https://ntrs.nasa.gov/search.jsp?R=19850026906 2020-03-20T17:15:48+00:00Z

General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

Produced by the NASA Center for Aerospace Information (CASI)

JPL PUBLICATION 85-53

Noiseless Coding for the Gamma Ray Spectrometer

Robert F. Rice **Jun-ji Lee**

(NASA-CR-176201) NOISELESS CODING FOR THE GAMMA RAY SPECTROMETER (Jet Propulsion Lab.) 37 p HC A03/MF A01 CSCL 22B

N85-35219

Unclas
G3/19 27479

June 1985

National Aeronautics and
Space Administration

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

TECHNICAL REPORT STANDARD TITLE PAGE

1. Report No. JPL 85-53	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Noiseless Coding for the Gamma Ray Spectrometer		5. Report Date June 1985 6. Performing Organization Code
7. Author(s) Robert Rice and Jun-ji Lee		8. Performing Organization Report No.
9. Performing Organization Name and Address JET PROPULSION LABORATORY California Institute of Technology 4800 Oak Grove Drive Pasadena, California 91109		10. Work Unit No. 11. Contract or Grant No. NAS7-918 13. Type of Report and Period Covered JPL External Publication
12. Sponsoring Agency Name and Address NATIONAL AERONAUTICS AND SPACE ADMINISTRATION Washington, D.C. 20546		14. Sponsoring Agency Code RE223 BP-838-10-04-54-00

15. Supplementary Notes

16. Abstract

The payload of several future unmanned space missions will include a sophisticated gamma ray spectrometer. Severely constrained data rates during certain portions of these missions could limit the possible science return from this instrument. This report investigates the application of universal noiseless coding techniques to represent gamma ray spectrometer data more efficiently without any loss in data integrity. Performance results demonstrate compression factors from 2.5:1 to 20:1 in comparison to a standard representation. Feasibility was also demonstrated by implementing a microprocessor breadboard coder/decoder using an Intel 8086 processor.

17. Key Words (Selected by Author(s))	18. Distribution Statement
Spacecraft Communications, Command and Tracking Communications Geosciences and Oceanography (General) Mathematical and Computer Sciences (Gen)	Unclassified; unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	29	

JPL PUBLICATION 85-53

Noiseless Coding for the Gamma Ray Spectrometer

Robert F. Rice **Jun-ji Lee**

June 1985

NASA
National Aeronautics and
Space Administration

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

The research described in this publication was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.

ABSTRACT

The payload of several future unmanned space missions will include a sophisticated gamma ray spectrometer. Severely constrained data rates during certain portions of these missions could limit the possible science return from this instrument. This report investigates the application of universal noiseless coding techniques to represent gamma ray spectrometer data more efficiently without any loss in data integrity. Performance results demonstrate compression factors from 2.5:1 to 20:1 in comparison to a standard representation. Feasibility was also demonstrated by implementing a microprocessor breadboard coder/decoder using an Intel 8086 processor.

ACKNOWLEDGEMENT

The authors wish to thank Ron Draper and William Purdy for their programmatic support of this work; Al Metzger and Jack Trombka for their technical advice; and Bruce Parham for demonstrating the feasibility of implementing these results in microprocessor form.

CONTENTS

Figures

Table

NOISELESS CODING
FOR THE GAMMA RAY SPECTROMETER

INTRODUCTION

Both the Mars Observer and the Mariner Mark II Comet Rendezvous Asteroid Flyby (CRAF) missions are expected to fly "spectrum" instruments. Such instruments must periodically communicate spectrum samples in the form of histograms. This report investigates the application of **universal noiseless coding** techniques to represent such data more efficiently without any loss in data integrity. While investigation focuses on the demanding task of efficiently coding 8192 bin Gamma Ray Spectrometer spectra for the Mars Observer mission, the techniques and analysis provided should more generally apply to other spectrum instruments with little modification.

SUMMARY

A single coding algorithm which will efficiently represent gamma ray spectrometer spectra at any spectrum update rate from 5 seconds up to 5 minutes is identified. Data rate compression factors from 20:1 to 2.5:1 compared to a standard spectrum representation were demonstrated over this range using representative Mars Observer Gamma Ray Spectrometer test data. Looking at these results in another way, spectrum updates can be accomplished 7 to 23 times more frequently with noiseless coding than with a standard representation if both approaches use the same data rate, R, (where $250 \leq R \leq 800$ bits/s). Whereas spectrum updates every 30 seconds would require roughly 2500 bits/s by standard means, spectrum updates using noiseless coding could be accomplished every 10 seconds using only 500 bits/s.

The presentation of these results will begin with discussions leading to the standard data representation for instrument generated spectra to be used as a means of comparison. We will then develop the structure and performance criteria for a Gamma Ray Spectrometer Noiseless Coder and finally graph and tabulate the results of performance runs on a representative test set. Additional techniques are developed to improve performance under unusual burst mode conditions.

