverse Fourier transform of $\Psi(\omega)$.

Now, the case of an optimum predictive filter will be considered for continuous and sampled data signals. If a predictive filter is desired, in Equation (5):

$$\Phi_d(\omega) = \Phi(\omega)\, e^{j\omega\alpha}, \tag{8}$$

where $\alpha$ is the interval of prediction. From Equation (6)

$$\Phi_d(\omega) = \Psi(\omega)\,\overline{\Psi}(\omega)\,e^{j\omega\alpha} \tag{9}$$

and

$$\frac{\Phi_d(\omega)}{\overline{\Psi}(\omega)} = \Psi(\omega)\,e^{j\omega\alpha}. \tag{10}$$

Equation (5) then becomes (Ref. 2 p. 64)

$$H_{opt}(\omega) = K(\omega) = \frac{1}{2\pi\Psi(\omega)} \int_0^\infty e^{-j\omega t}\,dt \int_{-\infty}^{\infty} \Psi(\omega)\,e^{ju(t+\alpha)}\,du. \tag{11}$$

One now desires to find $K(\omega)$ for sampled data signals. Let the input be the sampled data function

$$f_i^*(t) = \sum_{n=-\infty}^{\infty} f(nT)\,\delta(t-nT) \tag{12}$$

where $\delta(t-nT)$ is a delta function

$$\delta(t-nT) = 0, \quad t \neq nT \tag{13}$$

$$\int_{-\infty}^{\infty} \delta(t-nT)\,dt = 1, \tag{14}$$

and T is the sampling interval.

The correlation function is

$$\phi(\tau) = \lim_{T_1 \to \infty} \frac{1}{2T_1} \int_{-T_1}^{T_1} f_i^*(t+\tau)\, f_i^*(t)\, dt. \tag{15}$$

Since

$$f_i^*(t+\tau) = \sum_{k=-\infty}^{\infty} f(kT)\, \delta(t+\tau-kT), \tag{16}$$

$$\phi(\tau) = \lim_{T_1 \to \infty} \frac{1}{2T_1} \sum_{k=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f(kT)\, f(nT) \int_{-T_1}^{T_1} \delta(t+\tau-kT)\, \delta(t-nT)dt. \tag{17}$$

Let

$$T_1 = (N + \tfrac{1}{2})\, T. \tag{18}$$

The properties of delta functions imply that the integral is zero for $n > N$ and $n < -N$, therefore

$$\phi(\tau) = \lim_{N \to \infty} \frac{1}{(2N+1)T} \sum_{k=-\infty}^{\infty} \sum_{n=-N}^{N} f(kT)\, f(nT)\, \delta(\tau+nT-kT). \tag{19}$$

Now, let

$$k = n + m. \tag{20}$$

The correlation function becomes

$$\phi(\tau) = \frac{1}{T} \sum_{m=-\infty}^{\infty} \left[ \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} f(nT+mT)\, f(nT) \right] \delta(\tau - mT)$$

$$= \frac{1}{T} \sum_{m=-\infty}^{\infty} \phi_m \delta(\tau - mT)\,, \tag{21}$$

where

$$\phi_m = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} f(nT+mT)\, f(nT)\,. \tag{22}$$

Factorization (Ref. 1 p. 376) of $\Phi(\omega)$, the Fourier transform of $\phi(\tau)$, gives

$$\Phi(\omega) = \frac{1}{T} \sum_{m=-\infty}^{\infty} \phi_m e^{-jm\omega T}$$

$$= \frac{P(e^{j\omega T})\, P(e^{j\omega T})}{Q(e^{-j\omega T})\, Q(e^{j\omega T})} = \Psi(\omega)\, \overline{\Psi}(\omega)\,, \tag{23}$$

whence

$$\Psi(\omega) = \frac{P(e^{-j\omega T})}{Q(e^{-j\omega T})} = \sum_{m=0}^{\infty} \psi_m e^{-jm\omega T}\,, \tag{24}$$

where P and Q are polynomials. The terms in the expansion for $\Psi(\omega)$ may be obtained from the Fourier integral

