NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Technical Memorandum 33-695

Channel Coding and Data Compression System Considerations for Efficient Communication of Planetary Imaging Data

Robert F. Rice

(NASA-CR-139362) CHANNEL CODING AND DATA COMPRESSION SYSTEM CONSIDERATIONS FOR EFFICIENT COMMUNICATION OF PLANETARY IMAGING DATA (Jet Propulsion Lab.) CSCL 09F G3/07

N74-29496

Unclas 44991

JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

June 15, 1974

Reproduced by
NATIONAL TECHNICAL INFORMATION SERVICE
US Department of Commerce
Springfield, VA. 22151

PRICES SUBJECT TO CHANGE

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Technical Memorandum 33-695

Channel Coding and Data Compression System Considerations for Efficient Communication of Planetary Imaging Data

Robert F. Rice

JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

June 15, 1974

REPRODUCTION RESTRICTIONS OVERRIDDEN
NASA Scientific and Technical Information Facility

Copyright © 1974
Jet Propulsion Laboratory
California Institute of Technology

Override
W/10-74

Prepared Under Contract No. NAS 7-100
National Aeronautics & Space Administration

PREFACE

The work described in this report was performed by the Astrionics Division of the Jet Propulsion Laboratory.

ACKNOWLEDGMENT

The author expresses gratitude to his colleague Edward E. Hilbert for rewarding technical discussions during the writing of this material and in its development.

CONTENTS

I. INTRODUCTION 1

II. CHANNEL CODING WITHOUT DATA COMPRESSION 5

The Uncoded Channel 5

Relationship of Rate and E_s 7
P_s vs. E_s/N_0 for the Uncoded Channel 8

Two Coded Systems 10

Source Data 12

Error Considerations for PCM 14

The Jupiter/Saturn Channel 15

III. THE DATA COMPRESSION PROBLEM UNDER NOISELESS CHANNEL CONDITIONS 17

Picture Compression 18

Rate 18
Quality 20
A Hypothetical Super System (SS) 26
Super System as a Measure of Quality 27

Imaging Sequences 31

Sequencing Examples 34
More Complex Sequencing 44

Changing the Quality Function 55

Introduction to RM2 56

IV. CHANNEL CODING FOR DATA COMPRESSION 58

The Jupiter/Saturn Channel and Compressed Imaging Data 58

A Review for Uncompressed PCM 58
Transmission of Compressed Data 61

The Odenwalder Channel 66

Reed-Solomon Coding 68
Interleaving 71
Effect of a Codeword Error 78
Acceptable Values of P_{RS} 82
Uncompressed PCM 84
Performance Curves 84
Other Code Combinations 86
Data Other Than Imaging 92
Imperfect Phase Tracking 95

CONTENTS (cont'd)

AGC 101
Slow Drifts in E_b/N_0 102
Summary of Characteristics 103

WHY NOT SEQUENTIAL DECODING 104

V. INTRODUCTION TO AICS 107

APPENDIX A: DECIBEL REPRESENTATION 109

APPENDIX B: RS CODE BLOCK SYNCHRONIZATION 113

REFERENCES 120

ILLUSTRATIONS

1. The Uncoded Channel 6
2. Binary Symmetric Channel 7
3. Bit Error Probabilities for the Uncoded Channel 9
4. A Coded System 10
5. Ideal Performance Curves: Uncoded, Block and Viterbi 13
6. The Jupiter/Saturn Channel 16
7. Source Encoding Introduction 21
8. Open Loop Fidelity Judgements 23
9. General Characteristics, Super System 28
10. Using Super System to Measure Quality for a Given Data Class $\{D_k\}$ 29
11. Sensitivity to Data Variations for Individual Algorithm 31
12. A Hypothetical Imaging Sequence 32
13. Sequence Values, I 35
14. Sequence Values, II 41
15. Sequence Values, III 43
16. Interactive Terminal for Visualizing Sequencing Tradeoffs 46
17. Initializing the Terminal for Fixed Data Class 49
18. Test Sequence Generated for PCM Operation 50
19. Multiple Data Classes 51
20. Source Block Losses Due to Random Errors 64

ILLUSTRATIONS

21. Source Block Losses Due to Error Burst 66
22. Inserting the Reed-Solomon Block Code 67
23. Basic RS Structure 69
24. Basic RS Codeword Structure, J=8, E=16 71
25. Interleave A, Structure 73
26. Interleave B, Structure 75
27. Effect of RS Word Error, Interleave A 79
28. Performance Curves 85
29. Degradations Due to Phase Tracking Errors 98
30. Introduction to AICS 108
A-1 Decibel Conversion 110
A-2 Decibel Conversion: Expanded Scale Around Zero 111
B-1 Sync Configuration 1 117

TABLES

1. Comparison of Interleave Methods 81
2. Number of Pictures Between Source Block Errors 83

ABSTRACT

This report presents end-to-end system considerations involving channel coding and data compression which could drastically improve the efficiency in communicating pictorial information from future planetary spacecraft. In addition to presenting new and potentially significant system considerations, this report attempts to fill a need for a comprehensive tutorial which makes much of this very subject accessible to readers whose disciplines lie outside of communication theory.

Much of this material has been the basis of proposals for future Mariner and Pioneer missions under the title "Advanced Imaging Communication System (AICS)."

I. INTRODUCTION

This report deals with system considerations which could drastically improve the efficiency in communicating pictorial information from future planetary spacecraft. It is taken for granted that this is a desirable goal either in the form of more information or long term cost benefits. The interactive system elements which potentially afford these improvements are channel coding and data compression, the principal subjects of this report.

In addition to presenting new and potentially significant "end-to-end" system considerations, this report attempts to fill a need for a comprehensive tutorial which makes much of this very subject accessible to readers whose disciplines lie outside of communication theory. A sincere attempt is made to make this material "readable" to a wide audience. Chapter II provides basic terminology and background information on the development of planned deep space coded communication capabilities for planetary imaging experiments. This development was guided primarily by the desire for efficient communication of uncompressed imaging data. It has culminated with a proposed coded system for the Mariner Jupiter/Saturn '77 missions employing Viterbi decoded convolutional codes.

Although quite powerful when used to transmit uncompressed imaging data, this coded channel generally exhibits a classic interaction with compressed imaging data (and highly error sensitive data from other experiments). The basic consequence of this interaction is that, to make use of data compression at all, the error rate must be much lower than for uncompressed imaging data. This can be accomplished only by lowering the transmission rate by as much as a factor of two. Thus some of the advantages that compressing the data might offer are lost because of this necessary transmission rate reduction. A practical and powerful solution to this

problem is proposed in Chapter IV. We will discuss this subject in a moment after we have clarified the contents of the intervening Chapter III and its relationship to Chapter IV.

Chapter III provides an extensive new look into the overall system aspects of applying data compression to planetary imaging. Its only reliance on Chapter IV is the result that, for all practical purposes, the channel can be viewed as noiseless. Practically speaking this is significant, but to the reader it means that Chapter III can be read without referring to Chapter IV. On the other hand, the first few pages of Chapter III provide some basic definitions (e.g., compression factor, block structure) which are necessarily referred to in the discussions of Chapter IV. We wish to make clear, however, that the vast majority of Chapter III is not required if the reader wishes to pursue the channel problem directly.

The description of algorithms in Chapter III is maintained at a very general "black box" level and no algorithm is discussed in detail. The primary questions addressed here involve the identification of tradeoffs which assess how well potential black box candidates fit into a mission environment when looked at from an overall system point of view. This approach leads to the definition of an "ideal black box" and some desirable properties for advanced data compression algorithms. Included are some suggestions on how to make use of these properties. Many of these considerations have motivated recent JPL data compression research. Preliminary results of this research are briefly introduced at the end of the chapter (RM2).

The principal result of Chapter IV is that, based on first and second order considerations, there is a straightforward and practical way to supplement the considerable investment in existing coded communication systems such that the classic interaction between the channel and compressed data

disappears. Other less significant benefits are also provided. The solution involves the concatenation of Reed-Solomon block codes with Viterbi decoded convolutional codes for which the principal reference was a study done by Linkabit Corporation.[1] The overall system considerations presented here are not intended to necessarily tie down the precise performance and design parameters of a final communication system configuration. However, they do represent an extensive and thorough assessment of available information and should therefore provide a satisfactory basis for future simulations and study.

The block diagram of an Advanced Imaging Communication System (AICS) incorporating the Reed-Solomon concatenation system and recent JPL data compression research (RM2) is given in Chapter V.

II. CHANNEL CODING WITHOUT DATA COMPRESSION

Channel coding offers one means of improving the rate of information return with or without data compression.[1] In fact, historically the implementations or proposals of coding/decoding systems for Mariner missions have been made without regard to their compatibility with data compression. Three principal systems have resulted.

In addition to the uncoded channel, a (32, 6) block code was implemented for the Mariner '69 mission and a Viterbi decoded convolutional code has been proposed for the Jupiter/Saturn missions in the post 1977 period. The latter decoding system will almost definitely be implemented at the Deep Space Network (DSN) receiving stations. All three systems exhibit similar (and classic) characteristics when used to transmit compressed imaging data. The latter problem will be discussed in Chapter IV where "a solution" is demonstrated in the form of a straightforward, practical addition to the Jupiter/Saturn Viterbi system. The present discussion will focus on providing the necessary technical background for the uncoded, block and convolutional systems. Following historical precedent, this chapter will restrict attention to the transmission of uncompressed imaging data.

The Uncoded Channel

When we say "uncoded channel", we are really lumping many elements as shown in Fig. 1. The modulation systems currently envisioned for advanced Mariners and Pioneers employ both S-band and X-band carriers and a PSK squarewave subcarrier.[2] The telecommunications channel is accurately

[1] We will use the terms source encoding (or source coding) interchangeably with data compression. The former are used extensively in the theoretical literature.

[2] Some reasonable arguments for the choice of PSK modulation for coded systems is given in Ref. 2. Extensive information on the JPL operated Deep Space Network can be obtained in Ref. 3.

Preceding page blank

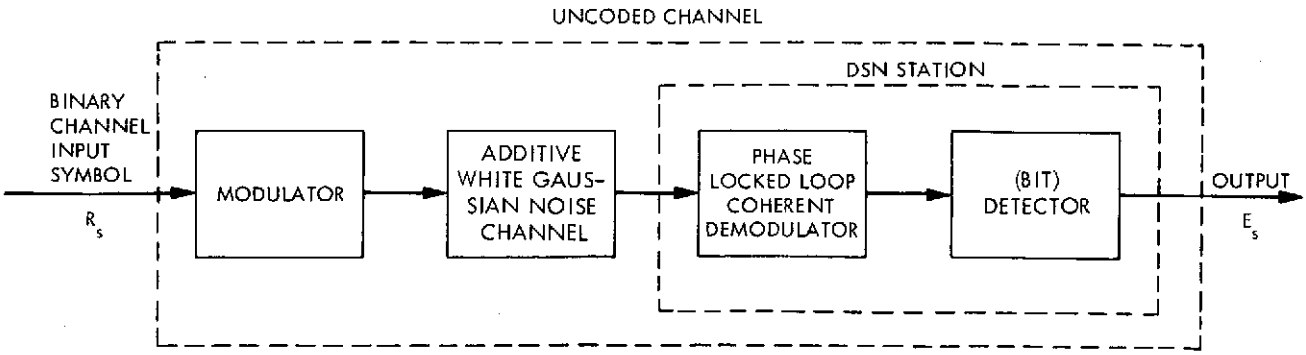


Fig. 1. The Uncoded Channel

modeled as white gaussian and the received data is coherently demodulated. There is, of course, an enormous amount of details and subtleties involved with operating this system which Fig. 1 does not do justice to. However, these considerations are of secondary importance and would serve only to obscure the main thrust of this report. Figure 1 can be reduced to a few critical parameters.

For each binary input symbol to the channel, the demodulator produces an output signal. The detector treats each such "noisy" signal individually, making a binary 0 or 1 decision. Because of the noise, the detector occasionally makes an error. The parameters required to characterize these "independent" errors are

E_s = Energy per received channel symbol (bit), (1)

N_0 = Single-sided noise power spectral density, (2)

P_s = Probability that an individual binary output symbol is in error. (3)

Because of the white Gaussian noise model, we can write[4]:

$$P_s = Q\left(\sqrt{\frac{2E_s}{N_0}}\right) \tag{4}$$

where

$$Q(\alpha) = \int_{\alpha}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\beta^2/2} d\beta \tag{5}$$

Thus, for a given signal to noise ratio, the uncoded channel can be modeled by the familiar memoryless binary symmetric channel shown in Fig. 2 with transition probability P_s (see Chapter IV of Gallager [5]). Consistent with earlier discussion, the diagram means that zeroes or ones at the input are independently caused to be in error at the output with probability, P_s.

Relationship of Rate and E_s

In reality, a channel symbol is transmitted over a time interval, τ. The transmission rate would then be $R_s = 1/\tau$ bits/sec. If the average received power is PWR, then E_s is given by

$$E_s = (PWR) \cdot \tau = \frac{PWR}{R_s}$$

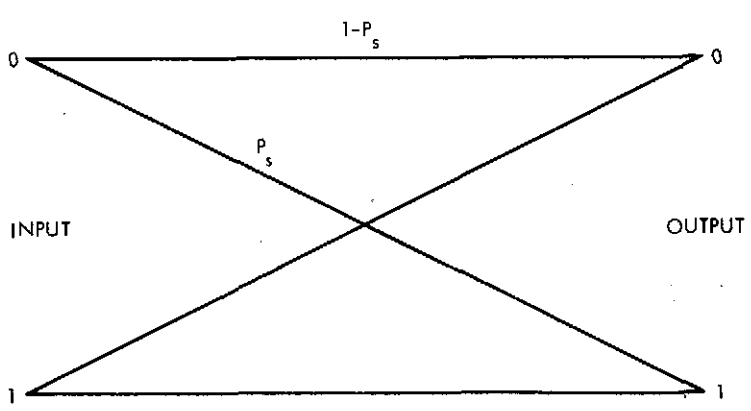


Fig. 2. Binary Symmetric Channel

Dividing by the constant noise spectral density N_0, we get

$$\frac{E_s}{N_0} = \frac{1}{N_0}(PWR \cdot \tau) = \frac{1}{N_0}\left(\frac{PWR}{R_s}\right) \tag{6}$$

Received power, PWR, depends upon a host of factors such as transmitter power, antenna gains, space loss, etc., all of which we don't want to get involved with here. It suffices to note that (E_s/N_0) may be increased by lengthening the interval τ. This, of course, means decreasing the transmission rate R_s. Ideally, if received power were to decrease (e.g., transmission distance increases), the signal to noise ratio could be kept constant by adjusting R_s accordingly. In practice, adjustments in R_s (and therefore E_s/N_0) have been limited to discrete steps, however, we will for the most part assume an arbitrary capability to adjust transmission rate.

Decibel representation. It is standard practice in communications to represent signal to noise ratios, rates, etc., in decibels (db). Unfortunately, this causes some confusion to those working in other areas. A useful example is provided by referring to Eq. 6 where increasing (or decreasing) E_s by x db corresponds to decreasing (or increasing) transmission rate by the same amount. The conversion of x to the equivalent multiplicative factor is given in Appendix A.

P_s vs. E_s/N_0 for the Uncoded Channel

Assuming synchronized phase coherent conditions at the DSN receiver, the probability, P_s of (4), that "an individual binary channel symbol entering the uncoded channel in Fig. 1 is detected improperly" is plotted as a function of symbol energy to noise ratio, E_s/N_0, in Fig. 3. Further discussion of the uncoded channel is deferred until we have introduced similar curves for two coded systems.

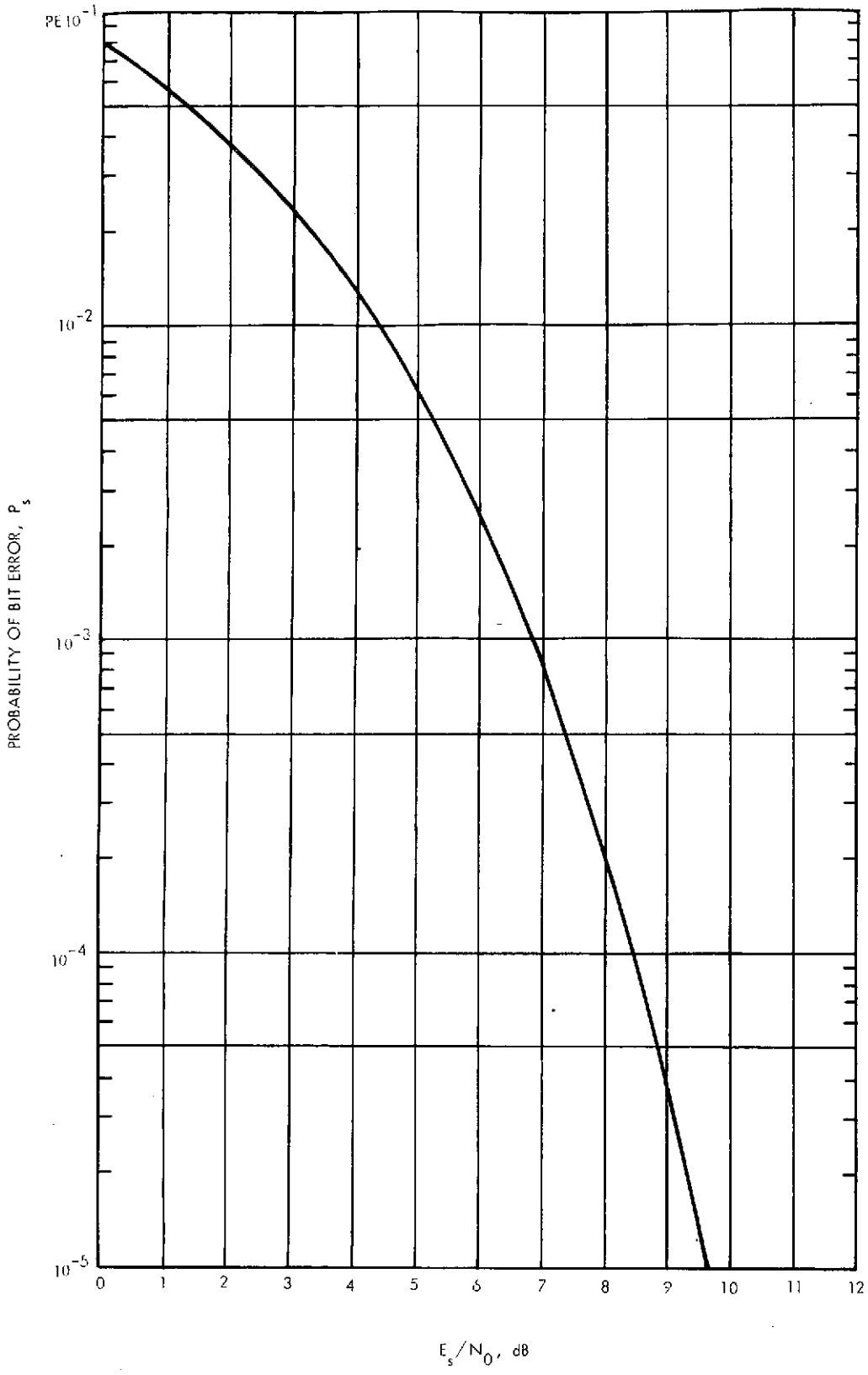


Fig. 3. Bit Error Probabilities for the Uncoded Channel

Two Coded Systems

A simplified block diagram of a coded system is shown in Fig. 4.

Points x and y locate the uncoded channel just discussed with bit symbol rate R_s and received energy per channel bit, E_s. In general, for every M information bits entering the encoder, there will be $M \cdot \nu$ coded bits leaving the encoder, where we define $1/\nu$ as the code rate. The exact relationship between input information bits and encoded output bits depends on the particular channel encoder. For any system, the information rate in and out of the coded system, R_b, is related to the uncoded channel rate, R_s, by:

$$R_b = \frac{R_s}{\nu} \tag{7}$$

Similarly, the received energy per information bit out of the coded system is:

$$E_b = E_s \cdot \nu \tag{8}$$

R_b and E_b are related as before through (6), (7) and (8).

Since the noise power hasn't changed, the critical parameter for the coded system is E_b/N_0. Clearly, the uncoded system is just a special case of a coded system in which $E_b = E_s$ and $R_b = R_s$. Consequently, we will continue with the new notation with R_b called <u>transmission rate</u> or information rate.

Coding theory says that for any transmission rate less than capacity, there exists coding schemes for which the error probability can be made

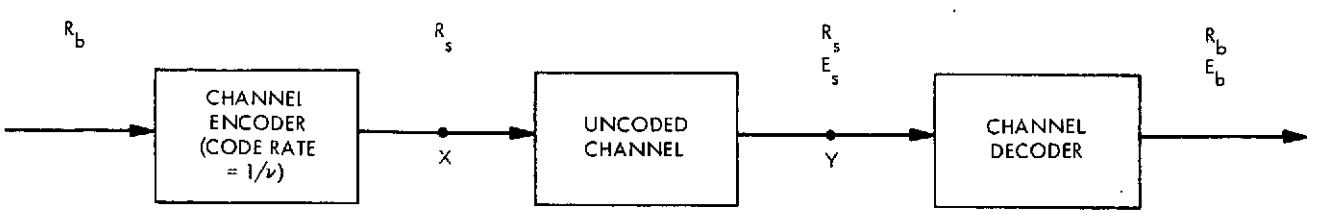


Fig. 4. A Coded System

arbitrarily small.[5] The underlying channel in this case is the infinite bandwidth, white gaussian noise, constant phase channel shown as a part of Fig. 1. Capacity for this channel can be shown to be -1.6 db in terms of E_b/N_0[2]. Such capacity theorems say nothing about complexity or how to find the systems. However, they provide a convenient means of determining how well a practical system is really performing.

In practice, the basic motivation for channel coding has been to reduce the frequency of errors in the "output information bit stream" for a given signal to noise ratio, E_b/N_0, or conversely, to increase the transmission rate, R_b, at which information can be transmitted with a given average error probability.

The general motivation takes a more specific form when a coding system is to be implemented for Mariner type planetary missions. We will investigate two such systems and their interaction with source encoded data. Details of these systems is superfluous to this report; the reader may consult the references.

The Mariner '69 mission implemented a (32, 6) block code with a decoder capable of operating at information bit rates up to 16 kbps. Primarily motivated by a requirement for higher decoding rates, a Viterbi decoded[3] $K = 7$, $\nu = 2$ (code rate = 1/2) convolutional code is planned for the Mariner Jupiter/Saturn missions in the late 1970's. In addition to providing improved coding gain over the block code, the latter decoder is available as an off the shelf item from Linkabit Corp. Its general applicability would seem to assure its inclusion at the DSN stations.

[3] An excellent tutorial on Viterbi decoding is given in Ref. 6 and the reader can find extensive performance characteristics in Ref. 7.

The performance curves for the uncoded, the (32, 6) block code, and the Viterbi K = 7, $\nu = 2$ system[4] are shown in Fig. 5. The ordinate, $\overline{P}_b$, is "average" probability of a bit error rather than simply probability of bit error as in Fig. 3. This is because for the coded systems, errors are no longer independent.

Error dependence for the block code is quite straightforward. At the encoder sequential groups of 6 information bits are mapped into 32 channel bits making up a codeword. When the decoder makes a codeword error, any of the corresponding 6 information bits could be wrong. Thus, errors can be thought of as occurring in bursts spanning 6 data bits.

Error dependence for the Viterbi system is considerably more involved and an explanation would require a detailed consideration of decoder structure. The length of error bursts in the decoded bit stream is random at any signal to noise ratio and can be as large as several times the constraint length K at low E_b/N_0 ($\approx$2 db). The "burstiness" diminishes as $\overline{P}_b$ is decreased (E_b/N_0 increased) and for our purposes in Chapter IV effectively disappears for $\overline{P}_b < 10^{-4}$.[5]

Source Data

Regardless of the type of sensor or scan technique, the end result of TV imaging is the representation of a 2-dimensional scene by an array of

---

[4]The performance curves in Fig. 5 assume ideal phase coherent receiver operating conditions. Non-ideal conditions will be discussed in Chapter IV. The Viterbi decoder is a Linkabit model LV7026 or LV7015C using 8 levels of soft quantization. Soft quantization means a quantized output of the demodulator in Fig. 1 is used by the Viterbi decoder rather than the detector output (hard quantization). This results in approximately 2 db improvement in coding gain. The performance curve for the LV7026 is slightly pessimistic relative to those given in [7] and reflects the results of more recent tests.

[5]A reasonable explanation for this phenomenon is obtained by noting the dominant terms in the equations for error bounds given in [6].

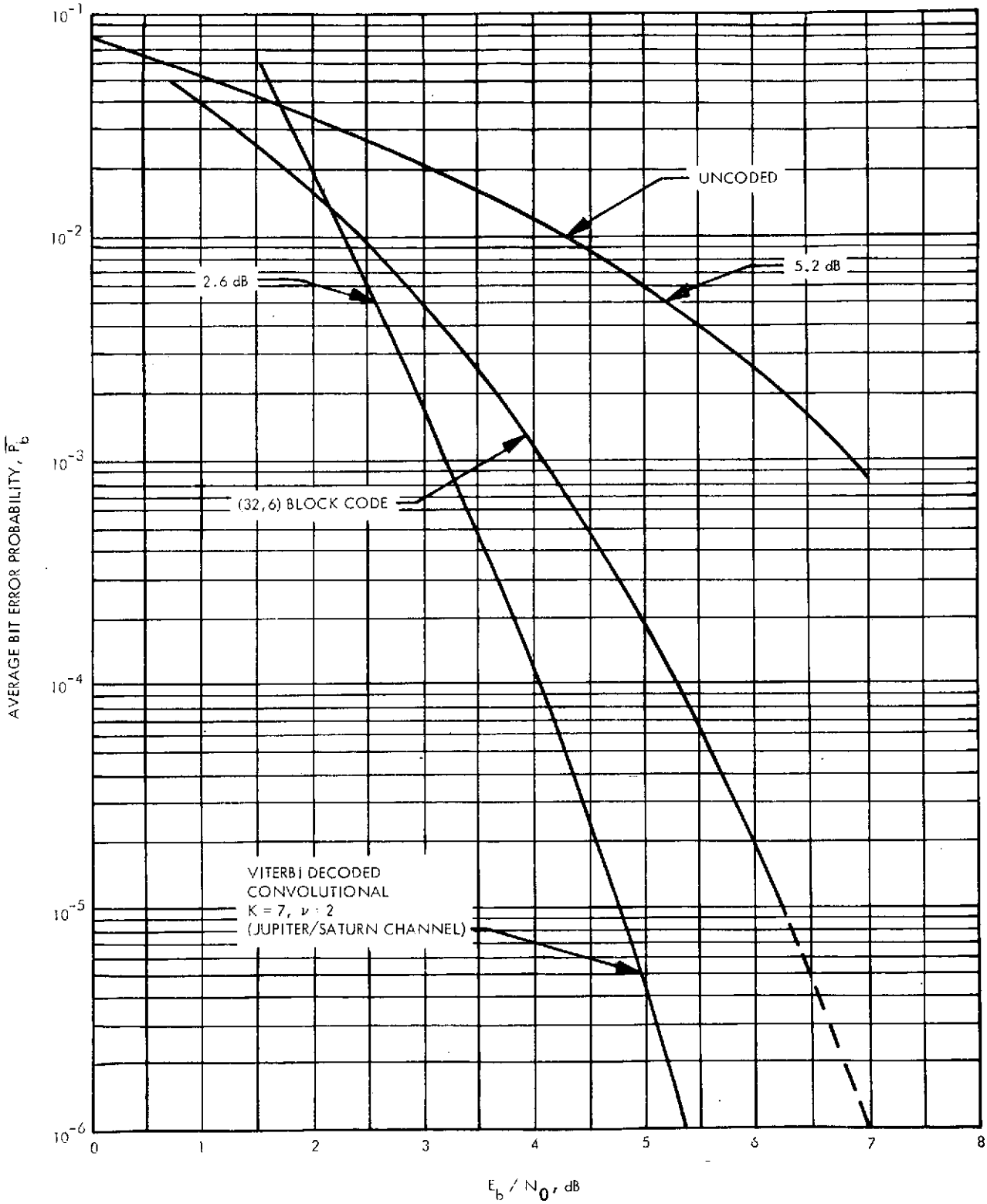


Fig. 5. Ideal Performance Curves: Uncoded, Block and Viterbi

numbers (we will call pixels). These numbers are quantized approximations to the brightness at a point or locality in the scene as sampled by the sensor. As an example, the proposed generator of these arrays for the Jupiter/Saturn missions is a line scanned vidicon with 800 lines, 800 pixels per line, and linear quantization to 8 bits/pixel (256 shades of grey). Advanced Pioneer missions employing imaging are anticipated to use a different approach. However, the only important consideration in this report is that a sampled and quantized version of a 2-dimensional scene must be transmitted back. Without data compression, it will require 6, 7, or 8 bits for each quantized pixel transmitted. Therefore, we lump the uncompressed output of all these alternative systems under the heading, PCM imaging data.