THE BASIC INSTRUMENT

A "spectrum instrument" monitors the rate of arrival of photons or particles which have energies lying within one or more instrument energy ranges. This is accomplished by counting the number of hits in each "energy bin" over a known sampling interval. The average rate for a given energy bin is simply the number of hits divided by the sampling interval. A **spectrum** can be viewed as the composite of the counts or arrival rates of all the instrument energy bins (essentially a histogram).

"Raw" Data Representation

Let

$$N \tag{1}$$

be the number of possible energy bins detectable by the spectrum instrument. And let[†]

$$n = \lceil \log_2 N \rceil \text{ bits} \tag{2}$$

be the number of bits required in a fixed length binary code to identify any one of the N bins. Of practical significance is the case where

$$N = 2^n$$

as for the proposed Mars Observer Gamma Ray Spectrometer Instrument where n = 13 bits and N = 8192 bins.

Now assume that all the hits occurring in a time interval of

$$\tau \text{ seconds} \tag{3}$$

are collected in a buffer in order of occurrence. Letting

$$\alpha_\tau \tag{4}$$

[†] $\lceil x \rceil$ is the smallest integer greater than or equal to x.

be the number of such hits and a_i the bin number of the i^{th} hit, the buffer contents (prefaced by the value of α_τ) would appear as the sequence

$$\tilde{B} = \alpha_\tau\, a_1 a_2 a_3 \cdots a_{\alpha_\tau} \quad . \tag{5}$$

$\tilde{B}$ is viewed as the "raw data" mode of representation for a spectrum instrument. By communicating $\tilde{B}$ every τ seconds we will know the order of occurrence of each hit in the interval but we will not know precisely when they occurred.

Consider the raw data rate. Note that the number of hits in an interval of length τ is not deterministic. The number of arrivals is in fact well modeled as Poisson[1] with parameter

$$\bar{\lambda} \text{ hits/s} \tag{6}$$

as the composite average arrival rate for all the energy bins. Then, the expected number of hits in interval τ is given as

$$E\{\alpha_\tau\} = \bar{\lambda}\tau \text{ hits} \quad . \tag{7}$$

For a given $\bar{\lambda}\tau$, we use

$$L \text{ bits} \tag{8}$$

to represent α_τ, where any number greater than $2^L - 1$ is set to $2^L - 1$ and the remaining samples of α_τ are discarded. L is chosen so that there is little chance that $2^L - 1$ is exceeded.

The total average bit rate required to communicate $\tilde{B}$ in (5) is then

$$R_R = \frac{n\bar{\lambda}\tau + L}{\tau}$$

$$= n\bar{\lambda} + L/\tau \text{ bits/s} \tag{9}$$

where we see that the contribution for L/τ becomes negligible as τ increases.

For the Mars Observer primary operating environment we have $\bar{\lambda} = 250$ hits/s, $n = 13$. If we take $\tau = .05$ seconds, the expected number of hits is 12.5. By using $L = 8$, the number of hits would have to exceed 255 before any loss in data occurred. The required "raw" data rate is then, from (9),

$$R_R = 3410 \text{ bits/s} \quad . \tag{10}$$

Under rare burst mode conditions (solar flares) $\bar{\lambda}$ may increase by an order of magnitude, Then taking $L = 12$, we have

$$R_R = 32740 \text{ bits/s} \tag{11}$$

under burst mode conditions.

Standard Spectrum Representation

Let

$$\tilde{S} = s_1 s_2 \cdots s_N \tag{12}$$

be the complete sampled spectrum generated over a τ second interval where

$$s_i = \begin{cases} \text{count of the} \\ \text{number of hits} \\ \text{in energy bin } i \end{cases} \tag{13}$$

We will later focus on minimizing the average number of bits needed to represent $\tilde{S}$ for any choice of τ. Here we will define the standard representation of $\tilde{S}$ which will later be used as a means of comparison.

Let $m_i(\tau)$ be the number of hits in bin i in a time interval of τ seconds. If by observation or probability model we can say that

$$\Pr\left[m_i(\tau) > M_i^{max}(\tau)\right] < \epsilon \tag{14}$$

where ϵ is very small, we need no more than

$$b_i(\tau) = \lceil \log_2 M_i^{max}(\tau) \rceil \text{ bits} \qquad (15)$$

to represent almost any occurrence of hits in bin i in an interval of τ seconds. As a backup, if a count were larger than $2^{b_i(\tau)} - 1$ (with probability $< \epsilon$) we could, at worst, truncate the identification of number of counts to $2^{b_i(\tau)} - 1$.