$$\psi_m = \frac{T}{2\pi} \int_{-\pi/T}^{\pi/T} \Psi(\omega)\, e^{jm\omega T} d\omega. \tag{25}$$

the inverse Fourier transform of $\Psi(\omega)$ gives

$$\psi(t) = \sum_{m=0}^{\infty} \psi_m \delta(t-mT) , \tag{26}$$

therefore,

$$\psi(t) = 0 , \quad t < 0 \tag{27}$$

as required. Expression of $\Phi(\omega)$ in the rational polynomial form of Equation (23) may first require the use of a method, such as Prony's method (Ref. 3), to approximate $\phi_m$ by a sum of exponentials in m.

Now, one can proceed to find $K(\omega)$ for sampled data functions. From Equation (11)

$$K(\omega) = \frac{1}{2\pi\Psi(\omega)} \int_0^{\infty} e^{-j\omega t} dt \int_{-\infty}^{\infty} \Psi(u)\, e^{ju(t+\alpha)} du. \tag{28}$$

The integral on the right is just the inverse Fourier transform

$$\frac{1}{2\pi} \int_{-\infty}^{\infty} \Psi(u)\, e^{ju(t+\alpha)} du = \psi(t+\alpha). \tag{29}$$

and from Equation (26)

$$\psi(t+\alpha) = \sum_{m=0}^{\infty} \psi_m \delta(t+\alpha-mT) . \tag{30}$$

Because only sampled values are of interest, one may let

$$\alpha = \ell T. \tag{31}$$

Then

$$K(\omega) = \frac{1}{\Psi(\omega)} \int_0^\infty e^{-j\omega t} dt \sum_{m=0}^{\infty} \psi_m \delta(t+\ell T-mT)$$

$$= \frac{1}{\Psi(\omega)} \int_0^\infty e^{-j\omega t} dt \sum_{m=\ell}^{\infty} \psi_m \delta(t+\ell T-mT), \tag{32}$$

for, since $t \geq 0$ in the integral, the omitted terms of the sum are zero.

Letting

$$m = k + \ell \tag{33}$$

in Equation (32), one has

$$K(\omega) = \frac{1}{\Psi(\omega)} \int_0^\infty e^{-j\omega t} dt \sum_{k=0}^{\infty} \psi_{k+\ell} \delta(t-kT)$$

$$= \frac{1}{\Psi(\omega)} \int_{-\infty}^\infty e^{-j\omega t} dt \sum_{k=0}^{\infty} \psi_{k+\ell} \delta(t-kT), \tag{34}$$

due to the fact that the integral is zero for $t < 0$. The indicated Fourier transform of Equation (34) yields

$$K(\omega) = \frac{1}{\Psi(\omega)} \sum_{k=0}^{\infty} \psi_{k+\ell} e^{-jk\omega T}, \tag{35}$$

and Equation (25) gives

$$\psi_{k+\ell} = \frac{T}{2\pi} \int_{-\pi/T}^{\pi/T} \Psi(u) e^{j(k+\ell) uT} du \tag{36}$$

from which

$$K(\omega) = \frac{T}{2\pi\Psi(\omega)} \sum_{k=0}^{\infty} e^{-jk\omega T} \int_{-\pi/T}^{\pi/T} \Psi(\omega) e^{j(k+\ell)uT} du . \tag{37}$$

When T = 1, Equation (37) becomes

$$K(\omega) = \frac{1}{2\pi\Psi(\omega)} \sum_{k=0}^{\infty} e^{-jk\omega} \int_{-\pi}^{\pi} \Psi(u) e^{j(k+\ell)u} du . \tag{38}$$

Equation (38) checks with Wiener's result (Ref. 2 p. 79) except for the absence of the parameter $\ell$ in his equation. Since $\ell = 0$ implies that $K(\omega) = 1$, one would assume that Equation (38) is correct.