Error Considerations for PCM

When one or more errors occur in the bits making up a pixel, the reconstructed brightness will be wrong for that pixel. The magnitude of a given reconstruction error can be large or small. A statistical characterization is of no consequence here. The important, and obvious, consequence of these errors is that the overall quality and information content of a reconstructed picture decreases as the frequency of these errors increases. Based on observed reconstructed pictures for the three systems of Fig. 5, a rule of thumb has developed in judging allowable error rates. For average bit error rates below 5×10^{-3}, reconstructed quality is considered good to excellent. At the other end of the scale, quality is considered poor to unusable with $\overline{P}_b$ greater than about 1/20.[6]

[6] Actually, at any $\overline{P}_b$ above 5×10^{-3}, reconstructed quality resulting from the Viterbi system is slightly better than that resulting from the block code which in turn is slightly better than that from the uncoded system. This is a consequence of the relative "burstiness" of the systems discussed earlier. However, the slight differences are of secondary importance.

The Jupiter/Saturn Channel

Returning to Fig. 5, we note that at the $\overline{P}_b$ which corresponds to good to excellent quality, 5×10^{-3}, both the block code and the Viterbi system offer substantial advantages in terms of E_b/N_0 (and therefore transmission rate). For all ranges of reasonable quality, the Viterbi system is superior to the block coded system. Considering this factor and including its availability and other subtle advantages, its choice for Mariner missions (which don't include data compression) is a good one. In any case, its inclusion at the DSN stations is a virtual certainty at this time.[7] Summarizing, dictated primarily by the requirements of uncompressed imaging data, the future high rate telecommunications link structure will have the form given in Fig. 6. This is consolidated under the single heading "Jupiter/Saturn Channel". It should be noted that this structure represents considerable investment, far exceeding the cost of individual decoders.

---

[7] Many other more powerful Viterbi decoded systems have been simulated.[7] Increasing the constraint length K by one will increase performance at $\overline{P}_b = 10^{-5}$ by approximately .5 db. Unfortunately, this approximately doubles the decoder complexity. Going to a $\nu = 3$ code also substantially increases overall decoder complexity, but produces a gain of about .5 db at $\overline{P}_b = 10^{-5}$. For both cases, the improvements are less at the higher values of $\overline{P}_b$. A K = 7, $\nu = 3$ coder/decoder is being investigated for the shuttle spacecraft. It is also a likely possibility at the DSN stations.

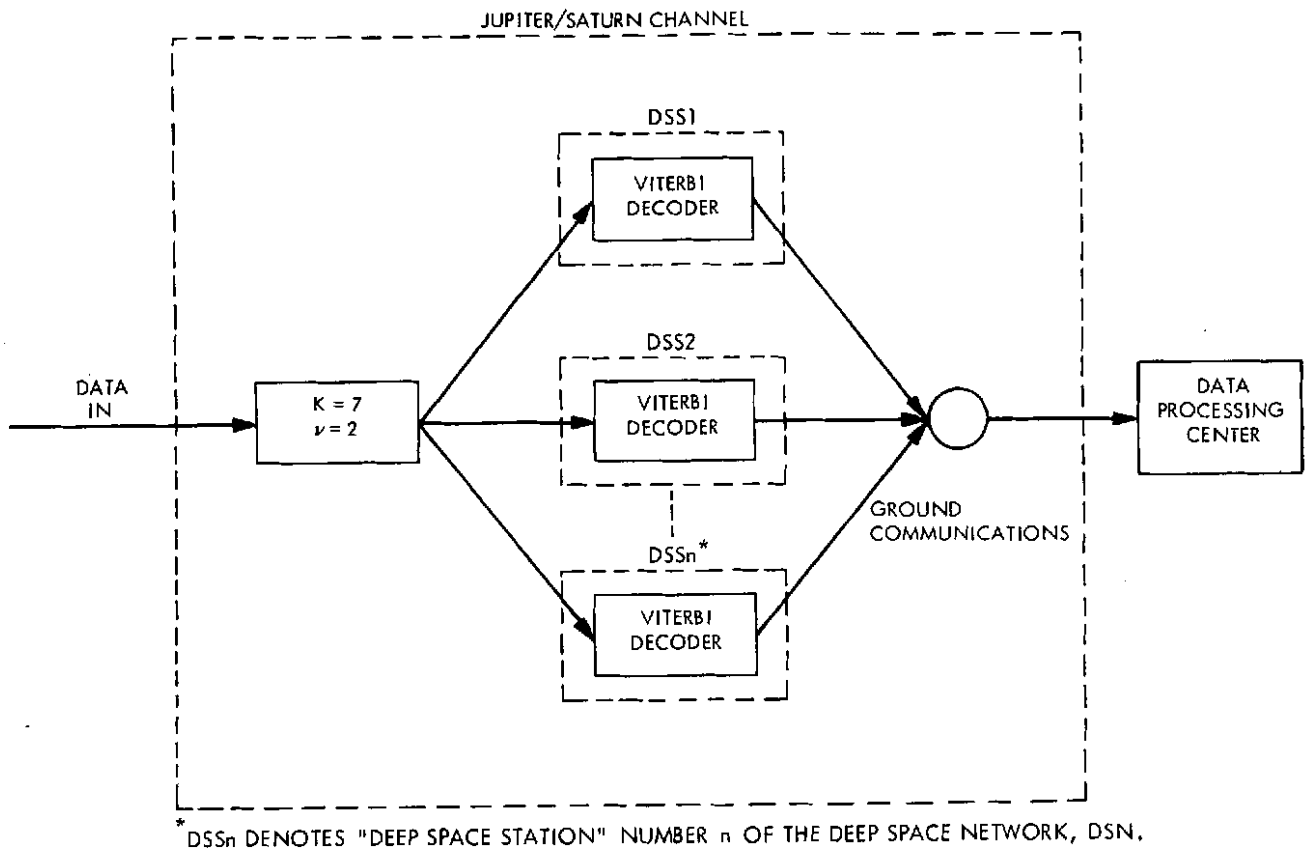


Fig. 6. The Jupiter/Saturn Channel

III. THE DATA COMPRESSION PROBLEM UNDER NOISELESS CHANNEL CONDITIONS

The vast majority of effort in image data compression has been expended in the context of a picture phone type of application. Applications to planetary imaging experiments are less well known and may be totally unfamiliar to many researchers outside of the space program. The two problems have a great deal of similarity, particularly with regard to specific algorithms and techniques used to "compress" data internal to individual TV frames. However, when looked at from an overall system viewpoint, the problems take on a different perspective.

This chapter looks, in considerable detail, at the system aspects of applying data compression to planetary imaging. The characterization of algorithms is maintained at a very general "black box" level. This approach allows considerable insight into the overall problem which might otherwise be obscured by detail. A discussion on picture rate/quality tradeoffs involving subjective judgements leads to an experimentally defined lower bound to performance. The extension of this concept to the source encoding of sequences of pictures (the real problem) points clearly to desirable properties for advanced compression algorithms. The latter considerations have been the prime motivation in recent JPL data compression research. Preliminary results of this research is briefly introduced at the end of this chapter.

Real-time applications are emphasized here and noiseless channel (error free) conditions are assumed throughout. In Chapter IV, the interaction of compressed data and the real telecommunications channels of Chapter II is treated. We'll find that by suitably modifying Fig. 6, the noiseless channel assumptions used here are quite adequate and practical. Only the

most basic source encoding definitions, given in the introductory portions of this chapter, are required.

PICTURE COMPRESSION

Rate

Consider first an individual PCM picture which we recall is an array of quantized numbers called pixels. Given a particular camera system, the number of bits representing an individual pixel is fixed by the number of levels of quantization. Consequently, the number of bits representing a block of PCM data (e.g. an array, a line segment or a complete picture) is fixed and depends only on the number of pixels in the block.

We will assume that all source encoders considered here code blocks of PCM data called source blocks. Practically speaking, the number of pixels in a source block can be assumed to be much less than for a complete picture. That is, many source blocks make up a complete picture. We will use $\mathcal{R}^{B}_{pcm}$ to denote the rate of a PCM source block, B. When discussing rates for complete pictures, we will simply omit the superscript, B. Such rates can be expressed in bits/source block (bits/sb), bits/picture (bits/p) or the more familiar normalized, bits/pixel.

Continuing, at the coding end when a source block of PCM data enters the source encoder, it performs its algorithms and produces a compressed version of the source block as output. That is, the source encoder produces a sequence of bits which represents the original source block. When the next source block enters, the process starts all over. Of course, in general, the results of coding one source block could influence the parameters of algorithms used in coding subsequent source blocks.

We define the rate of a compressed source block by $\mathcal{R}^{B}_{c}$. Fundamentally, the units used to express $\mathcal{R}^{B}_{c}$ doesn't matter, provided it is consistent. When

we do require units we will primarily use bits/sb. In this case, a compressed source block can be thought of as a sequence of bits, $\mathcal{R}_c^B$ bits in length. The more familiar expression of rate in bits/pixel is simply obtained by dividing by the number of pixels in a source block. Observe that $\mathcal{R}_c^B$ is not necessarily fixed for a given picture.

The familiar compression factor for a block, B, is given by

$$CF^B = \frac{\mathcal{R}_{pcm}^B}{\mathcal{R}_c^B} \tag{9}$$

The principal motivation for data compression is, of course, to obtain compression factors greater than 1 (i.e., $\mathcal{R}_c^B < \mathcal{R}_{pcm}^B$).

Exactly the same arguments hold when dealing with complete pictures. We simply omit the superscript B. As a guide to the reader, this chapter will be primarily concerned with picture rates (no superscript) whereas Chapter IV will be concerned with source block rates.

Each compressed source block, a sequence of bits, is transmitted over a communication channel and in this chapter we will assume this channel is error free. At the decoding end a compressed source block is "decompressed" or "decoded" to form a representation, $\hat{B}$, of the original PCM source block B. In general $\hat{B}$ is only an approximation to B. This is the subject of the next section.

A summary of the discussion thus far is given in Fig. 7 using two-dimensional source blocks as an example.

The following assumptions <u>are of no consequence in this chapter</u> since we assume noiseless channel conditions throughout. However, in Chapter IV we will investigate a more realistic situation in which some errors may occur.

To facilitate that discussion, we will assume that long synchronization sequences (sync words), having a negligible effect on overall data rate, are placed between (or included in) compressed source blocks.[8] Further, we will initially assume that the reconstruction of each compressed source block can be completed without supplemental information from other source blocks. In essence these assumptions limit the extent of an error's effect to a single source block. We emphasize again that the discussions in Chapter IV will demonstrate that disregarding the effect of errors in this chapter is a completely adequate and practical assumption from an overall system point of view. The preceding discussions and definitions are sufficient to permit the (unconvinced) reader to pursue Chapter IV first if he desires.

Quality

Rate or compression factor only partially defines performance. In general, a reconstructed version of a source block, $\hat{B}$, is not the same as the original B. The data has been distorted. Thus the missing quantity is a measurement of this distortion, a term widely followed in source coding literature. Instead we will primarily use its inverse, quality or fidelity, to define how good a block or picture is. There is no fundamental difference.

Quantitatively, researchers have used rms error and related measures to describe fidelity. These are sometimes useful, but inadequate, and frequently don't correlate well with subjective judgements. The problem is especially difficult in an environment of many different scientific users. Having a good quantitative measure for block or picture fidelity is desirable, but incomplete. As we shall see by example, the real source can be all PCM pictures in view of the camera in a particular time period, not a single picture. There is a tendency not to consider this bigger problem because no agreement can be reached in defining exactly how to analytically measure quality at the block or picture level. This unnecessarily obscures desirable

[8] Just about any existing practical algorithm can be made to meet this condition by combining smaller source blocks into a single large one.

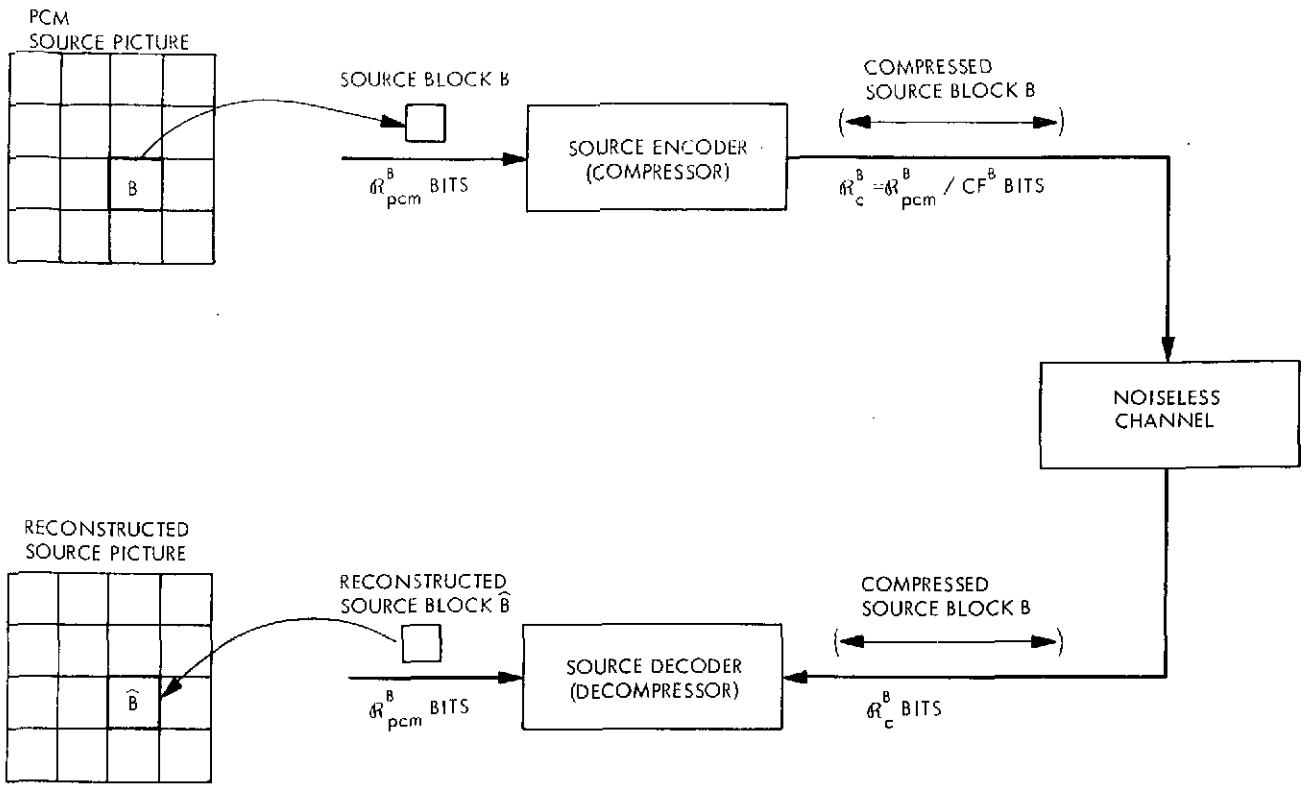


Fig. 7. Source Encoding Introduction

properties of real data compression systems which have a first order effect on the overall source coding problem.

We shall not pretend to have solved the difficult problem of finding a really useful all inclusive quantitative measure of fidelity. Instead, a more practical approach will be followed.

There is one case in which there is no question about quality. If we restrict each reconstructed source block to be exactly the same as the original, then the quality is limited only by the camera system producing the PCM pictures. We say the compression system is Information Preserving. An adaptive variable length coding system (RM1) that provides near optimal performance under this criteria for a wide range of data types is described in [8] and [9] . Rate performance depends on the data, typically ranging between picture compression ratios of 2 to 4 on 8-bits/pixel PCM source data.

The more active, detailed scenes require more bits to preserve the information and thus have lower compression ratios. This system employs either one or two-dimensional blocks.

To obtain rates lower than the minimum rate for an information preserving coder, it is no longer possible to guarantee that reconstructed data will be the same as the original. This is where the problem of defining fidelitey begins. We first consider picture quality and later will treat the idea of "value" of a picture sequence.

Figure 8 arranges the interacting parameters necessary for an open loop comparison of picture fidelity to (1) provide a basis for comparison of candidate real algorithms preceding a planetary mission and (2) provide a basis for predicting performance of a selected system during a mission. The first step indicated is to generate a sufficiently broad set of PCM sample pictures, D_1, D_2, ..., which typify potential characteristics expected to be encountered on the mission in question. Ideally, these include not only scene variations, S_1, S_2, ..., but also any significant variations in camera system parameters, P_1, P_2, ... In most situations, an adequate set D_1, D_2, ... can be obtained directly from prior missions.[9] For each viable algorithm, a decompressed version of each member of the test set could be generated. For fixed picture rate algorithms, this would be required for each operable rate.[10] We denote this collection of operable rates by $\mathcal{R}_c(1)$, $\mathcal{R}_c(2)$, ..., $\mathcal{R}_c(j)$, ...

[9] We will use the notation $\{D_k\}$ to define the class of data represented by the specific PCM picture D_k.

[10] Almost all existing compression algorithms are designed to operate at at most few fixed rates. We will assume that all such algorithms can be made to operate at a continuum of rates by using filler bits which contribute nothing to improving quality but simply increase picture rates to desired values. By operable rate we mean the design rate. If the algorithm actually operated at a continuum of rates, then a number of reasonably spaced rates would be selected.

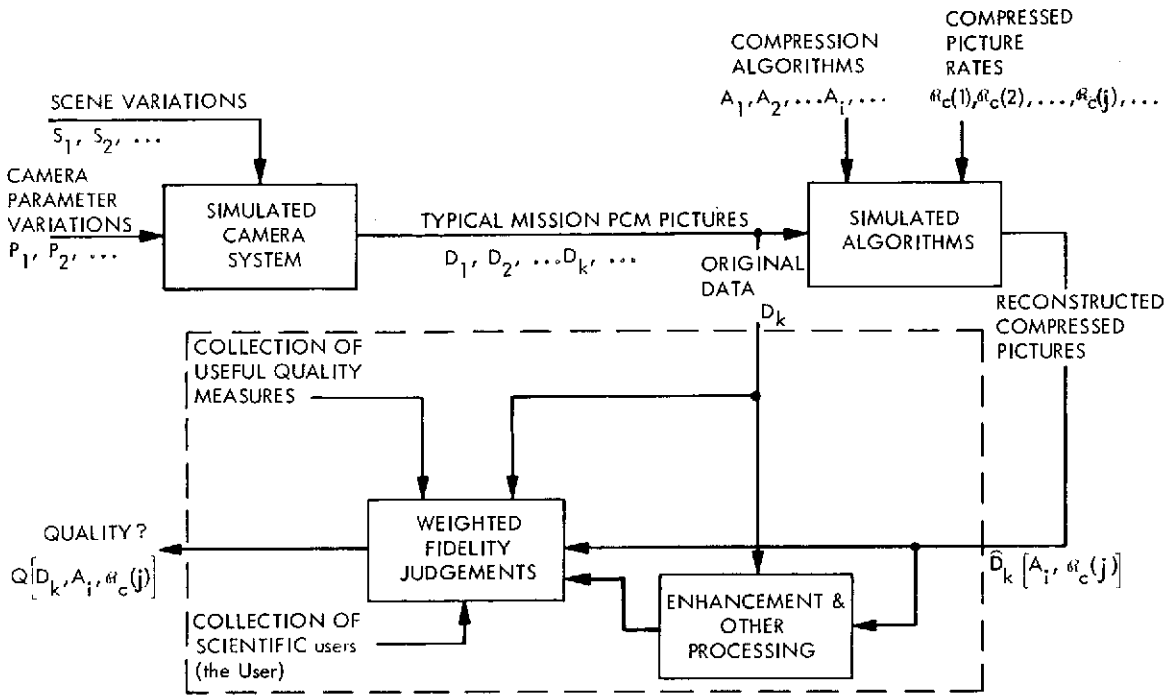


Fig. 8. Open Loop Fidelity Judgements

Decompressed pictures along with versions which have been further processed to bring out visual content could then be compared with the original data set by a collection of scientific users. Each of the users, who may each be from a separate discipline, can use and weight whatever existing methods of comparing quality which best serve his own scientific objectives. These techniques could include such things as purely visual subjective quality, crater counts, rms error, difference pictures, photometric measurements, etc. The weighted judgement of each user would then determine an overall quality rating or comparison (e.g., a geologist's opinion might receive a greater weighting when viewing cratered landscapes rather than clouds). For the present we will assume that all such weightings are fixed for a given $\{D_k\}$.

Henceforth we will use a capital "u" in User to denote the collection of scientific users.

With D_k a fixed sample picture representing a class of data, $\{D_k\}$, the implied quality function, $Q[D_k, A_i, \mathcal{R}_c(j)]$ in Fig. 8 is nonstatistical on a picture to picture basis. Hopefully, the D_k are chosen such that the quality resulting from a reconstructed picture is sufficiently "typical" of its class to give the User a good prediction of quality for any member of $\{D_k\}$. We are in effect sampling and quantizing the true quality function.

A structure such as in Fig. 8 is somewhat idealistic and probably familiar to most readers who have been even remotely involved with data compression. More realistically, an approximation to the experiment implied here is available in the form of extensive results in the literature, [10] conference proceedings, JPL research, etc. Keeping the structure of Fig. 8 in mind, we will use these results to obtain a first order, and practical, characterization of the interacting variables which influence picture quality. In pursuing this course, we will assume that visual subjective judgement of information content is heavily weighed in judging picture quality.

On an absolute scale, one can certainly identify reproduced pictures whose quality is Excellent. This probably includes any images that would yield to the User almost all the information available in the original. At the other end of the scale are images which are barely usable, Poor. All other useful pictures lie somewhere between these extremes. Trying to assign a scale directly to in-between qualities is a difficult, if not unrewarding, task. These comparisons are being made between a PCM original and reconstructed compressed pictures. If the reconstructed pictures are considerably distorted, it is extremely difficult to assign a meaningful absolute number to

quality (remember we are heavily weighting subjective evaluation). This "resolution in absolute judgement" problem is reduced by comparing pictures with other than an original. One can conclude such things as the qualities are about the same or the quality of one is slightly better than another, and so on. The latter comparisons of relative quality improve the definition of the quality function by crudely defining its gradient (i.e., by determining which direction the function moves when the variables D_k, A_i, and $\mathcal{R}_c(j)$ are changed).

To clarify this point somewhat, we consider an analogous situation. In an experiment, a collection of viewers are first presented with a blank field reference picture at maximum brightness and a relatively dark blank field picture A. Each viewer must subjectively guess at the brightness of A on a scale of 1 to 32. Later, he repeats this on blank field picture B which is actually slightly brighter than A. It is easy to imagine that a reasonable percentage of the time picture A would receive the same or greater brightness value than B. However, if in each case where an error was made the viewer was given a second chance and allowed to compare A and B directly, in almost all cases he would conclude that at worst, A and B have the same brightness. In the reduced number of cases in which an error persisted, the comparison of A and B could be "enhanced" to the viewer's eye by stretching the brightness scale. The brightness value on an absolute scale might still be wrong, but in all cases, the viewer would correctly decide that picture B was brighter than picture A.

In the same manner, the comparison of pictures which have about the same quality can be improved visually by enhancement techniques (Fig. 8)[10] which reveal to the User in more detail how much of what is important to him remains in the reconstructed pictures.

A Hypothetical Super System (SS)

Using this practical approximation to quantifying picture quality, we consider the concept of a Super System which provides us with a practical bound on performance at the picture level. The basic idea is to collect all the existing best performing algorithms into one system. Best here means "as determined by the experiment in Fig. 8" which includes the practical assumption that subjective judgements of picture quality are involved. It should become clear that the existence of an acceptable all inclusive quantitative measure of quality would simplify the definition and experimental determination of Super System.

For each operable rate, $\mathcal{R}_c(1)$, $\mathcal{R}_c(2)$, ... and each of the distinct data class representative source pictures D_1, D_2, ..., collect samples of reproduced pictures, $\hat{D}_k[A_i, \mathcal{R}_c(j)]$, which were "the best" of all algorithms. That is, we choose the $\hat{D}_k$ which satisfy

$$\max_{A_i} Q[D_k, A_i, \mathcal{R}_c(j)] \tag{10}$$

where, of course, $\hat{D}_k$ depends on A_i and $\mathcal{R}_c(j)$ (see Fig. 8). Because of the limitations in evaluating Q, even by this comparison of relative quality, more than one picture may satisfy (10). We denote each set of sample pictures that result by

$$\text{Pbest}[D_k, \mathcal{R}_c(j)] \tag{11}$$

We define Super System as one which contains all algorithms necessary to obtain one member of $\text{Pbest}[D_k, \mathcal{R}_c(j)]$ for each D_k and $\mathcal{R}_c(j)$. Assuming that the data class were known, such a system could operate at each operable rate of all individual algorithms and produce the maximum expected rate/quality performance in each case. In fact, with the assumption of filler bits, Super System could operate at a continuum of rates. Thus, the performance of this system could provide a useful lower bound to realizable

rate/quality performance.[11] We will further assume that Super System is also capable of classifying the type of picture data, $\{D_k\}$, (Pattern Recognition) and selecting out of its repertoire of algorithms the proper one. Thus, the only User input to this system would be the desired rate. Of course, we are disregarding any implementation considerations here. This factor could be introduced by discarding algorithms or collections of algorithms which exceed a given complexity, but we will not pursue this course further here.

What general characteristics would we observe in Super System?

Picture rate $\mathcal{R}_{ss}$ is arbitrarily selectable. (12)

For any type of source data, decreasing rate (increasing compression ratio) corresponds to monotonically non-increasing quality. (13)

Higher activity, detailed data will require more rate to preserve what is important to the User. For example, both the information preserving rate, $\mathcal{R}_{info}$, and the rate at which quality has degraded to barely usable, $\mathcal{R}_{poor}$, will generally increase as the data source is changed from low detail to high detail content. (14)

The characteristics described by paragraphs 12-14 are shown graphically in Fig. 9.

Super System as a Measure of Quality

Once the characterization of Super System has been established a more quantitive comparison of the relative performance of algorithms, with specific operable design rates, can be obtained by determining the rate

[11] As a lower bound to achievable performance, it may still be possible to find algorithms which perform better. Super System would simply be redefined by adding the improvements. In this sense Super System could be considered to provide a "practical" upper bound to performance which couldn't be exceeded without some work. In contrast, Rate Distortion theory [5], [10], [11] is an analytic approach to finding an absolute upper bound to expected rate/quality performance. Existing solutions to rate distortion bounds for imaging data are useful but suffer from a lack of adequate source modeling and acceptable quantitative measures of quality (distortion).

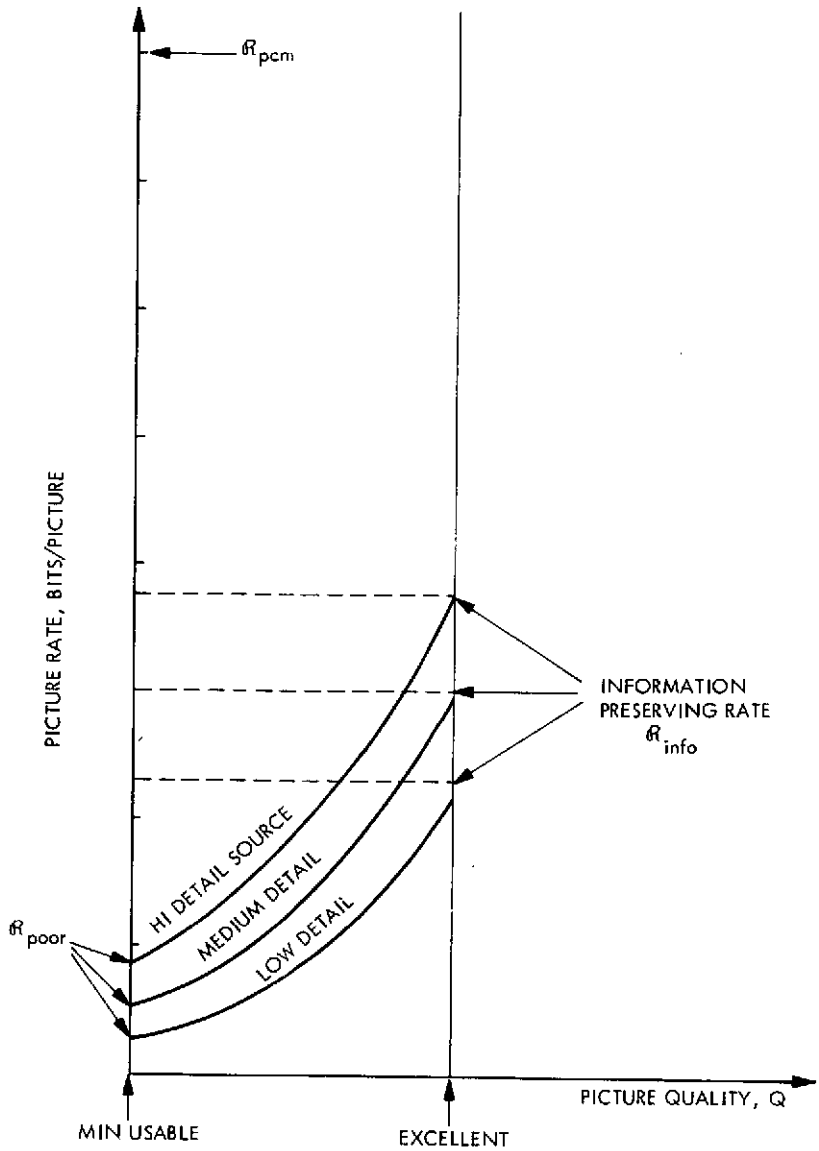


Fig. 9. General Characteristics, Super System

required by Super System to obtain about the same quality as the particular algorithm. This again takes advantage of a relative comparison of pictures of approximately the same quality. In this way the output rate of Super System can be used to quantitatively define quality. To clarify this, consider Fig. 10.