Obviously, a smaller interval, τ, and or a lower hit rate means there is less chance of the number of hits exceeding some threshold. Thus to achieve a given level of protection from count truncation (ϵ in (14)) the number of bits required in (15) can be quite different for different i and τ. This is particularly true for the Gamma Ray Spectrometer. However, standard spectrum instrument data handling does not capitalize on the broad disparity in quantization requirements across all the bins and possible spectrum intervals. Basically, the quantization requirements for all bins, b*, is chosen as

$$b^* = \underset{i,\tau}{MAX} \lceil \log_2 M_i^{max}(\tau) \rceil \quad . \qquad (16)$$

That is, b* satisfies the worst case requirements for all bins and the longest expected spectrum interval. The communication of each spectrum then requires

$$b^* \text{ bits/bin} \quad . \qquad (17)$$

By (1) there are N bins so that spectra communicated every τ seconds require

$$R_S(\tau) = \frac{Nb^*}{\tau} \text{ bits/s} \qquad (18)$$

which we define as the "Standard Spectrum Data Rate".

A plot of $R_S(\tau)$ and R_R is shown in Fig. 1 where we have taken

$$b^* = 9 \text{ so that } R_S(\tau) = \frac{73700}{\tau} \text{ bits/s} \qquad (19)$$

for the Mars Observer Gamma Ray Spectrometer.

We have marked some particular points of interest on the graph:

1) The standard spectrum representation is better than a raw data representation at spectrum update intervals in excess of about 21 seconds.

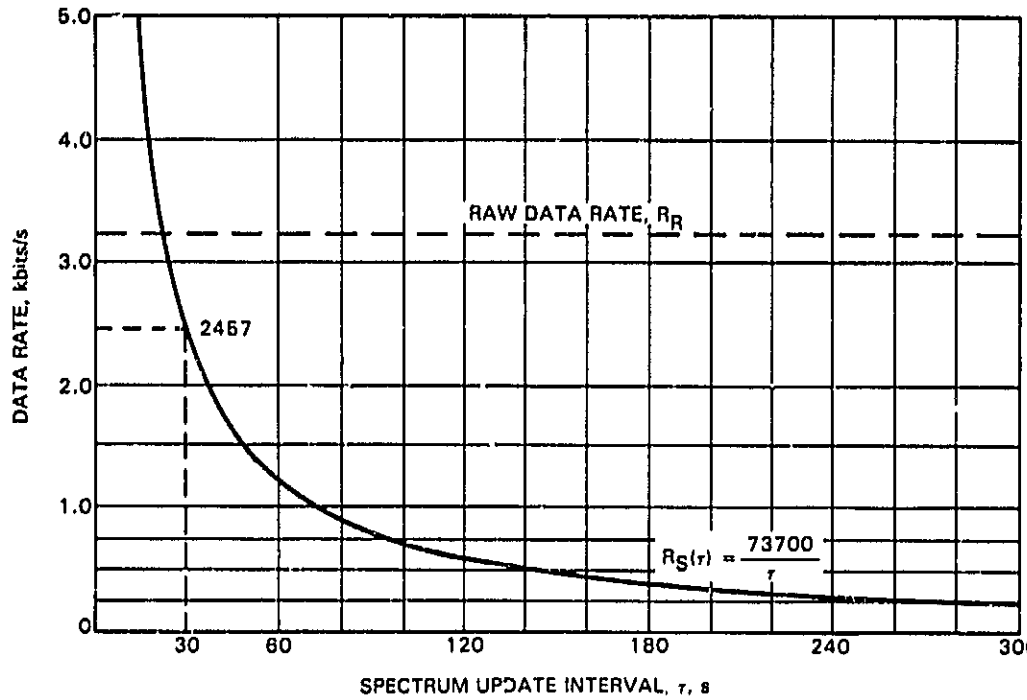

Fig. 1. Data Rates for Standard Spectrum Representation, $R_S(\tau)$.

2) Spectrum updates as frequently as every 30 seconds are highly desirable from a science point of view since update rate determines ground resolution on Mars Observer. Spectrum updates at 30 second intervals would require almost 2500 bits/s which is an order of magnitude more than may be available during certain portions of the Mars Observer mission.

STATISTICAL CHARACTERISTICS (Gamma Ray Spectrometer)[1]

The arrival of hits in the i^{th} bin is made up of two independent components: **background** (mostly cosmic rays), and the spectral emissions from the elements and compounds which the instrument hopes to detect. The latter category we label as the **science** component.

The arrival of hits from both components can be closely modeled as Poisson with parameters

$$\lambda_i^B \tau \quad \text{and} \quad \lambda_i^S \tau \tag{20}$$

respectively.

The probability of k hits in time interval τ by a Poisson process with arrival rate λ is

$$p_k(\tau) = \frac{(\lambda\tau)^k}{k!} e^{-\lambda\tau}, \; k \geq 0 \, , \; \tau \geq 0 \tag{21}$$

and the expected number of hits is

$$E\{k\} = \lambda\tau \text{ hits} \quad . \tag{22}$$

The overall arrival process for the i^{th} bin is Poisson with parameter $\lambda_i\tau$ given as

$$\lambda_i\tau = \left(\lambda_i^B + \lambda_i^S\right)\tau \quad . \tag{23}$$

Can we consider the arrival process independent from bin to bin? The background process is clearly independent from bin to bin. On the other hand, a single compound may generate hits in several bins (characteristic lines) so that this is not entirely true for the science component. However

$$\lambda_i^B >> \lambda_i^S \tag{24}$$

and the coding process we will consider later will deal with intervals much too short to distinguish spectral lines (if they were even there; the science component may or may not be present). Then for all practical purposes we can take the overall arrival process as Poisson and independent from bin to bin.