The factor $e^{j\omega t}$ appearing in these equations is rather cumbersome. The expressions may be simplified and a certain insight may be gained if one employs the z-transform.

$$z = e^{j\omega T}. \tag{39}$$

Making this substitution in Equation 24, one has

$$\Psi(z) = \frac{P(z^{-1})}{Q(z^{-1})} = \sum_{m=0}^{\infty} \psi_m z^{-m} . \tag{40}$$

This is a Laurent series (Ref. 4 p. 141), the general term of which is

$$\psi_m = \frac{1}{2\pi j} \int \Psi(z)\, z^m \frac{dz}{z} , \tag{41}$$

where the contour integral is taken around the origin. Equation (41) is just an explicit expression for the inverse z-transform. Now, Equation (35) becomes

$$\begin{aligned} K(z) &= \frac{1}{\Psi(z)} \sum_{k=0}^{\infty} \psi_{k+\ell} z^{-k} \\ &= \frac{1}{2\pi j \Psi(z)} \sum_{k=0}^{\infty} z^{-k} \int \Psi(z')(z')^{k+\ell} \frac{dz'}{z'} . \end{aligned} \tag{42}$$

There is an interesting similarity between this equation for the optimum sampled data filter and Equation (28) for the optimum continuous filter. Equation (42) may also be derived from Equation (37) by the substitutions

$$z = e^{j\omega t} \tag{43}$$

and

$$z' = e^{jut} . \tag{44}$$

To derive K(z) from $\Psi(z)$, one may write

$$\Psi(z) = \sum_{m=0}^{\ell-1} \psi_m z^{-m} + z^{-\ell} \sum_{k=0}^{\infty} \psi_{k+\ell} z^{-k}. \tag{45}$$

Solving Equation (45) for the desired expression, one has

$$\sum_{k=0}^{\infty} \psi_{k+\ell} z^{-k} = z^{\ell} \left[ \Psi(z) - \sum_{m=0}^{\ell-1} \psi_m z^{-m} \right]. \tag{46}$$

From Equation (40) and algebra

$$\begin{aligned} K(z) &= \frac{Q(z^{-1})}{P(z^{-1})} z^{\ell} \left[ \frac{P(z^{-1})}{Q(z^{-1})} - \sum_{m=0}^{\ell-1} \psi_m z^{-m} \right] \\ &= \frac{Q(z^{-1})}{P(z^{-1})} \frac{z^{\ell} \left[ P(z^{-1}) - Q(z^{-1}) \sum_{m=0}^{\ell-1} \psi_m z^{-m} \right]}{Q(z^{-1})}. \end{aligned} \tag{47}$$

Finally,

$$K(z) = \frac{z^{\ell} P(z^{-1}) - Q(z^{-1}) \sum_{m=0}^{\ell-1} \psi_m z^{\ell-m}}{P(z^{-1})}, \tag{48}$$

where K(z) is a rational polynomial in $z^{-1}$.

In a manner similar to the derivation of Equation (42) for the predicting filter, the more general optimum sampled data filter may be shown to be

$$H_{opt}(z) = \frac{1}{2\pi j \Psi(z)} \sum_{k=0}^{\infty} z^{-k} \int \frac{\Phi_d(z')}{\overline{\Psi}(z')} (z')^k \frac{dz'}{z'} . \tag{49}$$

the specific z-transform may be found by an equation analogous to Equation (48).

REFERENCES


1. Y. W. Lee, "Statistical Theory of Communication," John Wiley 1960.

2. Norbert Wiener, "Extrapolation, Interpolation, and Smoothing of Stationary Time Series," the MIT Press 1949.

3. T. Y. Young, "Representation and Analysis of Signals Part X. Signal Theory and Electrocardiography," Johns Hopkins University Report, May 15, 1962.

4. Wilfred Kaplan, "Operational Methods for Linear Systems," Addison Wesley 1962.

Appendix F

# FUTURE TELEMETRY SYSTEM REQUIREMENTS

The purpose of studying future telemetry requirements is not to classify and organize the telemetry needs of a single space mission or series of space missions within the constraints of space and ground equipment requirements, mission goals, vehicle destination, or the scientific and physiological experimental program. This task is itself, with different degrees of emphasis and orientation, the major problem considered under various study programs. The intent here is simply to consider all of the forseeable future telemetry data requirements for the purpose of characterizing the data that will be in use.