In the upper part of the diagram a sample picture representing class $\{D_k\}$ is operated on by Algorithm A* at operable rate $\mathcal{R}^*$ producing

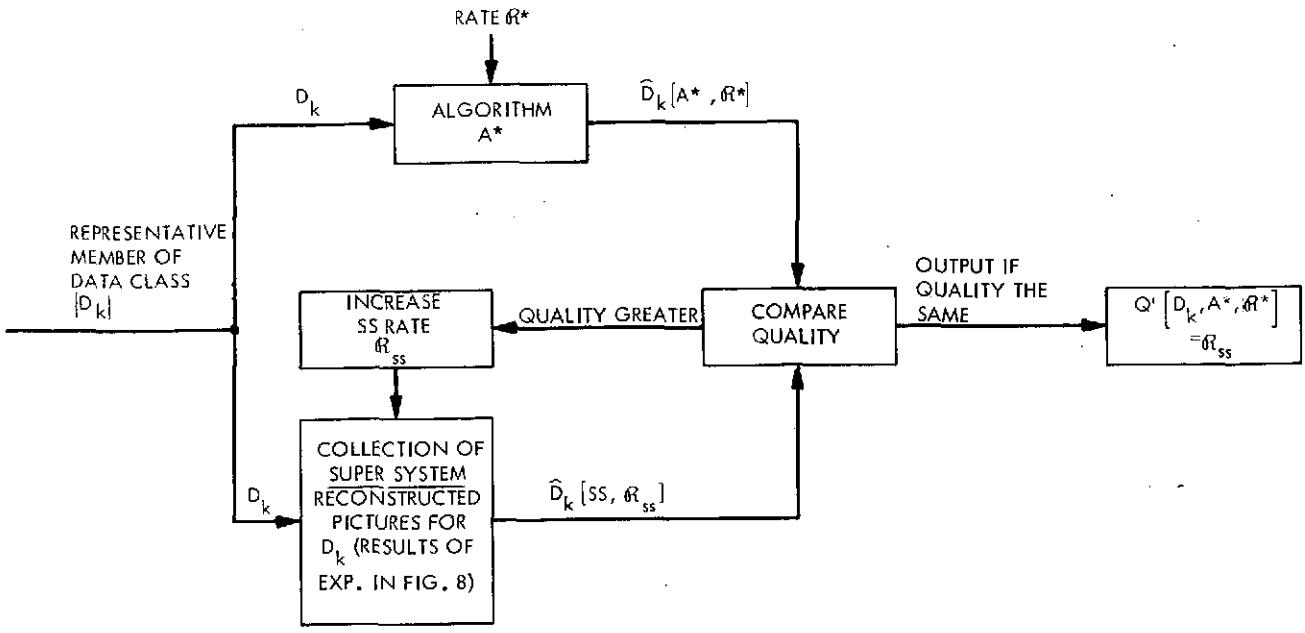


Fig. 10. Using Super System to Measure Quality for a Given Data Class, $\{D_k\}$

reconstructed picture $\hat{D}_k[A^*, \mathcal{R}^*]$. This is compared with pictures similarly reconstructed from Super System, $\hat{D}_k[SS, \mathcal{R}_{ss}]$. But since we have assumed that Super System has already been defined by Fig. 8 a collection of reproduced pictures (and other pertinent test results) is available in a file for input picture D_k and all operating rates. With progressively increasing Super System operating rates, $\mathcal{R}_{ss}$, reproduced pictures (and other results) are compared with the picture from algorithm A* until quality is considered about the same. The output of this experiment is the corresponding Super System output rate when the match occurs.

By the definition of Super System, if A* is not a new algorithm, we must have $\mathcal{R}_{ss} \leq \mathcal{R}^*$. Knowing $\mathcal{R}_{ss}$ uniquely defines quality for Algorithm A* in the form of a catalogued output picture, $\hat{D}_k[SS, \mathcal{R}_{ss}]$, etc. from Super System.

Therefore, the output of Fig. 10 becomes a quantitative quality measure $Q'[D_k, A^*, \mathcal{R}^*]$.[12] This concept more clearly illustrates the bounding provided by Super System.

The preceding outlines a systematic plan for putting numbers to quality when subjective judgements are involved. As noted earlier by referring to results in the literature,[10] we can easily make some general observations about existing algorithms which approximates the experiments in Figs. 8 and 10. In particular, we are interested in how these algorithms compare with the hypothetical Super System. Because of its unfamiliarity, we will not directly pursue the concept of Fig. 10 in future discussions. A return to a more heuristic treatment is better suited to our main pursuit. We note, however, that it is available as a potentially useful tool for quantifying quality.

In comparing individual algorithms with Super System we would observe the following:

An algorithm that performs well (relative to Super System) on one class of data may do poorly on another. This point is illustrated in Fig. 11. (15)

[12] With quality defined as the experiment in Fig. 10, a plot of rate vs. quality for Super System would be a 45° straight line, regardless of data source. To take into account variations in quality in going from one type of source to another (Fig. 9) would require a weighting which depended on the class of data. Generally, this weighting would be larger for low detail data than for high detail. This might be accomplished by using the structure of Fig. 10 to compare Super System with itself on different data sources.

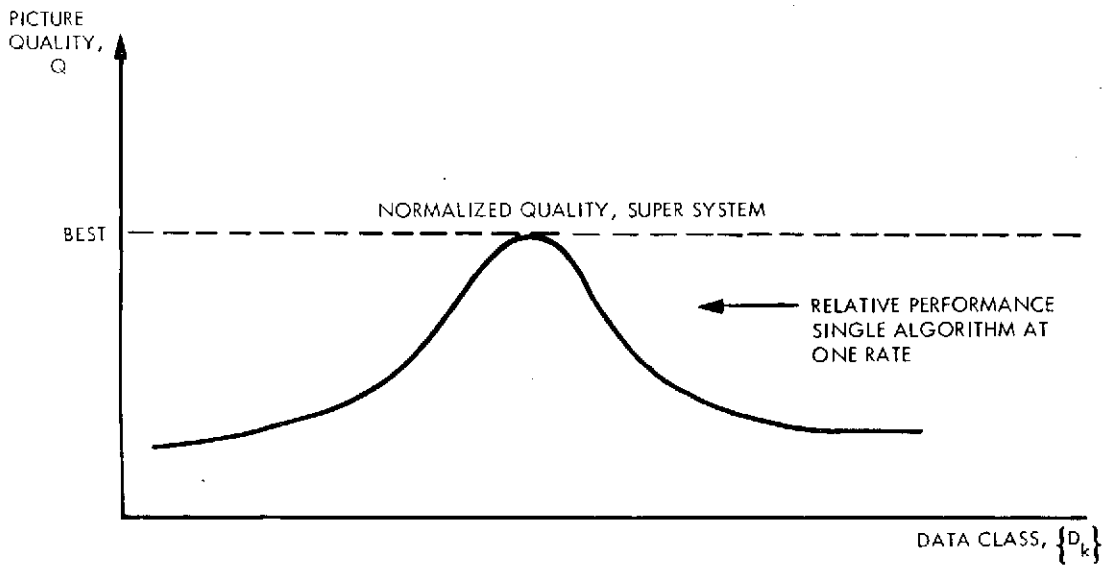


Fig. 11. Sensitivity to Data Variations for Individual Algorithm

In addition, an algorithm that does well at one operable rate may, if it has another operable rate at all, do poorly at another. (16)

Higher performance, especially at the lower picture rates, generally is paid for with increased complexity. (17)

IMAGING SEQUENCES

The limitations of individual algorithms described in paragraphs (15) and (16) can have severe ramifications on the real source coding problem which involves sequences of images. To see this more clearly, we will extend our discussion to the use of source encoding algorithms in a mission environment.

We introduce the heuristic term, VALUE of an imaging sequence and denote the function by SV(•). As in the case of picture quality we lay no claim to be able to precisely quantify the term in a way which everyone will agree. However difficult to define in an absolute sense, much can be said about the interaction of parameters which influence their relative increase or decrease. This observation gives us an approximation to the gradient of SV(•) even if a

hard quantitative number cannot be assigned. Finding local maxima of SV(•) in this problem means finding where the critical parameters (which include available onboard source encoding options) produce a zero gradient. By following this approach, we can gain considerable practical insight.

Consider the hypothetical, but not unrealistic, situation shown in Fig. 12. Here we assume that from a distant observation point (1), a single low resolution image (frame) of a large area of a planet is transmitted to Earth. Earth observers use this **a priori information to plan imaging sequences** to be used during a high resolution observation period (3). Commands must be received before this period begins (2). The User will try to optimize the use of onboard source encoding options he has available in order to maximize the VALUE of data returned during the sequence. We have a problem of source encoding with feedback.

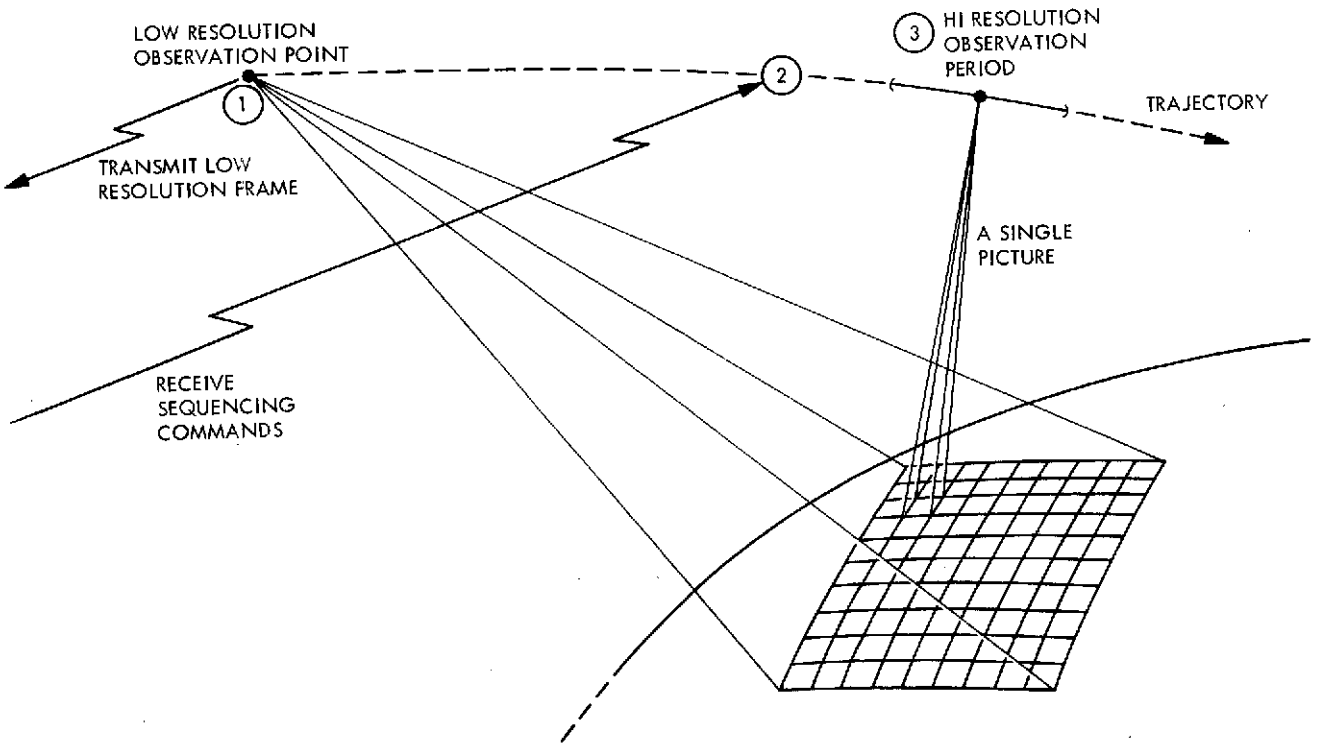


Fig. 12. A Hypothetical Imaging Sequence

Before continuing we need some further definitions and terminology. The word "coverage" in a colloquial sense refers to the area of the surface of the planet "covered" by a picture, a sequence or even a mission. For our purposes Maximum Coverage during the high resolution imaging sequence in (3) of Fig. 12 means that all possible images were transmitted (90 in this example). This limitation might result because of camera pointing restrictions or simply because of a maximum camera output rate. PCM Coverage refers to the maximum number of standard PCM pictures which could be transmitted during the sequence time. This is simply the number of bits available on the real-time channel divided by the number of bits in a PCM picture. Since we are presently assuming the camera system to be fixed, PCM Coverage is really a minimum coverage. Thus, we define Normalized Coverage, denoted NSC(•), as the ratio of Actual Coverage to PCM Coverage. It is easy to see that Normalized Coverage is another way of saying: average compression ratio during the sequence. The terminology emphasizes the real tradeoff being made.

In principle, the User must consider situations like those in Fig. 12 before a mission in order to assess the potential usefulness of data compression systems he may select for the mission. He can make "best estimates" of the situations to be encountered as in Fig. 12 based on his Earth based knowledge. However, he should also consider the possibility that the low resolution observation in (1) may significantly alter these **a priori assumptions**. In the same manner, the way in which the User assigns priorities to parameters which affect sequence values (e.g., data class, coverage, quality) may be much different than originally envisioned by the time an actual encounter occurs. This might be the result of "new information" from the low resolution observation in (1) or simply a rethinking of scientific objectives.

To attack this problem in a reasonable way, we will start with a sufficiently constrained example that allows us to make some basic observations.

Sequencing Examples

- Assume that from the low resolution observation picture, the User was able to accurately predict that all high resolution PCM pictures would belong to class $\{D_k\}$. However, the User was unable to obtain any detailed information.
- Assume a severely rate limited situation so that the maximum coverage possible was considerably greater than the PCM coverage.
- Assume that a single data compression algorithm, A*, is available which operates at a single picture rate, $\mathcal{R}^*$ (Rate $\Rightarrow$ Compression Factor $\Rightarrow$ Coverage) and is a member of Pbest $[D_k, \mathcal{R}^*]$. Since A* is one of the best algorithms (see (11)) for this rate and class of data we can assume it is one of those included in the definition of Super System.

Note that, to start with, we are considering a situation in which the mission is on. There is no recourse to redesign at this point.

To investigate the potential usefulness of algorithm A* under these initial conditions, consider the three-dimensional graph in Fig. 13 which plots sequence value, SV, as a function of picture quality, Q, and Normalized Sequence Coverage, NSC.

Starting with the boundary conditions, the minimum SV occurs at the origin with NSC = 1 and Q corresponding to minimum usable. This condition might result because of camera system failure mechanisms.

In general, if either Q or NSC is increased, SV will tend to increase (and will not decrease). In particular, moving along the Q axis, SV increases to a maximum at point A_2. This corresponds to sending PCM. For this

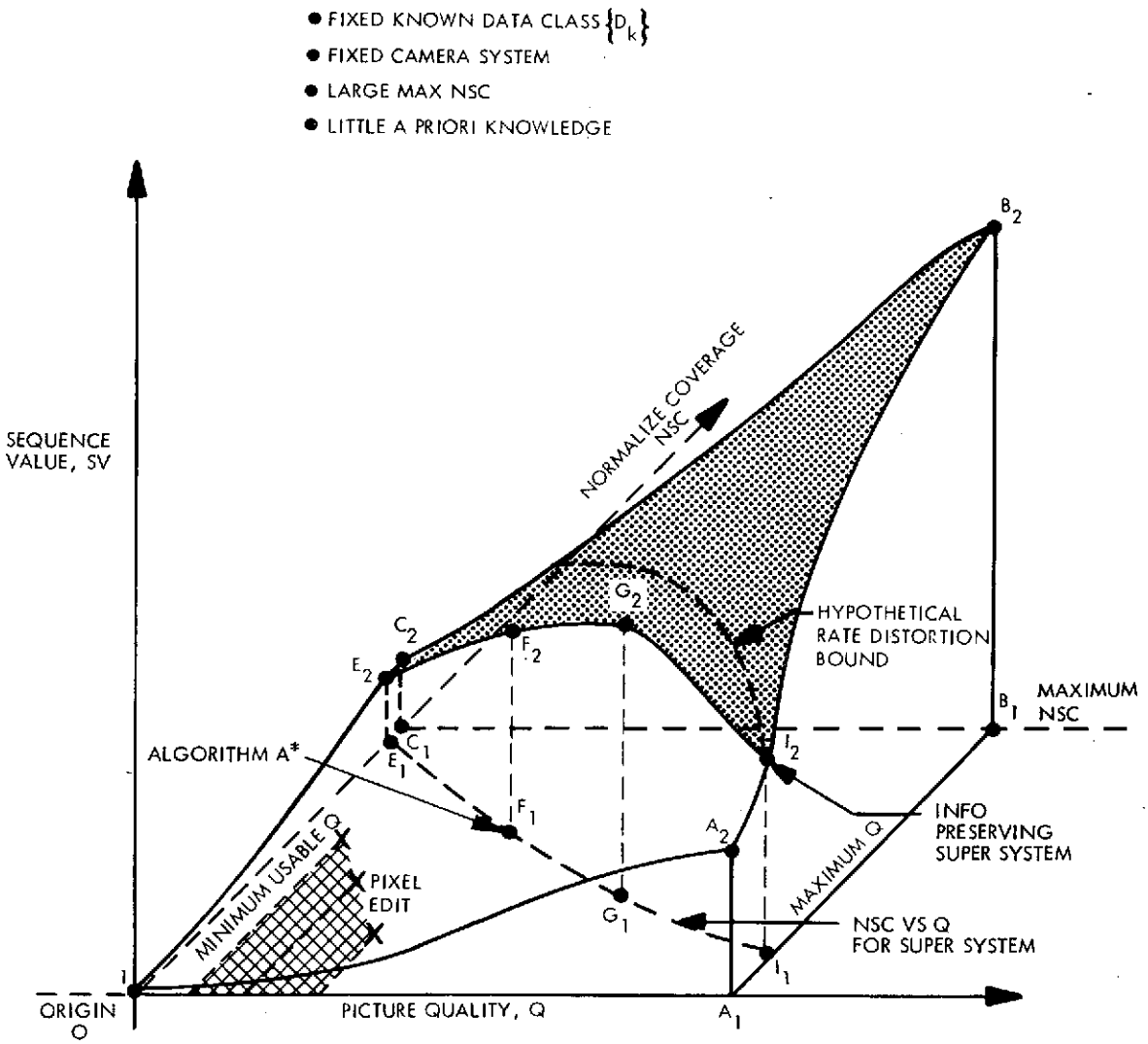


Figure 13. Sequence Values, I

hypothetical example, the rate at which SV increases with Q is shown to be decreasing as point A_2 is approached. This represents a case of diminishing returns which might not be true under different circumstances. Continuing around the boundary from A_2 to B_2, holding Q constant while increasing NSC to a maximum, SV again tends to increase. We make similar observations in traversing the boundary from the origin to point C_2 to B_2.

Comparing points A_2 and C_2 reveals that, for this specific example, it is considered more useful to send PCM at minimum coverage than obtaining maximum coverage with minimum usable quality.

Shown plotted in the NSC, Q plane is the performance curve for Super System (points E_1-F_1-G_1-I_1). Quantitatively, this might have been obtained by the experiment in Fig. 10 (remember NSC is inversely related to picture rate). Point F_1 on this curve corresponds to the operating point for algorithm A* and I_1 corresponds to the information preserving operating point where quality is the same as for PCM.

The Super System NSC vs. Q curve has been projected onto the SV surface where E_1, F_1, G_1, and I_1 become E_2, F_2, G_2, and I_2, respectively. Since the rate of Super System is arbitrarily selectable, this curve represents a lower bound to realizable sequence value, SV. For this hypothetical example, a maximum is obtained at G_2.

A heuristic idea of how Rate Distortion Theory (see footnote 11, p. 27) might fit in here is also shown in Fig. 13 where a hypothetical rate distortion bound of picture rate vs. quality (for data class $\{D_k\}$) has been projected onto the SV surface. This curve, shown dashed, would provide an absolute upper bound to obtainable sequence values. Ideally, the theory would be applied in a more direct manner to the complex source coding problems we are posing in this chapter.

With only algorithm A* and PCM to choose from means that the User must operate at one of two operating points in Fig. 13, F_1-F_2, and A_1-A_2. In this case, it is shown to be more valuable to use A* than PCM.

In making this judgement, the User is using his prediction of data class $\{D_k\}$ from the low resolution picture (1 in Fig. 12) to select an output of the experiment in Fig. 8 which is then used to predict the reconstructed picture quality produced by A* on members of $\{D_k\}$ (at its single operating rate).

The User need not have constructed the three-dimensional surface in Fig. 13 to make the binary decision between two operating points. However, if he did, he would conclude that, at least for data class $\{D_k\}$ and single operable rate algorithms, he should have originally selected an algorithm which operates best at point G_1 instead of F_1 (see (11)). The apparently erroneous choice of algorithm A* can be explained in a number of ways.

(1) Algorithm A* also works best on data class $\{D_\ell\} \neq \{D_k\}$ not only in terms of Q, but also in terms of SV (by a similar construction to Fig. 13 for data class $\{D_\ell\}$). Then a choice of algorithm A* would have been best if a priori knowledge before launch predicted a higher frequency of occurrence of $\{D_\ell\}$ than $\{D_k\}$ and/or higher priority was placed on $\{D_\ell\}$. Clearly, it would be more desirable to have both operating points.

(2) An easier explanation is simply that the surface representing SV in Fig. 13 has changed since the original choice of A* was made. This reflects a change in priorities which is certainly not hard to imagine in a mission which may take several years to complete.

Whereas point F_2 corresponds to an Earth based a priori maximum, point G_2 might represent the maximum after receiving the low resolution observation picture or simply a rethinking of scientific objectives during the course of the mission. In general, the maxima on the Super System SV curve could be located anywhere. Before a mission, the User can only predict a "most likely" location. The variations could be more substantial when looked at from a multi-mission, multi-planet viewpoint.

The principle implications of cases (1) and (2) above are that, even for this very constrained example, the advantages of having multiple operating points is clear. Having the full continuum of operating points provided by Super System is, of course, ideal. The latter provides the User with the maximum flexibility to adapt to the mission as it develops. Again, the bounding nature of Super System is apparent.

Low performance algorithms. It is not just the continuum of operating points that gives Super System such flexibility, but also its high performance. By the use of filler bits (which simply reduce coverage, but don't affect quality) the User could operate anywhere on the SV surface of Fig. 13 below E_2-F_2-G_2-I_2.[13] In the (NSC, Q) plane, this corresponds to the area (origin-A_1-I_1-G_1-F_1-E_1-origin). On the other hand, consider a relatively low performance algorithm such as Pixel Edit[14] used on the Mariner 10 Venus/Mercury flybys and planned for the Mariner Jupiter/Saturn 77 missions. The approximate location of typical operating points for 2:1, 4:1, and 8:1 Edit Schemes are shown in the (NSC, Q) plane as X's. The placement is based on JPL research involving experiments similar to that in Fig. 10. Even if we assumed that we could operate at all intermediate qualities between Editing Schemes, the User could still only operate in the cross-hatched region shown.

Although the Pixel Edit Schemes give the User three choices of coverage, it may still be less useful than the single rate, high performance

[13] We are neglecting the fact that Super System might have discrete quality points.

[14] A very simple algorithm to implement onboard, an N to 1 Edit Scheme means that only 1/Nth of the original PCM samples are transmitted. Linear interpolation schemes are used to reconstruct the missing samples.

algorithm A*. In the example of Fig. 13, the Q of algorithm A* is shown to be slightly better than 2:1 Edit. Hence, by the use of filler bits, any coverage that can be obtained by the Editing Schemes can also be achieved by algorithm A*, but at better quality and, therefore, at a greater value to the User.

Changing the data. This is not always true, however. Recall that thus far we have constrained the data to be from a fixed class $\{D_k\}$. If the User expects to encounter data from some other class $\{D_m\}$, then he must look at the problem all over again. The same arguments we have just made would again apply. We would draw similar conclusions in comparing Super System with Pixel Editing. However, the high performance algorithm A* which was one of the best for data class $\{D_k\}$ may perform poorly on $\{D_m\}$ (See Fig. 11). This data sensitivity exhibited by A* may make Pixel Editing more valuable when encountering $\{D_m\}$. These observations point out another desirable feature of Super System in that it is capable of performing well on the full range of data to be encountered.[15] This becomes even more important when the User is trying to trade off different data classes which occur in the same imaging sequence. In another situation, the User may not be able to accurately predict the data class (an assumption we've made so far). In such a case, it is obviously desirable that the available algorithm not fall apart.

Better apriori knowledge, lower NSC. To point out some other variations, we consider a modification of the SV graph in Fig. 13. In modifying our initial assumptions, we will assume first condition A, then A and B below.

[15] In the definition of Super System, we assumed that 1) it could recognize the data class it was operating on, and 2) used one of the best algorithms for that data class and picture rate. Thus, A* would not be used if the data were from $\{D_m\}$.

A. The low resolution picture in (1) produces much more information about the area to be observed in (3) than simply the data class $\{D_k\}$.

B. The transmission rate situation is significantly improved over our initial assumptions (e.g., X-band instead of S-band). This means that the number of PCM frames that can be transmitted in a given time interval is increased so that the Maximum Normalized Sequence Coverage, NSC (a comparison with PCM), is reduced.

A first order approximation to the changes to Fig. 13 introduced by assumption A is given in Fig. 14. Basically, it amounts to passing a horizontal plane through Fig. 13 to reflect the fact that returned images are worth more when you a priori know very little of their content then when you already have considerable information. This is shown in Fig. 14 by moving the origin from O to O' and replacing SV by SV^a. Points such as A_2, G_2, I_2, and B_2 are shown unchanged. Of course, they have new values given by SV^a. Relationships in the horizontal plane which contains the new origin O' are unchanged from the equivalent plane containing O. Equivalent points are noted using primes.

As shown, the (NSC, Q) location which achieves maximum SV^a for Super System is the same for both figures (G_1 or G_1'). Of course, the sequence value obtained has been reduced (SV^a instead of SV).

On the other hand, the modified surface in Fig. 14 shows that, for this example, the Pixel Edit schemes (the X's) can no longer improve on the information already obtained from the low resolution observation pictures.

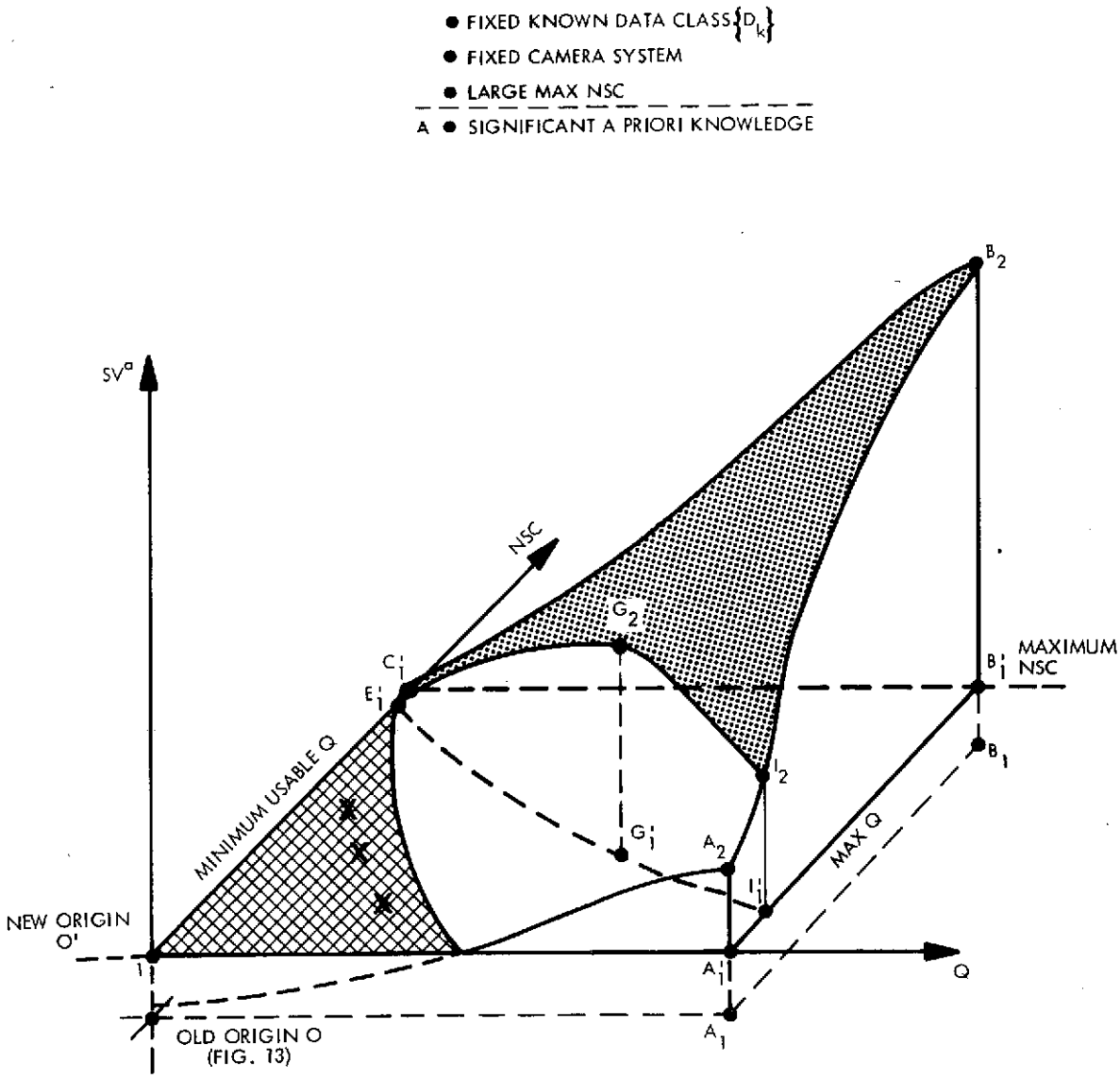


Fig. 14. Sequence Values, II

The introduction of Assumption B further alters the graph to that shown in Fig. 15. Clearly, increasing the available transmission rate cannot be harmful. Any absolute coverage that was obtainable in Fig. 14 with a given quality is now increased, resulting in increased sequence values. In terms of Normalized Sequence Coverage, NSC, the effect is to move the origin from O' to O'' and replacing NSC by NSC^b to reflect this shift. Many points from Fig. 14 have been retained to show the relationships. PCM operation in the (NSC^b, Q) plane has moved from A_1' to A_4, with increased SV^a indicated by the relative length of lines A_4-A_5 and A_1'-A_2. The new location for the Pixel Edit options (the X's) again gives them value, but still almost insignificant relative to PCM for this example.

