JIS 70-252-1

# TIME AND SOURCE ENCODING FOR MULTIPLEXED COMPRESSED SIGNALS

by
Carlo Joseph Broglio

Thesis submitted to the Faculty of the Graduate School
of the University of Maryland in partial fulfillment
of the requirements for the degree of
Master of Science
1969

VITA

Name: Carlo Joseph Broglio

Permanent address: 15703 Pinecroft Lane, Bowie, Maryland

Degree and date to be conferred: Master of Science, 1969

Date of birth: [redacted]

Place of birth: [redacted]

Secondary education: Pius X 1961

| Collegiate institutions attended | Dates | Degree | Date of Degree |
|---|---|---|---|
| University of Scranton | 9/61 - 6/63 | | |
| University of Detroit | 9/63 - 8/65 | BSEE | 1966 |
| University of Maryland | 9/65 - 6/68 | MS | 1969 |

Major: Electronic Engineering

Minor: Mathematics

Publications: Phase I Simulation Program, NASA Technical Memorandum X-564-67-177, April, 1967

Positions held: Aerospace Technologist; Goddard Space Flight Center, Greenbelt, Maryland

## APPROVAL SHEET

Title of Thesis: Time and Source Encoding for Multiplexed Compressed Signals

Name of Candidate: Carlo J. Broglio
Master of Science, 1969

Thesis and Abstract Approved: ______________________________

Dr. Alan B. Marcovitz
Associate Professor
Department of Electrical Engineering

Date Approved:

## ABSTRACT


Title of Thesis: Time and Source Encoding for Multiplexed Compressed Signals

Carlo J. Broglio, Master of Science, 1969

Thesis directed by: Associate Professor Dr. Alan B. Marcovitz

Sensor data transmitted from a spacecraft to the ground contain much redundancy. Efforts to remove this redundancy, called data compression, tend to be in the form of polynomial predictors in which the data are "curve fitted" to the longest straight line within a prescribed error tolerance. Once compressed, a major problem in reconstructing the data is that of identifying the time of occurrence of each data point.

This thesis proposes five methods for identifying the time of occurrence of a data point. Two methods are simulated on a CDC 3200 computer and the other three are calculated by using a probability of occurrence table measured by applying zero-order and linear predictors to spacecraft data taken from the Orbiting Geophysical Observatory B satellite. A noise-free environment and a prescribed error tolerance were assumed in the process of obtaining the probability of occurrence table. These proposed methods are analyzed and compared by the use of information theory and the results are presented in this Thesis.

PRECEDING PAGE BLANK NOT FILMED.

## TABLE OF CONTENTS

# LIST OF TABLES

## LIST OF ILLUSTRATIONS

# CHAPTER I

## INTRODUCTION

With the advent of satellites, the volume of data obtained and processed each year has increased enormously. For example, one Orbiting Ground Observatory (OGO) transmits to earth over 5 billion binary digits per day, thereby overloading the satellite-to-earth data channel. Consequently, it has become necessary to reduce the data bandwidth necessary for transmission.

One method under consideration is data compression--the removal of redundancy from the data. Redundancy occurs in many forms; *natural redundancy* is that redundancy which is intrinsic to the signal being observed. An example of this is the measurement of temperature, voltage, or current. These parameters tend to remain constant over long periods of time.

*Forced redundancy* is inherent in the design of the spacecraft. Examples of this kind of redundancy are oversampling, subcommutation identification, and spacecraft clocks.

*Correlation redundancy* arises because of its space relationship to other samples. An example of this is television picture data. If one considers the picture an $(n \times m)$ matrix, each interior point is related to the eight samples surrounding it. This type of redundancy is beyond the scope of this paper.

When data compression is used in a telemetry system, the following problems are encountered:

1. How to remove the redundancy,

2. How to control the error introduced by compression,

3. How to identify the time of occurrence of the data received,

4. How to transmit at nonuniform data rates,

5. How to recover the data when errors are made during transmission.

Furthermore, in a multiplexed telemetry system, there is the additional problem of identifying the sensor that produced the received data quantity.

The problem of redundancy removal has been studied on a one-source-one-output basis in the past (References 1 through 10). Some of the solutions considered in the past are polynomial prediction, polynomial interpolation, functional curve fitting, bit plane encoding, and depiction of data in a periodic manner.

The problem of controlling the introduced error must be solved by the user of the telemetry compression system since he is the one who best knows the limitations of the data. The problem of identifying the time of occurrence of the received data sample has also been studied in the past. Some of the solutions have been run-length encoding, ordering the raw telemetry samples, and numbering the raw telemetry samples.

The nonuniform rate of the compressed data stream (known as the buffering problem) has been the subject of many previous studies. It has been considered mainly a design problem in feedback control system theory (References 2, 3, and 4). The recovery of data after transmission is a problem of error detection and correction coding theory.

The problem of identifying sources in a multiplexed data stream has received very little attention in the past. The solutions proposed so far tend to treat this subject as a time identification problem. Furthermore, the problem of whether to compress the data and then multiplex it, or to multiplex and then compress, has not been investigated.

The object of this paper is to study the problems of redundancy removal, time sequence identification, and sensor identification with the main emphasis on the last two problems. The problem of buffering is not considered because it can be treated better in feedback control theory and the problem of transmission errors is not considered because it is handled best within the context of error correction coding; these two theories are beyond the scope of this paper. The previously mentioned solutions do not represent a complete list of the possible methods considered, but are intended as representative in each problem study area.

The problem of redundancy removal is approached using zero-order and first-order polynomial predictors. These methods were chosen because past studies (References 2 through 4) have shown them to be the most promising methods when applied to the sensor data used throughout this study.

The problem of introducing errors by compression was assumed solved by allowing a maximum average error rate of one quantization level. This assumption is meant to be interpreted as a presentation of results at this error level rather than as a solution to the problem of error control.

Five methods of coding time and sensor identification are compared. This paper differs from past work in that it considers a multiplexed data stream, suggests a method of data identification based on an existing data processing system, and is based on actual data from the OGO-B spacecraft now orbiting the earth.

The following considerations are of major importance: simplicity in spacecraft design, minimum changes in the current ground recovery hardware, and minimum changes in current data processing. These considerations are necessary to reduce the cost of converting to a data compression system so that the advantages of data compression will not be overcome by the cost of data recovery.

It is assumed that:

1. The data are in a noise-free environment,
2. The compressor is to be flown on the spacecraft,
3. Data from many experiments are to be multiplexed before transmission,
4. The experiments are mutually independent.

The assumption of a noise-free environment is perhaps unrealistic; however, the nature of the source is best studied under this condition, and it is hoped that the results of this study can be used to solve the noise problem, perhaps through coding.

Two types of compression algorithms were used—the zero-order predictor (ZOP) and the linear predictor (LP). A detailed description of their operation appears later. These two algorithms were applied to data from the OGO-B spacecraft, and the results were evaluated to determine a combination that would allow high compression with reasonable error rates. After completion of this phase of the study, attention was focussed on time and sensor identification. Two methods of time identification were simulated on the CDC 3200 computer, and three others were compared theoretically. The methods simulated on the computer encode the number of samples skipped within the frame matrix as identification, whereas the theoretical methods encode each position of the frame matrix in one case and each source in the other. In all cases, the frame synchronization code is assumed to be transmitted uncompressed. These methods will be discussed in detail later.

The frame matrix was maintained throughout this study because of the simplicity in applying orbit data to the frame sequence to determine the exact position of the spacecraft at the time the data were transmitted.

# CHAPTER II

# THEORETICAL BACKGROUND

## Telemetry

For the purpose of the present research, the telemetry system is a time-sampled digital system with a quantization precision set by a k-bit binary code. The data source is assumed to be some analog function of time which is sampled at a constant rate.

In a time-multiplexed telemetry system, different sources are sampled in a definite order of intermixed, time-shared sampling called *commutation*. The commutator may be mechanical or electronic, and its operation may either be inflexible (according to a predetermined pattern), changeable (according to different patterns, each brought about by ground command) or automatic, depending on the data. The output of a commutator is a sequence of samples from different sources, the pattern of samples repeating in some time period which is typically large in comparison to the time period between samples.

When a time-multiplexed telemetry signal is received, the samples are "sorted" into sources from the time-multiplexed sequence. This process, called *decommutation*, depends on frame synchronization and word synchronization in the multiplexed data sequence for reliable operation. Word synchronization uses an easily recognizable "sync" word that is inserted in the sampling sequence once every period or fractional period of the multiplexed sequence. The details of a simple, multiplexed pattern with its included word synchronization are illustrated in Figure 1.

In Figure 1, the letters A through L signify data sources such as space experiments, attitude sensors, spacecraft subsystem parameters (voltages, currents, temperatures, etc. associated with the spacecraft and not individual experiments), and an on-board clock. The subcommutator count (SCC) word has its lowest value at the beginning of the longest repetitive cycle in the multiplexed data format and has its highest value at the end of this longest cycle, which is called the "main frame." In Figure 1, the main frame constitutes eight rows of the pattern; two main frames are shown. Each row is called a minor frame, and the synchronization word, called a frame sync pattern (FSP), occurs once each minor frame. Source A, occurring three times each minor frame, is said to be *supercommutated*, that is, sampled more than once each minor frame. Source B is sampled once every minor frame. Sources C and D are sampled once every other minor frame, and these are subcommutated—sampled less than once each minor frame. Sources D through L are also subcommutated, and since their subcommutation pattern has the longest period, this pattern determines the subcommutator count.

Usually each major and minor frame are much larger than in this example. For example, the OGO-B telemetry format (Reference 11) has a (128 × 128) major frame matrix, that is, 128 minor frames, each with 128 samples.

WORD POSITION NUMBER

TIME INCREASING

| MINOR FRAME NUMBER | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 1 | SYNC | A | SCC 1 | B | C | A | E | A |
| 2 | SYNC | A | SCC 2 | B | D | A | F | A |
| 3 | SYNC | A | SCC 3 | B | C | A | G | A |
| 4 | SYNC | A | SCC 4 | B | D | A | H | A |
| 5 | SYNC | A | SCC 5 | B | C | A | I | A |
| 6 | SYNC | A | SCC 6 | B | D | A | J | A |
| 7 | SYNC | A | SCC 7 | B | C | A | K | A |
| 8 | SYNC | A | SCC 8 | B | D | A | L | A |
| 1 | SYNC | A | SCC 1 | B | C | A | E | A |
| 2 | SYNC | A | SCC 2 | B | D | A | F | A |
| 3 | SYNC | A | SCC 3 | B | C | A | G | A |
| 4 | SYNC | A | SCC 4 | B | D | A | H | A |
| 5 | SYNC | A | SCC 5 | B | C | A | I | A |
| 6 | SYNC | A | SCC 6 | B | D | A | J | A |
| 7 | SYNC | A | SCC 7 | B | C | A | K | A |
| 8 | SYNC | A | SCC 8 | B | D | A | L | A |

A, B, C, D, E, F, G, H, I, J, K, L = SAMPLED SOURCES

SYNC = SYNCHRONIZATION WORD

SCC = SUBCOMMUTATION COUNT

**Figure 1 – A time-multiplexed telemetry pattern.**

## Data Processing Techniques

After the data are time-multiplexed, they are transmitted to the ground tracking stations and recorded on tape. The first stage of processing is to obtain "bit sync" and "frame sync" from the recorded bit stream. For this purpose, the computer later uses analog-to-digital processors. The output of these processors is called a "buffer tape" that contains the information transmitted to the ground in a computer-acceptable format.

The computer analyzes the FSP to determine the data error quality, thereby providing data quality control. The computer also merges the data with orbit position data calculated from the spacecraft clock and orbit projection programs. These data are then stored on tape in the data archives. This tape, called an *edit tape*, is the input to the next stage of processing, decommutation. In decommutation and subsequent data processing, it is necessary only to locate absolute time as a reference point somewhere within a main frame: the absolute times of all samples can then be derived from their positions in the main frame (see Figure 8a).

## Data Compression

There are several methods for compressing data (Reference 3). Perhaps the simplest is *polynomial prediction*, in which an n-th order polynomial is generated by the compressor using $(n + 1)$ consecutive samples. The next sample is derived by evaluating the polynomial. This prediction is then compared with the current data value. If the current data value is within $\pm K$ of the predicted value, the sample is not transmitted. The value of K is a parameter of the compression method and is chosen by the experimenter, based upon the acceptable peak error.

The simplest of these methods is the zero-order predictor (ZOP). The operation of the ZOP is based on predictions of future samples using a horizontal projection of a zero-order polynomial from the present sample (Reference 4). This method simply adds (or subtracts) the K value, which establishes a peak error, to the present sample. As long as subsequent samples fall within this range, they are considered redundant and are not transmitted (Figure 2). The value of the subsequent sample is then assumed by the receiver to be the same as the present sample and to fall on a horizontal line projected through the sample. When a future sample falls outside this range, it is transmitted as a nonredundant sample. The K value is then set around the new sample, and the process is repeated.