Note that by the same arguments, the arrival of instrument hits is Poisson with arrival rate

$$\bar{\lambda} = \sum_{i=1}^{N} \lambda_i \tag{25}$$

where $\bar{\lambda}$ is the same term introduced in (6).

Observations on the λ_i

The following observations will influence our spectrum coding approach in subsequent sections.

- The λ_i's tend to decrease with increasing bin value i. (26)

- The largest λ_i may be as much as three orders of magnitude larger than the smallest. (27)

- Adjacent λ_i's tend to be similar. That is, if $i \approx j$ then $\lambda_i \approx \lambda_j$. (28)

We are now ready to develop an efficient spectrum noiseless coder.

II. SPECTRUM NOISELESS CODER

NOTATION AND DEFINITION

The application of noiseless code operator

$$\psi[\bullet] \tag{29}$$

to sequence $\widetilde{X}$ produces the coded sequence $\psi[\widetilde{X}]$, and given $\psi[\widetilde{X}]$ we can retrieve the original $\widetilde{X}$ precisely. That is, we can recover the original "noiselessly".

In general we subscript and superscript $\psi[\bullet]$ to specify different choices of operators. The particular operator then means those operations that are performed to produce a coded sequence.

$$\pounds(\psi[\widetilde{X}]) \tag{30}$$

means the "length" of sequence $\psi[\widetilde{X}]$ so $\pounds[\bullet]$ means the calculation of the length of some sequence.

When emphasizing the concatenation of two or more items we will use an asterisk, *.

PREPROCESSING

The basic form for a noiseless coder is shown in Fig. 2[2], [3].

The first steps are to perform instrument-specific "preprocessing" operations which seek to produce a sequence of independent non-negative integers, $\widetilde{\delta}$, such that a smaller integer tends to occur more often than a larger one. That is,

$$\Pr[\delta = 0] \geq \Pr[\delta = 1] \geq \Pr[\delta = 2] \geq \ldots . \tag{31}$$

The second instrument-independent step, indicated by $\psi_\ell[\bullet]$ in Fig. 2, is to efficiently represent such sequences of integers. The underlying technique is to assign shorter code words to the more likely smaller integers and larger code words to the less likely larger integers such that, on the average, fewer bits are used. We will not need to worry about designing $\psi_\ell[\bullet]$. References 2 and 3 document a broad class of adaptive algorithms (code operators) which will provide efficient coding of a preprocessed δ sequence under virtually any practical situation. We will later merely identify the operator we elect to use.

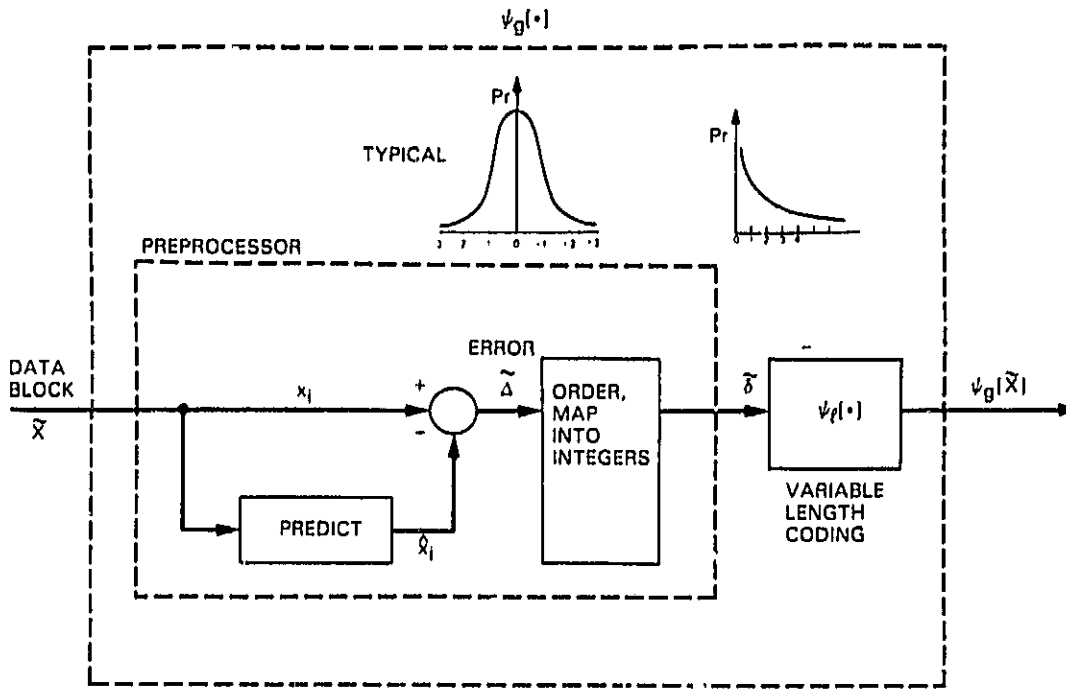

Fig. 2. Basic Noiseless Coder Structure.