The shift in the Super System NSC^b vs. Q curve results in a new location for a maximum SV^a, shown as G_5 on the SV^a surface and G_4 in the (NSC^b, Q) plane. Thus, for this example, if the User had Super System (or a system which approximated it), he would shift his operating point to increase picture quality while actually increasing absolute coverage obtained. For other situations, the new maxima for Super System might be located quite differently. Again, the full range of alternatives provided by Super System (or its approximation) would allow the User to adjust for changes in his interpretation of the SV^a surface right up to the last moment.

Camera system changes. The situation in Fig. 15 shows that because of the increased data rate, the range over which even Super System is applicable (i.e., in the NSC^b, Q) plane) has been reduced. This would appear to be of no consequence since the sequence value situation is better than before. This can lead to fallacious reasoning if we carry this argument further by assuming an available data rate so high that almost all pictures possible could

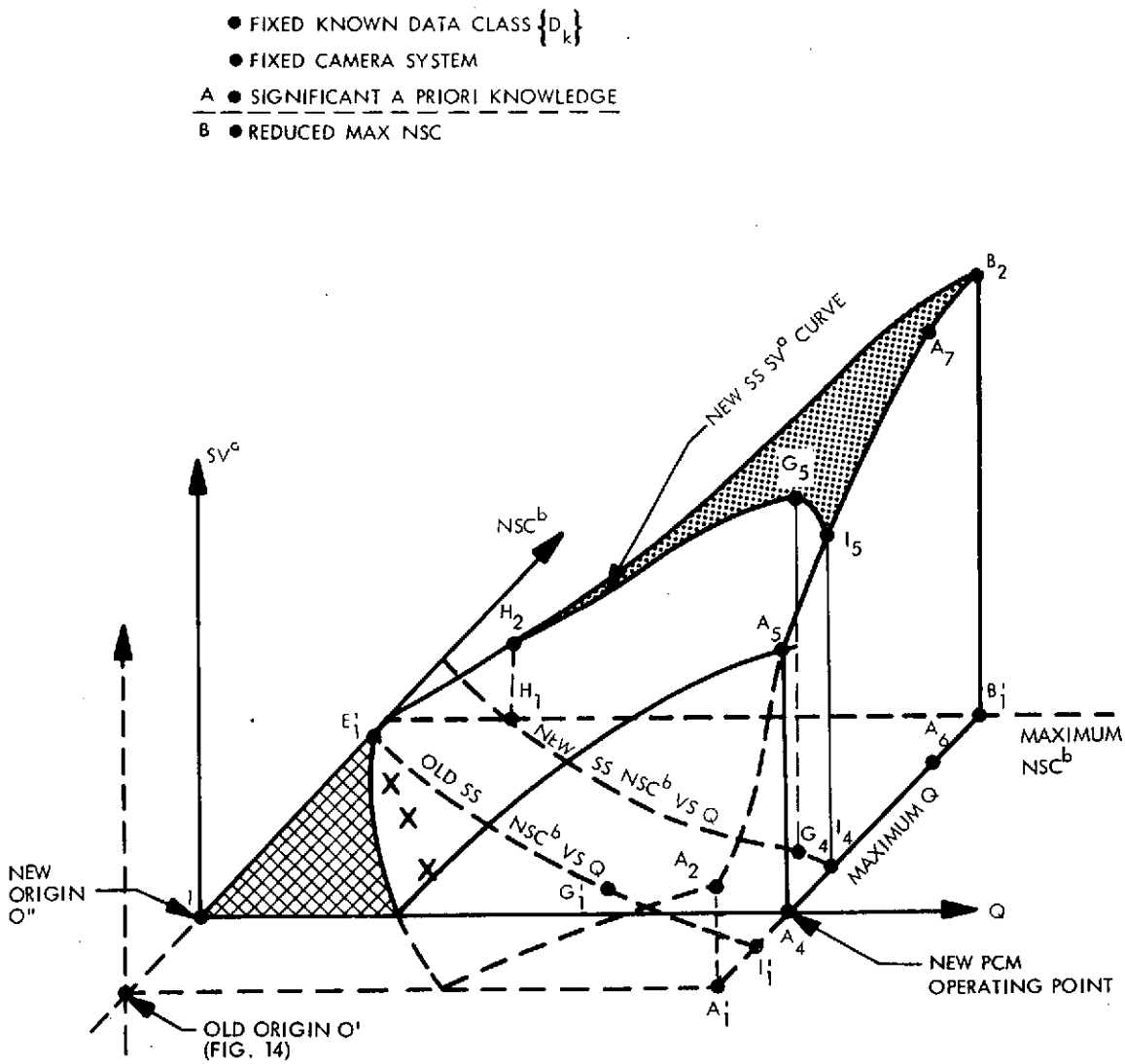


Fig. 15. Sequence Values, III

be transmitted using PCM alone. For example, this might move the PCM operating points to A_6 and A_7 in Fig. 15. The User's options available during the mission have essentially been reduced to PCM or nothing. But so what? If he is getting almost full coverage at maximum quality, who needs data compression? Such statements ignore the fact that all along, we have arbitrarily fixed the camera system. Once a camera system's parameters have been juggled and fixed at launch, the User can't change his mind about increasing picture quality later. In this latest example, decreasing Q doesn't buy him anything either. Assuming that an approximation to Super System was available and ignoring economic considerations, the User might instead choose a higher performance camera system (e.g., more resolution, multispectral data, etc.) to effectively extend the SV graph and regain the flexibility to trade off coverage and quality during a mission.

One can contrive many similar situations. Certainly it may be practical and economical criteria which turn out to be the limiting factors in camera design. However, the point is that designing camera systems solely around PCM operation may limit the potential advantages that data compression might offer during a mission.

More Complex Sequencing

Since Super System is basically a collection of all the best existing algorithms for each data class, operating rate, etc., it is clearly the most powerful system in terms of performance.[16] However, we have ignored the significant (if not impossible) implementation problems implied by Super System. The User cannot have Super System for nothing. Thus, to develop a new algorithm

[16] For existing algorithms, that is. See footnote 11, page 27.

which emulates Super System, but is constrained in complexity, the desirability of each of the various properties of Super System should first be looked at individually to assess their relative importance. The examples discussed so far would suggest that it might be more desirable to try and be "good" everywhere rather than "best" under a few restricted conditions.

The latter point becomes even more apparent by considering some potentially more sophisticated sequencing situations. Our basic example depicted in Fig. 12 will generally involve large transmission turn-around times. To take maximum advantage of the flexibility of a source encoding system which approximates Super System's characteristics, the User must be able to rapidly make decisions. This becomes increasingly more difficult as the complexity of sequencing tradeoffs increase. Certainly, it is desirable that the User's decision-making be made as straightforward as possible. Thus, before proceeding to these more complex situations, we need to establish the rudimentary definition of a computer controlled interactive terminal which will permit the User to instantly visualize the impact of his tradeoffs.

Interactive terminal. A block diagram of the general structure for an interactive terminal is given in Fig. 16. The principal aim of the terminal is to permit rapid but complex sequencing decisions during a mission. Its more general applicability should be obvious.

The initial input to this system is the Low Resolution Observation picture and the basic constraints placed on the imaging sequence the User is considering. The User then enters commands to the terminal (in a language specially designed for this purpose) which calls up desired information from the vast collection of test results generated by the experiments in Figs. 8 and 10, and displays them in various forms. One principle visual display would be actual sequences using reconstructed compressed pictures derived

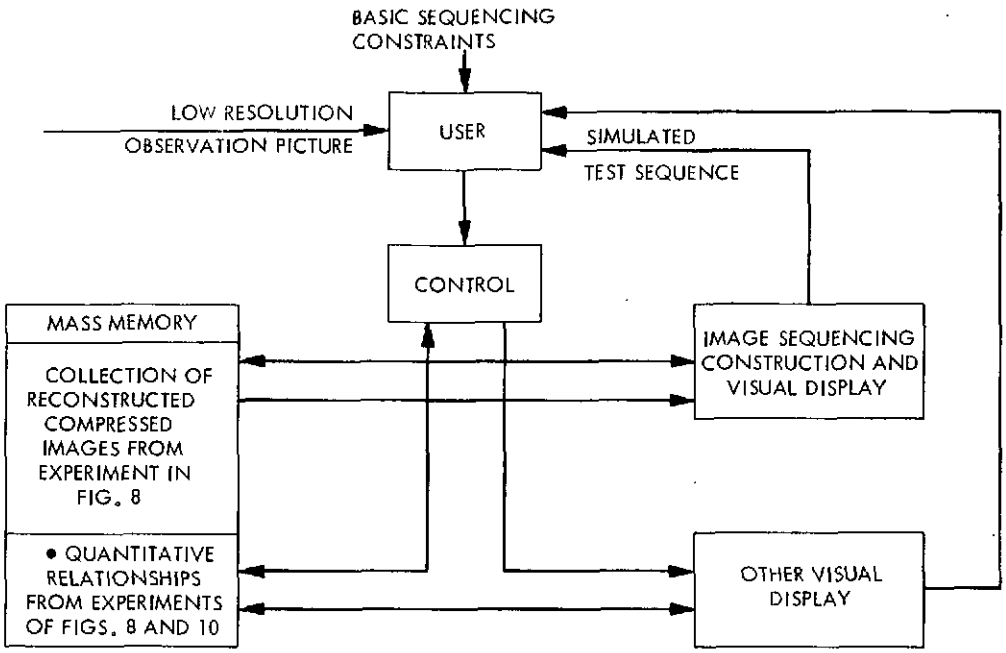


Fig. 16. Interactive Terminal for Visualizing Sequencing Tradeoffs

in the experiment of Fig. 8. Each such reconstructed test picture displayed would in essence predict the subjective quality for the corresponding picture to be obtained during the actual sequencing. In general, each such picture might correspond to a different data class, picture rate, etc. The collection of pictures patched together to form a "Test Sequence" or test mosaic would predict the subjective results the User might expect from the actual sequence. Each test sequence would be supported by other test result information such as the quantitative measures of quality, but the User would probably rely heavily on the pictorial information.[17]

[17] Recall that picture quality is a weighted collection of quality measures, both quantitative and subjective. The weighting is made by the User who is again a weighted collection of individual scientific users.

The terminal would be capable of providing a test sequence for all possible alternatives the User might select under all potential sequencing situations a mission(s) might provide. Of course, the User does not want to look at test sequences for each possible alternative, but instead wants to be able to iterate on his alternatives in a way that rapidly converges on a "good" set of parameters to use for the actual sequence.

We will further illustrate the basic concepts of this Interactive Terminal by examples. This will by no means specify all the desirable features that could be included, but should serve as an adequate introduction.

A good starting point for discussion is the example of Fig. 13. In this example, we have assumed that all potential images are from data class $\{D_k\}$ and have equal **a priori** importance. However, the terminal must know which frames it should include in constructing a simulated test sequence. In reality, the User will be influenced by the desire to have frames he does receive connected or even overlapping. Therefore, we will assume that each potential image is numbered by the User in order of priority.

The terminal must know the classification of the data. It could assist the User here by selecting and displaying test pictures D_k representing the data classes $\{D_k\}$ until a match with the low resolution observation data is made. Pattern recognition techniques could also be employed to speed the process, especially in more involved situations.

For this relatively simple case, an "original" test sequence consists of an array made up of many of the same original test pictures, D_k. The terminal would be capable of generating this original for comparisons along with the Low Resolution Observation picture on the same scale.

It should be possible to enter such information as transmission rates, picture rates, coverage, etc., in whatever form is most convenient to him, letting the computer make any necessary conversions and calculations. These are details that would evolve during an actual development of a terminal.

To tie down our example and review our points so far, assume that the sequence to be considered has 64 potential frames instead of the 90 in Fig. 12. Fig. 17 summarizes the initialization of the terminal for this example.

Suppose that the User wants to visually observe PCM operation, points A_1-A_2 in Fig. 13. The terminal would generate a test sequence such as that shown in Fig. 18 where eight PCM frames are assumed possible. The blank areas, which would not receive any high resolution pictures, might be filled in with the corresponding data from the low resolution observation picture. The eight PCM frames would all be represented by the original test picture, D_k.

If the User now wants to find points G_1-G_2 in Fig. 13, he might move along the Super System curve by successively entering lower and lower picture rates (or say, average bits per picture element) to the terminal. Each time the terminal would generate a test sequence with an increasing number of frames (following the ordering established by the User) being represented by simulated versions of D_k reconstructed from compressed data at the corresponding picture rate (these, of course, being retrieved from mass memory by the terminal). It can be assumed that the terminal will adjust picture rates to account for the discrete number of pictures. For example, if the picture rate selected by the User implied a leftover fraction of a frame, the terminal would adjust all picture rates upward until the frame number came out even. This would be straightforward with Super System (or its approximation) since the picture rates are arbitrary.

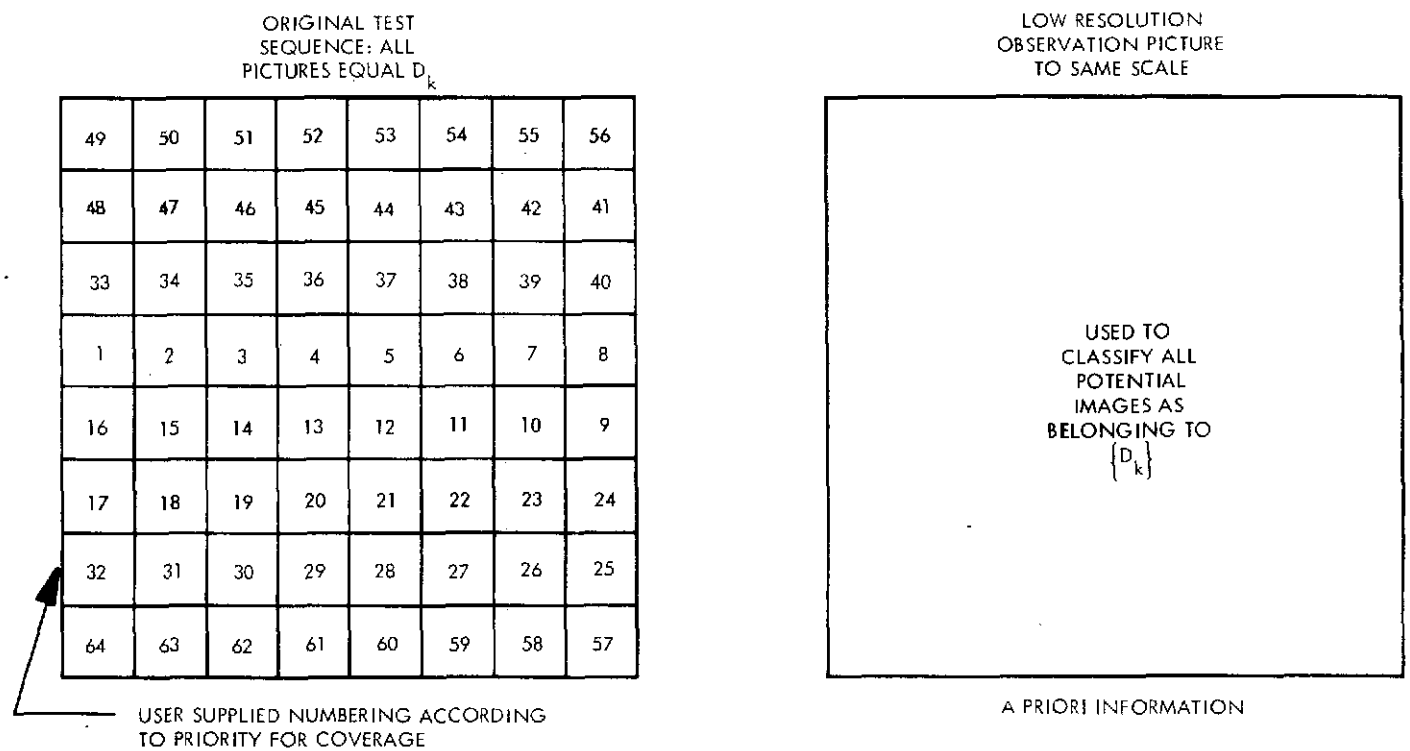


Figure 17. Initializing the Terminal for Fixed Data Class

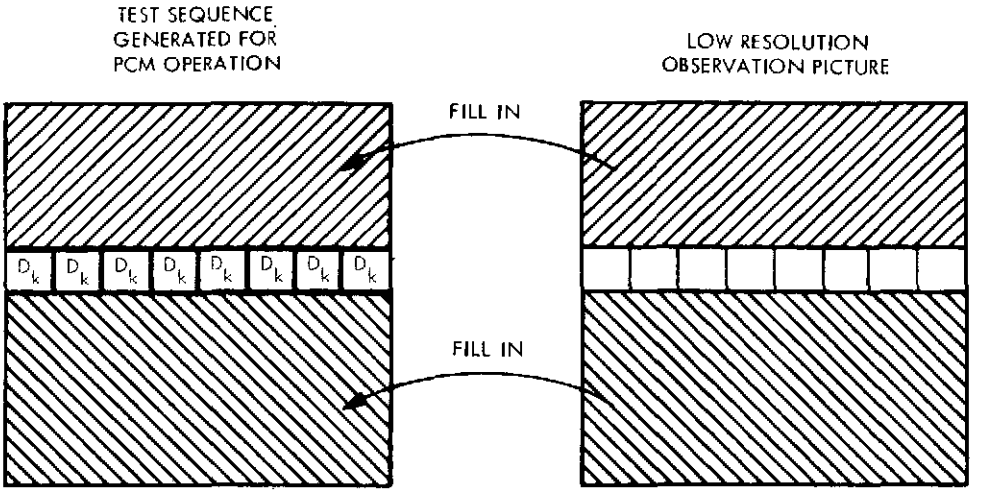


Fig. 18. Test Sequence Generated for PCM Operation

The User might accomplish the same thing by entering the desired coverage instead of picture rate. Again, the roundoff problem would be avoided using Super System (or its approximation) because of the arbitrary rate capability.

The User would continue to select new test sequences until he felt he had found a combination of coverage and quality which was most valuable to him. In practice, if the User actually had Super System, he would probably guess at G_1-G_2 first to avoid testing the complete Super System operating range.

If the User was testing algorithm A* and comparing it to PCM operation, then he need only check two points. In both cases, the roundoff problem exists.

Suppose that in some specific situation, a certain quantitative quality measure(s) was of particular interest to the User. He could enter selected values for this parameter and have the terminal perform the necessary search of its stored test data to determine what test sequence is possible.

We now turn to a more sophisticated tradeoff situation where we assume that the User is making use of Super System or an approximation to it.

Trading off data classes. Suppose that the observed area from the low resolution image in (1) of Fig. 12 was represented by many data classes

instead of just one as has been assumed thus far. Again the terminal could assist in this classification. An example is shown in Fig. 19 where data classes $\{D_{k_1}\}$, $\{D_{k_2}\}$, ..., $\{D_{k_5}\}$ are shown representing distinct regions of the surface observed in 1. We will also assume the same numbering of frames as that in Fig. 17.

Whereas before we assumed that each potential high resolution image had approximately the same importance, the addition of each data class adds a new dimension to the tradeoffs involved. In general, information derived from the distinct data classes may have different User priorities. This point is obscured when data rates are so high that all frames can be returned using PCM. We need some new notation.

Let F_k^{max} represent the number of potential high resolution frames from data class $\{D_k\}$ (e.g., in Fig. 19, $F_{k_1}^{max} = 11$, $F_{k_2}^{max} = 15$). The total number of potential frames (maximum coverage) is then

$$F^{max} = \sum_k F_k^{max} \tag{18}$$

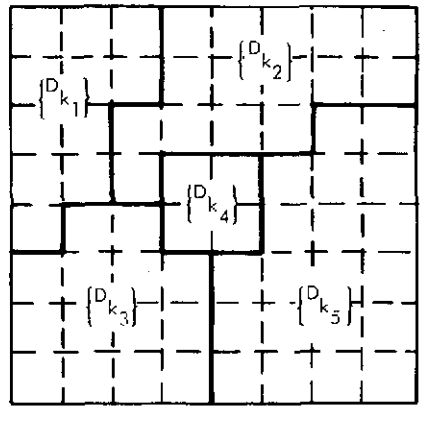


Fig. 19. Multiple Data Classes

For example, $F^{max} = 64$ in Fig. 19. The fraction of potential frames represented by data class $\{D_k\}$ is then given by

$$f_k = \frac{F_k^{max}}{F^{max}} \tag{19}$$

Let N^{max} equal the total number of bits available during the complete imaging sequence. With $\mathcal{R}_{pcm}$ equal to the rate of a PCM picture in bits/p, the PCM coverage possible is

$$C_{pcm} = \frac{N^{max}}{\mathcal{R}_{pcm}} \tag{20}$$

Thus, F^{max}/C_{pcm} would be Maximum Normalized Sequence Coverage used in earlier examples.

Let N^k denote the total number of bits assigned to data class $\{D_k\}$. Then PCM Coverage for data class $\{D_k\}$ would be given by

$$C_{pcm}^k = \frac{N^k}{\mathcal{R}_{pcm}} \tag{21}$$

As usual the problem facing the User is to determine how to distribute the total bits N^{max} in a way which he feels will yield him the most value. The terminal can generate a simulation of all possible alternatives but of course it is desirable to converge on a good choice as quickly as possible. With five data classes in this example it is not as easy to guess a good choice at the start.

Suppose the User did not take into account the relative importance of different data classes in his initial inputs to the terminal. He might then assign $N^k = (f_k)(N^{max})$ bits to data class $\{D_k\}$. The distribution of the N^k bits among the frames belonging to data class $\{D_k\}$ becomes a quality/coverage tradeoff similar to the examples we've already discussed in detail. The priority for coverage within a data class would be based on the order of numbers assigned to frames within that class (e.g. 4, 5, 12, 13 for data class $\{D_{k_4}\}$ in Fig. 17).

If there was a significant difference in the relative importance of the different data classes then it would become clear from the test sequences generated. For example, if say $\{D_{k_4}\}$ was considerably more important than the other data classes, the User would eventually instruct the terminal to put more bits into test pictures representing $\{D_{k_4}\}$ at the expense of other data classes. That is, increasing N^{k_4} to a prescribed amount while decreasing N^{k_1}, N^{k_2}, N^{k_3}, and N^{k_5} so that the sum, N^{max}, remains constant.

A better way to get to the same point is obtained by initially taking into account the relative preference between data classes. Let α_i define a User priority for data class $\{D_{k_i}\}$ where $\sum_i \alpha_i = 1$.

Then when instructing the terminal to generate a first test sequence, the User would assign

$$N^{k_i} = (\alpha_i)(f_{k_i})(\gamma) \text{ bits} \tag{22}$$

where

$$\gamma = \frac{N^{max}}{\sum_i \alpha_i f_{k_i}} \tag{23}$$

As before, $\sum_i N^{k_i} = N^{max}$, but now we have initially included the relative priorities between data classes. The User is now more likely to land

close to a satisfactory test sequence than before. This would, of course, take practice. He can proceed from the initial test sequence as before and could in principle carry the sequencing refinements as far as desired, where in general, each picture could have a different rate.

Observe that we have wholly emphasized real-time transmission problems. If the spacecraft has mass memory, it provides the User with an additional tradeoff parameter which could be incorporated into the terminal. In fact, it should be obvious that the basic structure of the terminal would allow expansion to include almost any tradeoff parameters, including costs. This could go as far as simulating operations for complete missions.

Enter pattern recognition. The sequencing discussions above elaborate on the various alternatives and tradeoffs involving source encoding the User may make prior to an imaging sequence. Once a sequence begins, the User cannot change his mind because of the large transmission turn-around times. However, the combination of an approximation to Super System and pattern recognition would permit on the spot refinements in the User's sequencing commands. For example, suppose the pattern recognition device was capable of detecting certain general features (e.g., data classes) which were of particular interest to the User. When these features were detected during the actual sequencing, more bits (higher quality) could be reassigned to those frames containing the desirable features. A reduction by the same total number of bits would be distributed amongst all other frames remaining in the sequence. The general sequencing criteria established by the User initially would be preserved, but certain especially interesting features would be enhanced. In the same manner, if other scientific experiments on board the spacecraft (usually making up at most 20% of the total transmission rate) suddenly have a particularly large burst of critical data, they may be allocated

more bits at the expense of all images (or a few images). The effect on the imaging sequences would be "epsilon".

Other problems. In all the examples and discussions, we have ignored the practical implications of a more complex spacecraft command structure and potentially difficult camera pointing problems. These need to be looked at on an individual basis (Pioneer, Mariner, specific missions, etc.). Here, we will only note that a) picture rate changes which were not commanded by the User are the most difficult to implement. Examples of this are the use of information preserving modes (data dependent output rates) and changes instituted by pattern recognition control; b) for these most difficult situations the major difficulties are significantly reduced by the existence of large buffers or mass memory (non-real-time transmission).

CHANGING THE QUALITY FUNCTION

In all the deliberations above, we have assumed a fixed User quality function. That is, the User's weighting of the various quantitative and subjective measures of picture quality have been assumed fixed for a given data class. Super System was defined in Fig. 8 under this assumption. However, there is no reason why this definition could not be extended to include variations in the User's assessment of what picture quality means to him. After all, the User is assumed to be a weighted collected of scientific users and this composition may change. In fact, this extension was implied in the discussions on the Interactive Terminal. However, it was considered an unnecessary complication and of secondary importance to include in the main discussions.

The primary impact of this extended definition is to simply add another desirable feature to the performance characteristics of Super System. Briefly, the User may wish to alter the emphasis placed on the reproduction

of certain features internal to a picture (e.g., high frequency vs. low frequency). Super System could adjust to any new preferences, provided it is told about the new preferences, by switching to another algorithm if necessary. The effect is to reallocate a fixed number of bits internal to a picture in a way which enhances features which have been given an increased priority. The arguments are in essence a small scale version of our discussions on sequencing.

INTRODUCTION TO RM2

The system concepts discussed in this chapter have been the motivating force behind current TV data compression research at JPL. The latest product of this research is in the form of a system called RM2. Although still in the research stage, preliminary results demonstrate characteristics which emulate those of Super System.

An information preserving mode is provided which is essentially the same as developed in earlier research, RM1[8], [9]. It adaptively provides near optimum performance under this kind of constraint for any expected data. The variable length coding employed here performs a similar function in a second, rate controlled mode.

The rate controlled mode permits nearly any arbitrary picture rate to be selected on a frame to frame basis. In terms of picture quality and Super System as a measure of comparison,[18] good performance has been observed at all rates on representative data from a wide range of data classes (with desired picture rate as the only input parameter). Changes in User priorities for both spatial and frequency features can be accommodated by additional inputs. Estimates of implementation complexity are quite reasonable for spacecraft applications.

[18] As estimated from the literature.[10]

Complete detailed documentation of the rate controlled mode is not available at this time. However, [12] describes a new two-dimensional transform which plays a fundamental role.