The second method to be considered is *linear prediction* (Reference 1). The linear predictor (LP) uses a first-order extrapolation polynomial of the form

$$y_t = y_{t-i} + \left(y_{t-i} - y_{t-i-1}\right)(i) \pm K \ ,$$

where $y_{t-i}$ is the last sample sent and $y_{t-i-1}$ is the value prior to $y_{t-i}$ assumed by the receiver. Thus, if the previous sample was not transmitted, the predicted value of $y_{t-i-1}$ is used.

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 (Figure 3). If the new data point is within ±K of the predicted value, then that point is not transmitted. If the new data point is outside the aperture, then that

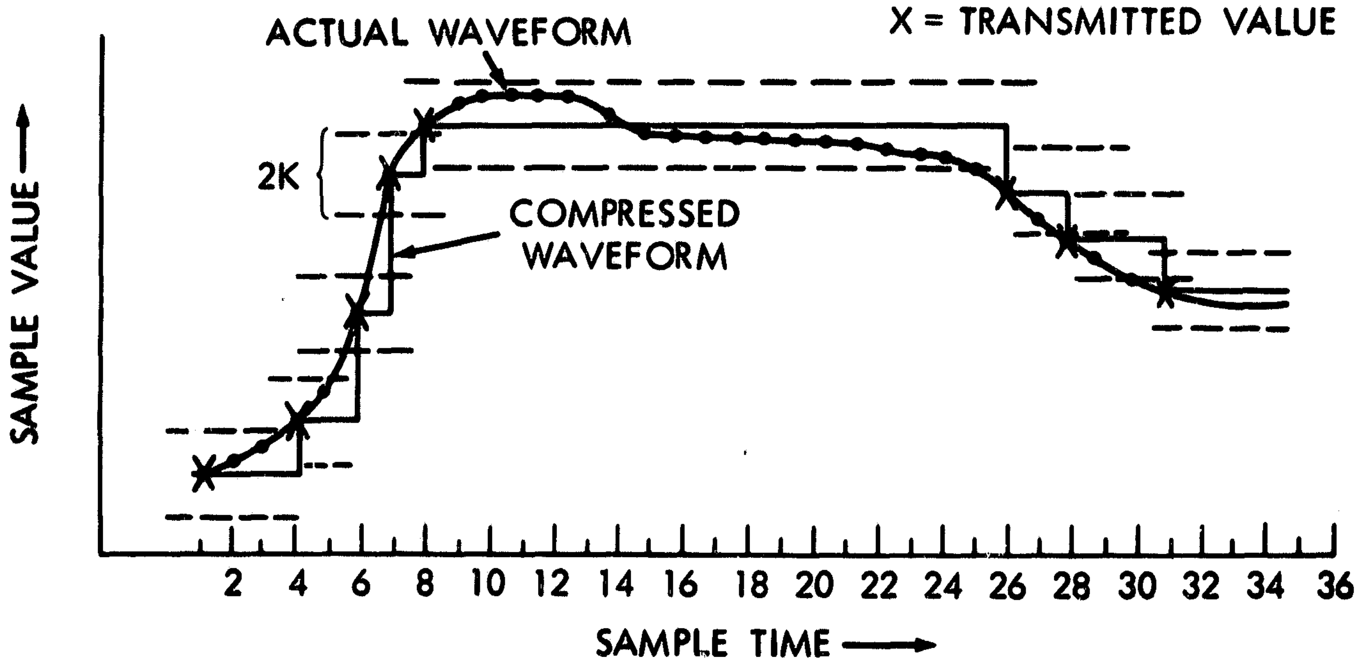


Figure 2 – Zero-order predictor compressed waveform.

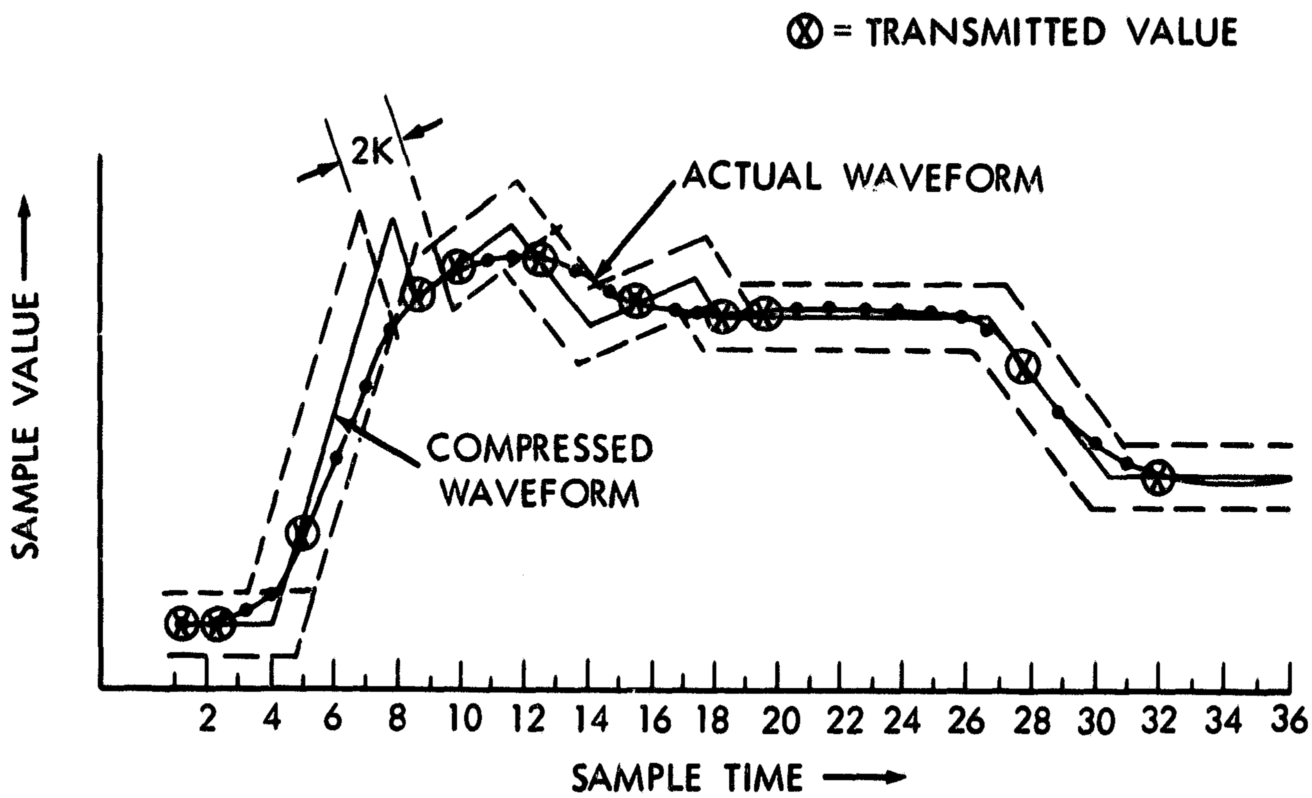


Figure 3 – Linear predictor compressed waveform.

point is transmitted, and a new prediction line is drawn through the present data point that was transmitted and the previously predicted data point.

Assigning a tolerance value K to the compression algorithm subjects the quantized waveform to error in reconstruction. Assume that the value of K is 3 (Figures 2 and 3); then any sample that falls within three or less counts of the predicted value would not be sent. Thus, an error is introduced into the transmitted data value (Reference 3).

The methods of data compression mentioned previously are by no means a complete list of possibilities (see References 12 through 21). They were chosen because of their ease of implementation and data reconstruction. These methods offer the most promise when the impact on ground data processing is considered.

When data compression is used, the continuity of the input data waveform is lost. Hence, a requirement to reestablish the space relationship of the data is imposed on the compressed data stream. This requirement is satisfied by the use of a time identification encoding scheme. The choice of such a scheme fixes the cost of identifying the data.

When a multiplexed data source is considered, the problem increases because now not only the continuity of the data source must be reconstructed, but also the source itself must be identified. Figure 4 shows the telemetry pattern in Figure 1 after compression. The blank slots represent missing data points caused by compression.

In minor frame 1, some means of identifying the fact that word 6 is missing is necessary. This may be done by identifying the word position number of each data word sent, by identifying the source of each data word sent, or by ordering

WORD POSITION NUMBER

TIME INCREASING

| MINOR FRAME NUMBER | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 1 | SYNC | A | SCC 1 | B | C | | E | A |
| 2 | SYNC | | SCC 2 | | D | A | F | |
| 3 | SYNC | A | | | | | G | A |
| 4 | SYNC | | | B | D | A | H | |
| 5 | SYNC | | | | C | A | I | |
| 6 | SYNC | A | | | | | J | A |
| 7 | SYNC | | | | | A | K | |
| 8 | SYNC | | | | | | L | A |
| 1 | SYNC | | SCC 1 | | C | | | |
| 2 | SYNC | A | SCC 2 | B | | | | A |
| 3 | SYNC | | | | | A | G | |
| 4 | SYNC | A | | | D | | | A |
| 5 | SYNC | | | | | | I | A |
| 6 | SYNC | | | B | | A | | |
| 7 | SYNC | A | | | C | | | A |
| 8 | SYNC | | | B | | A | L | |

A, B, C, D, E, F, G, H, I. J, K, L — SAMPLED SOURCES
SYNC — SYNCHRONIZATION WORD
SCC — SUBCOMMUTATOR COUNT

Figure 4 – A received time-multiplexed compressed telemetry pattern.

the data words and identifying the number of words skipped since the last transmitted sample.

Thus, each word sent could have a 3-bit prefix representing a binary number of its position in the minor frame (for example, word 4 of minor frame 1 would be 100A), or it could have a 4-bit source identification followed by a 2-bit repetition factor where necessary: That is, each of the 12 data sources A through L could be represented by a unique 4-bit pattern, and for sources appearing more than once (such as A), the number of the output for that source in that frame could appear. For example, word 8, frame 1, could be prefixed by 000111, where 0001 represents source A, and 11 represents the third output of source A in frame 1.

Finally, the number of words since the last transmitted data word could be encoded into a binary number of fixed length such as four bits, and sent with the next transmitted sample. For instance, in the first appearance of minor frame 8, the telemetry word 7 that is transmitted could be prefixed by 0101. These methods and others will be studied in more detail later.

## CHAPTER III

## THEORY OF OPERATION

Two terms necessary for comparing compression algorithms are raw compression ratio and actual compression ratio. The raw compression ratio is the number of data bits transmitted divided by the number of data bits in the uncompressed data stream. The actual compression ratio is the number of data bits plus identification bits transmitted divided by the number of data bits in the uncompressed data stream. The second value includes the cost of identifying the data.

Five compression schemes will be considered. In all five cases, a set of samples $xi_1, xi_2, xi_3, \cdots xi_n$ in a minor frame is sent; that is, $n \leq 128$ samples of the frame. They are ordered such that $1 \leq i_1 \leq i_2 \leq i_3 \leq \cdots \leq i_n \leq 128$.

In the first three schemes A, B, and C, advantage is taken of the ordering; in the last two schemes D and E, it is not. Hence, from information theory, one would expect schemes A, B, and C to be better, since schemes D and E have a larger range to encode than the other three. That is, it is more probable that some of the 128 words will appear rather than that all 128 will be skipped. This hypothesis will be examined later.

First, the two compression schemes that were submitted to computer simulation will be examined. Both schemes employ the idea of minor frames and word position in the frame in data recovery, as is presently done. Both attempt to identify the time of occurrence in the spacecraft multiplexer rather than at the experiment sensor. Time information is recovered by transmitting every FSP. Also, both methods employ a parameter T which allows T-compressed major frames to elapse, then transmits a full major frame of uncompressed data, which allows all measurements to be reestablished without error. In the case of constants such as voltages and currents, which would be absent for hours, T allows a periodic check on their value. Another use of the parameter T is to increase the redundancy when the noise level of the spacecraft-to-ground communication channel is increased because of electrical storms, solar activity, etc.

In the computer simulation, a 7-bit word was inserted after the FSP to represent the number of words appearing before the next FSP. This required the spacecraft to store an entire frame of data before transmission and minimized the need for ground equipment modification since it enabled the equipment to predict the time of occurrence of the next FSP. This removes some of the burden of data error control from the computer since it can be preprocessed to a limited extent in the frame synchronizer (Reference 22). Since this 7-bit word could be added to any of the schemes to be discussed, it will be omitted from the analysis of the schemes.

The first scheme (A) is designed for minimal analog-to-digital equipment modification. Each transmitted data word in scheme A comprises a 3-bit identification section, a 0- to 7-bit tag, and a 9-bit data value. The 9-bit value is the level of quantization used by OGO-B (Reference 23).

The 3-bit identifier represents a binary count of the number of bits in the variable-length section following it. This enables the data processor to count the number of bits in a word and thereby separate data words (see Figure 5). This method was chosen because it represents only a slight change in the current data processors that are designed for uncompressed data streams.

The 0- to 7-bit tag represents the number of data words missing from the minor frame format since the previous data word. This enables the computer to reconstruct the frame format; further processing may then be done with no change in existing programs. Hence, time and source identification are accomplished in the same way as they are now done.