The problem here is to identify the instrument-specific preprocessing that is appropriate for the Gamma Ray Spectrometer.

Prediction and Ordering

Letting $\hat{x}_i$ be the prediction of the actual number of hits x_i, the prediction error is given as

$$\Delta_i = x_i - \hat{x}_i \quad . \tag{32}$$

Then assuming we know λ_i for the i^{th} bin we take $\hat{x}_i$ as the closest integer to the expected number of hits, that is from (22)† and (23)

$$\hat{x}_i = [\lambda_i \tau] \quad . \tag{33}$$

The resulting error distribution is given directly by the Poisson distribution

†[a] is the closest integer to a.

$$Pr[\Delta_i = j] = Pr[k = [\lambda_i \tau] + j] \quad \text{for any } j \tag{34}$$

where the latter term can be calculated by substitution in (21). Using this distribution we can get the exact ordering which would allow us to map the error Δ_i into the integers such that condition (31) is met.

However, a prediction about the mean, $\lambda_i \tau$, generally results in a unimodal error distribution around zero. When $\lambda_i \tau >> 1$ the standard error mapping in Table 1 generates the same ordering (for all terms of consequence) as using the Poisson distribution itself. This can be modified to account for the boundary conditions (when $\lambda_i \tau$ is near zero) resulting from the fact that the number of counts cannot be less than zero[1].

Table 1. Basic Mapping of Δ_i into the Integers δ_i.

PREDICTION ERROR Δ_i	INTEGER δ_i
0	0
+1	1
−1	2
+2	3
−2	4
+3	5
−3	6
.	.
.	.
.	.

Specifically, with x_{MAX} equal to the largest allowed count value, we can map the error Δ_i as

$$\delta_i = \begin{cases} 2\Delta_i - 1 & \text{if } 0 < \Delta_i \leq \hat{x}_i \\ x_i & \text{if } \Delta_i > \hat{x}_i \\ 2|\Delta_i| & \text{if } x_i - x_{MAX} \leq \Delta_i \leq 0 \\ x_{MAX} - x_i & \text{if } \hat{x}_i - x_{MAX} > \Delta_i \end{cases} . \tag{35}$$

This mapping yields almost identical results, under all conditions, as that produced by direct use of the Poisson distribution.

Based on these observations, all future discussions will presume one of the following preprocessing options:

- We do not attempt to predict at all and simply take

$$\delta_i = x_i \tag{36}$$

  for each sample (bin).

- We predict x_i as the closest integer to $\lambda_i\tau$ and then use the mapping in (35) to convert the prediction errors into integers. The $\lambda_i\tau$ are derived from long term average statistics. (37)

PERFORMANCE MEASURE, ENTROPY

Consider a single bin spectrum driven by a Poisson process with parameter $\lambda\tau$. Then the probability of k hits, $p_k(\tau)$, is given by Eq. 21. The entropy of this distribution is given as

$$H(\lambda\tau) = -\sum_{k=0}^{\infty} p_k(\tau) \log_2 p_k(\tau) . \tag{38}$$

This expression represents the **minimum** possible average rate (averaged over many spectra) that **any noiseless coder** could code the counts of this single bin spectrum. A plot of $H(\lambda\tau)$ vs. $\lambda\tau$ is given in Fig. 3.