IV. CHANNEL CODING FOR DATA COMPRESSION

When uncompressed PCM imaging data is transmitted over a telecommunications channel, the effect of a single bit error is constrained to the particular pixel in which it occurs. On the other hand, the effect of a single bit error on compressed data will characteristically propagate over many pixels during the reconstruction process. This "error sensitivity" is the underlying cause of the classic data compression problem. The classic problem itself is simply that, with channels such as those discussed in Chapter II, the transmission rate must be reduced (which increases the signal to noise ratio, E_b/N_0) by significant factors in order to "clean up the channel" and reduce the number of error events to a tolerable level. As a consequence, much of the potential gains offered by data compression in the form of pictures/sec, coverage, etc. (see Chapter III) may be lost.

In this chapter we will first discuss this classic problem in more detail, restricting attention to the best of the basic Mariner channels, the Jupiter/Saturn Channel in Fig. 5. We then will describe a straightforward and practical way to supplement the considerable investment in the Jupiter/Saturn Channel such that the classic data compression problem disappears. Since we are primarily interested in first order effects, the reader may assume ideal receiver operating conditions unless noted otherwise.

THE JUPITER/SATURN CHANNEL AND COMPRESSED IMAGING DATA

A Review for Uncompressed PCM

The general composite effect of an increasing frequency of bit errors on uncompressed PCM imaging data is a gradual loss in information content. This point can't be ignored no matter how much filtering is done to make the reproduced pictures "look nice". As noted in Chapter II, a rule of thumb has developed for bounding the range of subjective quality resulting from

transmitting PCM images over the basic Mariner channels. For average bit error rates below 5×10^{-3}, reconstructed quality is considered good to excellent. At the other end of the scale, quality is considered poor to unusable with $\overline{P}_b > 1/20$. No doubt one could construct elaborate experiments (such as those in Chapter III) to investigate the subjective quality lying between these extremes. For the Jupiter/Saturn Channel such elaborate experiments are of questionable value. We note from Fig. 5 that in going from excellent quality ($\overline{P}_b = 5 \times 10^{-3}$) to poor quality ($\overline{P}_b = 1/20$) the required E_b/N_0 changes from 2.6 db to 1.6 db. Using the decibel conversion in Appendix A, we see that this amounts to a transmission rate increase of only 25 percent. In addition, this apparent rate/quality tradeoff is one which is controllable by the User only in a very limited sense. In practice he can only select rates in discrete steps which up to now have been much larger than the complete 25 percent. In reality, the User's only tradeoff consideration during a mission is how bad should he allow the data to get before he reduces the rate (by a discrete step) when off nominal fluctuations in receiver signal to noise ratio reduces E_b/N_0 below 2.6 db. One could argue the preciseness of these statements, but would be making a mountain out of a mole hill. The major points should be clear: a) the degradation in quality internal to a picture caused by random errors is a phenomenon which is not controllable by the User, b) the potential improvements in transmission rate in going from excellent to poor quality is on the order of only 25 percent using the Jupiter/Saturn Channel, c) this potential rate/quality tradeoff is primarily controlled by fluctuations in the communication link not by direct User intervention.

Pixel editing. Pixel editing is more closely related to uncompressed PCM than to what is usually considered data compression and thus we will

mention it here. Basically an N to 1 pixel edit scheme transmits only 1/Nth of the original PCM samples.[19] Reconstruction of the missing pixels is accomplished by linear interpolation. Under noiseless conditions reproduced pictures look like the original PCM pictures with reduced sampling rates in two dimensions. Thus degradation in picture quality due to editing alone is essentially a resolution loss.

When an error occurs its effect is no longer limited to a single pixel. Instead the error effect is spread over all those missing pixels which are reconstructed (by linear interpolation) using the "bad" pixel. The extent of this error propagation is quite limited compared to more sophisticated algorithms (e.g. only four pixels for N=2). In addition, by definition of the interpolation process, the effect of an error diminishes as the distance between an interpolated pixel and an error increases. Further, an error in an interpolated pixel does not represent the same level of information loss as an error in a transmitted pixel. This is because an interpolated pixel is really only a best guess. Thus the degradation caused by individual errors on pixel edited data is quite similar to PCM, increasing with N but not dramatically.

Recall that for PCM data $\overline{P}_b = 5 \times 10^{-3}$ on the Jupiter/Saturn Channel is the approximate error rate below which the effect of channel errors is considered negligible. From the discussions above, it is not surprising that the corresponding operating points for edited data are not significantly different. At worst a $\overline{P}_b \approx 5 \times 10^{-4}$ is necessary for negligible error degradation on 8 to 1 edited data. This difference in operating points on the

[19] Such algorithms were flown on the 1974 Mariner flybys of Venus and Mercury and have been proposed for the Jupiter/Saturn Mariner missions. Variations on this basic theme using averages produces very similar results.

Jupiter/Saturn performance curves (see Fig. 5) amounts to a transmission rate difference of about 22 percent. This figure is correspondingly less for values of N less than eight. This 22 percent reduction in transmission rate necessary to achieve negligible error degradation when using 8 to 1 editing instead of PCM is rather insignificant compared to the 800 percent compression factor. This is the reason for statements such as "there is little interaction between pixel editing and the Jupiter/Saturn Channel (relative to PCM, that is)." This factor, along with its simplicity, are the prime virtues of editing.

As for PCM, each editing scheme (used on the Jupiter/Saturn Channel) has a narrow range of signal to noise ratios over which degradation due to errors goes from negligible to intolerable. Thus operationally the use of editing or PCM on this channel is nearly identical. One might argue with the precision of these statements and formulate extensive experiments to better define these characteristics. This might be justified in a limited sense if the User's options were only PCM, editing and the Jupiter/Saturn Channel. However, in light of the results of this chapter, they would not be useful.

Transmission of Compressed Data[20]

There are other algorithms besides pixel editing which are not much more sensitive to random transmission errors than uncompressed PCM

[20] To help avoid confusion to the uninitiated reader, transmission rate refers to the rate in bits/sec at which individual information bits, compressed or not, are transferred over the channel (see Chapter II). On the other hand, rates for compressed data are often discussed in terms of average bits/pixel, bits/source block, bits/picture. These terms avoid the element of time which is convenient when you are working on the data compression problem by itself (see Chapter III). If desired, rates such as source blocks/sec, pictures/hr, etc. could be obtained by combining terms.

(e.g. certain transform techniques). However, this is the exception rather than the rule. Generally compressed data exhibits significant sensitivity to transmission errors relative to that experienced by uncompressed PCM. The well known consequence of this sensitivity when using channels such as the Jupiter/Saturn Channel is that the transmission rate must be reduced by significant factors in order to reduce the number of error events to a tolerable level. Each individual algorithm will, of course, exhibit its own particular form of quality degradation when used on the Jupiter/Saturn Channel. This fact is often met with proposals for exhaustive simulations. This approach is unfortunately looking in the wrong direction for a solution.

In subsequent sections we will demonstrate a solution to the problem for virtually <u>any</u> compression algorithm. Assuming the worst possible sensitivity to individual bit errors, it is shown that, <u>at virtually all transmission rates</u> for which uncompressed PCM can be transmitted over the Jupiter/Saturn Channel with negligible degradation due to errors ($E_b/N_0 \geq 2.6$ db), compressed data can also be transmitted with negligible added degradation <u>due to errors</u>. In preparation for these results we need only deal in very general terms.

<u>Source blocks</u>. Henceforth we will assume that data compression algorithms take on the source block structure described in the early pages of Chapter III. We will continue with the <u>assumption</u> that each source block is independent of other source blocks during the reconstruction process (later we will back off on this). Further, if we <u>assume</u> fairly large source blocks, then correspondingly large sync words placed at the start of compressed source blocks will have a negligible effect on overall data rate (e.g., a 32 bit sync sequence will alter the rate of a 4096 pixel source block by less than 0.01 bits/pixel). Just about any existing practical algorithm

can be placed in this form by combining smaller source blocks into a single large one.

We will assume the worst possible sensitivity to errors: if a single bit error occurs anywhere within a compressed source block, including the sync word, then that block is assumed completely lost. This is obviously overdoing it in most cases, but if we can handle this situation then we can certainly handle all cases in which the effect of individual errors is not really so devastating.

Conversely, if a compressed source block and its sync word are error free, then that source block can be decoded. Distortion is due only to the source encoding algorithm. These statements make use of the assumption that the decoding of any compressed source block does not depend on information from other source blocks and that the correct location of the start of any (error free) compressed block can always be determined with very high probability. The long sync word assures the latter.[21]

With this background we can take another heuristic look at the difficulties with transmitting compressed data over the Jupiter/Saturn Channel. Figure 20 illustrates the effect of randomly occurring errors on compressed imaging data under the worst case assumptions given above. The large square on the left represents a PCM picture whereas the smaller squares represent two dimensional source blocks (e.g. the source blocks might be 64 by 64 pixel arrays and the complete picture, 512 by 512 pixels). A small square with an "X" means that the corresponding compressed representation of that source block has an error somewhere in it. By definition, regardless

[21] The subject of synchronization is discussed in Appendix B. However, it is highly recommended that this subject be deferred until completion of Chapter IV.

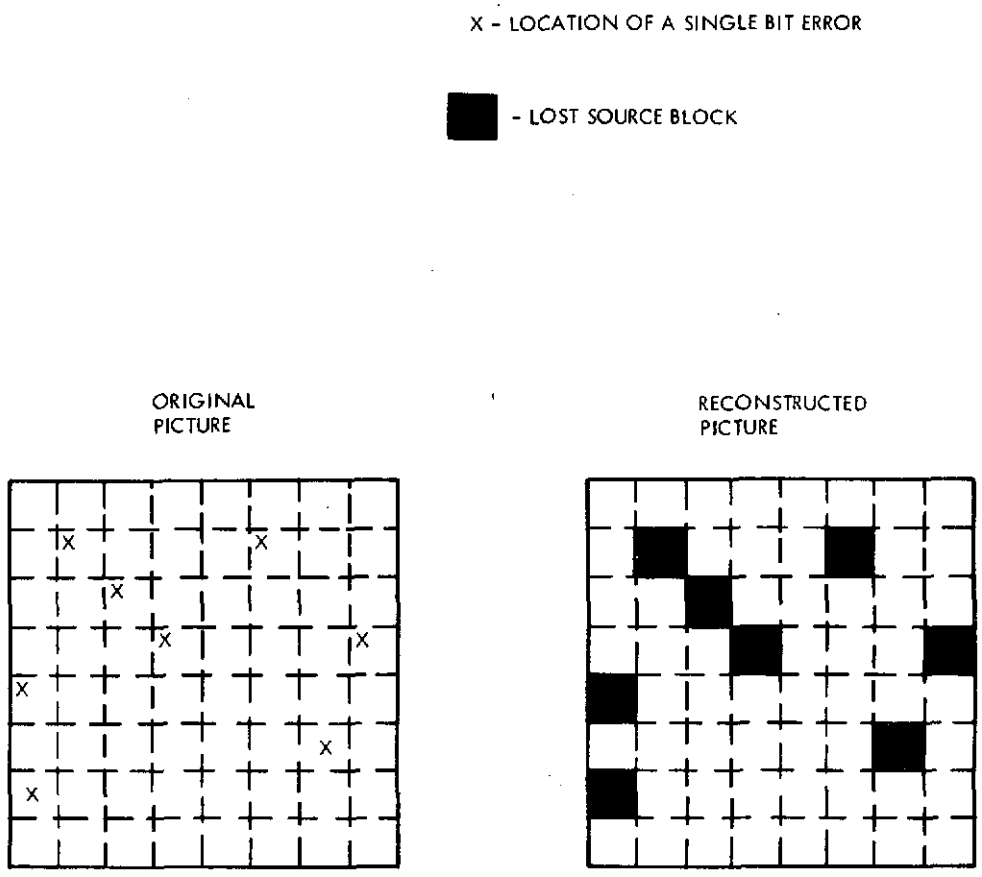


Fig. 20. Source Block Losses Due to Random Errors

of where an error occurs within a compressed source block, the block is assumed to be completely lost. This is indicated in the large square to the right where a darkened array denotes a lost source block. The quality of reconstructed data for all other source blocks is determined solely by the particular data compression algorithm.

A key point in this example is that because the location of bit errors was generally distributed uniformly throughout the compressed data, each error appeared in a different compressed source block. Consequently, each error caused the loss of a source block. At average bit error rates, $\overline{P}_b$, low enough to even talk about using the Jupiter/Saturn Channel to transmit compressed data under these worst case assumptions, errors will tend

to occur in approximately this random uniform fashion.[22] By low enough we mean values of $\overline{P}_b$ between 10^{-6} and 10^{-7}. Referring to the Jupiter/Saturn performance curves in Fig. 5, we see that the increase in signal to noise ratio (beyond that required for uncompressed PCM, $E_b/N_0 = 2.6$ db) necessary to achieve these low error probabilities is about 3 db. Or using Appendix A, this means that to obtain an acceptably low frequency of lost source blocks the transmission rate must be reduced by a factor of about two. Thus a net gain cannot be obtained from the data compression and Jupiter/Saturn Channel unless the average compression factor exceeds approximately two.

However precise the factor of two quoted above, be it really 1.7 or 2.3, is not important. The main point is that it is significant. For most algorithms which are less sensitive to individual errors than we have assumed above, the required transmission rate reduction factor would be less, but still significant. We will not attempt to assign numbers here. Instead, in the following sections, we will provide a practical means of reducing this factor to approximately zero for all algorithms.

Before proceeding, note that two main properties of the Jupiter/Saturn Channel are responsible for the dilemma. The first and most obvious is that the performance curves (Fig. 5) just aren't steep enough. That is, to lower $\overline{P}_b$ far enough requires large increases in E_b/N_0. The second and more subtle property is the generally random distribution of individual bit errors at low values of $\overline{P}_b$. As an aid to the reader's intuition here, consider Fig. 21 which is identical to Fig. 20 except in one respect. The eight

[22] The lack of precision in this statement is not crucial. It is well known that the severe burstiness experienced by the Viterbi decoding algorithm at high bit error rates ($\overline{P}_b > 5 \times 10^{-3}$) greatly diminishes at low values of $\overline{P}_b$. We'll see from Fig. 21 that the assumption of uniformity, at worst, simply bounds the performance of the Jupiter/Saturn Channel.

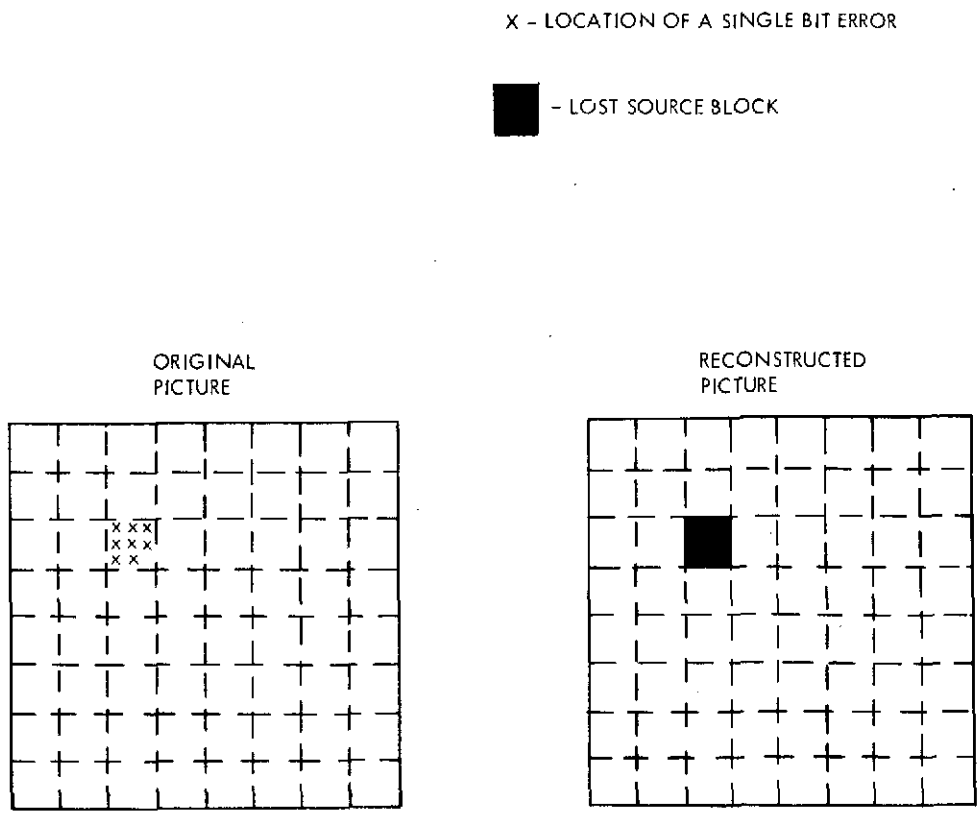


Fig. 21. Source Block Losses Due to Error Burst

bit errors which were shown uniformly distributed among the compressed source blocks in Fig. 20 are shown as all occurring in the same (compressed) source block in Fig. 21. As a result only one source block is lost instead of eight. The first error in a compressed source block causes all the damage and any others are of no consequence. Thus in general, for a given average bit error probability, it is desirable that errors occur in bursts.

THE ODENWALDER CHANNEL

The proposed solution to the problem we have posed is provided by the insertion of a Reed/Solomon block code into the communication system as indicated in Fig. 22. A key to the simplicity of this configuration is that the Reed/Solomon decoder need not involve the DSN stations (see Fig. 6).

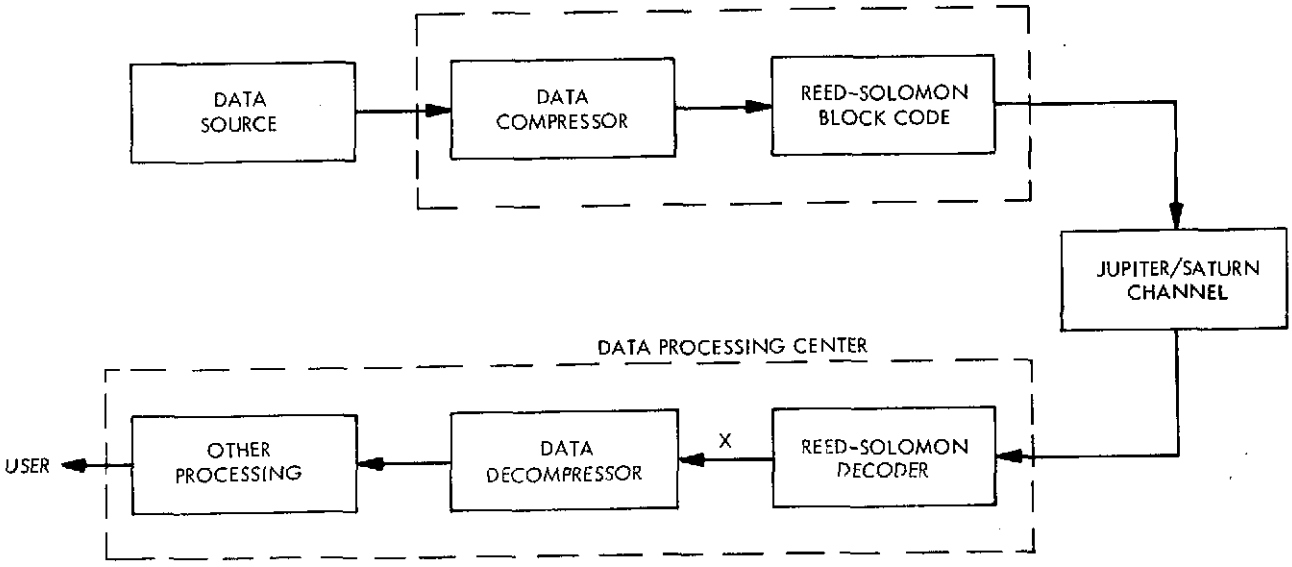


Fig. 22 Inserting the Reed/Solomon Block Code

Thus physically, the Reed/Solomon coding might best be considered part of the source encoding operations as implied in Fig. 22. This line of thought also coincides with our goal to provide a solution to the problem of transmitting compressed data over the Jupiter/Saturn Channel. However, to demonstrate that this is indeed a solution, our purposes are much better served by following the historical approach in which the Reed/Solomon is part of a concatenated channel coding system.

Combining Reed/Solomon block coding with Viterbi decoded convolutional codes was first investigated by Joseph Odenwalder in his Ph.D. dissertation.[13] Subsequently, this work was extended in a study for Ames Research Center by Odenwalder and other members of Linkabit Corporation.[1] We will rely very heavily on the latter results.

The Linkabit study investigated many Reed/Solomon codes coupled with a number of Viterbi decoded convolutional codes in which constraint length, K, and code rate, $1/\nu$, were the main parameters. We will almost immediately zero in on one particular combination. The choice for the convolutional code is obviously directed by the anticipated future existence of the Jupiter/Saturn Channel. The primary choice of Reed/Solomon coding parameters is directed by both performance and the implementation and speed requirements of the decoder. Our approach will be to treat the system impact of this particular concatenated channel coding system in detail. Later we'll return to the question of coding parameters and find that perturbations in these parameters are of secondary importance and have no impact on the overall results.

Reed/Solomon Coding

We emphasize again that our interest are at an overall system level and consequently we need not get involved with the intracacies of coding and decoding algorithms. These details are extensively treated in the references. Of course the primary reference is the Linkabit study.[1] However, perhaps of more fundamental interest to the reader uninitiated in algebraic coding is Chapter VI of Gallager.[5] This well written chapter actually provides all the background necessary for the reader to design his own Reed/Solomon coder and decoder. Gallager was in fact followed closely in the Linkabit study. The reader would find that a Reed/Solomon code is really a BCH code with a specific set of parameters. Gallager's general discussion of BCH decoding procedures needs improvement in only one area, E. R. Berlekamp's Iterative Algorithm. A more clearly written discussion of that concept is given by Massey.[14]

Proceeding with our own less detailed development, we first wish to establish the basic code parameters. A very simple block diagram of a Reed/Solomon (RS) block coder is shown in Fig. 23.

The first thing to notice is that the RS code is non-binary. An RS symbol consists of a sequence of J bits so that there are 2^J possible RS symbols. All coding and decoding operations involve RS symbols, *not* individual bits. Fortunately our interests here lie in the results of these operations, not in their details. In this area, the Gallager reference is excellent.

Returning to Fig. 23, $2^J - (1+2E)$ information symbols (or $J\left[2^J - (1+2E)\right]$ information bits) from some data source enter the RS Coder to the left. The result of coding operations is a <u>codeword</u> of length $2^J - 1$ symbols of which the first $2^J - (1+2E)$ are the same symbols as those entering to the left. This makes the code <u>systematic</u>. The remainder of the codeword is filled in with <u>2E parity symbols</u>.

An RS symbol is in error if any of the J bits making up the symbol are in error. E represents the number of correctable RS symbol errors in an RS codeword. That is, if E or less RS symbols are in error in any way, the decoder will be capable of correcting them. Actually some

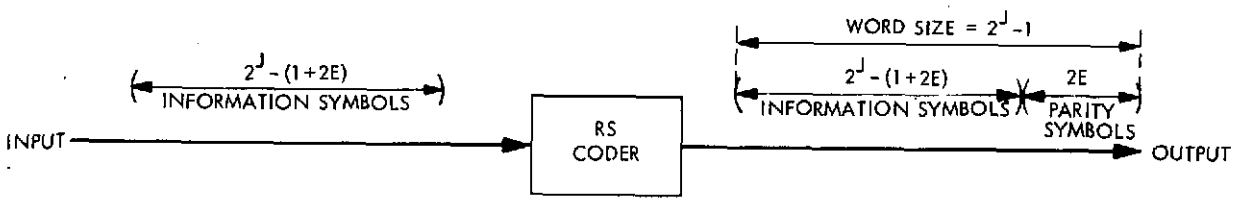


Fig. 23. Basic RS Structure

additional errors could be corrected, but this capability is difficult to provide in the decoder and as we'll see, is certainly not worth the effort.

Linkabit performed simulations for various values of the parameters J and E.[23] Eventually, they focused attention on the specific system with J=8, E=16. We will do the same since this system is well suited to our goals for basically the same reasons. The tradeoffs involved will make more sense after first looking at the impact of this one system in detail.

Low overhead. The basic codeword structure for this specific code with J=8, E=16 is given in Fig. 24. The diagram is self-explanatory. Note that the overhead associated with the parity symbols is only around 15 percent. From an onboard storage point of view (mass memory applications), it requires only 15 percent more memory to store data protected by RS coding than without. More significantly, the low overhead means that ground communications are not severely affected by transmitting RS coded data. Consequently, an RS decoder need only be placed at a single destination, not at each DSN station (see Figs. 6 and 22). If desired a "quick look" at the data (information bits) would still be possible since the code is systematic. The low overhead also influences, in a positive way, the implementation of both coder and decoder. The reader is referred to the references for details. The impact of these observations will not be diminished by the inclusion of interleaving and synchronization.

---

[23] For those readers already versed in algebraic coding, the generator polynomials for all codes investigated were taken as

$$g(D) = \prod_{i=1}^{2E} (D - \lambda^i)$$

where λ is a primitive element of $GF(2^J)$. For all practical purposes this leaves J and E as the sole parameters defining each code.

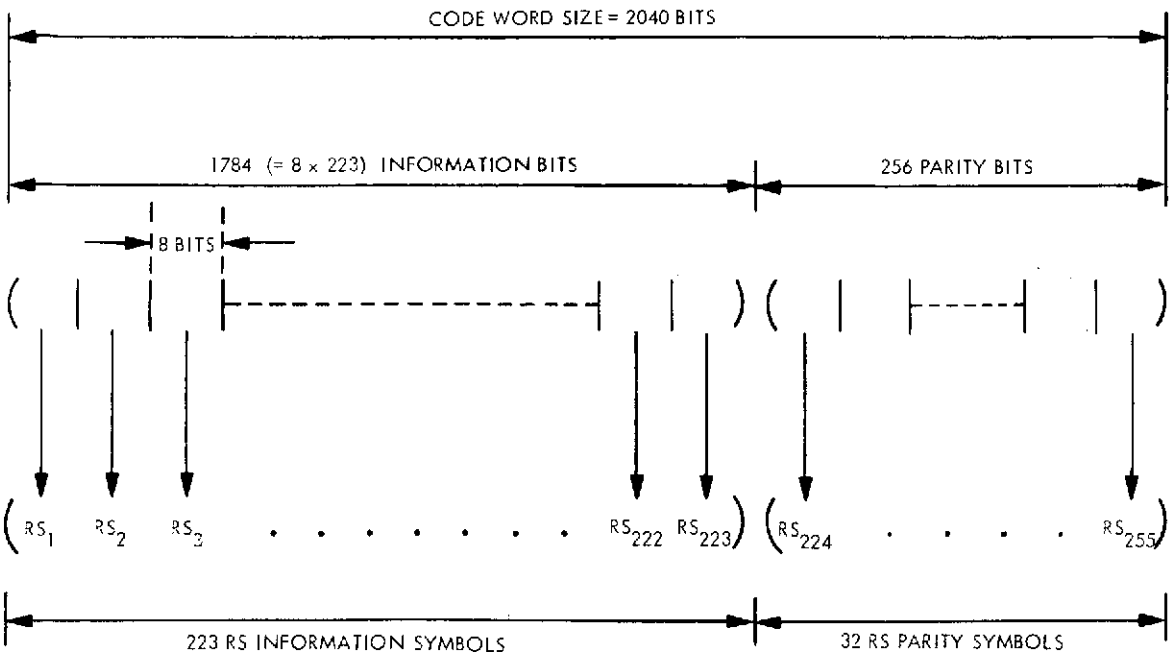


Fig. 24. Basic RS Codeword Structure, J=8, E=16

Interleaving

To make the most effective use of the power of RS coding when concatenated with Viterbi decoded convolutional codes requires interleaving. This is because of the extreme burstiness in error events experienced by Viterbi decoders at values of E_b/N_0 of interest (between 2.0 and 2.5 db).[24] Without interleaving Viterbi decoder burst error events would tend to occur within one RS codeword. That one codeword would have to correct all of these errors. Thus over a period of time there would be a tendency for some codewords to have "too many" errors to correct (i.e. greater than 16)

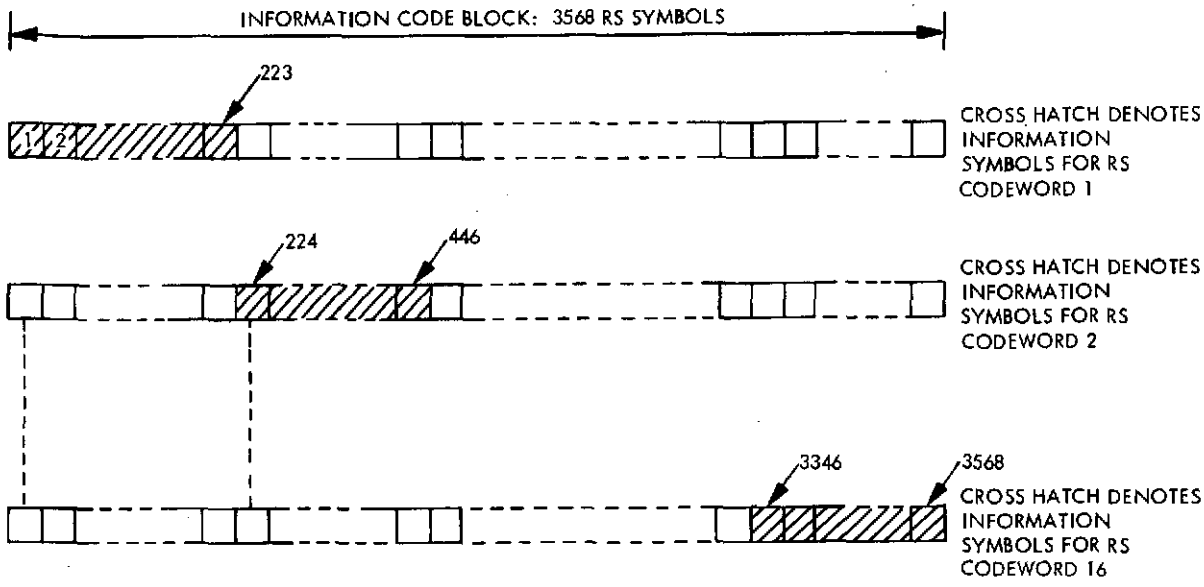
[24] From Figs. 6 and 22 we see that, except for the very unlikely errors caused onboard or during ground communications, errors seen by the RS decoder are characterized by the Viterbi decoder. Note that because of the 15% overhead for parity symbols the Viterbi decoder must operate at an effective E_b/N_0 which is approximately .6 db below that of the overall concatenated system.

while the remaining would have "too few" (i.e. much less than 16). This situation does not make effective use of the capabilities of the RS coding. The effect of interleaving is to spread these bursty error events over many codewords so that the RS decoder tends to work uniformly hard on all the data.