Among the reports, articles, and papers studied, references 1 through 18, a paper entitled, "Space Data Handling (With Emphasis on Data Compaction)" by M. A. Hyman, IBM, Federal Systems Division, deals most directly with the problem of characterizing future space data requirements. In this paper, the author analyzes the space data handling requirements for the decade 1965-1975 and discusses some recently developed techniques for data compaction. Of particular interest is a table reproduced here as Table F-1, which presents space data according to its quantitative nature (such as scientific, physiological, graphic, etc.) and according to the link over which the data is to be transmitted.

The following categories are enumerated:

a. Tracking and control
b. Scientific
c. Physiological
d. Voice
e. Graphics

*Table F-1. Estimated Data Rates, Compaction Ratios, Channel Capacities*

| | Compaction Ratio (1965) | Ground to Ground | Ground to Near-Space (to Satellites) | Near-Space to Near-Space (Sat-to-Sat) | Ground to Mid-Space (to Moon) | Mid-Space to Mid-Space LEM & Mother) | Ground to Far-Space (to Mars) |
|---|---|---|---|---|---|---|---|
| Tracking and Control | 1 → 3 | | ←100→200 b/s (av)<br>←1000 b/s (peak) | → | 0→20 b/s (av)<br>100 b/s (peak) | 100→200 b/s (av)<br>1000 b/s (peak) | 1→2 b/s (av)<br>10 b/s (peak) |
| Scientific | 1 → 50 | | $10^8 \rightarrow 10^9$ b/day<br>$10^4 \rightarrow 10^5$ b/s (peak) | $10^3 \rightarrow 10^4$ b/s (av) | $10^2 \rightarrow 10^3$ b/s (av)<br>$10^7 \rightarrow 10^8$ b/day | $10^3 \rightarrow 10^4$ b/s (av) | 10→100 b/s (av)<br>$10^6 \rightarrow 10^7$ b/day |
| Physiological | 2 → $10^3$ | | $10^8 \rightarrow 10^9$ b/day<br>$10^5$ b/s (peak) | $10^6 \rightarrow 10^7$ b/day | $10^8 \rightarrow 10^9$ b/day<br>$10^4$ b/s (peak) | $10^8 \rightarrow 10^9$ b/day | $10^7 \rightarrow 10^8$ b/day |
| Voice | 1 → 20 | | $10^4 \rightarrow 3x10^4$ b/s (Real Time) | $10^4 \rightarrow 3x10^4$ b/s (Real Time) | $10^3 \rightarrow 3x10$ b/s (Slow Time) | $10^4 \rightarrow 3x10^4$ b/s (Real Time) | 100→300 b/s (Slow Time) |
| Graphic | 2 → 100 | | ←$10^6$ $10^7$ b/s (TV)<br>←$10^5$ $10^6$ b/picture | → | | | → |
| Channel Capacity for All Uses (1965) | | $5x10^3$b/s (modem)<br>$3x10^4$b/s (voice)<br>$5x10^6$ b/s (TV)<br>per channel | $10^7$ b/s | $5x10^3$ b/s (modem)<br>$3x10^4$ b/s (voice)<br>$3x10^6$ b/s (TV)<br>per channel | $10^4 \rightarrow 10^5$ b/s | $3x10^4$ b/s (voice)<br>per channel | 10 b/s |

Of these categories, it appears that tracking and control, voice and graphics are areas where the characteristics of the data will not change greatly in the near future. This hypothesis permitted the use of current representative data to be used in the simulation program described in this report.

The characteristics of scientific and physiological data for future manned space flights will possibly change from present comparable data requirements. Because the number and variety of experiments will undoubtedly increase, it appears that a corresponding increase in the types and characteristics of the data will follow. Therefore, it will be necessary to continually check all data sources in these two areas and develop compression systems with flexibility to avoid obsolescence.

REFERENCES


1. "Earth-Orbital Mission Definition Document," NASA Contract NASW 1215, 30 July 1965.