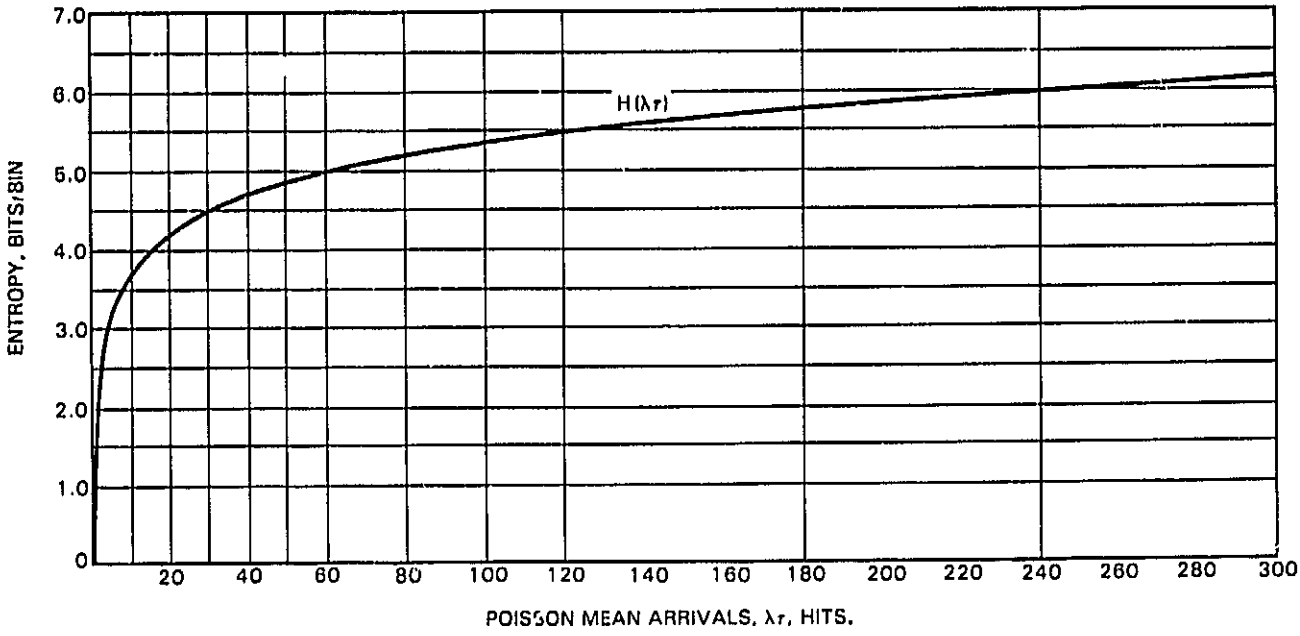

Fig. 3. Poisson Entropy, $H(\lambda\tau)$.

If $\lambda\tau$ were fixed and known then a long sequence of counts taken at τ second intervals could be represented with no fewer than $H(\lambda\tau)$ bits/bin count. A coder which coded closely to $H(\lambda\tau)$ would be considered efficient.

Now suppose we applied a **separate noiseless coder** to each of the N = 8192 independent bins of the Gamma Ray Spectrometer. We could expect to code no better than

$$H_{\lambda}(\tau) = \frac{1}{N}\sum_{i=1}^{N} H(\lambda_i\tau) \text{ bits/bin} \quad . \tag{39}$$

Using long term average values for the $\{\lambda_i\}$ we call this term the **Gamma Ray Average Entropy.**

On the other hand, if we restricted ourselves to a **single fixed noiseless coder** for all bins, we could expect to code no better than

$$H\left(\frac{\bar{\lambda}\tau}{N}\right) \geq H_{\lambda}(\tau) \tag{40}$$

where $\bar{\lambda}$ is the average arrival rate over all bins given in (24).

For the Gamma Ray Spectrometer we must also worry about efficiently coding each of the different bins for a broad range of τ. The performance goal we're shooting for is to code close to $H_\lambda(\tau)$ in (39) for any τ of interest.

A plot of the various entropies derived from representative Mars Observer test data is shown in Fig. 4.

These same curves have been converted to the equivalent data rate by the formula (see 18)

$$\text{data rate} = \frac{N \cdot \left[\begin{array}{c}\text{rate in} \\ \text{bits/bin}\end{array}\right]}{\tau} \text{ bits/s} \qquad (41)$$

and are plotted again in Fig. 5. $H(\lambda\tau/N)$ and $H_\lambda(\tau)$ have been replaced by $H'(\lambda\tau/N)$ and $H'_\lambda(\tau)$ respectively.