Two methods of interleaving will be investigated here. We will call them Interleave A and Interleave B. The first exhibits a slight performance advantage in the transmission of compressed data whereas the second offers an advantage in memory requirements for the onboard RS coder. In both cases we will assume Linkabit's choice of interleaver depth, I=16.

Interleave A. A diagram illustrating Interleave A is shown in Fig. 25. The consecutive numbers 1, 2, ..., 3568 denote labeling of consecutive information symbols which are to be interleaved and coded into 16 RS codewords. This is just the compressed or uncompressed data (grouped into 8 bit symbols) as it would enter the RS Coder. We call this sequence of bits an Information Code Block to distinguish it from a Code Block which also includes parity symbols. The length of an Information Code Block is (16) (223) = 3,568 RS symbols or (8) (3568) = 28,544 bits.

The crosshatched regions specify which information symbols belong to each of the 16 codewords. As specified, the first 223 form the information symbols of codeword 1, the second 223 information symbols belong to codeword 2, and so on. Without interleaving these symbols, along with their 32 parity symbols, would be transmitted over the Jupiter/Saturn Channel in the order in which they appear. Thus a particularly long burst of errors from the Viterbi decoder would tend to affect the symbols of only one codeword. With Interleave A the order of RS information symbol transmission is (1, 224, ..., 3346), (2, 225, ..., 3347), ..., (223, 446, ..., 3568).



* ORDER OF RS INFORMATION SYMBOL TRANSMISSION OVER JUPITER/SATURN (VITERBI) CHANNEL IS (1, 224, ..., 3346), (2, 225, ..., 3347), ... (223, 446, ..., 3568).

** SIMILARLY FOR PARITY SYMBOLS.

Fig. 25. Interleave A, Structure

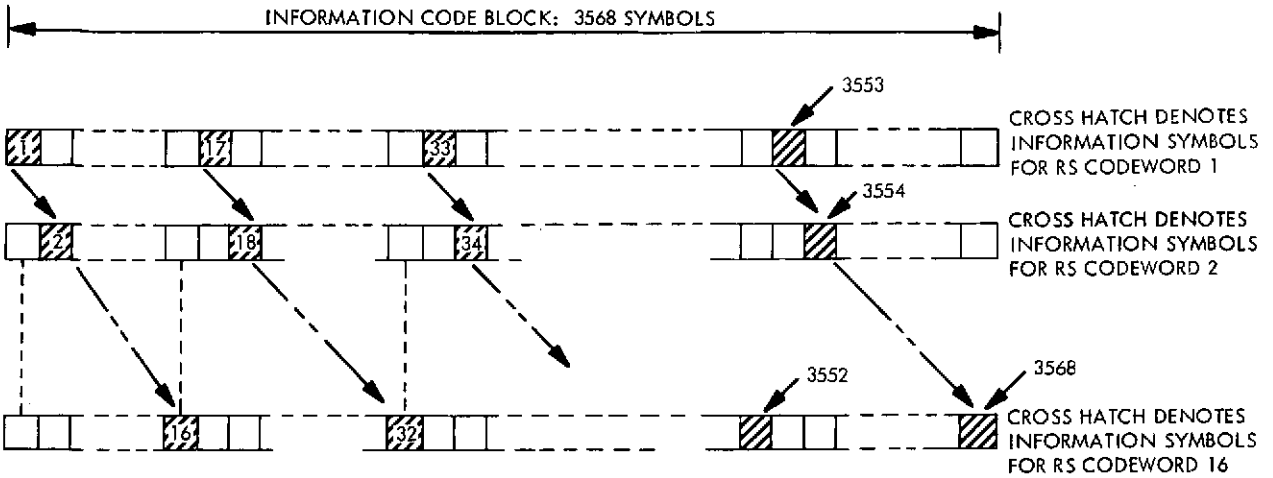
That is, the first symbol from codeword 1, the first symbol from codeword 2, ..., the first symbol from codeword 16, the second symbol from codeword 1, and so on. The parity symbols would follow in the same manner. With this arrangement it should be clear that a burst of errors that spans $k \leq 16$ RS symbols (128 bits) will be distributed among k different codewords.

Since the information symbol 3346 is the 16th symbol to be transmitted, memory for the complete Information Code Block must be provided in addition to that required for parity symbol generation. This much working memory today is really insignificant. For example, Advanced Pioneer mission planners are presently assuming at least 10^6 bits of working memory. Single solid state chips are available off the shelf with 4096 bits

of random access memory. However, we point out that the second interleave method, Interleave B, does offer an advantage in this area by requiring memory only for the parity symbols.

If 16 or less RS symbols of a codeword are in error before entering the decoder, then all information symbols of that codeword leaving the decoder will be correct. No decoding error is made. On the other hand, if more than 16 RS symbols of a particular codeword are in error before decoding, then a decoding error will occur and the output information symbols may have many errors. If we interpret Fig. 25 as describing an output Information Code Block we see that the effect of a decoding error on a particular codeword is constrained to the corresponding crosshatched region for that codeword. Thus for Interleave A the effect of an RS decoding error is confined to consecutive symbols. An RS decoding error will appear as a burst of errors of up to 223 symbols in length (1784 bits). Earlier we pointed out that this bursty property is desirable for the transmission of compressed data. We will see that it is the relatively greater burstiness of Interleave A over Interleave B that gives Interleave A a slight performance advantage.

Interleave B. Before investigating the specific effects of RS codeword errors on compressed data, we need to establish the basic structure of Interleave B. This is shown in Fig. 26. Again the consecutive numbers 1, 2, ..., 3568 denote the labeling of consecutive information symbols as they would enter the coder. Also as in Fig. 25, the crosshatched regions specify which information symbols belong to each of the 16 codewords. Note that for each codeword, adjacent symbols are separated by 15 other symbols in the Information Code Block. For example, the information symbols for codeword 1 are made up of Information Code Block Symbols 1, 17, 33,



* DIAGONAL ARROWS INDICATE THE ORDER OF RS SYMBOL TRANSMISSION OVER THE JUPITER/SATURN (VITERBI) CHANNEL.
** SIMILARLY FOR PARITY SYMBOLS.

Fig. 26. Interleave B, Structure

..., 3553. As indicated by the arrows, the order of transmission of RS information symbols (over the Jupiter/Saturn Channel) is in exactly the same way they appear in the Information Code Block 1, 2, ..., 16, 17, ..., 3568. Parity symbols would follow in the same manner. It is easy to see that this accomplishes the desired interleaving (e.g. a burst error event from a Viterbi decoder would have to span symbols 2 through 16 in order to affect adjacent symbols 1 and 17, of codeword 1). In addition this ordering means that no memory is required for the Complete Information Code Block since this data can be transmitted, unchanged, as it arrives. Thus significantly less memory is required for this form of interleaving.

Just as we did in Fig. 25 we can interpret Fig. 26 as describing an output Information Code Block so that, as before, the effect of a decoding error on a particular codeword is specified by the crosshatched regions for that codeword. Unlike Interleave A, we note that these crosshatched

regions are spread throughout the Information Code Block rather than constrained to a consecutive string of 223 symbols. In a moment we will see the consequence of this.

Error probabilities. The choice of interleaver I=16 was selected to achieve statistical independence between RS symbols of individual codewords "before decoding". That an interleaver depth of 16 is sufficient to make any dependencies negligible for our specific concatenated coding system is highly plausible. Error bursts from a Viterbi decoder exceeding 120 bits (15 RS symbols) are extremely unlikely for the K=7, ν=2 code for E_b/N_0 values as low as 1.4 db ($< 10^{-5}$). It was primarily such observations which led Linkabit to choose I=16 (along with the fact that 16 is a power of 2). This choice would seem to even be overdoing it for the specific code of the Jupiter/Saturn Channel, particularly under nominal phase coherent receiver operating conditions (for which our interests will be restricted to Viterbi decoder E_b/N_0 values greater than about 2 db). Perhaps the major point to keep in mind is that even doubling interleaver depth to 32 does not severely impact the implementation of either coder or decoder. It is not a big issue.

We will continue, as Linkabit did, with the assumption that enough interleaving is provided to make the assumption of independent RS symbol error events a valid one. An interleaver depth of no more than I=16 should be completely adequate in this sense. From a more practical point of view I=16 may not be necessary.

Then, with $\bar{\pi}$ equal to the average probability of an RS symbol error leaving the Viterbi decoder (groups of 8 bits), the probability of an RS codeword error (using Interleave A or B) is given by

$$P_{RS} = \Pr\left[\begin{array}{l}\text{more than 16}\\ \text{independent symbol}\\ \text{errors}\end{array}\right]$$

$$= \sum_{k=17}^{255} \binom{255}{k} \bar{\pi}^{k} (1-\bar{\pi})^{255-k} \tag{24}$$

Thus P_{RS} is determined entirely by $\bar{\pi}$. Linkabit determined $\bar{\pi}$ first by directly monitoring the correctness or incorrectness of RS symbols emanating from simulated Viterbi decoders at various signal to noise ratios. In a less direct method, they used Viterbi burst error statistics to obtain the same results. A performance curve (P_{RS} vs E_b/N_0) which we will present later was derived from Eq. 24 and the experiments which produced the various values of $\bar{\pi}$. These results do not, therefore, represent a complete simulation of the concatenated system as a single unit. However, the precision of these results (using Eq. 24) under nominal phase coherent receiver operating conditions rests only on how good the assumption of independent RS symbol errors is. We have indicated that this is a very good one. Much can also be said about some second order effects such as imperfect carrier phase tracking, and we will do so later. Motivated much by the considerations of this chapter, more complete simulations were recently initiated at Linkabit in a second study.

RS Code Block synchronization. In Linkabit's study an I=16 symbol (128 bit) synchronization sequence was assumed to separate each RS Code Block. This configuration is potentially unacceptable from several viewpoints. We discuss this topic in Appendix B and suggest some "not necessarily optimum" alternatives which imply that RS Code Block synchronization

is not a problem although further study supported by simulations is desirable.[25] Certainly we can afford to be sloppy in meeting desired performance requirements. Because the RS Code Blocks are so long, the additional overhead of even two 128 bit sequences is less than "1" percent.

Burst error correcting capability. The reader may check that the Reed Solomon decoder is capable of correcting any single error burst in an RS Code Block (32,640 bits) of up to 2,041 bits in length ($\approx E \cdot I \cdot J$).

Effect of a Code Word Error

Here we restrict attention to source blocks originating from 4096 pixels (e.g. 64 by 64 pixel arrays). This choice is desirable as our results will show, but is not crucial. In the early portions of Chapter III, we defined $\mathcal{R}_c^B$ as the rate of a compressed source block. Here we are interested in the units, bits/sb. That is, the sequence of bits representing a compressed source block is $\mathcal{R}_c^B$ bits long (including a large sync sequence). $\mathcal{R}_c^B$ is related in the usual way to the corresponding rate for uncompressed PCM, $\mathcal{R}_{pcm}^B$, through compression factor, CF^B, in Eq. 9. If we divide $\mathcal{R}_c^B$ or $\mathcal{R}_{pcm}^B$ by 4096 we obtain an average rate in bits/pixel. This is probably a more familiar representation although bits/sb is more directly related to our pursuits here.

Figure 27 illustrates the effect of an individual RS codeword error on sequences of compressed source blocks when Interleave A is employed. At the top of the figure is shown an output Information Code Block in much the same manner as in Fig. 25. The subsequences of decoded information bits for each of the 16 codewords are indicated by the parentheses and are labeled from 1 to 16. Each subsequence is 1784 bits long for a total of 28,544 bits.

[25] It is suggested that the reader defer reading Appendix B until Chapter IV has been completed.

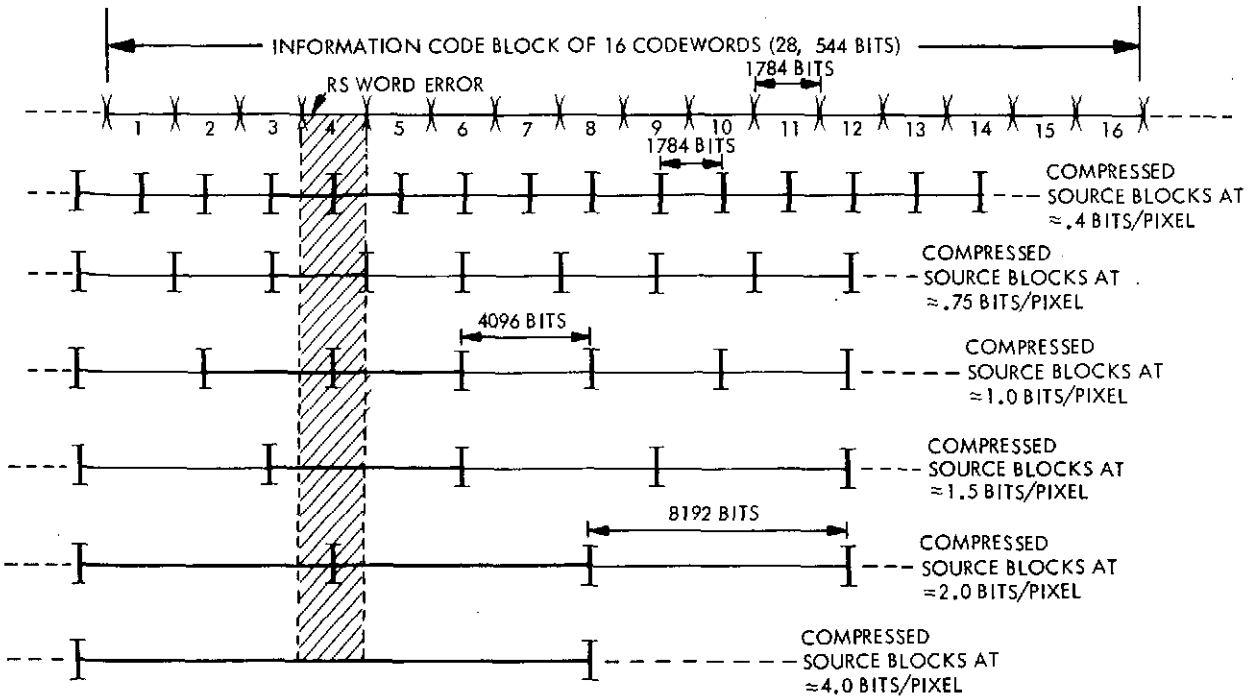


Fig. 27. Effect of RS Word Error, Interleave A

At this point the output Information Code Block represents compressed data still to be "decompressed" (point X in Fig. 22). The number of compressed source blocks making up the 28,544 bits depends on the distribution of compressed source block rates, $\mathcal{R}_C^B$. That is, how many bits it takes to represent each compressed source block. We will look at the simpler case in which each compressed source block in the sequence is represented by a fixed number of bits.

Shown immediately below the Information Code Block in Fig. 27 is a sequence of compressed source blocks which each require 1784 bits. This is equivalent to about 0.4 bits/pixel assuming 4096 pixel source blocks. Each compressed source block is indicated by brackets. Note that the start of the first RS Codeword is not (necessarily) synchronous with the start of a compressed source block. Thus the Information Code Block contains data from 17 compressed source blocks.

Below this example are shown several similar illustrations for increasing compressed source block rates (lower compression ratios) starting with average rates of 0.75 bits/pixel and increasing up to 4.0 bits/pixel. Note that because of the increasing number of bits to represent a compressed source block the Information Code Block represents fewer and fewer source blocks. At 4 bits/pixel a compressed source block is over 16,000 bits long so that an RS Information Code Block only "overlaps" 2 or 3 compressed source blocks.

To investigate the effect of an RS codeword error, we restate some earlier results and assumptions. First we assume that if any error occurs in a compressed source block, that complete source block is lost but no more. We add to this by assuming that if an RS codeword is in error after decoding, all decoded information bits are in error for that codeword. Finally we recall from Fig. 25 that when Interleave A is used, the effect of a codeword error is constrained to a consecutive sequence of information bits (symbols). In Fig. 27 these potential error sequences are those enclosed by parentheses and labeled 1 to 16. In that diagram we have assumed that codeword 4 was in error. By our assumptions above, any compressed source block which is represented by this sequence of wrong bits is lost. In Fig. 27 this corresponds to any compressed source block which falls in the crosshatched region. In all cases we observe the following: using Interleave A, the number of source blocks lost due to an RS codeword error is 1 or 2.

To obtain similar results for Interleave B, we recall from Fig. 26 that when a single RS codeword error occurs the effect is spread uniformly across the complete Information Code Block. Thus the typical number of lost source blocks is simply the number of compressed source blocks represented by the

Information Code Block. Extending our earlier observations using Fig. 27 results in a summary comparison of Interleave A and B in Table 1.

A subtle point. The assumption of complete independence in the decoding of adjacent source blocks is not necessary to avoid the catastrophic propagation of errors (complete loss in data) from one source block to another. For example, by essentially replacing the words "complete independence" by "slightly influenced" would only slightly modify the worst case results in Table 1 for the RM2 data compression system (mentioned at the end of Chapter III). Each stated result for Interleave A and B would include an added " ··· plus some slight additional degradation in the reconstruction of data immediately adjacent to those source blocks which were completely lost."

Table 1. Comparison of Interleave Methods

Rate of Compressed Source Block in Bits	Rate in Bits/ Pixel	Error Event: Typical No. of Lost Source Blocks due to RS Word Error	
		Interleave A	Interleave B
1,784	≈0.4	1 or 2	15 or 16
4,096	1.0	1 or 2	9 or 10
8,192	2.0	1 or 2	5 or 6
16,384	4.0	1 or 2	2 or 3

*Source block contains 4096 pixels.

Acceptable Values of P_{RS}

The discussions just completed describe the effect of individual RS codeword errors in terms of lost source blocks. The next question to a address is the determination of the largest value of P_{RS} for which the overall impact of these error events is considered negligible. More simply, how often can we let these error events occur.

With an RS codeword error rate given by P_{RS}, on the average, a source block error event would occur every $1/P_{RS}$ RS codewords.

But the number of source blocks per RS codeword is given by

$$\gamma = \frac{1784 \text{ information bits/RS word}}{\mathcal{R}_c^B \text{ bits/source block}} \tag{25}$$

Thus, on the average, a source block error event would occur every

$$N_{SB} = \gamma/P_{RS} \text{ source blocks} \tag{26}$$

To carry this point further to a situation which is more readily visualized, assume that our 4096 pixel source blocks are 64 by 64 pixel arrays. Further, assume that the frame size for a picture is 512 by 512 pixels making up a total of 64 source blocks as in the example of Fig. 21. Using (26) we can then say that, on the average, a source block error event would occur every

$$N_p = \frac{N_{SB}}{64} \text{ pictures} \tag{27}$$

Eq. 27 is evaluated for three values of P_{RS} in Table 2.

Table 2. Number of Pictures Between Source Block Errors

Source Block Rates in Bits/Pixel	N_p (Eq. 27) = Average Number of Pictures Between Source Block Error Events (see Table 1)		
	$P_{RS} = 10^{-3}$	$P_{RS} = 2 \times 10^{-4}$	$P_{RS} = 10^{-4}$
≈0.4	13.6	68	136
1.0	6.8	34	68
2.0	3.4	17	34
4.0	1.7	8.5	17

*Source Block contains 4096 pixels
**Picture Size: 512 by 512 pixels (64 source blocks)
***P_{RS} = Probability of an RS codeword error

A lengthy contemplation of Table 2 or even a more extensive plotting of data points is just not a paying proposition. In a moment we will look at the overall performance curve for the concatenated system (P_{RS} vs E_b/N_o) and find that changing P_{RS} by an order of magnitude requires only 0.1 db. Thus the selection of the highest acceptable value of P_{RS} is not a critical issue. However, in order to continue our discussion, we will choose $P_{RS} = 10^{-4}$ as the value of RS codeword error probability below which the effect of lost source blocks can be considered negligible for both interleave methods. This choice has met with harmonious agreement during several presentations of this material.

In support of these conclusions, we note that with this choice of $P_{RS} = 10^{-4}$ and a source block rate of 4.0 bits/pixel, typically only 1 out of 17 pictures would have any degradation due to the channel. That is, the quality of 16 out of 17 pictures would be controlled solely by the characteristics

of the particular data compression algorithm. Typically, every 17th picture would suffer the loss of 1 or 2 source blocks with Interleave A, or 2 or 3 source blocks with Interleave B.

Decreasing the source block rate (increasing the compression ratio) lengthens the interval between source block error events. Specifically, with $P_{RS} = 10^{-4}$ and a source block rate of 0.4 bits/pixel, we see that typically only 1 out of 136 pictures would have any loss in quality associated with the channel. Every 136th picture or so would suffer the loss of 1 or 2 source blocks if Interleave A were used or 16 to 17 source blocks if Interleave B were used (see Table 1).

Uncompressed PCM

When an RS codeword error occurs during the transmission of uncompressed PCM, the result is a burst of errors extending over 1784 bits using Interleave A or spread more thinly over 28,544 bits using Interleave B. If we assumed 8 bits/pixel for each PCM sample, then these error bursts would occur typically once every 8 pictures or so if $P_{RS} = 10^{-4}$. Any imagined advantage to accepting a higher frequency of these error bursts in order to increase transmission rate should be tempered by the fact that changing P_{RS} by an order of magnitude requires only 0.1 db (as we shall see). Therefore, we will also choose $P_{RS} = 10^{-4}$ as the maximum RS word error probability below which degradation to uncompressed PCM data can be considered negligible.

Performance Curves

The performance curves for the Jupiter/Saturn Viterbi decoded K=7, ν=2 convolutional code and our particular choice of concatenated systems is shown in Fig. 28. Both curves maintain the assumption of nominal synchronized phase coherent receiver operation. The Viterbi performance curve is

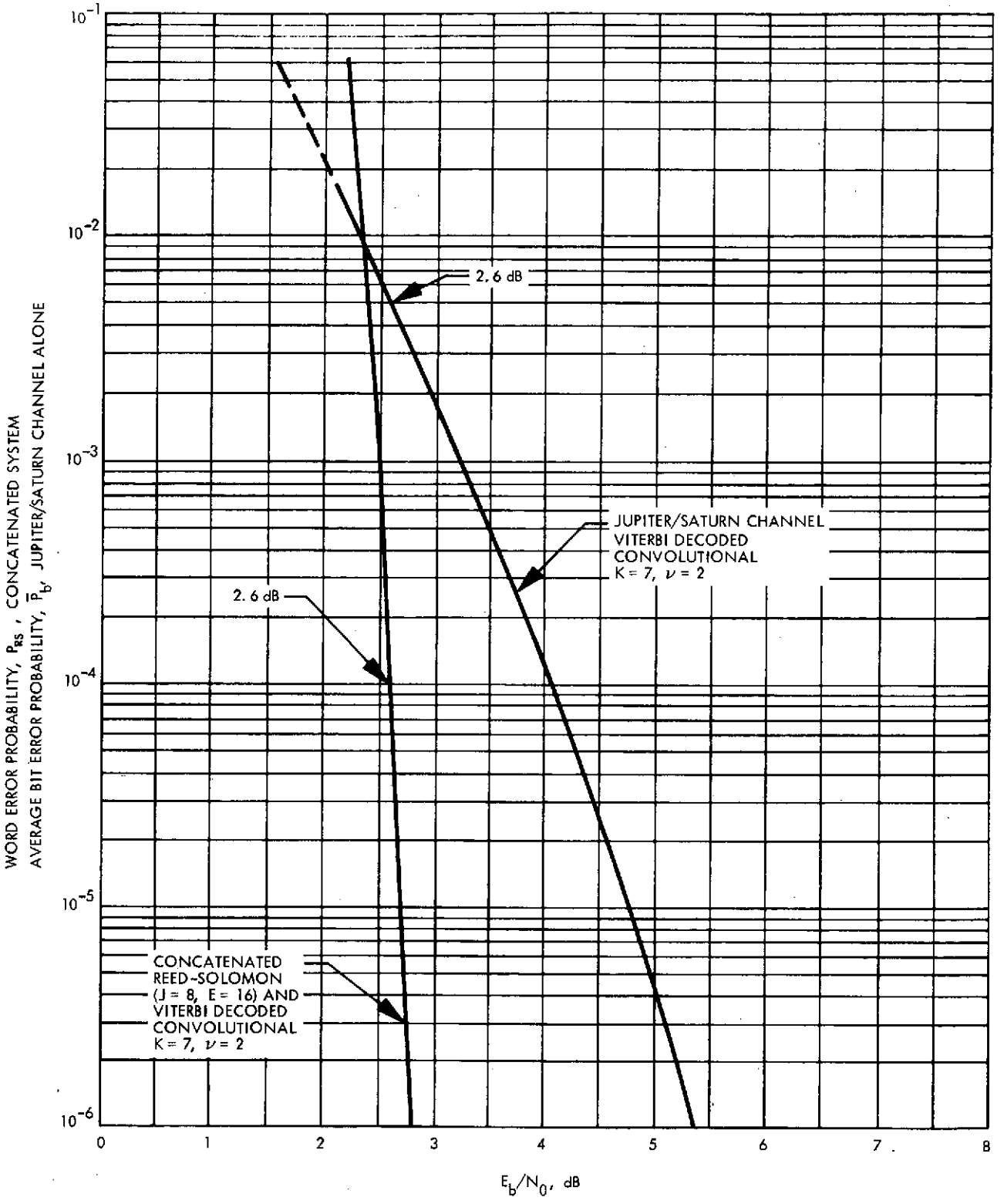


Fig. 28. Performance Curves

the same as that appearing in Fig. 5. The performance curve for the concatenated system is a plot of Eq. 24, for which the qualifications relating to interleaving are discussed in that section.

Recall that for uncompressed PCM data transmitted over the Jupiter/Saturn Channel, we established that $P_b = 5 \times 10^{-3}$ is the approximate value of average bit error probability below which the effect due to errors can be considered negligible. This corresponds to an E_b/N_0 of 2.6 db. Similarly we just established that for both compressed or uncompressed data transmitted on the concatenated channel $P_{RS} = 10^{-4}$ is a reasonable choice of RS word error probability below which any effects due to the channel can be considered negligible. This also corresponds to an E_b/N_0 of approximately 2.6 db. Thus uncompressed data on the Jupiter/Saturn channel and both compressed and uncompressed on the concatenated channel can be transmitted at (about) the same rate with negligible degradation due to channel errors. By our worst case assumptions for the error sensitivity of compressed data, this statement includes virtually any data compression algorithm.

The fact that the $\overline{P}_b = 5 \times 10^{-3}$ operating point for the Jupiter/Saturn Channel and the $P_{RS} = 10^{-4}$ operating point for the concatenated system were determined by "reasonable judgements" and not rigorous mathematical definitions of quality, is utterly without practical significance. The major point is that we no longer have to give up significant transmission rate in order to "use" data compression.

Other Code Combinations

We have noted that the Linkabit study involved many combinations of Viterbi decoded convolutional codes and Reed-Solomon code parameters. Having established the operating characteristics for a particular combination, we can better understand the tradeoffs involved.

Fixing the convolutional code. Of primary practical significance is our emphasis on the K=7, ν=2 convolutional code. As noted in Chapter II there is considerable momentum into the installation of Viterbi decoders for such a code at the DSN stations. This led us to the definition of the Jupiter/Saturn Channel in Fig. 6 and our emphasis on this code.