The second scheme (B) is similar to scheme A except that the word identification is modified. The transmitted data word is now composed of a 1- or 4-bit identifier, a 0- to 6-bit tag and a 9-bit data value. The first bit of the identifier represents whether or not any data words have been omitted from the minor frame format. If its value is zero, then no words are absent. If it is a one, however, the next three bits represent the number of bits in the tag section and also the scale factor for the tag section.

The 0- to 6-bit tag section represents a binary value that must be added to 2 raised to the power represented by the last three bits of the identifier to obtain the number of words skipped. For example, if one word is skipped, the data identifier is 1000; for two words, the data identifier is 10010; and, for 125 words, the data identifier is 1110111101. Analyzing the last case reveals: 1 signifies that some word(s) has been skipped; 110 signifies that six bits are in the tag and that the value of the tag must be added to $2^6$, or 64. The 6-bit tag 111101 is

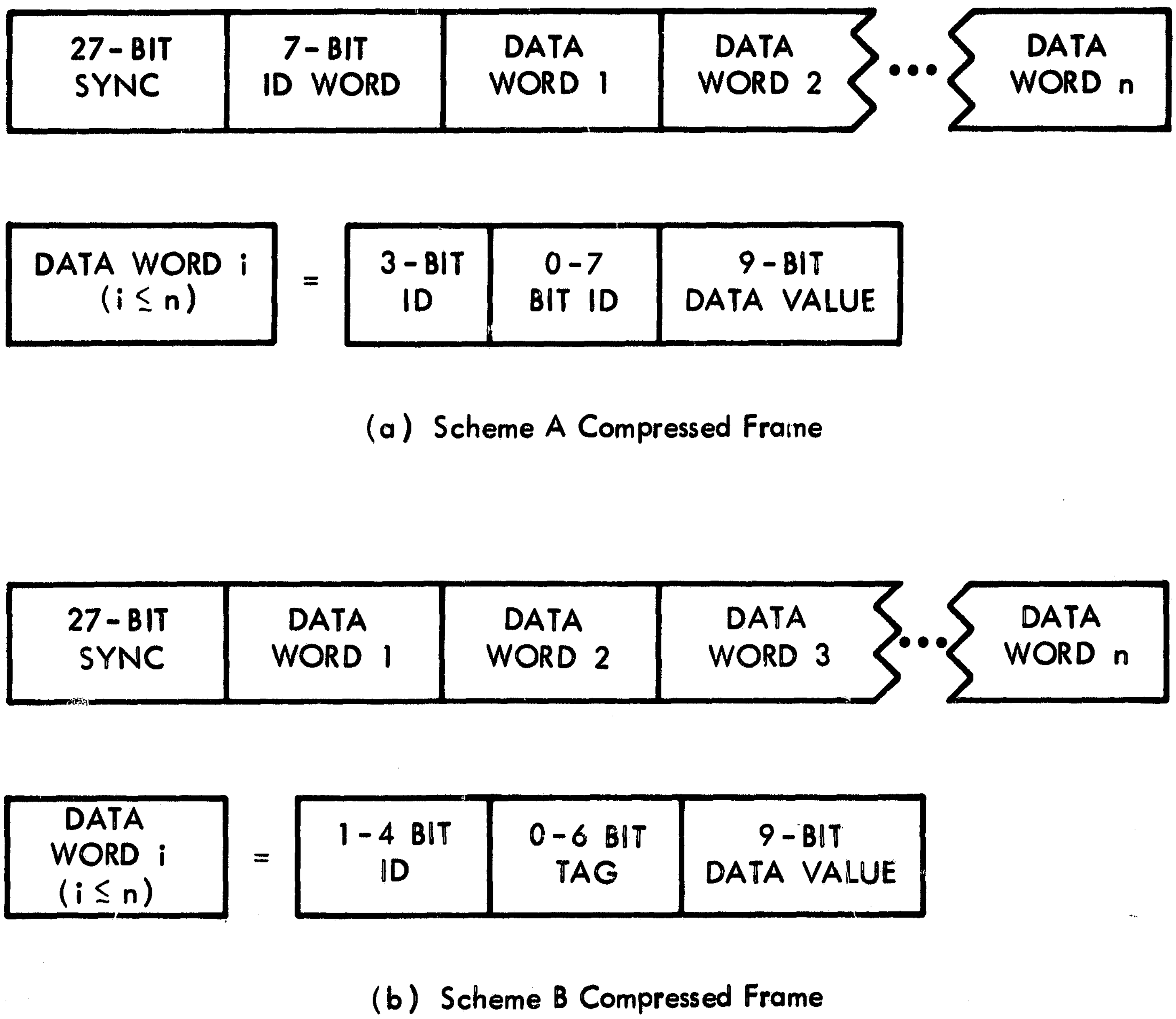


**Figure 5 – Simulated compression schemes.**

decoded as $1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 61$. Thus, $61 + 64 = 125$ words are skipped (see Figure 6).

The implementation of these schemes is exemplified in Figures 7 and 8. In Figure 7, the dashed-line enclosure represents the additional amount of design and development needed in the spacecraft. The dashed line encloses a large portion of the control box, signifying the necessity for increased complexity in this area, and the portion outside the dashed line corresponds to the current design.

Figure 8a represents the current methods for recovery of the transmitted data; Figure 8b shows the increased effort necessary to handle compressed data in the current system. The new box represents merely a subprogram added to the current buffer tape processing programs. Neither scheme will require additional equipment at the receiving station. However, both schemes require minor modifications of the current analog-to-digital data processors.

A theoretical calculation of the worst case, that is, all words of a minor frame present, shows the following actual compression ratios: $C_A = 0.767$ and $C_B = 0.926$, where $C_A$ refers to scheme A and $C_B$ refers to scheme B. These calculations were based on the OGO-B spacecraft format, which was used throughout the study. These schemes are optimal when the probability of not skipping a word is 0.5 for scheme B and 0.125 for scheme A and for both schemes when the probability of skipping one word is 1/16; skipping two or three words is 1/32; skipping four to seven words is 1/64; skipping eight to 15 words is 1/128; skipping 16 to 31 words is 1/256; and skipping 64 to 125 words is 1/1024. Computation of these probabilities showed, however, that the foregoing schemes were not optimal for the data source

| Number of words skipped | Scheme A code | | Scheme B code | | |
|---|---|---|---|---|---|
| | ID | Tag | Flag | ID | Tag |
| 0 | 000 | – | 0 | – | – |
| 1 | 001 | 0 | 1 | 000 | – |
| 2 | 001 | 1 | 1 | 001 | 0 |
| 3 | 010 | 00 | 1 | 001 | 1 |
| | | | | | |
| | | | | | |
| 34 | 110 | 100010 | 1 | 101 | 00010 |
| 35 | 110 | 100011 | 1 | 101 | 00011 |
| | | | | | |
| | | | | | |
| 125 | 111 | 1111101 | 1 | 110 | 111101 |
| 126 | 111 | 1111110 | 1 | 110 | 111110 |