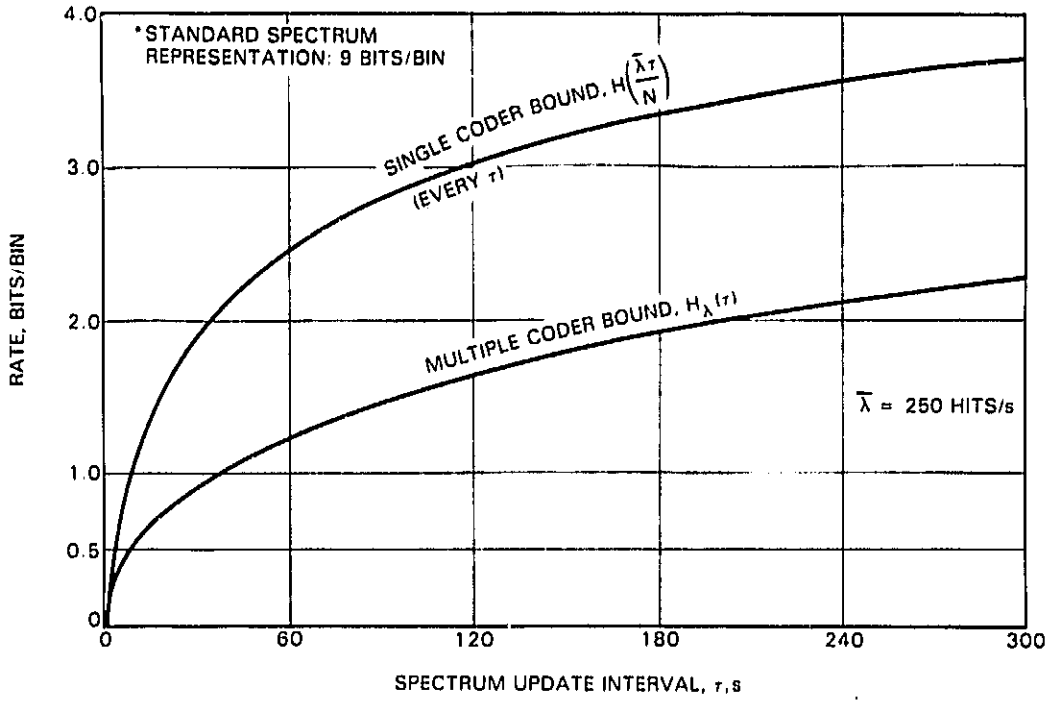

Fig. 4. Spectrum Coder Entropy Performance Bounds.

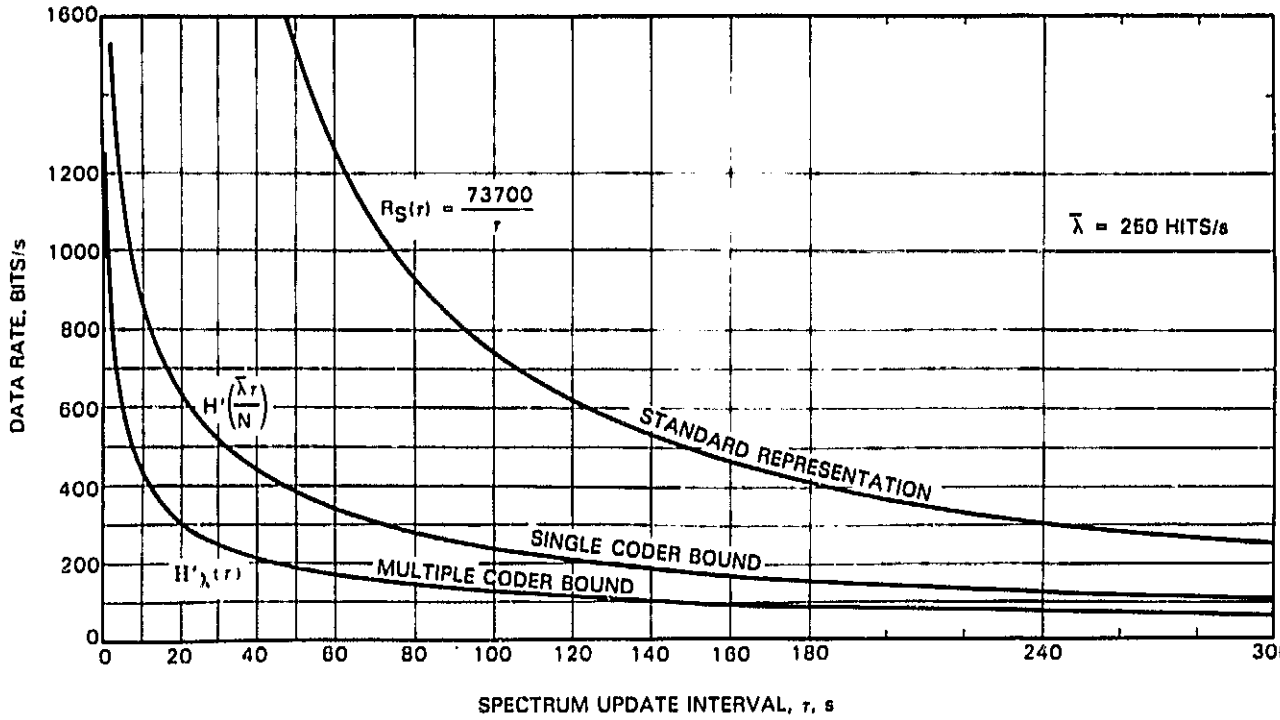

Fig. 5. Spectrum Coder Performance Bounds in bits/s.

The first observation from these curves is that the lower curve for $H'_\lambda(\tau)$ is dramatically lower than the corresponding rate for a standard representation. Second, the curve for $H'_\lambda(\tau)$ is significantly lower than that of $H'(\bar{\lambda}\tau/N)$. Unfortunately, the apparent implications for achieving the performance of the lower curve is to build a separate coder for each $\lambda\tau$ combination. This will not be necessary. In the following sections we will complete the definition of a coder which approaches the performance of the lower curve, for all τ, with complexity closer to that of a single coder.