2. "Apollo Earth Orbit Preliminary Experiment Payload Packages (Advanced Copy)," 8 March 1965.

3. "NASA Experiment Descriptions for Extended Apollo Earth-Orbit Flights," 15 March 1965.

4. "Ground Operational Support Systems Interface Report," Section I, The Apollo Spacecraft, 15 February 1964.

5. NASA,"Derivation of a Comprehensive Engineering Experiment Program for Manned Earth-Orbital Missions," August 1965.

6. "Summary of Results of Engineering Research Program for Manned Earth-Orbital Missions," August 1965.

7. NASA, "Allocation of Space Resources Aboard an Orbital Laboratory," August 1965.

8. "Analysis and Preliminary Design of Engineering Experiments for Manned Earth-Orbital Missions, Part I," Space Flight Requirements for Satellite Recovery, August 1965.

9. "Analysis and Preliminary Design of Engineering Experiments for Manned Earth-Orbital Mission, Part II," Four Experiment Designs for the Manned Earth-Orbiting Laboratory, August 1965.

10. Jack Cohen and A. Adelman, "Data Management Systems for Manned Space Laboratories," IBM paper, XVth Congress, Athens, Greece.

11. N. N. Berger and R. W. Ulrickson, "Programmable Telemetry for Aerospace Missions," paper presented at 1965 Wescon Show.

12. E. I. Mueldorf, "Communications System for a Manned Orbiting Station," IBM, Federal Systems Division.

13. "Computer Centered Spacecraft Study," IBM Proposal.

14. "Bio-Medical Data Representation for the MORL, A Feasibility Study," IBM Document, August 1964.

15. SCD, "Methods of Manual Analysis of Multisource, Continuously Recorded Bio-Medical Data," 30 June 1963.

16. "MOL Bio-Medical Experiment," Preproposal Definition.

17. "General Human Performance in Space," Experiment P-11.

18. M. A. Hyman, "Space Data Handling (With Emphasis on Data Compaction)," IBM, Federal Systems Division. Presented at the twelfth East Coast Conference on Aerospace and Navigational and Electronics, October 27 through 29, 1965, Baltimore, Maryland.

Appendix G

# A CRITIQUE OF BIT-PLANE ENCODING COMPRESSION

## G.1 INTRODUCTION

Bit Plane Encoding[1] is a source encoding technique in which consecutive samples from a particular sensor are divided into subgroups so that some of the subgroups can be represented in an abbreviated form (compressed) and thus reduce the binary data required to describe the samples. The term bit plane evolves from a cuboid magnetic core memory in which the bits of a given order from the various samples of a sensor are arranged such that they are parallel to each other. Then the $n^{th}$ order bits of each of the sample words form a horizontal plane called a bit plane. Bit plane encoding is accomplished by reading words into the columns of the memory, then reading out and encoding the planes.

## G.2 Characteristics of the Bit-Plane Encoding Algorithm

A representative compression system using bit-plane encoding is shown in Figure G-1. To obtain consecutive samples of any sensor, the PCM output of the analog-to-digital converters is demultiplexed for storage in the central memory. While the data is stored, the monitor makes measurements on each of the bit planes to determine the method of encoding the various planes. The monitor controls the readout of the memory on a bit plane basis, and then selects the encoding operations in accordance with the bit-plane measurements, to be performed by the parallel encoder or the code box. The monitor identifies all monovalued planes and describes them summarily by noting the value assumed by all the bits in the plane. Another operation performed by the monitor is to

[1] R. C. Barker, J. W. Schwartz, "Bit Plane Encoding: A Technique for Source Encoding," IEEE Trans. on Aerospace and Electronic Systems, Vol. AES-2, No. 4, pages 385-392, July 1966.