**Figure 6 – Time identification codes for schemes A and B.**

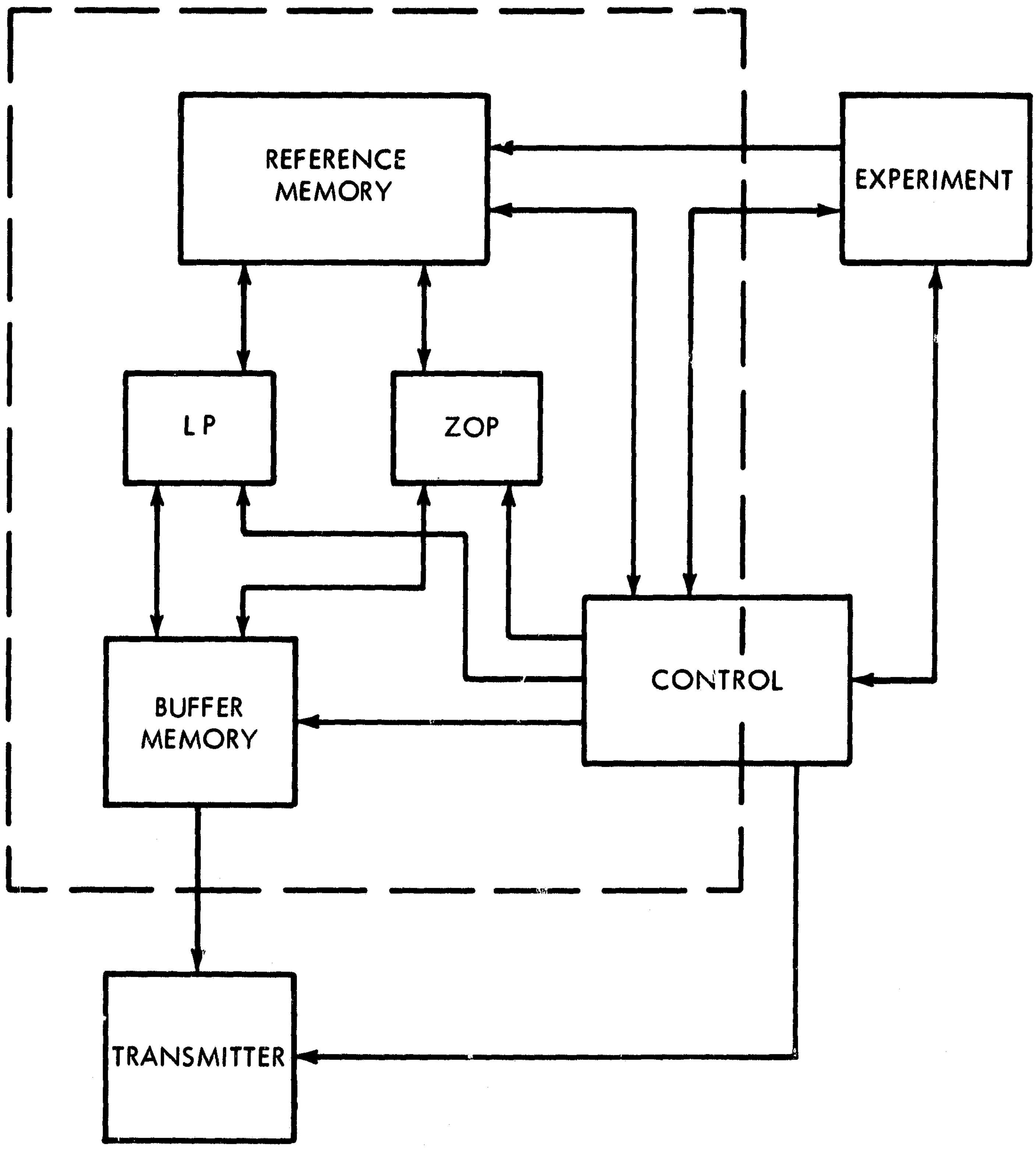


**Figure 7 – A data compression system.**

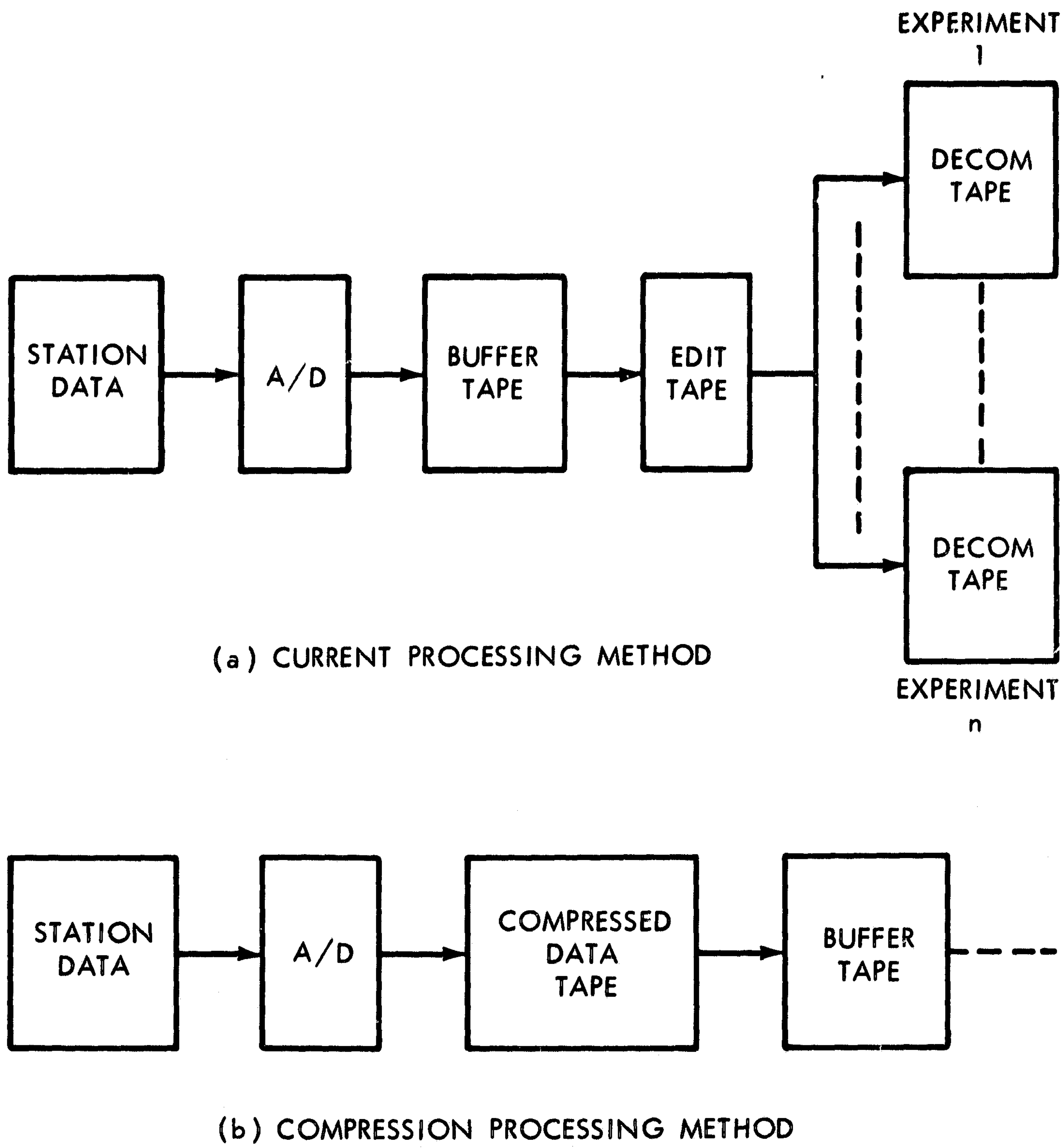


**Figure 8 – Data processing methods.**

used. The results of this computation appear in Table A4. Since the source was neither of these forms, scheme C was derived as an optimal code by using the Huffman method (Reference 24). The probabilities used were derived from the word compression ratios measured by the computer for schemes A and B. Hence, scheme C represents an optimal code for methods A and B. Scheme C was not simulated on the computer; its derivation appears in Appendix A. Figure 9a shows a frame format using scheme C as the time identification algorithm.

The last two theoretical methods, which were not simulated on the computer, as mentioned earlier, will be described next. The first method (D) employs a probability of occurrence table to derive an optimum code for each word of the frame (see Appendix B). This table is derived by observing the raw compression ratio for each word. The data words of the minor frame are assumed to be independent in their occurrence although compression is done on an experimental basis. Hence, if an experiment's output occurs in more than one word, then all words belonging to the experiment are compressed as a single source. Thus, each output from the source is assumed to be independent of the previous output from that source.

The probability of occurrence for a word is found by: $\left(1/C_{Ri}\right) = P_{Ai}$, where $C_{Ri}$ is the raw compression ratio for word i and $P_{Ai}$ is the probability that word i occurs in a frame. Since these values represent the word activity, they do not sum to unity. The table may be normalized by dividing each value by

$$\sum_{i=1}^{128} P_{Ai}$$

The derivation of the code from this table is described in Reference 24.

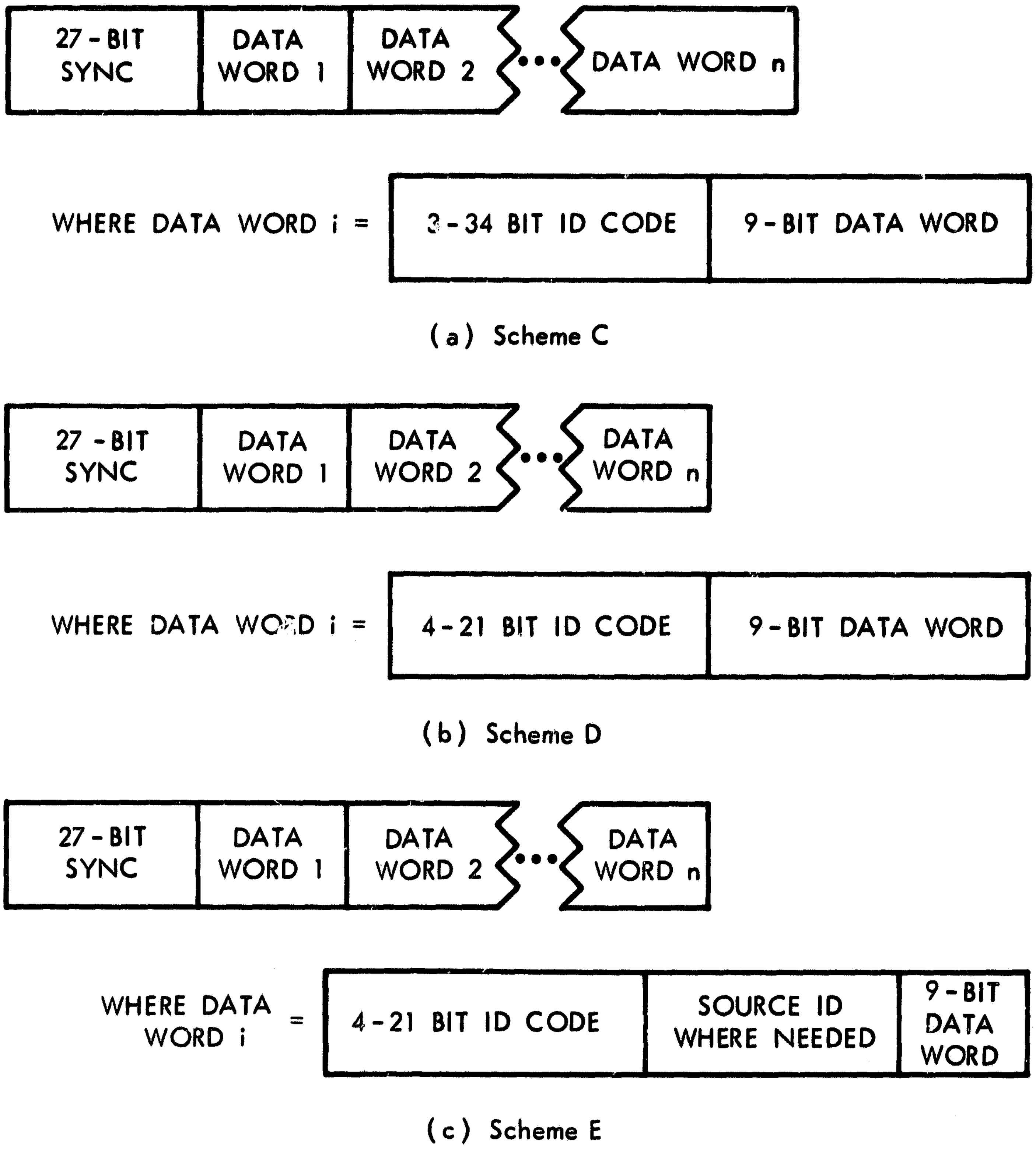


**Figure 9 – Data formats for compression schemes C, D, and E.**

Now each word has its own label that defines it every time the word appears (see Figure 9b). Once the word is defined, the frame can be reconstructed, and processing is identical to current methods.

The last method (E) attempts to find an optimum code for each experiment sensor, based on a probability of occurrence table for the experiment sensor (see Appendix C). This table is found by summing all the compressed output bits for each sensor and dividing by the total number of bits for that sensor in the uncompressed data stream, yielding a raw compression ratio for each sensor. The probability of occurrence is then found as in scheme C, and the code identification derived. As each word is transmitted, as required by the compression algorithm, this identification tag, modified where necessary, is also sent (see Figure 9c).

For sensors with more than one word per frame, an ambiguity can arise concerning which word is being sent. This problem is increased when the position of the sensor words in the main frame is such that the words are close together. In order to minimize the possibility of confusing sensor outputs, additional bits are added to the foregoing code when the possibility of confusing the outputs is greater than 0.6 percent. Hence the frame can again be reconstructed but at the expense of increasing the reconstruction program complexity.

# CHAPTER IV

## RESULTS AND DISCUSSION

The first phase of the study shows that the ZOP achieves a higher raw compression ratio with less errors than does the LP, for most of the sensors. These values were derived bv subjecting the real spacecraft data output to each compression algorithm for K values, ranging from zero to ten over the same time period. The K values were chosen in such a manner as to provide reasonable error rates (less than one quantization level RMS error) while yielding good compression. These K values are shown in Table A1 along with the type of algorithms used.

Table A2 represents the RMS error introduced for each data word in the minor frame. Note that data words 1, 2, 3, 97, 98, and 99 do not appear here. Words 1, 2, and 3 are the FSP, which is not compressed, and words 97, 98, and 99 are the subcommutator data words, which represent many different sensors. Also notice that certain words appear with K values of zero. These words represent the spacecraft status, clock, binary subcommutator counter, and certain experiments for which no transmission errors can be allowed. These data words are predictable or constant most of the time.

Having arbitrarily set an average error level of less than one quantization level, it will be assumed to be a reasonable one. This fixes the peak error rates for the purposes of this study by selecting K values for each sensor. However, the experimenter must choose these K values because he has a clearer understanding of the limitations of his instruments and the usefulness of his data.

Now the problem becomes one of choosing an identification scheme for each data point. For the data used, the five previously described schemes show the following overall compression ratios: scheme A = 5.47, scheme B = 5.58, scheme C = 5.97, scheme D = 5.58, and scheme E = 5.55.

The foregoing results for schemes A and B were obtained by computer simulation. In order to compare schemes A and B with the other schemes, a theoretical calculation was accomplished by use of the following equation: the average number of identification bits for a particular word, k, equals $(n_i \times P_i)$ 4000, where $n_i$ represents the number of bits needed to encode the fact that i words were skipped prior to word k, and $P_i$ is the probability of skipping i words prior to word k; $P_i$ is found from

$$P_i = P_{Ak}\left[1 - P_{A(k-1)}\right]\left[1 - P_{A(k-2)}\right] \cdots \left[1 - P_{A(k-i+1)}\right]\left[P_{A(k-i)}\right]$$

where $P_{Ak}$ is the probability of occurrence for word k and is equal to $1/C_k$, where $C_k$ is the raw compression ratio of word k. The number 4000 represents the number of minor frames over which the sample was taken.

The results of this calculation show $C_A$ = 5.44 and $C_B$ = 5.55—close agreement with the simulated results. These values are found under the assumption that the probability of occurrence of a particular word is independent of the probability of the word just prior to it. This assumption seemed to be valid since the computed compression closely agrees with the simulated ones. Since schemes A and B are not optimal, an optimal code was derived for the data source. The data for this derivation and the code appear in Appendix A. Scheme C represents the result of calculating the compression for this code. As expected, this scheme shows an improvement in compression over the nonoptimal methods.

The hypothesis suggested earlier that schemes A, B, and C are better than schemes D and E is not completely valid. Although the optimum method used in scheme C did yield a slight improvement over schemes D and E, the improvement was not significant. On the other hand, schemes A and B, since they were not optimum, yielded a poorer compression ratio than schemes D and E. This is explained by the fact that the probability distribution of the number of words skipped was more evenly distributed than expected.

Since much of the past work on data compression concerned time-identification of a single source, the last two schemes were used to compare the effects of carrying this work over to a multiplexed source. In scheme D, each word of the minor frame is treated as a data source for time-identification purposes. In scheme E, each sensor is identified separately in the multiplexed data stream.

The data for these derivations and the respective codes appear in Appendix B for scheme D and Appendix C for scheme E. These schemes are representative of past study on a single-source model, and they yield about the same compression as schemes A and B. Further, when scheme C is considered, it is seen that schemes D and E are suboptimal.

Also in scheme E, one encounters the problem of ambiguous data since a sensor with more than one output in a minor frame may appear twice with no other sensor data separating them in the compressed data stream. Hence, when this source is received, one may not know which output the data represent. The result quoted for scheme E represents a probability of ambiguity of 0.006. It is assumed that this is an acceptable level of ambiguity. Table C2 represents the effect on compression of removing individually those sensors that have a possible ambiguous output. Table C2 also shows the different probabilities of ambiguity associated with each ambiguous sensor.

# CHAPTER V

# CONCLUSIONS

The use of premultiplexer compression offers promise because the nature of the source is most prominent at this point. The need for transmitting only the most meaningful information is dramatized by the present and foreseen telemetry limitations at planetary distances. With reasonable ground facilities, an 8-foot antenna reflector would permit less than 50 bits per second to be transmitted to earth from a spacecraft at 5 astronomical units distance. Even with foreseen improvements in the telemetry system, this figure would only increase to a few thousand bits per second. Since such a telemetry system cannot be allowed to overload with redundant data, some form of compression will be essential.

Two types of compression algorithms were studied, the ZOP and the LP. Table A1 shows that the ZOP was heavily favored (only two LP's out of 128). Actually this figure is deceptive since Table A1 refers to the frame matrix, and several sensors appear more than once in this matrix. A more realistic figure is two LP's out of 89 sensors.