In an earlier section, the RS code parameters were defined by J and E. Using the K=7, ν=2 convolutional code, Linkabit investigated concatenated systems for which the parameter J was varied from 6 to 9 and E was set to 2, 4, 8, 12, 16, 24, 32, 48, and 64.[26] Of all these codes only two outperformed the one we have emphasized here (J=8, E=16) at values of P_{RS} in the vicinity of 10^{-4}. The RS code with J=9, E=48 offers an advantage of 0.05 db while the J=9, E=32 code offers about 0.1 db.[27]

JPL has estimated (assuming CMOS technology) that something less than 50 chips would be required to implement an RS coder for the J=8, E=16 code employing Interleave B (Fig. 26). The impact of 50 chips or so is relatively insignificant compared to the requirements for on-board data handling and contemplated data compression algorithms. The (very likely) availability of significant on-board working memory in future spacecraft would reduce this impact further for either interleave method. Thus, even for the more complex J=9, E=32 and J=9, E=48 codes, we will concentrate on the more crucial questions relating to implementation of the concatenated system on the ground. We will find that although neither of these more

---

[26] The performance curves presented in the Linkabit study,[1] are actually plots of bit error probability estimates which are less than P_{RS} (Eq. 24) by a factor of about $E/2^J$. That is, to obtain the value of E_b/N_0 for a given P_{RS} for these curves, the reader should select a bit error probability equal to $P_{RS}(E/2^J)$.

[27] Altering Code Block size by a factor of two does not significantly alter the assumptions on source block error events.

complex codes is out of the question, there is a significant practical advantage in choosing J=8 which far outweighs the rather marginal performance advantages.

It is not clear whether the Reed-Solomon decoding at the destination Data Processing Center (see Fig. 22) should be done in hardware, software or some combination of both. Tradeoffs involve costs and time of development, maximum decoder operating speed, impact on the Data Processing Center, etc. These questions cannot be answered now, but some potential gross inefficiencies in design can be avoided by making some simple observations.

The parameter J denotes the length of a Reed-Solomon symbol in bits (see Fig. 24) and the coding and decoding of RS codewords strictly involves operations with these symbols (see Gallager[5]). Since the memory of any modern minicomputer is structured in powers of two, with a byte size of 8-bits the most common, the choice of J=8 is ideal for software decoding applications. The potential advantage in efficiency, both in writing the necessary programs and in operating them, cannot be overstated. For exactly the same reasons, present telemetry standards request that data be grouped into 8-bit bytes. Hardware implementations would be similarly affected since they involve much the same components used in computer design. Clearly, these advantages in choosing J=8 far outweigh the small performance gains of the J=9 codes. Further, the choice of E=16 provides a slight potential advantage in software decoder operating speed since the computation requirements per codeword is dominated by an E^2 term.[1]

If we now look in the other direction at codes which do not perform as well as the J=8, E=16 code, we are certainly not interested in any codes that are more complex. Of those codes which are simpler, the two best offer nearly

identical performance which is inferior by about 0.2 db for values of P_{RS} in the vicinity of 10^{-4}. In addition, these codes have reduced burst error correcting capabilities for the same interleaving depth.

The first of these, J=8, E=8 offers potential advantages over the J=8, E=16 code in two areas without giving up the desirable J=8 feature. Because the equation for computation load per codeword is dominated by an E^2 term, software decoder implementations may more easily achieve high rates. A second advantage is provided by the slightly reduced overhead associated with fewer parity symbols.

A second code offering about the same performance as the J=8, E=8 code has the parameters J=7, E=8. This code is clearly inferior to the J=8, E=8 code. First, the desirable J=8 property is lost. The potential computation advantage over the J=8, E=16 code is diminished because the decoder has less than half as long to do the reduced number of computations (because the codeword size has been reduced). For the same reason the advantage of reduced overhead disappears. Thus the J=7, E=8 code is not a viable alternative.

The next code in order of performance is a J=7, E=4 code which gives up another 0.25 db. Also, burst error correcting capability is further reduced to about 1/8th of the J=8, E=16 code. It might pick up a slight improvement in computation advantage over the J=8, E=8 code if it weren't for the potential inefficiencies introduced by the 7 bit symbols. Thus it offers only a disadvantage when compared to the J=8, E=8 code.

Continuing, a J=6, E=4 code gives up slightly less than 0.1 db further, but reduces burst error correcting capability by another factor of two. It offers only disadvantages compared to the J=7, E=4 code in either computation or overhead. More significantly, the much reduced codeword size will

start affecting our assumptions on source block error events. We can easily disregard this one.

Thus at this point we are left with only two viable contenders, the J=8, E=16 code and the J=8, E=8. Linkabit's study of implementation alternatives suggested that a properly micro-programmed mini-computer could probably achieve decoding speeds for the J=8, E=16 code in the region of 100 kbps although it might be close. This, of course, needs further investigation. A better solution might be to build a hardware decoder for which higher decoding rates are more easily achieved. Linkabit estimated that about 145 off-the-shelf TTL and MOS chips could accomplish this design. This leaves the J=8, E=16 code as "prime" candidate since there is no sense in giving up 0.2 db if you don't have to.

Changing the convolutional code.[28] For our purposes, consideration of a vast assortment of convolutional codes more powerful than the K=7, ν=2 code is of very low priority. The Viterbi decoders for most of these have been discarded as impractical for inclusion at the DSN stations for various reasons. In general, one can expect improvements in performance by increasing K or ν at the expense of implementation complexity and other related difficulties (e.g. increasing ν beyond 3 presents horrendous tracking difficulties for diminishing improvements in performance).[7] One additional code which is seriously being considered is a K=7, ν=3 code which offers between 0.3 and 0.5 db over the K=7, ν=2 code with improvements largest at lower values of $\overline{P}_b$. To avoid the effect of bandwidth expansion at the DSN stations if implemented, the use of the K=7, ν=3 code (3 channel symbols for each information bit) might be restricted to transmission rates below

[28] 3-bits of receiver symbol quantization should be assumed for Viterbi decoders considered here.

C-2

70 to 80 kbps, leaving operation of the K=7, ν=2 code to higher transmission rates. Thus conceivably both codes could be onboard the same spacecraft. Our main purpose here is not to discuss the merits or demerits of installing Viterbi decoders for the K=7, ν=3 code at the DSN stations, but to recognize the possibility and note whether this has any impact on our choice of Reed-Solomon coding parameters.

We can make some reasonable estimates on what to expect with a K=7, ν=3 code by using the results of a K=8, ν=2 code, a K=8, ν=3 code and a K=8, ν=7 code obtained in the initial Linkabit study. Taken collectively these three codes represent a greater perturbation on convolutional code parameters (from the K=7, ν=2 code) than does the K=7, ν=3 code.

For each of these codes, we would make the identical assessment of Reed-Solomon code parameters, and for the same reasons. Again we are left with the two alternative RS codes with J=8, E=16 and J=8, E=8, in all three cases separated in performance near $P_{RS} = 10^{-4}$ by about 0.2 db as before. It is not unreasonable to expect very similar conclusions for the potential DSN candidate convolutional code with K=7, ν=3.

Equally important is the fact that the Viterbi decoder performance improvement obtained by going from the K=8, ν=2 convolutional code to the K=8, ν=3 code is passed on to the concatenated systems (about 0.4 to 0.5 db at $P_{RS} \approx 10^{-4}$). This is not surprising since, given that sufficient interleaving is provided, the performance of an RS decoder depends only on the average probability of RS symbol errors exiting a Viterbi decoder. Thus we can expect a similar result in going from the K=7, ν=2 convolutional code to the K=7, ν=3 code.

A summary conclusion of these observations is not one that ties down the final system configuration or performance, but one which guides the

assignment of priorities for the next level of investigations. The prime candidate RS code parameters are J=8, E=16 since there is no point in giving up 0.2 db if you don't have to. On a first order basis, this choice is virtually independent of the two candidate convolutional codes of which a K=7, ν=2 code is itself the prime candidate at this time (Jupiter/Saturn Channel).

Bandwidth limited applications. The application to deep space telecommunications we have been investigating here is predominantly a power limited rather than bandwidth limited problem. For other applications in which both constraints are severe, the combination of the J=8, E=8 Reed-Solomon code with a high code rate convolutional code (ν smaller) might provide a powerful and practical solution. The general insensitivity of RS code parameters noted in the initial Linkabit Study would certainly lead one to expect "good" results.

Data Other than Imaging

Any scientific mission to the planets will include data other than that provided by imaging experiments. This includes both general science and engineering measurements. Some of this data is considered much more sensitive to channel errors than uncompressed (or pixel edited) PCM imaging data. We will first look at the difficulties this imposes on the proposed Jupiter/Saturn missions.

As we discussed for compressed data, just a few errors can severely degrade a complete block of science data for some experiments. It is quite clear that the transmission of such data over the Jupiter/Saturn Channel at a 5×10^{-3} average bit error rate produces totally unacceptable degradation. A "cleaner" channel is required for this data.

During cruise operations, when science and engineering data totally monopolize the telecommunications channel, an acceptable but not desirable

alternative is provided by simply lowering the transmission rate (increasing E_b/N_0) until the error rate is low enough. As we noted previously, decreasing the transmission rate by a factor of two on the Jupiter/Saturn Channel will reduce the average bit error rate from 5×10^{-3} to about 10^{-6}. However, during a close planetary encounter general science and engineering must "share" the channel with imaging. Imaging experiments are typically allocated between 80 and 90 percent of the total transmission capability during such encounters. Reducing the transmission rate by a factor of two to obtain very low error rates is clearly unacceptable for uncompressed PCM imaging experiments since they only require bit error rates in the vicinity of 5×10^{-3}. The proposed solution to this problem for the Jupiter/Saturn Mariner missions is to put additional error protection on the general science and engineering data using a modified Golay block code. Let's look at this solution.

The basic binary (23, 12) Golay block code (see Berlekamp[15]) was modified to a (24, 12) code for the Jupiter/Saturn application. The codeword length is 24 bits with 12 information "bits" and 12 parity bits. Thus there is a 100 percent overhead associated with the parity bits.

The nominal mode of operation during a close encounter will be to operate the Jupiter/Saturn Channel (see Fig. 6) at the usual 5×10^{-3} bit error rate ($E_b/N_0 \approx 2.6$ db). Uncompressed or edited PCM imaging data would be transmitted directly, but science and engineering data would first be "Golay encoded".[29] Because of the 100 percent overhead due to the Golay parity bits, when the Jupiter/Saturn Channel is operating at an

[29] For the concatenated Golay/Viterbi coding system, interleaving of Golay codewords is necessary for the same reasons that interleaving of Reed-Solomon codewords is required. However, for a given Viterbi decoder error rate, it is more critical because the Golay codewords are almost two orders of magnitude smaller.

E_b/N_0 = x db, the overall concatenated Golay/Viterbi coding system is operating at x + 3 db. Another way of saying the same thing is that when the Jupiter/Saturn Channel is operating at a transmission rate of R bits/sec, the general science and engineering data is really getting through at only R/2 bits/sec.

The additional coding provided by the Golay does accomplish a lowering of the bit error rate on the science and engineering data. It does so quite inefficiently when looked at from an overall coding system viewpoint. For average bit error rates above about 10^{-6}, the concatenated Golay/Viterbi coding system actually requires a higher E_b/N_0 to achieve a given error rate than the Viterbi system alone. The inefficiency is acceptable in this application because it applies to only a small percentage of the total data and solves the problem of error sensitivity for this data. However, this inefficiency becomes a greater concern as the overall transmission rate diminishes (e.g. missions beyond Saturn).

As noted, the Golay does accomplish an acceptable lowering of bit error rates for science and engineering data when the Jupiter/Saturn Channel is operating at a 5×10^{-3} average bit error rate. However, it does so marginally. That is, operating the Jupiter/Saturn Channel at bit error rates only slightly above 5×10^{-3} results in bit error rates out of the concatenated Golay/Viterbi system which is considered intolerable for some scientific experiments. Operationally this means that the E_b/N_0 = 2.6 db at which the Jupiter/Saturn Channel achieves $\overline{P}_b = 5 \times 10^{-3}$ is a fairly tight threshold.

The fact that the Reed-Solomon/Viterbi concatenation systems we have discussed are ideally suited to the requirements of general science and engineering data should not need elaboration. Using our principal

candidate RS code with J=8, E=16, all data can be transmitted through this concatenated system at an overall E_b/N_0 of 2.6 or 2.7 db with negligible degradation due to errors. The Golay can be discarded. Further, those bits which had contributed to the 100 percent overhead of the Golay code can instead be allocated directly to general science or TV. The higher performance also means that during cruise mode the available transmission rate for general science and engineering data is increased. Note also that it is now possible to generally apply data compression techniques to general science and engineering without worrying about a disastrous effect from errors.

Finally, for future reference, we define the Jupiter/Saturn Communications System as the combination of the Jupiter/Saturn Channel (Fig. 6) and the interleaved Golay coding (used exclusively for general science and engineering data).

Imperfect Phase Tracking

Linkabit's initial study and all of our deliberations so far have assumed virtually ideal receiver operating conditions for which carrier phase is known exactly. In practice this is not always the case.

A phase locked loop tracking a noisy received signal will generally provide a phase reference for demodulation which is imperfect. This causes a degradation in system performance. The greater the signal to noise ratio in this carrier tracking loop (which we will call α) the better the reference signal. The purpose of this section is to obtain a reasonable idea of what degradation to expect for the concatenated system as α is decreased.

Before continuing it is important to put the problem in proper perspective, noting what we are intending to accomplish, and perhaps more important, what we are not intending to accomplish. The latter point is the easiest. The arguments we make are in no way intended to replace the

extensive simulations necessary to establish precise performance characteristics. Some of these simulations will be performed in a second study recently initiated. Our intentions here are geared to showing that going to a concatenated system is not likely to introduce any new serious problems.

We have noted many times that the proposed installation of Viterbi decoders at the DSN stations is a very serious proposal. Hence our emphasis on the K=7, ν=2 convolutional code and our definition of the Jupiter/Saturn Channel in Fig. 6. Such proposals must necessarily take into account the phase tracking problem. Therefore, our interests are well served by arguments which suggest that the problem is less severe with the concatenated system.

We will make use of some performance curves generated analytically by Heller and Jacobs[7] for the K=7, ν=2 code of primary interest to us. One major assumption made in this analysis was that a phase error, ϕ, could be considered constant over the length of almost any Viterbi decoder error burst. This is a good assumption under many conditions, but not all those that can be expected for either Mariner or Pioneer missions. However, we are primarily interested in performance trends indicated by these curves and in how they relate to the concatenated system.

Heller and Jacobs noted that the performance curve for the K=7, ν=2 Viterbi decoder under perfect phase coherent conditions (where $\alpha = \infty$) could be written parametrically as a function[30] of E_b/N_0

$$\overline{P}_b(\phi=0) = f(E_b/N_0) \tag{28}$$

[30] The function f we assume in Eq. 28 corresponds to the Viterbi performance curves in Figs. 5 and 28. As noted in Chapter II these curves are slightly pessimistic compared to the results in Ref. 7. This reflects the results of more recent tests on actual hardware. It can be expected that this slight shift in performance will be transferred to the concatenated system when more complete simulations are completed. Practically speaking, this is of no consequence to our discussions and conclusions here.

Then the bit error probability for a constant phase error ϕ could be written as

$$\bar{P}_b(\phi) = f\left(\frac{E_b}{N_0} \cos^2 \phi\right) \tag{29}$$

They then assumed that for a second-order phase locked loop, ϕ is a random variable with distribution given by

$$p_\alpha(\phi) = \frac{e^{\alpha \cos \phi}}{2\pi I_0(\alpha)} \quad , \quad \alpha >> 1 \tag{30}$$

where $I_0(\cdot)$ is the zeroth order modified Bessel function and α is the loop signal to noise ratio.[16] Integrating over ϕ using (29) and (30) Heller and Jacobs obtained the average bit error probability

$$\bar{P}_b^\alpha = \int_{-\pi}^{\pi} p_\alpha(\phi)\, \bar{P}_b(\phi)\, d\phi \tag{31}$$

where now $\bar{P}_b^\alpha$ is a function of both α and E_b/N_0. These curves are shown for several values of α in Fig. 29.[31] Values of $\bar{P}_b^\alpha$ above 10^{-3} have been extrapolated.

The Viterbi curve shown for $\alpha = \infty$ is the same as that given in Figs. 5 and 28. The trend that we wish to make particular note of is that the effect of decreasing α is much more severe at lower values of $\bar{P}_b^\alpha$ than at the higher values. For example, an increase in E_b/N_0 of about 0.75 db is required to maintain an average bit error probability of 10^{-3} when α is

[31] Subsequently we will leave off the α in $\bar{P}_b^\alpha$ when we are referring to ideal phase coherent conditions with $\alpha = \infty$. This is consistent with our earlier notation.

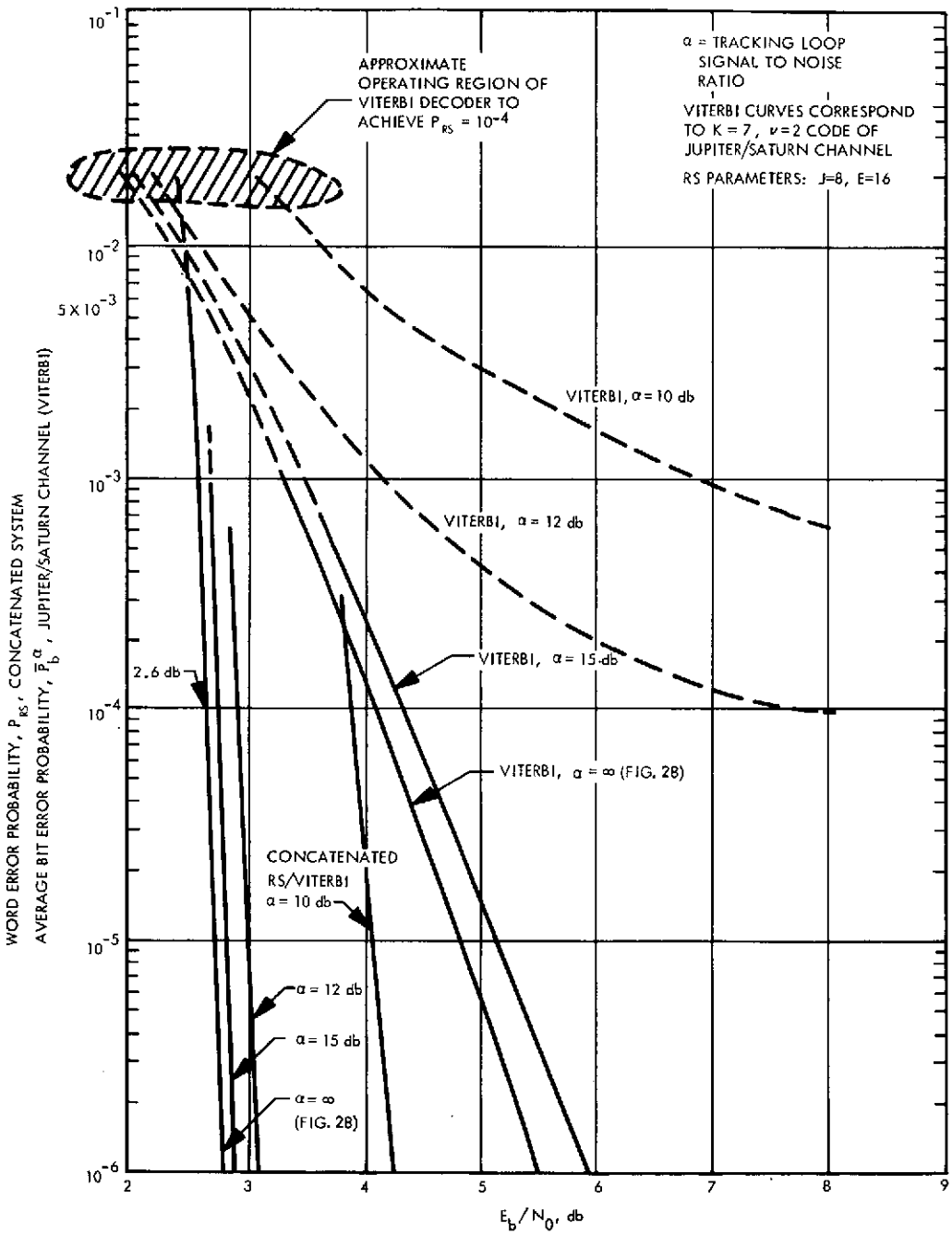


Fig. 29. Degradations Due to Phase Tracking Errors

decreased from 15 db to 12 db. On the other hand, an increase of over 4 db is required to maintain $\overline{P}_b^{\alpha} = 10^{-4}$.

For the transmission of uncompressed PCM using this Viterbi system (Jupiter/Saturn Channel) our often stated rule of thumb criteria for negligible degradation due to the channel is simply that $\overline{P}_b^{\alpha} \leq 5 \times 10^{-3}$. Thus our main points of interest are the separation between the Viterbi curves in the vicinity of 5×10^{-3}. This tells us, for decreasing α, how much E_b/N_0 must be increased (transmission rate decreased) to maintain negligible degradation to the PCM data from channel errors. As a comparison we are interested in the corresponding increases in E_b/N_0 which would be required by the concatenated system to maintain negligible degradation due to errors for <u>both</u> uncompressed and compressed PCM.

Recall from earlier sections that a quite reasonable choice for Reed-Solomon word error probability, below which degradation due to the channel could be considered negligible for both compressed and uncompressed data, is $P_{RS} = 10^{-4}$. We noted that if sufficient interleaving was provided P_{RS} depended (through Eq. 24) only on $\overline{\pi}$, the average RS symbol error probability exiting a Viterbi decoder. This statement is unaffected by the introduction of time varying phase errors although the depth of interleaving required for "sufficiency" is probably larger. For a given code, the K=7, υ=2 code here, there is a monotonic relationship between $\overline{\pi}$ and the average bit error probability $\overline{P}_b^{\alpha}$. That is, we can interpret P_{RS} as a function of $\overline{P}_b^{\alpha}$. Consequently, we could rewrite Eq. 24 with P_{RS} as a function of $\overline{P}_b^{\alpha}$ instead of $\overline{\pi}$, say, $P_{RS} = h(\overline{P}_b^{\alpha})$. The critical value of $\overline{P}_b^{\alpha}$ of interest to us is the largest value which makes $h(\overline{P}_b^{\alpha}) \leq 10^{-4}$. The critical value is (approximately) the same for each α but the E_b/N_0 at which it occurs will be larger for smaller values of α.

Now, we already know that under phase coherent conditions with $\alpha = \infty$, $P_{RS} = 10^{-4}$ is obtained at an overall E_b/N_0 of 2.6 db. Because of the 15 percent overhead for the parity symbols of the J=8, E=16 RS code, the corresponding operating point for the Viterbi decoder is in the vicinity of $E_b/N_0 = 2$ db. From the Viterbi curve for $\alpha = \infty$ in Fig. 29 (or Figs. 5 and 28) we see that such an operating point gives a critical $\overline{P}_b \approx 1/50$. This same (approximate) average bit error probability is obtained for smaller α at higher values of E_b/N_0. The approximate operating region where this critical bit error probability is reached for each α is shown in Fig. 29 by the crosshatched region.

When the Viterbi decoder reaches this critical bit error probability somewhere inside the crosshatched region, $P_{RS} = 10^{-4}$ regardless of α. Thus as the loop signal to noise ratio α is decreased, the amount that E_b/N_0 must be increased (rate decreased) to maintain $P_{RS} = 10^{-4}$ is specified by the separation of the Viterbi curves within the crosshatched region. From this observation it is easy to plot the concatenated system performance curves for the different values of α as shown in Fig. 29.[32]

Noting that the separation of Viterbi curves is smaller in the crosshatched region than when $\overline{P}_b^{\alpha} = 5 \times 10^{-3}$ would lead to the conclusion that: as α is decreased, the concatenated system must increase E_b/N_0 by less than the Viterbi system alone to maintain negligible degradation due to channel errors. Observe that this conclusion did not depend on the precision of the performance curves, but only on the trend that the Viterbi

[32] Similar modeling at JPL produced performance curves in general agreement with those in Fig. 29. This "high rate" model was considered quite reasonable for transmission data rates in excess of 1 kbps. A "low data rate" model produced curves which maintained a constant separation as $\overline{P}_b^{\alpha}$ was varied. This would lead to the conclusion that degradation in performance due to imperfect phase tracking at low data rates would be about the same for the concatenated system and Viterbi alone.

performance curves become closer together (for different α) as bit error probability is increased. This result should, therefore, be less sensitive to the scrutiny of Heller and Jacob's initial assumptions than the performance curves themselves.

AGC

Following the discussion in[7], coded systems that make use of receiver outputs quantized to more than two levels require an analog-to-digital converter at the receiver matched filter output, with thresholds that depend on correct measurement of the noise variance. All Viterbi decoded systems we have discussed used 8-levels of quantization. Level settings are effectively controlled by automatic gain control circuitry (AGC) and thus it is of interest to understand the potential effect of an inaccurate AGC signal on performance. We can afford to be brief here. Linkabit tests[17] on their K=7, υ=2 Viterbi decoder indicated that (under phase coherent conditions) for AGC measurements off by as much as 3 db, the ideal value of average bit error probability $\bar{P}_b$ (obtained with perfect AGC), could be restored by an increase in E_b/N_0 of 0.1 db. This included all values of $\bar{P}_b$ of interest to us. As we have noted many times, for a given code combination with sufficient interleaving, P_{RS} depends only on $\bar{P}_b$ through $\bar{\pi}$ (see discussion on phase tracking).

Suppose $\bar{P}_b^*$ is the critical value of $\bar{P}_b$ which results in $P_{RS} = 10^{-4}$. Then if a 0.1 db increase in E_b/N_0 (at the Viterbi decoder) will restore $\bar{P}_b$ to $\bar{P}_b^*$ it will also restore P_{RS} to 10^{-4}. Again, we emphasize that these arguments are not intended to replace simulations. However, the conclusions are unmistakable. Degradation in performance due to imperfect AGC can be expected to be about the same for the concatenated Reed-Solomon/Viterbi system as for the Viterbi (Jupiter/Saturn) system alone. Further, this degradation can be expected to be minor.

Slow Drifts in E_b/N_0

In practice E_b/N_0 values at the DSN receivers may slowly drift about an expected nominal value. Because of the performance curve steepness of the RS/Viterbi concatenated system, the effect of a drift in E_b/N_0 values below 2.6 db (about 2 db for the Viterbi decoder part of the concatenated system) could be quite abrupt. One can avoid this problem with the addition of a buffer zone around 2.6 db by choosing a nominal operating point of say, $O_1 = 2.6 + x$ db.[33]

An almost identical situation exists for the proposed "Jupiter/Saturn Communications System" but for slightly different reasons. If one were concerned only with the transmission of uncompressed (or pixel edited) PCM imaging data directly over the Jupiter/Saturn Channel, the effect of drifts in E_b/N_0 below 2.6 db ($\overline{P}_b = 5 \times 10^{-3}$) would not be as abrupt. Channel errors do not render this data virtually useless until E_b/N_0 values in the vicinity of 1.6 db are reached ($\overline{P}_b \approx 1/20$). Thus one might be tempted into choosing a smaller buffer zone which permitted occasional drifts below 2.6 db. That is, choosing a nominal operating point of $O_2 = 2.6 + y$ db, where $y < x$. However, in a recent section we noted that the Jupiter/Saturn Communication System must also handle general science and engineering data. To handle this more error sensitive data, the Jupiter/Saturn Communication System also includes a Golay block code which is used exclusively on the general science and engineering data. Even with this additional error protection, the bit error rate resulting from operation of the Jupiter/Saturn Channel at $E_b/N_0 \approx 2.6$ db is considered barely adequate for some experiments. Thus operation of the Jupiter/Saturn Channel only slightly below 2.6 db is unacceptable, not because

[33] This nominal operating point can be chosen to account for degradations due to imperfect phase tracking and AGC errors, but we will assume the ideal performance curves in Fig. 28.

of imaging, but because of general science. Therefore, the rather questionable tradeoff in accepting very noisy PCM imaging data in return for a small transmission rate advantage does not really exist for the Jupiter/Saturn Communication System. This leaves $O_2 \approx O_1$.

Summary of Characteristics

Listed below is a brief summary of major characteristics we have attributed to a Reed-Solomon concatenated coding system aimed at applications to future Mariner or Advanced Pioneer missions employing imaging.[34] The reader is referred to the lengthy discussions above for elaboration and qualification of these statements.

- Under ideal receiver operating conditions, all data (uncompressed and compressed imaging, general science and engineering) can be transmitted at an E_b/N_0 of approximately 2.6 db with negligible degradation due to channel errors. For all but uncompressed PCM imaging data, this performance offers an advantage of approximately 3 db (factor of two) in transmission rate over the proposed Jupiter/Saturn Communication System[35] (during planetary encounter modes).
- Degradations in performance due to imperfect receiver phase tracking and AGC should be about the same as for the Jupiter/Saturn Communication System (i.e., for a Viterbi decoded convolutional K=7, ν=2 code with 8 levels of receiver (quantization).
- Significant burst error correcting capability in ground communications or on-board storage of data is provided.

[34] Recently, Chen[18] suggested the application of concatenated RS/Viterbi coding to low data rate atmospheric probes which do not include imaging experiments.

[35] Assumes worst case sensitivity to errors for compressed data.

- A Reed-Solomon decoder can be implemented at a single Data Processing Center, avoiding severe impact on the many DSN stations. The implementation complexity of a hardware decoder, capable of operating at up to 100 kbps, was estimated at 145 chips using available technology.