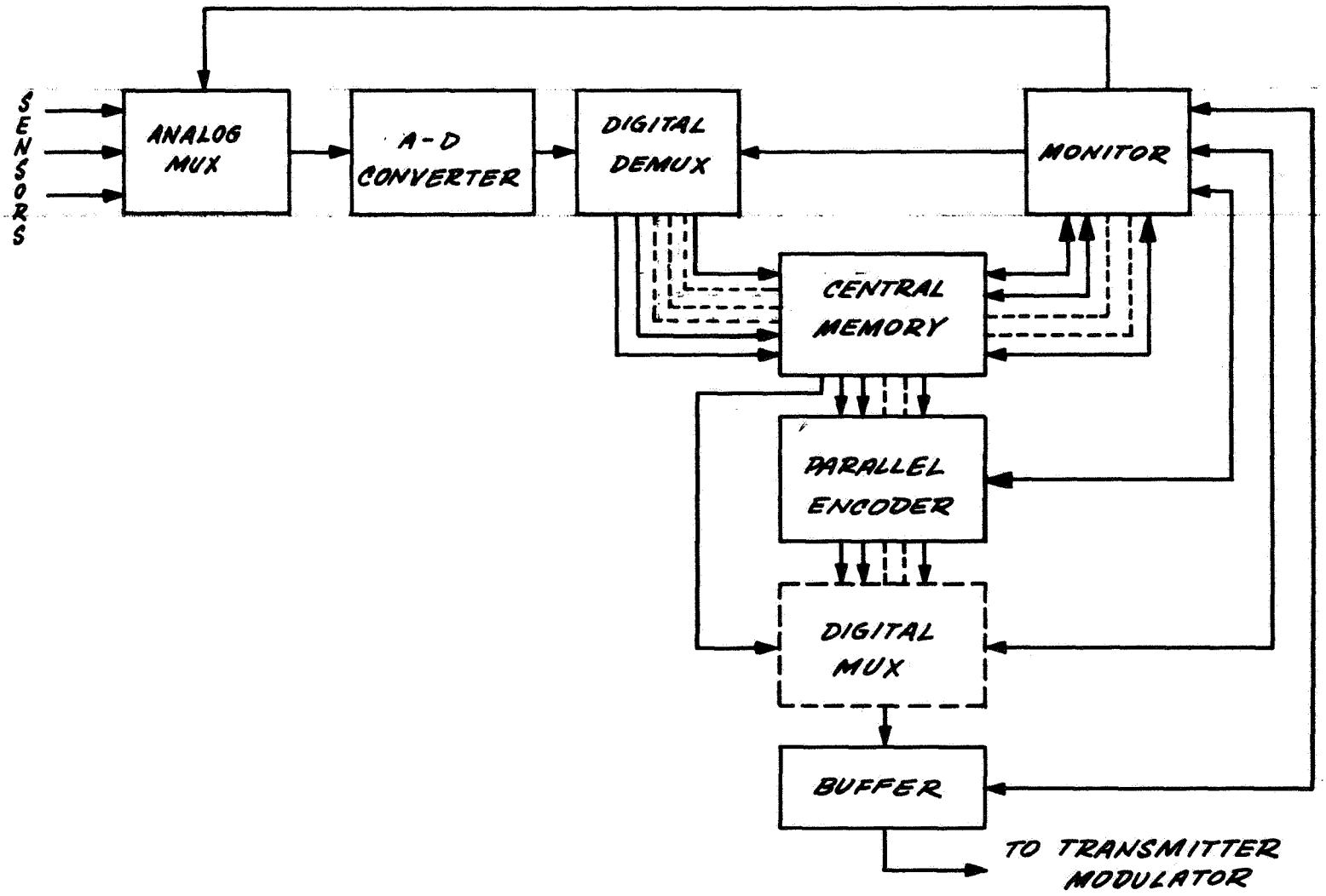


*Figure G-1. Bit-Plane Encoding Compression System*

determine the count-of-changes ($C_c$), which is the number of times adjacent bits in a bit plane are different. For M consecutive samples of a sensor output, run-end encoding is used if:

$$0\ C_c \leq \frac{M}{\log_2 M} - 1.$$

If:

$$C_c \geq \frac{M}{\log_2 M},$$

the bit plane is transmitted bit by bit. Run length encoding may be used in place of run-end, if desired.