Hence, the cost of implementing an LP seems to be too great, considering its usage. It would be better to sacrifice compression for this reduction in cost. This argument is strengthened further by the fact that both sources that use an

LP are binary counters and hence generate a saw-tooth waveform. Thus, they only need to be transmitted at their zero crossing points. It would be less expensive to implement such a transmitting function than an LP compressor.

Notice also in Table A1 that words 36, 101, and 107 have infinite K values. This is true because word 101 is a fill word in the OGO-B frame format used to keep the frame matrix constant; words 36 and 107 were stipulated by the experimenter as useless to him at 64,000 bps transmission, which was the case here. Thus, an advantage of a compression system is to remove useless data from the transmitted data stream.

This may seem a trivial case. Actually it is not since generally not all experiments on a spacecraft of this type are turned on at all times. The reasons for this are basically three-fold. First, because of the design of certain experiments and the need for saving weight in spacecraft design, the drain on the spacecraft power supply would be too great to permit continuous operation. Secondly, because of the nature of certain experiments and the spacecraft orbit, it is impossible to derive useful data throughout the entire orbit.

The third but less important reason is that because of the unknown factors of space parameters, certain experiments, when flown for the first time, are found to be designed below their useful range. An example of this was Dr. Van Allen's first flight package designed to measure energetic particles. Unaware at the time of the presence of the now famous belts, his instruments went to their maximum value and remained.

Five data methods compression have been proposed. In choosing which one to apply, the cost of implementation versus the compression must be kept in mind.

Implementing schemes C, D, and E would require the development of an entirely new system in the analog-to-digital phase of the processing. Also, extensive reprogramming would have to be done to handle the compressed data in these forms. The spacecraft design would be very complex compared to the current design. Therefore, long development times with great initial cost would be expected.

Schemes A and B, on the other hand, take full advantage of the current data processing programs since only a subprogram is needed to get back to the frame matrix. The frame matrix is important at present because of its usage in time-correlation of the experimenter's data relative to the orbit data. Also, it is important because existing edit and decom programs are based on it.

Furthermore, scheme A could be handled on existing equipment without modification, and scheme B requires only minor modifications. Scheme C, which represents an optimum coding scheme for schemes A and B, would require extensive modification. The slight gain in compression that it offers doesn't warrant its usage in a practical system.

Since the initial cost of an untried data compression system is one of the major deterrents to its usage, schemes A and B are proposed as a practical, first step toward more sophisticated methods. Their moderate initial cost would allow studies of compression algorithms to be made on the spacecraft. Then when optimum data compression methods are found, a gradual change to better time encoding schemes could follow. Thus, the development of better ground systems could be spread out over years while the advantages of data compression were being used in a somewhat more limited sense.

The results show that encoding the number of samples skipped is at least equivalent to encoding each source when considering a multiplexed data stream. In the optimal case, this method is slightly better than source identification, and the ease of implementing it compared to identifying each source, particularly for schemes A and B, leads to the conclusion that this type of time identification is better suited to multiplexed data.

Further, it has been shown that schemes A and B are at least equivalent to the more optimum methods; i.e., the improvement in compression of the optimum methods (C, D, E) is not significant. This fact is of major importance when it is considered that the latter three methods are very dependent on the probability distribution of the data, which may not be known in advance.

## APPENDIX A

## DERIVATION OF THE COMPRESSION RATIO FOR SCHEME C

The data of Table A4 were obtained from the first phase of the study, the selection of K values and compression algorithms. The total number of data bits needed to be output over 4000 frames was found to be 554,274. The total number of data bits in 4000 frames is 4,608,000, yielding a raw compression ratio of 8.31.

Table A4 represents the normalized, ordered probability of skipping the number of words in the leftmost column. These probabilities were found from the equation

$$P_k = \frac{\sum_{i=1}^{128} P_{ki}}{\sum_{k=1}^{128} P_k} ,$$

where $P_{ki}$ equals the probability of skipping k words prior to word i; $P_{ki}$ is found from

$$P_{ki} = P_{Ai}\left[1 - P_{A(i-1)}\right]\left[1 - P_{A(i-2)}\right]\cdots\left[1 - P_{A(i-k-1)}\right]P_{Ak} ,$$

where $P_{Ak}$ is the probability of occurrence for word K and is equal to $1/C_k$, where $C_k$ is the raw compression ratio of word K.

The identification code is derived by use of the Huffman Algorithm from Noiseless Coding Theory. The code and the associated number of identification bits are in Table A4.

The net cost of identifying the data points is found from the summation

$$\sum_{i=1}^{128} n_i P_{ki} 4000 .$$

This total is added to the number of data bits, yielding

$$554,274 + 218,094 = 772,368$$

total bits necessary for transmission.

Hence, the compression is computed from

$$C_c = \frac{4,608,000}{772,368} = 5.96 .$$

Table A1

K Values and Compression Algorithms for the OGO-B Frame Matrix

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | - | 0 | 0 | 0 | 1Z | 1Z | 1L | 1Z | 7Z | 2Z |
| 1 | 7Z | 0Z | 3Z | 1Z | 7Z | 7Z | 10Z | 1Z | 1Z | 1Z |
| 2 | 1Z | 1Z | 1Z | 1Z | 1Z | 1Z | 7Z | 3Z | 1Z | 1Z |
| 3 | 2Z | 4Z | 6Z | 7Z | 7Z | 10Z | ∞ | 7Z | 7Z | 10Z |
| 4 | 2Z | 3Z | 7Z | 1Z | 2Z | 7Z | 4Z | 7Z | 7Z | 1Z |
| 5 | 1Z | 1Z | 1Z | 1Z | 2Z | 7Z | 7Z | 7Z | 0Z | 0Z |
| 6 | 1Z | 7Z | 7Z | 10Z | 7Z | 0L | 0Z | 0Z | 3Z | 2Z |
| 7 | 2Z | 7Z | 2Z | 3Z | 7Z | 3Z | 3Z | 1Z | 3Z | 5Z |
| 8 | 3Z | 1Z | 3Z | 2Z | 1Z | 7Z | 7Z | 7Z | 10Z | 1Z |
| 9 | 1Z | 1Z | 2Z | 2Z | 2Z | 4Z | 6Z | - | - | - |
| 10 | 1Z | ∞ | 4Z | 7Z | 1Z | 3Z | 7Z | ∞ | 3Z | 3Z |
| 11 | 4Z | 7Z | 7Z | 1Z | 1Z | 1Z | 1Z | 7Z | 7Z | 10Z |
| 12 | 7Z | 1Z | 1Z | 1Z | 1Z | 2Z | 3Z | 2Z | 2Z | - |

Z = ZOP algorithm
L = LP algorithm

Table A2

RMS Error Table in Quantization Levels

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | - | 0 | 0 | 0 | 0.114 | 0.418 | 0 | 0.419 | 1.056 | 0 |
| 1 | 1.085 | - | 0 | 0.122 | 1.15 | 1.22 | 1.49 | - | - | 0 |
| 2 | 0.024 | 0.103 | 0.352 | 0.414 | 0.0272 | 0.16 | 1.22 | 0.424 | 0.315 | 0.454 |
| 3 | 0.222 | 0.67 | 0.899 | 0 | 0 | 0 | - | 1.305 | 1.299 | 1.457 |
| 4 | 0.479 | 0 | 1.22 | 0.01582 | 0 | 1.193 | 1.06 | 1.148 | 1.33 | 0.164 |
| 5 | 0.0134 | 0.0126 | 0.162 | 0.486 | 0.364 | 0.67 | 0.825 | 0.345 | 0 | 0 |
| 6 | 0 | 1.29 | 1.25 | 1.455 | 1.113 | 0 | 0 | 0 | 0.564 | 0.1395 |
| 7 | 0.5 | 0.419 | 0.585 | 0 | 1.145 | 0.54 | 0 | 0.126 | 0.525 | 0.296 |
| 8 | 0.377 | 0.0475 | 0.308 | 0.2318 | 0.0725 | 1.015 | 1.3 | 1.285 | 1.456 | 0.269 |
| 9 | 0.01582 | 0.491 | 0.156 | 0.0675 | 0.372 | 0.688 | 0.915 | - | - | - |
| 10 | 0.498 | - | 0.685 | 1.089 | 0.0174 | 0 | 1.08 | - | 0 | 0.528 |
| 11 | 1.185 | 1.17 | 1.286 | 0.01582 | 0.354 | 0.418 | 0.0524 | 1.39 | 1.312 | 1.528 |
| 12 | 1.023 | 0.465 | 0.05 | 0.0475 | 0.0593 | 0.431 | 0.42 | 0.295 | 0.0226 | - |

Table A3

Raw Compression and Probability of Occurrence for Each Data Word

| Data Word | Compression | Probability | Data Word | Compression | Probability |
|---|---|---|---|---|---|
| 1 | 1.00 | 1.000000 | 25 | 19.13 | 0.052270 |
| 2 | 1.00 | 1.000000 | 26 | 7.42 | 0.134770 |
| 3 | 1.00 | 1.000000 | 27 | 210.52 | 0.004750 |
| 4 | 2000.00 | 0.000500 | 28 | 5.81 | 0.172100 |
| 5 | 4000.00 | 0.000250 | 29 | 26.84 | 0.037257 |
| 6 | ∞ | 0.000000 | 30 | 4000.00 | 0.000250 |
| 7 | 4000.00 | 0.000250 | 31 | 4000.00 | 0.000250 |
| 8 | 8.31 | 0.120300 | 32 | 95.23 | 0.010500 |
| 9 | 8.06 | 0.124100 | 33 | ∞ | 0.000000 |
| 10 | 9.87 | 0.101317 | 34 | ∞ | 0.000000 |
| 11 | ∞ | 0.000000 | 35 | 27.39 | 0.036510 |
| 12 | 60.60 | 0.016501 | 36 | ∞ | 0.000000 |
| 13 | 16.06 | 0.062266 | 37 | 8.43 | 0.118620 |
| 14 | 7.44 | 0.134408 | 38 | 1.17 | 0.854700 |
| 15 | 1.13 | 0.884955 | 39 | 6.15 | 0.162600 |
| 16 | 6.51 | 0.153610 | 40 | 4000.00 | 0.000250 |
| 17 | ∞ | 0.000000 | 41 | 9.75 | 0.102564 |
| 18 | ∞ | 0.000000 | 42 | 9.85 | 0.101522 |
| 19 | 2000.00 | 0.000500 | 43 | 4000.00 | 0.000250 |
| 20 | 210.52 | 0.004750 | 44 | 23.25 | 0.043010 |
| 21 | 4000.00 | 0.000250 | 45 | 6.72 | 0.148810 |
| 22 | ∞ | 0.000000 | 46 | 7.60 | 0.131578 |
| 23 | 31.49 | 0.031756 | 47 | 2.34 | 0.427350 |
| 24 | 4000.00 | 0.000250 | 48 | 1.96 | 0.510200 |

Table A3—Continued

Raw Compression and Probability of Occurrence
for Each Data Word

| Data Word | Compression | Probability | Data Word | Compression | Probability |
|---|---|---|---|---|---|
| 49 | 17.62 | 0.056753 | 73 | 12.34 | 0.081037 |
| 50 | 20.00 | 0.050000 | 74 | 9.75 | 0.102564 |
| 51 | 21.62 | 0.046253 | 75 | 75.47 | 0.013250 |
| 52 | 800.00 | 0.001250 | 76 | 31.25 | 0.032000 |
| 53 | 108.10 | 0.009250 | 77 | 16.00 | 0.062500 |
| 54 | 4000.00 | 0.000250 | 78 | 43.47 | 0.023004 |
| 55 | 4.11 | 0.243300 | 79 | 30.30 | 0.033003 |
| 56 | 3.48 | 0.287360 | 80 | 7.72 | 0.129530 |
| 57 | 17.31 | 0.057770 | 81 | ∞ | 0.000000 |
| 58 | 86.95 | 0.011500 | 82 | 1000.00 | 0.001000 |
| 59 | 86.95 | 0.011500 | 83 | ∞ | 0.000000 |
| 60 | 13.60 | 0.073529 | 84 | 2000.00 | 0.000500 |
| 61 | 8.6 | 0.116279 | 85 | 6.42 | 0.155763 |
| 62 | 1.16 | 0.862068 | 86 | 7.47 | 0.133868 |
| 63 | 11.39 | 0.087796 | 87 | 1.15 | 0.869565 |
| 64 | 7.18 | 0.139275 | 88 | 7.98 | 0.125313 |
| 65 | 62.50 | 0.016000 | 89 | 36.36 | 0.027502 |
| 66 | ∞ | 0.000000 | 90 | 400.00 | 0.002500 |
| 67 | ∞ | 0.000000 | 91 | 4000.00 | 0.000250 |
| 68 | 9.66 | 0.103500 | 92 | 4000.00 | 0.000250 |
| 69 | ∞ | 0.000000 | 93 | 4000.00 | 0.000250 |
| 70 | 4000.00 | 0.000250 | 94 | ∞ | 0.000000 |
| 71 | 1333.33 | 0.000750 | 95 | ∞ | 0.000000 |
| 72 | 800.00 | 0.001250 | 96 | 102.56 | 0.009750 |