WHY NOT SEQUENTIAL DECODING?

Sequential decoding of long constraint length codes (see Chapter VI of Wozencraft and Jacobs,[4] and Jacobs[2] for an introduction) is another potential means of providing the necessary "clean" channel for compressed imaging data (and general science and engineering) at low values of E_b/N_0. Although recent Pioneer Missions employed a software decoded K=32, υ=2 convolutional code, the decoders can operate effectively at maximum decoding rates of 2 kbps or so and are therefore not generally applicable.

A study by Rice [19] investigated the applicability of a high speed sequential decoder[20], [21] to compressed imaging data. The study made use of many of the same arguments used here in this chapter. The assumed error sensitivity for compressed data was virtually the same "worst case" assumption used here. The principal error event of the sequential decoder (modified Fano Algorithm) was a "burst" of erasures up to 1024 bits in length, similar to the loss of a codeword using Interleave A in Fig. 27. The "ideal" theoretical performance curves assumed (erasure rate vs. E_b/N_0) were for a υ=3 code and were about 0.3 to 0.4 db better than a more practical υ=2 code (which Layland emphasized in his simulations). Comparisons were made with the transmission of uncompressed PCM imaging data using a Viterbi decoded K=6, υ=3 code at a 5×10^{-3} average bit error rate. This is much the same comparison we have emphasized here using the K=7, υ=2 code (which is about 0.3 db inferior to the K=6, υ=3 code at $\overline{P}_b \approx 5 \times 10^{-3}$).

The sequential decoder performance curves are much steeper than the Viterbi decoder performance curves but not as steep as the Reed-Solomon/Viterbi performance curves (see Fig. 28). Primarily this means that the bursty characteristic of error events for the two systems is worth more to sequential decoding than to the concatenated system.[36]

If we extrapolate the results of Ref. 19 we would conclude that: it is probably possible to build a hardware sequential decoder capable of operating at a maximum decoding rate in the vicinity of 100 kbps and which achieves performance considered comparable to the RS/Viterbi concatenated sytem under ideal receiver operating conditions.

It would be difficult to make a more precise statement without considerable elaboration primarily because, ideally, performance of a sequential decoder improves as data rate is decreased. However, the statement will suffice. There are more crucial practical considerations which, based on present knowledge, make the Reed-Solomon/Viterbi concatenation system a more cost-effective choice.

The vast majority of work on sequential decoding has been done under the assumption of ideal receiver operating conditions. For those intimately familiar with the practical aspects of both sequential decoders and Viterbi decoders there seems to be a universal rule that sequential decoding is considerably more sensitive to receiver imperfections such as AGC or phase tracking problems. This observation is loosely stated in many places, but a direct comparison which would help us here is unavailable. We will accept it as an unresolved issue. We have noted that the degradation to performance of the RS/Viterbi concatenated system from these effects is expected

[36] Layland showed that with proper buffer management, these curves can be made considerably steeper. Ref. 22.

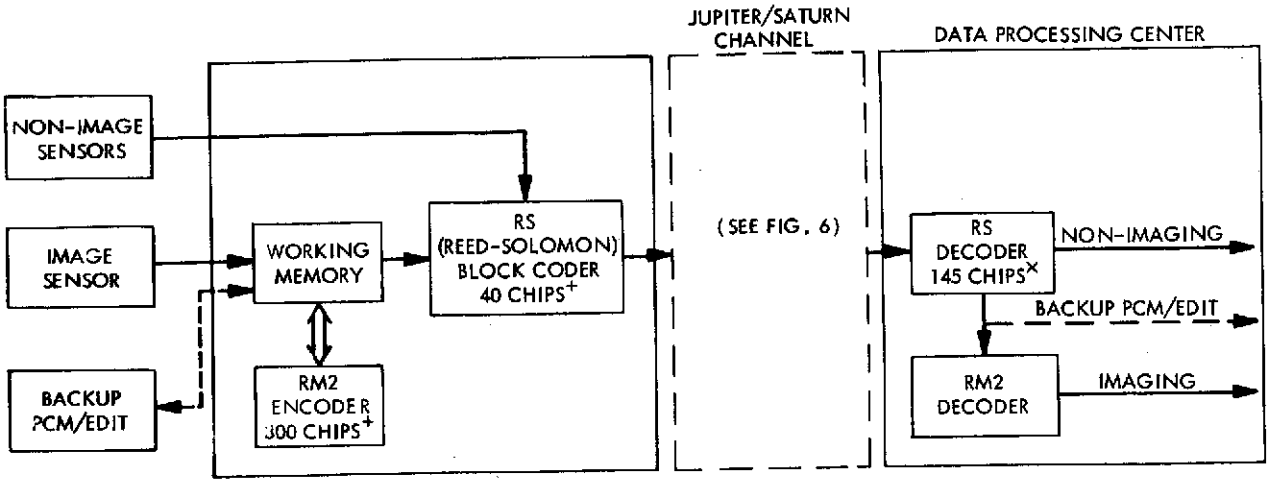
to be less than or equal to that of a Viterbi system alone. Thus, the concatenated system may clearly outperform a sequentially decoded system when receiver imperfections are taken into account.

Perhaps the major practical difference in systems is obtained by noting that to implement the concatenated system requires the installation of a single Reed-Solomon decoder at a single destination Data Processing Center whereas a sequentially decoded system requires new sequential decoders to be placed at each DSN station.

Other important but less significant advantages of the concatenated system include the considerable burst error correcting protection of data both on-board and through ground communications. The installation of more powerful Viterbi decoders at the DSN stations at some later time would map directly into improvements in performance for the concatenated system.

V. INTRODUCTION TO AICS

Many of the system concepts discussed in this report were consolidated into a series of presentations (given by the author and Ed Hilbert) which served as proposals for future Mariner and Advanced Pioneer missions. The intent of this very brief chapter is principally to identify, at a glance, the major system elements and nomenclature of these proposals. Following this intent, a block diagram of the proposed Advanced Imaging Communication System (AICS) is shown in Fig. 30. This figure is a more elaborate version of Fig. 22 where we first introduced the Reed-Solomon concatenation concept. The reader may obtain a lengthy development of that subject in Chapter IV. The RM2 data compression system specified in the diagram is a recent development still in the research stage. Complete documentation is not presently available. However, the system concepts discussed at length in Chapter III clearly motivated RM2 research. The discussions identify desirable properties for data compression systems and these properties have been exhibited in preliminary evaluations of RM2. Viewed from an overall system standpoint, results clearly indicate that AICS offers significant improvements in imaging capabilities over spacecraft which emulate the Mariner Jupiter/Saturn '77 configuration.



+ ESTIMATED ASSUMING CMOS TECHNOLOGY; INTERLEAVE B, I = 16
× ESTIMATED ASSUMING TTL, MOS TECHNOLOGY; INTERLEAVE B, I = 16
REED - SOLOMON CODE PARAMETERS (PRINCIPAL CANDIDATE): J = 8, E = 16

Fig. 30. Introduction to AICS

APPENDIX A

DECIBEL REPRESENTATION

Any quantity r can be converted to decibel form, denoted $\hat{r}$, by the equation

$$\hat{r} = 10 \log_{10} r \tag{A-1}$$

Multiplication of r by some factor β is given as $r' = \beta r$. This operation reduces to addition in decibel form

$$\hat{r}' = 10 \log_{10} r + 10 \log_{10} \beta = \hat{r} + \hat{\beta} \tag{A-2}$$

The correspondence between the factor β and its decibel representation is given in Figs. A-1 and A-2. Note that multiplicative factors of 2 and 1/2 correspond to +3 db and -3 db, respectively.

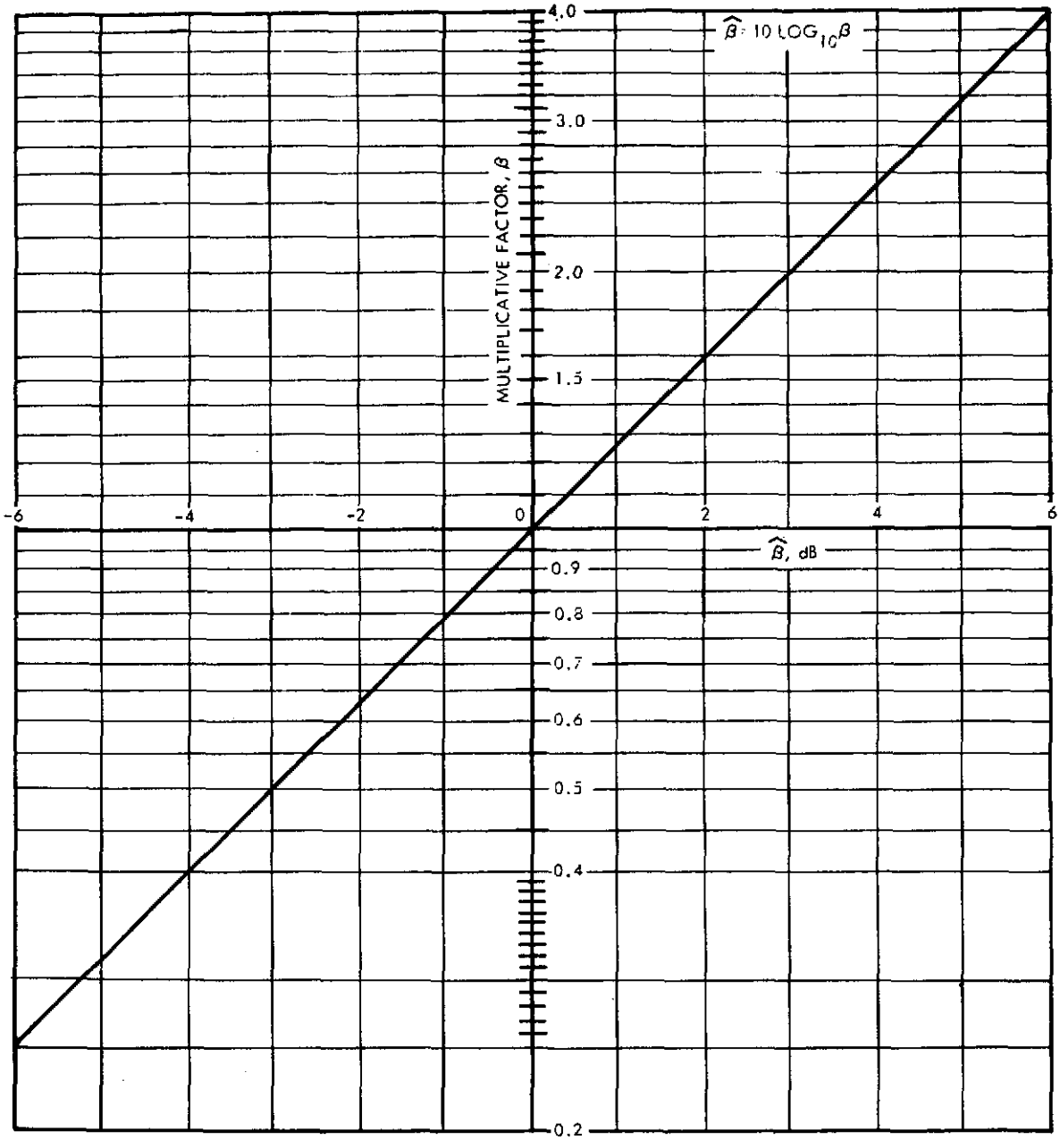


Fig. A-1. Decibel Conversion

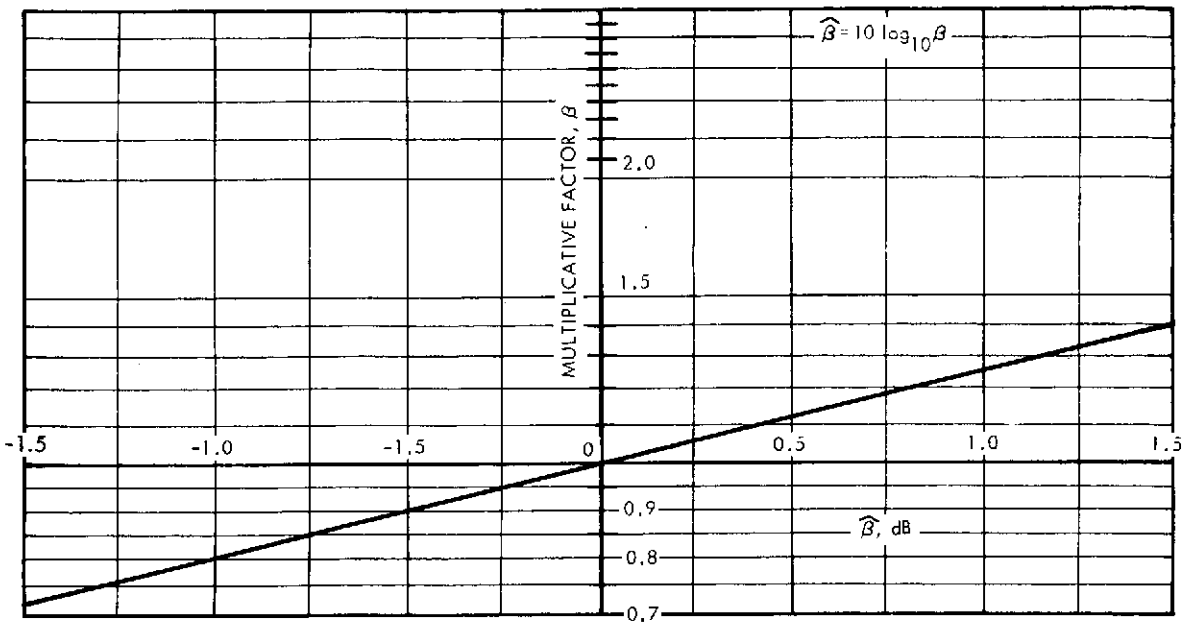


Fig. A-2. Decibel Conversion, Expanded Scale Around Zero

APPENDIX B

RS CODE BLOCK SYNCHRONIZATION

BASIC GOALS

The basic idea of synchronization here is to find and maintain the correct location of the starting point of RS Code Blocks in a long bit stream containing many such RS Code Blocks. Correctly identifying the starting point of an RS Code Block correctly locates all other bits for that code block (provided some weren't missing for some reason). Decoding of the RS code words can proceed.

During an acquisition phase a search is made for a known sequence of bits (the synchronization sequence which we will call SYNC) whose relative location to the start of an RS Code Block is also known. Correctly identifying SYNC, and therefore the start of an RS Code Block, means that the system is "locked up" or "synchronized". For the situation we are concerned with the actual synchronization sequence may be modified because of errors so that, in order to lock up, it is necessary to recognize not only SYNC itself, but also close approximations to it. If more errors occur than have been accounted for by these approximations, the system will not recognize the actual occurrence of SYNC. It will "miss lock". The latter is a very undesirable event and its likelihood should be made as small as possible.

The chances of missing lock can be reduced by recognizing a greater number of approximations to SYNC during search. Doing so, however, increased the chances that some other sequence of bits is incorrectly identified as SYNC. This event we call a "false lock" and is, or course, undesirable. Its likelihood of occurrence should also be made as small as possible.

Once synchronization is obtained, it must be periodically monitored (e.g., once every RS Code Block). During this monitoring phase, it is clearly

Preceding page blank

undesirable to make any decision that the system has lost lock when, in fact, it was correctly synchronized.

In the following more detailed discussions, the reader can assume that we are directing attention to the prime candidate coding system with parameters J=8 (i.e., 8-bit symbols), E=16 and an interleaving depth of I=16. Any restrictive statements could easily be generalized for other cases.

SEQUENCE CORRELATION

To make things simple, we'll define the correlation between two 16 symbol sequences, S_1 and S_2, as the number of correct "symbol" comparisons between the two and call it $C(S_1, S_2)$. Symbol comparisons are made with the sequences lined up: first symbol of S_1 is compared with the first symbol of S_2, second symbol with the second, and so on. Thus, $C(S_1, S_2)$ could be any number from 0 to 16.

SYNCHRONIZATION BASICS

Suppose we again let SYNC be the desired 16 symbol synchronization sequence and randomly chose each bit of S_1 to be a zero or one. The probability that S_1 will be chosen identical to SYNC (i.e., $C(S_1, \text{SYNC}) = 16$) is 2^{-128}, an incredibly small number. Thus, if we had an <u>error free</u> bit stream of random zeroes and ones (e.g., compressed data) with the synchronization sequence, SYNC, imbedded somewhere in it, the chances of finding SYNC at any other place would be virtually zero (false lock).

More realistically consider the case where symbol errors occur. We establish the rule: <u>decide sequence, S, is the synchronization sequence, SYNC, if $C(S, \text{SYNC}) > T$.</u>

Under this rule we would miss SYNC <u>if it was really there</u> only if there had been 16-T or more symbol errors. That is

$$P_{MS}^{T} = P_r \left[\begin{array}{l} \text{Missed SYNC} \\ \text{With Correlation} \\ \text{Threshold T} \end{array} \right] = P_r \left[\begin{array}{l} \text{16-T Or More} \\ \text{Symbol Errors} \\ \text{In SYNC} \end{array} \right] \tag{B-1}$$

Obviously, as <u>T is lowered P_{MS}^{T} decreases.</u>

In setting a threshold T we will decide any sequence S is really SYNC provided its correlation exceeds $T (T \leq 15)$. If we are wrong in this decision, a false lock results (during search mode) we can bound the probability of this event by

$$P_{FL}^{T} = \Pr \left[\begin{array}{l} \text{False Lock} \\ \text{With Correlation} \\ \text{Threshold T} \end{array} \right] \leq \sum_{j=0}^{T} \Pr \left[\begin{array}{c} \text{All Sequences, S,} \\ \text{With} \\ C(S, \text{SYNC}) \geq j \end{array} \right] \cdot \Pr \left[\begin{array}{l} \text{T + 1 - j Or} \\ \text{More Symbol} \\ \text{Errors In S} \end{array} \right] + \Pr \left[\begin{array}{c} \text{All Sequences, S,} \\ \text{With} \\ C(S, \text{SYNC}) > T \end{array} \right] \tag{B-2}$$

where we have taken advantage of the fact that the error process is independent of the process which produces each sequence, S.[37] Clearly, <u>P_{FL}^{T} increases as T is decreased.</u> In reality P_{FL}^{T} would have to be weighted by the number of sequences, S, that are compared with SYNC, during a search. This could depend on how well the location of SYNC was known (and on how elaborate a search algorithm was implemented). At worst the weighting factor would be the length of an RS Code Block ($\approx$32,000).

Once the system was locked up, the known position of SYNC (we have assumed one SYNC for each RS Code Block) could be monitored to check that the system is still synchronized. The same type of problem exists as in the

[37] SYNC must be carefully chosen so that cyclic shifts of SYNC <u>do not</u> have a high correlation. Otherwise, only a few errors might result in a decision to lock up on a shifted version of SYNC.

search mode. If a sufficient number of errors occurred in SYNC, the system would have to decide that synchronization had been lost. We'll call this event a false unlock and denote its probability by P_{FU}^{T}. This is a very undesirable event since it would initiate a potentially long search. P_{FU}^{T} could be determined by an equation such as (B-1).

Similarly, if the system had lost synchronization, then deciding that it was still locked up would be the equivalent of a false lock during acquisition. We'll denote the probability of this event by P_{FL2}^{T}. It could be determined by an equation such as (B-2). Note that the T in P_{FU}^{T} and P_{FL2}^{T} does not necessarily imply the same threshold as in P_{MS}^{T} and P_{FL}^{T}.

In general the optimization of thresholds would be preceded by weighting the probabilities P_{MS}^{T}, P_{FL}^{T}, P_{FU}^{T} and P_{FL2}^{T} by cost functions which assessed the impact of each event. The implied elaborate tradeoffs would seem to be out of place and unnecessary here. On a first order basis, it is likely that all of these terms can be made negligibly small without much difficulty. As we noted in the main text, even two 128 bit synchronization sequences affects data rate by less than one percent. So there is a lot of flexibility in achieving performance goals. In the following section, we discuss briefly several configurations which serve as suggestions for further simulations and analytic work.

SOME ALTERNATIVES

Let's first look at the basic configuration for synchronization which Linkabit assumed but did not investigate in their initial study. A single 16 symbol synchronization sequence, which we will again call SYNC, was assumed to separate each RS Code Block of 16 codewords as shown in Fig. B-1. This is probably the simplest configuration and is desirable for that reason, but it has some drawbacks. Because all the symbols of SYNC are

transmitted consecutively over the Jupiter/Saturn Channel, it is subject to the bursty error events characteristic of Viterbi decoders at high average bit error rates. This statement is true regardless of the type of interleave (A or B) or whether the system is locked up or not. Equations B-1 and B-2 could be evaluated analytically by modeling the error events from Viterbi burst error statistics and modeling the occurrence of sequences, S, with the assumption that each bit of S is chosen to be a zero or one with equal probability. More desirable simulations would be quite straightforward.

Now let's modify Configuration 1 slightly to improve its performance under synchronized conditions. Instead of making SYNC separate from an RS Code Block we chose it to be part of the Code Block. In particular, for both Interleave A and B, we let the first symbol of SYNC be the first symbol of codeword 1, the second symbol of SYNC be the first symbol of codeword 2, ..., the 16th symbol of SYNC be the first symbol of the 16th codeword. We'll call this Configuration 2. The reader will see from Figs. 25 and 26 that Configuration 2 means that SYNC is the first 16 symbols transmitted in an RS Code Block for both Interleave A and B. When the system is trying to find SYNC to lockup, the situation is the same as for Configuration 1 because all symbols of SYNC are transmitted consecutively. However, performance is considerably improved once the system is locked up. In a synchronized mode, SYNC would be monitored to check that the system was maintaining lock after RS decoding. Since each symbol of SYNC is an information symbol of a different codeword, each is therefore protected by the formidable error

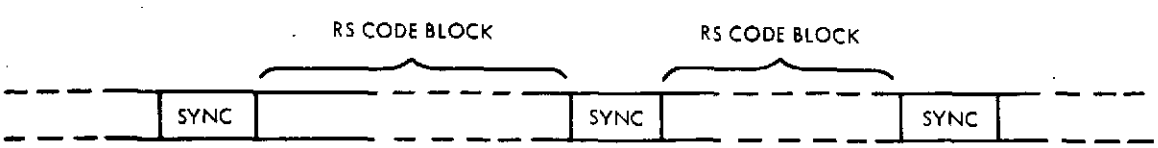


Fig. B-1. Sync Configuration 1

correcting capability of each RS codeword. It doesn't require much elaboration to see that P_{FU}^{T} and P_{FL2}^{T} can easily be made virtually zero.

Now let's try and improve performance during the acquisition phase. In Configuration 3 we let SYNC be the first 16 symbols of codeword 1. The reader may check Figs. 25 and 26 to see that this means that, for both Interleaves A and B, SYNC has been interleaved so that each symbol is separated on the Viterbi channel by 15 other symbols. Making our usual assumption for the sufficiency of an interleaver depth of 16, symbol errors in SYNC will occur independently with probability $\bar{\pi}$ (the average probability of a symbol error out of the Viterbi decoder). Unlike Configurations 1 and 2, the chances of missing SYNC or getting a false lock will no longer be dominated by error bursts. There will tend to be fewer long sequences of errors and the threshold T will have a more noticeable control on P_{MS}^{T} and P_{FL}^{T}. Equations B-1 and B-2 can be evaluated in a straightforward manner since error events are now binomial.

Configuration 3 will give up some protection in the synchronized mode since all symbols of SYNC belong to a single codeword. Thus if codeword 1 is ever wrong, there would be a tendency for a large number of errors to occur within SYNC, a potential for lost lock.

Configuration 4 retains the desirable attributes of both Configurations 2 and 3 while still using only one SYNC sequence. Here we let the first symbol of SYNC be the first symbol of codeword 1, the second symbol of SYNC becomes the second symbol of codeword 2, the third symbol of SYNC becomes the third symbol of codeword 3, ..., the 16th symbol of SYNC becomes the 16th symbol of the 16th codeword. The reader can see that each symbol of SYNC is protected by a separate codeword during the synchronized mode and protected from Viterbi error bursts during the acquisition phase.

Any of the schemes described above could be supplemented with another 16 symbol sequence with negligible impact on data rate. Thus it seems fairly certain that all of the relevant probabilities we have mentioned, P^T_{MS}, P^T_{FL}, P^T_{FU} and P^T_{FL2}, can be made negligible without altering the transmission rate capabilities to any degree.

Analytic evaluations and simulations are required to decide just what configuration is required. None of the configurations suggested above generates a severe impact on the overall Reed-Solomon decoder implementation, particularly a hardware implementation. The arguments here need to be extended to take into account the total environment of the Data Processing Center where the Reed-Solomon decoder would be located. However, they suggest that there are no major difficulties.

SYNCHRONIZATION OF SOURCE DATA

We have assumed in the text that source blocks (data frames) compressed or not (imaging data or not), would be separated by sync words. Transmitting data directly over the Jupiter/Saturn Channel means that source block synchronization is subject to the same basic problems we have just discussed for the synchronization of RS Code Blocks. It would serve no purpose to elaborate on the similarities and differences here. The point that we wish to make is that most of these difficulties would disappear when using the concatenated coding system. As Chapter IV clearly indicates, source data and the sync words separating source blocks would be virtually error free almost all the time when exiting a synchronized RS decoder. Under these conditions, the synchronization of source blocks is clearly a much simpler problem.

REFERENCES

1. J. P. Odenwalder et. al., "Hybrid coding systems study," submitted to NASA Ames Res. Ctr. by Linkabit Corp., San Diego, Calif., Final Rep., Contract NAS2-6722, Sept. 1972.

2. I. M. Jacobs, "Sequential decoding for efficient communication from deep space," IEEE Trans. Commun. Technol., vol. COM-15, Aug. 1967, pp. 492-501.

3. "Deep Space Network/Flight Project Interface Design Handbook," JPL Document 810-5, Rev. C. Jet Propulsion Laboratory, Pasadena, Calif., April 15, 1972 (JPL Internal Document).

4. J. M. Wozencraft and I. M. Jacobs, Principles of Communication Engineering. New York: Wiley, 1965.

5. R. G. Gallager, Information Theory and Reliable Communication. New York: Wiley, 1968.

6. A. J. Viterbi, "Convolutional codes and their performance in communication systems," IEEE Trans. Commun. Technol., vol. COM-19, part II, Oct. 1971, pp. 751-772.

7. J. A. Heller and I. M. Jacobs, "Viterbi decoding for satellite and space communication," IEEE Trans. Commun. Technol., vol. COM-19, part II, Oct. 1971, pp. 835-848.

8. R. F. Rice, "Television data compression for a grand tour of the outer planets," Proceedings of 4th Hawaii International Conf. on Sys. Science, Jan. 1971, pp. 595-597.

9. R. F. Rice and J. R. Plaunt, "Adaptive variable length coding for efficient compression of spacecraft television data," IEEE Trans. Commun. Technol., vol. COM-19, part I, Dec. 1971, pp. 889-897.

10. W. K. Pratt, "Bibliography on digital image processing and related topics," USCEE Report 453. University of Southern California, Los Angeles, Calif., Sept. 1, 1973.

11. T. Berger, Rate Distortion Theory: A Mathematical Basis for Data Compression. Englewood Cliffs, N. J.: Prentice-Hall, 1971.

12. R. F. Rice, "RM2: transform operations," Technical Memorandum 33-680. Jet Propulsion Laboratory, Pasadena, California, March 1, 1974.

13. J. P. Odenwalder, "Optimum decoding of convolutional codes," Ph. D. dissertation, Syst. Sci. Dep., Univ. California, Los Angeles, 1970.

14. James L. Massey, "Shift register synthesis and BCH decoding," IEEE Trans. Info. Theory, vol. IT-15, pp. 122-127, Jan. 1969.

15. E. R. Berlekamp, Algebraic Coding Theory. New York: McGraw-Hill, 1968.

16. A. J. Viterbi, Principles of Coherent Communication. New York: McGraw-Hill, 1966.

17. Instruction Manual for LV7015C Convolutional Encoder, Viterbi Decoder. Linkabit Corp., San Diego, California, 1972.

18. C. H. Chen, "Low data rate digital space communications," Southeastern Massachusetts Univ., North Dartmouth, Mass., Final Rep., NASA Grant NGR-22-031-002, Nov. 1973.

19. R. F. Rice, "Channel coding/decoding alternatives for compressed TV data on advanced planetary missions," Proceedings of 5th Hawaii International Conf. on Sys. Science, Jan. 1972, pp. 60-62.

20. W. A. Lushbaugh and J. W. Layland, "System design of a sequential decoding machine," SPS 37-50, Vol. II. Jet Propulsion Laboratory, Pasadena, California, pp. 71-78.

21. J. W. Layland, "Multiple-mission sequential decoder-comparing performance among three rate 1/2, K=32 codes," SPS 37-64, Vol. II. Jet Propulsion Laboratory, Pasadena, Calif., pp. 50-52.

22. J. W. Layland, "Performance of an optimum buffer management strategy for sequential decoding," Technical Report 32-1526, Vol. IX. Jet Propulsion Laboratory, Pasadena, Calif., pp. 88-96.

NASA — JPL — Coml., L.A., Calif.