The monitor unit, through its measurements on the bit planes, essentially divides each sampled sensor outputs into three groups of data for transmission to the ground. The first data group identifies the operations to be performed on the bit planes such as whether they are monovalued or run-end encoded. The second group designates the value of the first sample in the M consecutive samples being processed. The third data segment called the bit plane data group contains both the bit planes which are run-end encoded and those transmitted bit by bit. The sequencing of the foregoing three data groups into the multiplexer is is controlled by the monitor. Each bit plane which is not monovalued is read out from the memory through the parallel encoder and is either run-end/run-length encoded or left unaltered before being fed into the multiplexer and then to the output buffer.

## G.3 Implementation Problems of Bit Plane Encoding

One of the unique problems in implementing this technique is the choice of group size or the number of consecutive samples of the sensor which should be considered in the coding procedure. According to the developer of this technique,[2] the selection of the group size depends upon the relative and absolute duration of the periods of activity for the sensor (frequency spectrum) the multiplexing formats, transmission delays that can be tolerated, and the fading characteristics of the down-line.

---

[2] Ref. 1, p. 389.

The group size determines the storage required per sensor and the rate at which the sensors may be sequenced into the output buffer. The group size is also used for buffer control. The group size is decreased if buffer underflow exists and is increased when overflow occurs. The group size number of bits in the sample words, data transmission rate, and the number of sensors which can be handled simultaneously by the monitor unit are the principal factors in determining the size of the central memory. For a system required to operate on a large number of sensors with a reasonable large group size, the weight and power penalties may become excessive for the spacecraft environment. This would also be dependent on the time required for the monitor to make all the measurements on each bit plane, possibly store the operations and first value data, and sequence the respective data groups to either the buffer or the digital multiplexer.

Notice the digital multiplexer in Figure G-1 is blocked out with dashed lines. If the first value and operations data are stored in the monitor, it may be possible to eliminate the multiplexer by combining directly the two stored data groups in the monitor with the appropriate channel outputs of the parallel encoder. In essence, the multiplexing function is being done by the monitor unit.

A more conventional design would store the first value and operations data groups in the central memory. The two data groups would be read out of memory in sequence with the third (bit planes) data group out of the encoder. The multiplexer would then perform the sequencing of the three data groups from each sensor in accordance with a preselected transmission format.

The multiplexing problem with the selection of an optimum transmission format is considerably more difficult in the bit-plane encoding compression system than in a polynomial compressor with regard to timing because the frames and subframes must be elastic to accommodate the variation in the number of bits in any of the three data groups. The number of bits in the first value data group is dependent upon the group size selected. The number of bits in the operations data group may be held constant for a given number of operating sensors. When the number of operational sensors is changed, the size of the operations data group will likewise change. The maximum number of bits

in the bit planes (third) data group is dependent on the number of bit planes which simultaneously may exhibit periods of high activity. With the variation in size of the data groups the formatting technique must facilitate the marking of the end of each group and the insertion of the frame synchronization pattern at periodic times as required by the ground recovery circuitry. The timing problem in the spacecraft associated with multiplexing the three data groups could be relieved somewhat by sensor tagging each data group and letting the ground processor combine the appropriate groups.

The number of channels or sensors on which the monitor unit must simultaneously make bit-plane measurements is a tradeoff among memory size, group size of each sensor, the number of operational sensors, time required for the bit-plane measurements, the time necessary to encode or transmit bit by bit a maximum number of bit planes for a given sensor and the maximum transmission rate. Thus, the changing of group size of the number of operating sensors for buffer control directly effects the number of sensors which must be handled simultaneously by the monitor. Timing for buffer control and simultaneous measurements on the bit plane of a number of sensors are simplified if the sensors being handled have identical group size.

The complexity and cost of implementing bit-plane encoding for a spacecraft telemetry system are contingent upon the large amount of storage required in the spacecraft, the inefficiency in processing (multiplexing and demultiplexing operations are required), and the timing control required to handle a practical formatting system, channel identification, and buffer control. There are sufficiently large with present technology to justify ruling out this compression technique for the first generation of the ACT System.