Table A3—Concluded

Raw Compression and Probability of Occurrence
for Each Data Word

| Data Word | Compression | Probability | Data Word | Compression | Probability |
|---|---|---|---|---|---|
| 97 | 4.72 | 0.211800 | 113 | 2000.00 | 0.000500 |
| 98 | 3.79 | 0.263800 | 114 | 4000.00 | 0.000250 |
| 99 | 3.99 | 0.250600 | 115 | 19.23 | 0.052002 |
| 100 | 37.73 | 0.026504 | 116 | 285.71 | 0.003500 |
| 101 | $\infty$ | 0.000000 | 117 | 7.44 | 0.134408 |
| 102 | 57.97 | 0.017250 | 118 | 1.13 | 0.884955 |
| 103 | 6.96 | 0.143673 | 119 | 6.75 | 0.148148 |
| 104 | 19.13 | 0.052273 | 120 | 7.85 | 0.127388 |
| 105 | 10.41 | 0.096061 | 121 | 285.71 | 0.003500 |
| 106 | 8.84 | 0.113122 | 122 | 285.71 | 0.003500 |
| 107 | $\infty$ | 0.000000 | 123 | 285.71 | 0.003500 |
| 108 | 86.95 | 0.011500 | 124 | 266.66 | 0.003750 |
| 109 | 14.38 | 0.069541 | 125 | 181.81 | 0.005500 |
| 110 | 6.67 | 0.149925 | 126 | 800.00 | 0.001250 |
| 111 | 2.39 | 0.418410 | 127 | 166.66 | 0.006000 |
| 112 | 2.03 | 0.492610 | 128 | 16.19 | 0.061766 |

Table A4

Huffman Code for Number of Samples Skipped, Schematic C

| Number of Samples Skipped | Probability of Skipping | $n_i$* | Code |
|---|---|---|---|
| 1 | 0.173235868 | 3 | 000 |
| 6 | 0.092404574 | 3 | 100 |
| 12 | 0.076578412 | 4 | 0011 |
| 7 | 0.070025065 | 4 | 0101 |
| 2 | 0.067545370 | 4 | 0110 |
| 10 | 0.054338394 | 4 | 1101 |
| 5 | 0.047242803 | 4 | 1111 |
| 11 | 0.043795023 | 4 | 1010 |
| 23 | 0.041078904 | 5 | 00100 |
| 9 | 0.039457290 | 5 | 00101 |
| 8 | 0.035963183 | 5 | 01000 |
| 3 | 0.035715062 | 5 | 01110 |
| 13 | 0.031583676 | 5 | 11000 |
| 14 | 0.025718843 | 5 | 11100 |
| 4 | 0.023928569 | 5 | 11101 |
| 25 | 0.023762695 | 5 | 10110 |
| 24 | 0.015909861 | 6 | 011110 |
| 22 | 0.014325995 | 6 | 110010 |
| 15 | 0.014261874 | 6 | 110011 |
| 18 | 0.011157012 | 6 | 101110 |
| 19 | 0.009553423 | 7 | 0100100 |
| 17 | 0.008817485 | 7 | 0100110 |
| 16 | 0.008670054 | 7 | 0100111 |
| 20 | 0.005256475 | 7 | 1011111 |
| 21 | 0.004666306 | 8 | 01001010 |
| 31 | 0.0042[illegible]16 | 8 | 01111100 |

*$n_i$ = number of bits in the code

Table A4—Continued

Huffman Code for Number of Samples Skipped, Scheme C

| Number of Samples Skipped | Probability of Skipping | $n_i$* | Code |
|---|---|---|---|
| 35 | 0.002893870 | 8 | 10111100 |
| 30 | 0.002415339 | 9 | 010010110 |
| 32 | 0.002060067 | 9 | 011111010 |
| 26 | 0.002014548 | 9 | 011111011 |
| 29 | 0.001392703 | 9 | 101111010 |
| 33 | 0.001058470 | 10 | 0100101110 |
| 27 | 0.001050231 | 10 | 0100101111 |
| 36 | 0.000859126 | 10 | 0111111000 |
| 28 | 0.000857378 | 10 | 0111111001 |
| 39 | 0.000769617 | 10 | 0111111101 |
| 34 | 0.000755170 | 10 | 0111111110 |
| 37 | 0.000694837 | 10 | 1011110110 |
| 40 | 0.000487073 | 11 | 01111110101 |
| 38 | 0.000461838 | 11 | 01111110110 |
| 41 | 0.000431112 | 11 | 01111110111 |
| 49 | 0.000382027 | 11 | 01111111110 |
| 47 | 0.000339404 | 11 | 01111111111 |
| 42 | 0.000257572 | 11 | 10111101110 |
| 44 | 0.000235083 | 12 | 011111101000 |
| 48 | 0.000214510 | 12 | 011111110000 |
| 43 | 0.000203769 | 12 | 011111110001 |
| 50 | 0.000199969 | 12 | 011111110010 |
| 46 | 0.000141640 | 13 | 1011110111010 |
| 45 | 0.000135413 | 13 | 1011110111011 |
| 56 | 0.000122354 | 13 | 0111111010010 |
| 55 | 0.000048512 | 14 | 01111111001100 |

*$n_i$ = number of bits in the code

Table A4—Continued

Huffman Code for Number of Samples Skipped, Scheme C

| Number of Samples Skipped | Probability of Skipping | $n_i$* | Code |
|---|---|---|---|
| 54 | 0.000037403 | 15 | 011111101001100 |
| 53 | 0.000034501 | 15 | 011111101001101 |
| 51 | 0.000028684 | 15 | 011111101001110 |
| 52 | 0.000026756 | 15 | 011111110011110 |
| 59 | 0.000026387 | 15 | 011111110011100 |
| 57 | 0.000014995 | 16 | 0111111010011110 |
| 58 | 0.000013779 | 16 | 0111111100111110 |
| 62 | 0.000011165 | 16 | 0111111100111010 |
| 63 | 0.000011157 | 16 | 0111111100111011 |
| 61 | 0.000010342 | 16 | 0111111100110100 |
| 72 | 0.000009806 | 16 | 0111111100110110 |
| 64 | 0.000009205 | 17 | 01111110100111110 |
| 60 | 0.000009205 | 17 | 01111110100111111 |
| 70 | 0.000006534 | 17 | 01111111001111110 |
| 65 | 0.000005995 | 17 | 01111111001101010 |
| 71 | 0.000005547 | 17 | 01111111001101110 |
| 73 | 0.000002335 | 18 | 011111110011010111 |
| 66 | 0.000002048 | 18 | 011111110011011110 |
| 69 | 0.000001878 | 18 | 011111110011011111 |
| 67 | 0.000001860 | 19 | 0111111100111111100 |
| 68 | 0.000001376 | 19 | 0111111100110101100 |
| 74 | 0.000001270 | 19 | 0111111100110101101 |
| 84 | 0.000000838 | 20 | 01111111001111111101 |
| 80 | 0.000000754 | 20 | 01111111001111111110 |
| 77 | 0.000000495 | 21 | 011111110011111110100 |
| 82 | 0.000000466 | 21 | 011111110011111110110 |

*$n_i$ = number of bits in the code

Table A4—Continued

Huffman Code for Number of Samples Skipped, Scheme C

| Number of Samples Skipped | Probability of Skipping | $n_i$* | Code |
|---|---|---|---|
| 83 | 0.000000424 | 21 | 011111110011111111000 |
| 79 | 0.000000397 | 21 | 011111110011111111110 |
| 76 | 0.000000270 | 22 | 0111111100111111101010 |
| 78 | 0.000000186 | 22 | 0111111100111111101110 |
| 75 | 0.000000159 | 22 | 0111111100111111110010 |
| 96 | 0.000000122 | 23 | 01111111001111111010110 |
| 81 | 0.000000111 | 23 | 01111111001111111011110 |
| 95 | 0.000000080 | 23 | 01111111001111111111100 |
| 88 | 0.000000079 | 23 | 01111111001111111111101 |
| 90 | 0.000000070 | 23 | 01111111001111111111110 |
| 97 | 0.000000069 | 23 | 01111111001111111111111 |
| 85 | 0.000000056 | 24 | 011111110011111110111110 |
| 103 | 0.000000053 | 24 | 011111110011111110111111 |
| 94 | 0.000000047 | 24 | 011111110011111111001100 |
| 89 | 0.000000047 | 24 | 011111110011111111001101 |
| 91 | 0.000000043 | 24 | 011111110011111111001111 |
| 87 | 0.000000033 | 25 | 0111111100111111101011100 |
| 86 | 0.000000032 | 25 | 0111111100111111101011101 |
| 102 | 0.000000024 | 25 | 0111111100111111110011100 |
| 92 | 0.000000015 | 26 | 01111111001111111010111100 |
| 93 | 0.000000013 | 26 | 01111111001111111010111110 |
| 100 | 0.000000009 | 26 | 01111111001111111100111010 |
| 108 | 0.000000008 | 27 | 011111110011111110101111010 |
| 109 | 0.000000006 | 27 | 011111110011111110101111110 |
| 101 | 0.000000005 | 27 | 011111110011111111001110111 |
| 115 | 0.000000004 | 28 | 0111111100111111101011110110 |

*$n_i$ = number of bits in the code

Table A4—Concluded

Huffman Code for Number of Samples Skipped, Scheme C

| Number of Samples Skipped | Probability of Skipping | $n_i$* | Code |
|---|---|---|---|
| 107 | 0.000000003 | 28 | 0111111100111111101011111110 |
| 98 | 0.000000003 | 28 | 0111111100111111101011111111 |
| 104 | 0.000000003 | 28 | 0111111100111111110011101100 |
| 99 | 0.000000002 | 29 | 01111111001111111010111001110 |
| 106 | 0.000000002 | 29 | 01111111001111111010011101111 |
| 105 | 0.000000001 | 30 | 011111110011111111001110110100 |
| 114 | 0.000000001 | 30 | 011111110011111111001110110101 |
| 110 | 0.000000001 | 30 | 011111110011111111001110110110 |
| 112 | 0.000000000 | 34 | 0111111100111111110011101101110000 |
| 113 | 0.000000000 | 34 | 0111111100111111110011101101110001 |
| 116 | 0.000000000 | 34 | 0111111100111111110011101101110010 |
| 117 | 0.000000000 | 34 | 0111111100111111110011101101110011 |
| 111 | 0.000000000 | 34 | 0111111100111111110011101101110100 |
| 125 | 0.000000000 | 34 | 0111111100111111110011101101110101 |
| 119 | 0.000000000 | 34 | 0111111100111111110011101101110110 |
| 118 | 0.000000000 | 34 | 0111111100111111110011101101110111 |
| 120 | 0.000000000 | 34 | 0111111100111111110011101101111000 |
| 124 | 0.000000000 | 34 | 0111111100111111110011101101111001 |
| 122 | 0.000000000 | 34 | 0111111100111111110011101101111010 |
| 121 | 0.000000000 | 34 | 0111111100111111110011101101111011 |
| 123 | 0.000000000 | 34 | 0111111100111111110011101101111100 |
| 126 | 0.000000000 | 34 | 0111111100111111110011101101111101 |
| 127 | 0.000000000 | 34 | 0111111100111111110011101101111110 |
| 128 | 0.000000000 | 34 | 0111111100111111110011101101111111 |

*$n_i$ = number of bits in the code

# APPENDIX B

## DERIVATION OF THE COMPRESSION RATIO FOR SCHEME D

From the raw compression ratios of Table A3, the probability of occurrence table is found from the equation

$$P_{Ai} = \frac{1}{C_{Ri}} \quad \text{for } i = 4, \cdots, 128 .$$

This table is arranged in decreasing order, normalized by the factor

$$\frac{1}{\sum_{i=1}^{128} P_{Ai}}$$

and is presented in Table B1.

The Huffman code and associated code lengths are also presented here. The net cost of identifying the data points for this scheme are found by the summation

$$\sum_{i=4}^{128} P_i n_i$$

In Table B2 the first three words do not appear since they represent the FSP, which was not compressed. However, these 108,000 bits are present in the raw data bit count. The result of these calculations yields

$$554,274 + 270,340 = 824,614 .$$

The compression ratio is

$$C_D = \frac{4,608,000}{824,614} = 5.59 .$$

Table B1

Huffman Code for the Main Frame Word Positions, Scheme D

| Main Frame Word Number | Probability of Occurrence | $n_i$* | Code |
|---|---|---|---|
| 118 | 0.071370951 | 4 | 1011 |
| 15 | 0.070846610 | 4 | 1100 |
| 87 | 0.069999597 | 4 | 1101 |
| 62 | 0.068991248 | 4 | 1110 |
| 38 | 0.068708910 | 4 | 1111 |
| 48 | 0.040979309 | 5 | 10000 |
| 112 | 0.039668455 | 5 | 10010 |
| 47 | 0.034445206 | 5 | 00000 |
| 111 | 0.033739362 | 5 | 00001 |
| 56 | 0.023131520 | 5 | 01001 |
| 98 | 0.021276167 | 5 | 01110 |
| 99 | 0.020207317 | 6 | 100011 |
| 55 | 0.019602307 | 6 | 100111 |
| 97 | 0.017081434 | 6 | 101010 |
| 28 | 0.013874884 | 6 | 001001 |
| 39 | 0.013108539 | 6 | 001011 |
| 85 | 0.012564030 | 6 | 001100 |
| 16 | 0.012382527 | 6 | 001101 |
| 110 | 0.012080023 | 6 | 001110 |
| 45 | 0.011999355 | 6 | 010000 |
| 119 | 0.011938854 | 6 | 010001 |
| 103 | 0.011575848 | 6 | 010100 |
| 64 | 0.011233009 | 6 | 010110 |
| 26 | 0.010870004 | 6 | 010111 |
| 117 | 0.010829670 | 6 | 011000 |
| 14 | 0.010829670 | 6 | 011001 |

*$n_i$ – number of bits in the code

Table B1—Continued

Huffman Code fo. the Main Frame Word Positions, Scheme D

| Main Frame Word Number | Probability of Occurrence | $n_i$* | Code |
|---|---|---|---|
| 86 | 0.010789336 | 6 | 011010 |
| 46 | 0.010607833 | 6 | 011011 |
| 80 | 0.010446497 | 6 | 011110 |
| 120 | 0.010264994 | 7 | 1000100 |
| 88 | 0.010103658 | 7 | 1000101 |
| 9 | 0.010002823 | 7 | 1001101 |
| 8 | 0.009700319 | 7 | 1010000 |
| 37 | 0.009559150 | 7 | 1010011 |
| 61 | 0.009377647 | 7 | 1010010 |
| 106 | 0.009115476 | 7 | 1010110 |
| 68 | 0.008349131 | 7 | 0001000 |
| 74 | 0.008268463 | 7 | 0001010 |
| 41 | 0.008268463 | 7 | 0001011 |
| 42 | 0.008187795 | 7 | 0001100 |
| 10 | 0.008167628 | 7 | 0001110 |
| 105 | 0.007744121 | 7 | 0010000 |
| 63 | 0.007078611 | 7 | 0010100 |
| 73 | 0.006534102 | 7 | 0010101 |
| 60 | 0.005929093 | 7 | 0011110 |
| 109 | 0.005606421 | 7 | 0101010 |
| 77 | 0.005041764 | 8 | 10001000 |
| 13 | 0.005021579 | 8 | 10001001 |
| 128 | 0.004981245 | 8 | 10100010 |
| 57 | 0.004658573 | 8 | 10100011 |
| 49 | 0.004577905 | 8 | 10101110 |
| 104 | 0.004214899 | 8 | 00010010 |

*$n_i$ = number of bits in the code

Table B1—Continued

Huffman Code for the Main Frame Word Positions, Scheme D

| Main Frame Word Number | Probability of Occurrence | $n_i$* | Code |
|---|---|---|---|
| 25 | 0.004214899 | 8 | 00010011 |
| 115 | 0.004194732 | 8 | 00011010 |
| 50 | 0.004033397 | 8 | 00011011 |
| 51 | 0.003730892 | 8 | 00011110 |
| 44 | 0.003468721 | 8 | 00100011 |
| 29 | 0.003004880 | 8 | 00111111 |
| 35 | 0.002944379 | 8 | 01010110 |
| 79 | 0.002662042 | 8 | 01111100 |
| 76 | 0.002581374 | 8 | 01111110 |
| 23 | 0.002561207 | 8 | 01111111 |
| 89 | 0.002218368 | 9 | 101011110 |
| 100 | 0.002137700 | 9 | 000111110 |
| 78 | 0.001855362 | 9 | 001000100 |
| 102 | 0.001391522 | 9 | 010101110 |
| 12 | 0.001331021 | 9 | 010110110 |
| 65 | 0.001290687 | 9 | 011111011 |
| 75 | 0.001068850 | 10 | 1010111110 |
| 108 | 0.000927681 | 10 | 0010001010 |
| 59 | 0.000927681 | 10 | 0010001011 |
| 58 | 0.000927681 | 10 | 0001111110 |
| 32 | 0.000847013 | 10 | 0011111000 |
| 96 | 0.000786512 | 10 | 0011111010 |
| 53 | 0.000746178 | 10 | 0101011110 |
| 127 | 0.000484008 | 11 | 00011111110 |
| 125 | 0.000443674 | 11 | 00011111111 |
| 27 | 0.000383173 | 11 | 00111110010 |

*$n_i$ = number of bits in the code

Table B1—Continued

Huffman Code for the Main Frame Word Positions, Scheme D

| Main Frame Word Number | Probability of Occurrence | $n_i$* | Code |
|---|---|---|---|
| 20 | 0.000383173 | 11 | 00111110110 |
| 124 | 0.000302505 | 11 | 01010111110 |
| 123 | 0.000282338 | 12 | 101011111100 |
| 122 | 0.000282338 | 12 | 101011111101 |
| 121 | 0.000282338 | 12 | 101011111110 |
| 116 | 0.000282338 | 12 | 101011111111 |
| 90 | 0.000201670 | 12 | 001111100110 |
| 126 | 0.000100835 | 13 | 0011111001110 |
| 72 | 0.000100835 | 13 | 0011111001111 |
| 52 | 0.000100835 | 13 | 0011111011100 |
| 82 | 0.000080668 | 13 | 0011111011110 |
| 71 | 0.000060501 | 14 | 00111110111010 |
| 113 | 0.000040334 | 14 | 00111110111011 |
| 84 | 0.000040334 | 14 | 00111110111110 |
| 19 | 0.000040334 | 14 | 00111110111111 |
| 4 | 0.000040334 | 14 | 01010111111100 |
| 114 | 0.000020167 | 15 | 010101111111010 |
| 93 | 0.000020167 | 15 | 010101111111011 |
| 92 | 0.000020167 | 15 | 010101111111100 |
| 91 | 0.000020167 | 15 | 010101111111101 |
| 70 | 0.000020167 | 15 | 010101111111110 |
| 54 | 0.000020167 | 15 | 010101111111111 |
| 43 | 0.000020167 | 15 | 010101111110000 |
| 40 | 0.000020167 | 15 | 010101111110001 |
| 31 | 0.000020167 | 15 | 010101111110010 |
| 30 | 0.000020167 | 15 | 010101111110011 |

*$n_i$ = number of bits in the code

Table B1—Concluded

Huffman Code for the Main Frame Word Positions, Scheme D

| Main Frame Word Number | Probability of Occurrence | $n_i$* | Code |
|---|---|---|---|
| 24 | 0.000020167 | 15 | 010101111110100 |
| 21 | 0.000020167 | 15 | 010101111110101 |
| 7 | 0.000020167 | 15 | 010101111110110 |
| 5 | 0.000020167 | 16 | 0101011111101111 |
| 95 | 0.000000000 | 20 | 01010111111011100000 |
| 101 | 0.000000000 | 21 | 010101111110111000010 |
| 107 | 0.000000000 | 21 | 010101111110111000011 |
| 94 | 0.000000000 | 20 | 01010111111011100010 |
| 83 | 0.000000000 | 20 | 01010111111011100011 |
| 81 | 0.000000000 | 20 | 01010111111011100100 |
| 69 | 0.000000000 | 20 | 01010111111011100101 |
| 67 | 0.000000000 | 20 | 01010111111011100110 |
| 66 | 0.000000000 | 20 | 01010111111011100111 |
| 36 | 0.000000000 | 20 | 01010111111011101000 |
| 34 | 0.000000000 | 20 | 01010111111011101001 |
| 33 | 0.000000000 | 20 | 01010111111011101010 |
| 22 | 0.000000000 | 20 | 01010111111011101011 |
| 18 | 0.000000000 | 20 | 01010111111011101100 |
| 17 | 0.000000000 | 20 | 01010111111011101101 |
| 11 | 0.000000000 | 20 | 01010111111011101110 |
| 6 | 0.000000000 | 20 | 01010111111011101111 |

*$n_i$ = number of bits in the code

## APPENDIX C

## DERIVATION OF THE COMPRESSION RATIO FOR SCHEME E

In Table C1, the sensors are numbered by experiment and letter. The letters represent different sensors of the same experiment. In the cases where a sensor appears more than once in the main frame, the word position numbers of its outputs appear to the left. Table C1 is arranged according to the decreasing order of the probability of occurrence. The probability of occurrence is found by dividing the bits out by the bits in; then a Huffman code is derived for each sensor.

For those sensors with more than one output per frame, the probability of an ambiguous output is calculated by finding $P_{Ak}$, where $P_{Ak}$ is as defined in Appendix A and k is the number of words between outputs of the sensor. The Huffman code is then modified where necessary so that the probability of ambiguity is less than 0.006. These modifiers appear at the end of Table C1.

Table C2 represents the probability of ambiguity of multiple-output sensors and the effect of added coding to remove this ambiguity on the overall compression ratio. For the case of an ambiguity of less than 0.006, the compression ratio is found by the following procedure. The number of identification bits is found by the sum

$$\sum_{i=4}^{128} n_i P_i \, 4000 \; .$$

To this sum is added the number of data bits required. In Table C2, the first three words representing the FSP do not appear; however, they do appear in the sum of the output bits. The results of these computations show

$$C_E = \frac{4,608,000}{554,274 + 275,351} = \frac{4,608,000}{829,625} = 5.55 \; .$$

The other compression ratios in Table C2 were computed similarly.

Table C1

Huffman Code for the Experiment Sensor, Scheme E

| Main Frame Word Numbers | Sensor Number | Probability of Occurrence | Bits in | Bits out | $n_i$* | Code |
|---|---|---|---|---|---|---|
| 15: 38: 62; 87; 118 | 10B | 0.867550000 | 180000 | 156159 | 3 | 101** |
| 48; 112 | 17C | 0.499861111 | 72000 | 35991 | 3 | 011 |
| 47; 111 | 17B | 0.422625000 | 72000 | 30429 | 4 | 1100 |
| 56 | 4B | 0.286750000 | 36000 | 10323 | 4 | 0010 |
| 98 | SCC2 | 0.263750000 | 36000 | 9495 | 4 | 0001 |
| 99 | SCC3 | 0.250500000 | 36000 | 9018 | 4 | 0101 |
| 55 | 4A | 0.243000000 | 36000 | 8748 | 5 | 10000 |
| 97 | SCC1 | 0.211750000 | 36000 | 7623 | 5 | 10011 |
| 8; 26; 45; 64; 85; 103; 120 | 18 | 0.183500000 | 252000 | 34902 | 5 | 11110*** |
| 28 | 7B | 0.172000000 | 36000 | 6192 | 5 | 11111 |
| 46; 110 | 17A | 0.1406025 | 72000 | 10125 | 5 | 00000 |
| 16; 39; 63; 80; 88; 119 | 10C | 0.134416666 | 216000 | 29034 | 5 | 00001 |
| 14; 37; 61; 86; 117 | 10A | 0.127400000 | 180000 | 22932 | 5 | 01001 |
| 106 | 3D | 0.113000000 | 36000 | 4068 | 6 | 100100 |

*$n_i$ = number of bits in the code

Table C1—Continued

Huffman Code for the Experiment Sensor, Scheme E

| Main Frame Word Numbers | Sensor Number | Probability of Occurrence | Bits in | Bits out | $n_i$* | Code |
|---|---|---|---|---|---|---|
| 68 | 15E | 0.103500000 | 36000 | 3726 | 6 | 110101 |
| 74 | 3C | 0.102500000 | 36000 | 3690 | 6 | 110110 |
| 42 | 3B | 0.101500000 | 36000 | 3654 | 6 | 111000 |
| 10 | 3A | 0.101250000 | 36000 | 3645 | 6 | 111001 |
| 9; 73; 41; 105 | 13A | 0.100875000 | 144000 | 14526 | 6 | 111010 |
| 60 | 6C | 0.073500000 | 36000 | 2646 | 6 | 001101 |
| 109 | 15J | 0.069500000 | 36000 | 2502 | 6 | 001111 |
| 77 | 5B | 0.062500000 | 36000 | 2250 | 6 | 010000 |
| 13 | 5A | 0.062250000 | 36000 | 2241 | 7 | 1000100 |
| 128 | 12D | 0.061750000 | 36000 | 2223 | 7 | 1000101 |
| 57 | 4C | 0.057750000 | 36000 | 2079 | 7 | 1000110 |
| 49 | 7C | 0.056750000 | 36000 | 2043 | 7 | 1001010 |
| 104 | 5C | 0.052250000 | 36000 | 1881 | 7 | 1101000 |
| 25 | 5E | 0.052250000 | 36000 | 1881 | 7 | 1101001 |
| 115 | 8B | 0.052000000 | 36000 | 1872 | 7 | 1101110 |
| 50 | 7D | 0.050000000 | 36000 | 1800 | 7 | 1101111 |

*$n_i$ = number of bits in the code

Table C1—Continued

Huffman Code for the Experiment Sensor, Scheme E

| Main Frame Word Numbers | Sensor Number | Probability of Occurrence | Bits in | Bits out | $n_i$* | Code |
|---|---|---|---|---|---|---|
| 51 | 7E | 0.046250000 | 36000 | 1665 | 7 | 1110110 |
| 29 | 7I | 0.037250000 | 36000 | 1341 | 7 | 0011000 |
| 35 | SPC3 | 0.036500000 | 36000 | 1314 | 7 | 0011100 |
| 79 | 19B | 0.033000000 | 36000 | 1188 | 7 | 0100010 |
| 23 | 2G | 0.031750000 | 36000 | 1143 | 8 | 10001110 |
| 89 | 5F | 0.027500000 | 36000 | 990 | 8 | 10001111 |
| 100 | 9I | 0.026500000 | 36000 | 954 | 8 | 10001110 |
| 12; 44; 76; 108 | 13B | 0.025750000 | 144000 | 3708 | 8 | 11101110 |
| 78 | 19A | 0.023000000 | 36000 | 828 | 8 | 11101111 |
| 102 | 11G | 0.017250000 | 36000 | 621 | 8 | 00111010 |
| 65 | SCCU | 0.016000000 | 36000 | 576 | 8 | 01000110 |
| 75 | 5D | 0.013250000 | 36000 | 477 | 9 | 100011110 |
| 59 | 6B | 0.011500000 | 36000 | 414 | 9 | 001100100 |
| 58 | 6A | 0.011500000 | 36000 | 414 | 9 | 001100101 |
| 32; 96 | 11C | 0.010125000 | 72000 | 729 | 9 | 001100110 |
| 53 | 7J | 0.009250000 | 36000 | 333 | 9 | 001110110 |

*$n_i$ = number of bits in the code

Table C1—Continued

Huffman Code for the Experiment Sensor, Scheme E

| Main Frame Word Numbers | Sensor Number | Probability of Occurrence | Bits in | Bits out | $n_i$* | Code |
|---|---|---|---|---|---|---|
| 127 | 12C | 0.006000000 | 36000 | 216 | 10 | 0011001110 |
| 125 | 12A | 0.005500000 | 36000 | 198 | 10 | 0011001111 |
| 27 | 15B | 0.004750000 | 36000 | 171 | 10 | 0011101110 |
| 20 | 7H | 0.004750000 | 36000 | 171 | 10 | 0100011100 |
| 124 | 9E | 0.003750000 | 36000 | 135 | 10 | 0100011110 |
| 123 | 9D | 0.003500000 | 36000 | 126 | 10 | 1000111100 |
| 122 | 9C | 0.003500000 | 36000 | 126 | 10 | 1000111101 |
| 121 | 9B | 0.003500000 | 36000 | 126 | 10 | 1000111110 |
| 116 | 9A | 0.003500000 | 36000 | 126 | 10 | 1000111111 |
| 90 | 9F | 0.002500000 | 36000 | 90 | 11 | 00111011110 |
| 126 | 12B | 0.001250000 | 36000 | 45 | 12 | 001110111110 |
| 72 | 15I | 0.001250000 | 36000 | 45 | 12 | 001110111111 |
| 52 | 7F | 0.001250000 | 36000 | 45 | 12 | 010001110100 |
| 82 | 1B | 0.001000000 | 36000 | 36 | 12 | 010001110101 |
| 71 | 15H | 0.000750000 | 36000 | 27 | 12 | 010001111100 |
| 113 | 9G | 0.000500000 | 36000 | 18 | 13 | 0100011111010 |

*$n_i$ = number of bits in the code

Table C1—Continued

Huffman Code for the Experiment Sensor, Scheme E

| Main Frame Word Numbers | Sensor Number | Probability of Occurrence | Bits in | Bits out | $n_i$* | Code |
|---|---|---|---|---|---|---|
| 84 | 1D | 0.000500000 | 36000 | 18 | 13 | 0100011111011 |
| 19 | 7A | 0.000500000 | 36000 | 18 | 13 | 0100011101100 |
| 4 | 2A | 0.000500000 | 36000 | 18 | 13 | 0100011101101 |
| 114 | 8A | 0.000250000 | 36000 | 9 | 14 | 01000111011100 |
| 93 | 11F | 0.000250000 | 36000 | 9 | 14 | 01000111011101 |
| 92 | 11E | 0.000250000 | 36000 | 9 | 14 | 01000111011110 |
| 91 | 11D | 0.000250000 | 36000 | 9 | 14 | 01000111011111 |
| 70 | 15G | 0.000250000 | 36000 | 9 | 14 | 01000111111000 |
| 54 | 15D | 0.000250000 | 36000 | 9 | 14 | 01000111111001 |
| 43 | 9H | 0.000250000 | 36000 | 9 | 14 | 01000111111010 |
| 40 | 15C | 0.000250000 | 36000 | 9 | 14 | 01000111111011 |
| 24 | 15A | 0.000250000 | 36000 | 9 | 14 | 01000111111100 |
| 21 | 2E | 0.000250000 | 36000 | 9 | 14 | 01000111111101 |
| 7 | 2D | 0.000250000 | 36000 | 9 | 14 | 01000111111110 |
| 5 | 2B | 0.000250000 | 36000 | 9 | 15 | 010001111111110 |
| 30; 94 | 11A | 0.000125000 | 72000 | 9 | 16 | 0100011111111110 |

*$n_i$ = number of bits in the code

Table C1—Continued

Huffman Code for the Experiment Sensor, Scheme E

| Main Frame Word Numbers | Sensor Number | Probability of Occurrence | Bits in | Bits out | $n_i$* | Code |
|---|---|---|---|---|---|---|
| 31; 95 | 11B | 0.000125000 | 72000 | 9 | 17 | 01000111111111110 |
| 11; 17; 18 | 20 | 0.000000000 | 108000 | 0 | 20 | 01000111111111111000**** |
| 83 | 1C | 0.000000000 | 36000 | 0 | 20 | 01000111111111111001 |
| 81 | 1A | 0.000000000 | 36000 | 0 | 20 | 01000111111111111010 |
| 69 | 15F | 0.000000000 | 36000 | 0 | 20 | 01000111111111111011 |
| 67 | EG2 | 0.000000000 | 36000 | 0 | 20 | 01000111111111111111 |
| 66 | EG1 | 0.000000000 | 36000 | 0 | 20 | 01000111111111111110 |
| 34 | SPC2 | 0.000000000 | 36000 | 0 | 21 | 010001111111111111000 |
| 33 | SPC1 | 0.000000000 | 36000 | 0 | 21 | 010001111111111111001 |
| 22 | 2F | 0.000000000 | 36000 | 0 | 21 | 010001111111111111010 |
| 6 | 2G | 0.000000000 | 36000 | 0 | 21 | 010001111111111111011 |

*$n_i$ = number of bits in the code

Table C1—Concluded

Huffman Code for the Experiment Sensor, Scheme E

| | Sensor Number | Word Position | Modifier | Modified Code | $n_i'$ |
|---|---|---|---|---|---|
| ** | 10B | 15 | 000 | 101000 | 6 |
| | | 38 | 001 | 101001 | 6 |
| | | 62 | 010 | 101010 | 6 |
| | | 87 | 011 | 101011 | 6 |
| | | 118 | 100 | 101100 | 6 |
| *** | 18 | 8 | 000 | 11110000 | 8 |
| | | 26 | 001 | 11110001 | 8 |
| | | 45 | 010 | 11110010 | 8 |
| | | 64 | 011 | 11110011 | 8 |
| | | 85 | 100 | 11110100 | 8 |
| | | 103 | 101 | 11110101 | 8 |
| | | 120 | 110 | 11110110 | 8 |
| **** | 20 | 11 | 00 | 0100011111111111100000 | 22 |
| | | 17 | 01 | 0100011111111111100001 | 22 |
| | | 18 | 10 | 0100011111111111100010 | 22 |

where $n_i'$ = number of bits in the modified code

Table C2

Comparison of Ambiguity in Scheme E

| Sensor Number | Probability of Ambiguity Removed | Number of Identification Bits Needed | Total Number of Bits Output | Actual Compression Ratio |
|---|---|---|---|---|
| - | None | 221,964 | 776,238 | 5.936 |
| 10B | 0.333 | 264,017 | 818,291 | 5.631 |
| 18 | 0.0115 | 233,298 | 787,572 | 5.850 |
| 10C | 0.006 | 231,684 | 785,958 | 5.862 |
| 10A | 0.0011 | 229,608 | 783,882 | 5.878 |
| 13A | 0.00006 | 225,192 | 779,466 | 5.911 |
| 17C | 0.00005 | 225,963 | 780,237 | 5.906 |
| 17B | 0.00003 | 225,345 | 779,619 | 5.911 |
| 17A | 0.000002 | 223,089 | 777,363 | 5.927 |
| 13B | 0.000008 | 222,788 | 777,062 | 5.931 |
| - | All | 300,236 | 854,510 | 5.392 |

## BIBLIOGRAPHY


1. Stumpe, J. W.: Redundancy Reduction Techniques and Applications. Telecommunications, vol. 1, no. 2, T39, T43, October 1967

2. Bechtold, W. R., Medlin, J. E., and Weber, D. R.: PCM Telemetry Data Compression Study, Phase I, Lockheed Missiles and Space Company, M-62-65-2, October 1965

3. Bjorn, T., Jr., Friccero, W., and Medlin, J.: PCM Telemetry Data Compression Study, Phase II. Lockheed Missiles and Space Company, M-62-66-8, January 1967

4. Medlin, J., and Smith, W.: PCM Telemetry Data Compression Study, Phase III. Lockheed Missiles and Space Company, M-62-67-5, September 1967

5. Balakrishnan, A. V.: Adaptive Methods in Communications Systems. NASA Technical Memorandum X-55747, January 1967, pp. 9-27

6. Davisson, L. D.: Adaptive Prediction of Time Series. NASA Technical Memorandum X-55747, January 1967, pp. 27-41

7. Anderson, T., Eisenberger, I., Lushbaugh, W., and Posner, E.: Demonstration of a Quantile System for Compression of Data from Deep Space Prober. NASA Technical Memorandum X-55747, January 1967, pp. 55-77

8. Laughlin, C. R.: Data Compaction Studies, NASA Document Technical Memorandum X-55747, January 1967, pp. 109-117

9. Cliff, R. A.: The Use of a Stored-Program Computer on a Small Spacecraft. NASA Technical Memorandum X-55747, January 1967, pp. 129-135

10. Andrews, C. A., Davies, J. M., and Schwarz, G. R.: Adaptive Data Compression. Proceedings of the IEEE, 55(3):267-278, March 1967

11. Quann, John J.: Data-Processing Plan for Eccentric Orbiting Geophysical Observatory, OGO-B, Technical Memorandum X-55890, March 1966

12. Kanefsky, M.: Data Compression Using Bit-Phase Encoding. Final Report, 1966 Session of the Goddard Summer Workshop. June 15 to September 15, 1966, pp. 1-23

13. Davisson, L. D.: The Fan Method of Data Compression. NASA Technical Memorandum X-55742, Final Report, 1966 Session of the Goddard Summer Workshop, June 15 to September 15, 1966, pp. 23-31

14. Davisson, L. D.: The Concept of Energy Compression Ratio and Its Application to Run Length Coding. NASA Technical Memorandum X-55742, Final Report, 1966 Session of the Goddard Summer Workshop, June 15 to September 15, 1966, pp. 31-45

15. Kelley, R.: The Effect of Channel Errors on a Sample Compression System Employing Binary Sequence Time Encoding. NASA Technical Memorandum X-55742, Final Report, 1966 Session of the Goddard Summer Workshop, June 15 to September 15, 1966, pp. 45-65

16. Sobotkin, F.: Use of a Formal Language to Describe On-Board and Ground Station Processing of Satellite Experiment. NASA Technical Memorandum X-55742, Final Report, 1966 Session of the Goddard Summer Workshop, June 15 to September 15, 1966, pp. 121-135

17. Davisson, L. D.: The Theoretical Analysis of Data Compression Systems. Proceedings of the IEEE, 56(2):176-186, February 1968

18. Kortman, C. M.: Redundancy Reduction—A Practical Method of Data Compression. Proceedings of the IEEE, 55(3):253-263, March 1967

19. Ehrman, L.: Analysis of Some Redundancy Removal Bandwidth Compression Techniques. Proceedings of the IEEE, 55(3):278-287, March 1967

20. Abate, J. E.: Linear and Adaptive Delta Modulation. Proceedings of the IEEE, 55(3):298-308, March 1967

21. Bello, P., Lincoln, R., and Gish, H.: Statistical Delta Modulation. Proceedings of the IEEE, 55(3):308-320, March 1967

22. Lynch, Thomas J.: Data Compression With Error-Control Coding for Space Telemetry. NASA TR R-261, June 1967

23. OGO-Operations: OGO-B Experiment Operations Manual. Goddard Space Flight Center, Greenbelt, Maryland

24. Ash, Robert B.: Information Theory. Interscience Publishers, 1967