COMPLETING THE DEFINITION

Partitioning the Spectrum

The observations in (25)-(27) give us the necessary clues to avoid the implied individual coder for each bin (and τ). We need only partition the spectrum into subblocks and code each subblock separately. So let $S = s_1 s_2 s_3 \ldots s_N$ in (12) be rewritten as

$$\tilde{S} = \tilde{X}_1 * \tilde{X}_2 * \tilde{X}_3 * \ldots * \tilde{X}_M \tag{42}$$

where the $\tilde{X}_i$'s are consecutive subblocks of the samples of $\tilde{S}$. That is, if s_v is the last sample of $\tilde{X}_i$ then s_{v+1} is the first sample of the next subblock, X_{i+1}.

Gamma Ray Code Operator

We define the overall Gamma Ray Spectrometer spectrum noiseless code operator as $\psi_G[\cdot]$ where

$$\psi_G[\tilde{S}] = \psi_g[\tilde{X}_1]^*\psi_g[\tilde{X}_2]^* \ldots . \psi_g[\tilde{X}_M] \tag{43}$$

and where $\psi_g[\cdot]$ is a code operator that works on subblocks. In this report we take $\psi_g[\cdot]$ as the structure in Fig. 2 with the preprocessing assumptions in (36) and (37) and

$$\psi_\ell[\cdot] = \psi_{10}[\cdot] \tag{44a}$$

defined explicitly in Ref. 2 or

$$\psi_\ell[\cdot] = \text{8-option FAST Compressor} \tag{44b}$$

defined in Ref. 3. Both of these are too involved to go into here.

The next section will look at the performance of this configuration.

Breadboard

A breadboard coder/decoder of $\psi_G[\cdot]$ (using $\psi_\ell[\cdot] = \psi_{10}[\cdot]$) was recently built, and tested by Bruce Parham using an Intel 8086 microprocessor.[5] The breadboard coder requires roughly 1 Kbyte of memory for instructions and another 1 Kbyte for internal buffers. A decoder is about half this size. Both will handle throughput data rates of 20 kbits/s, far higher than the allowable data rates on Mars Observer.

SPECTRUM CODER PERFORMANCE RESULTS

Using $\psi_{10}[\cdot]$

Gamma Ray Spectrometer test data, highly representative of the expected Mars Observer environment, was provided by Jack Trombka of Goddard[4]. It consisted of roughly 1/2 hour of raw data acquired from a recent balloon flight.

The noiseless coder just defined was run on the entire data set for different values of τ, running from 5 seconds up to 5 minutes†. The performance results are displayed in Figs. 6-9.

Figure 6 shows performance in bits/bin with the Gamma Ray Spectrometer noiseless coder rate denoted by $R_G(\tau)$ and where

$$R_G(\tau) \approx 0.189\sqrt{\tau} \text{ bits/bin} \quad . \tag{45}$$

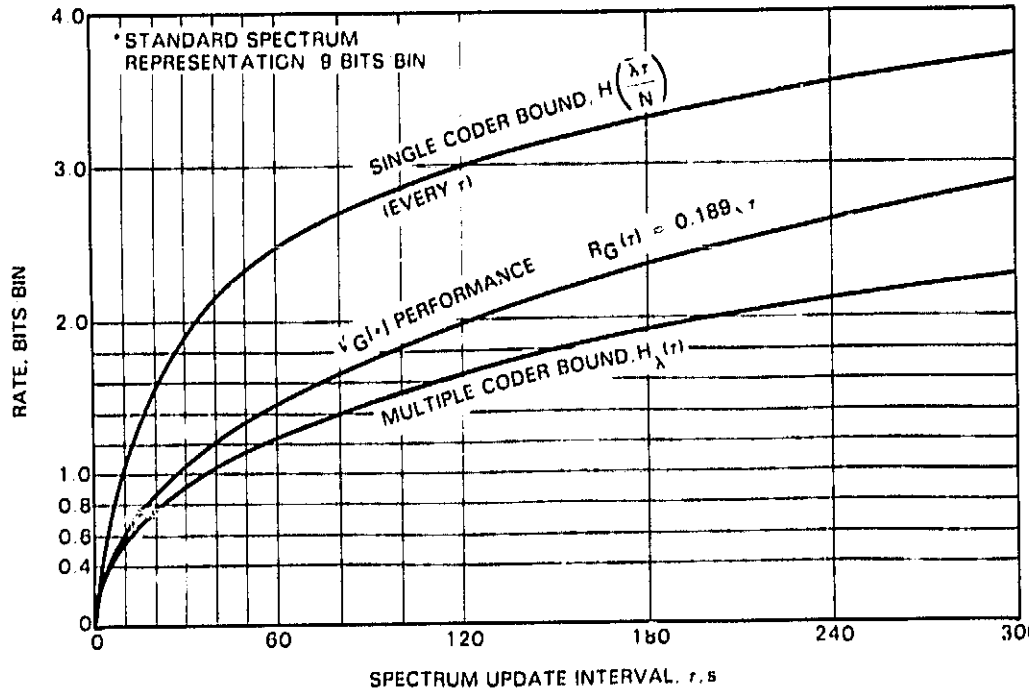

Fig. 6. $\psi_G[\cdot]$ Performance in bits/bin Using $\psi_{10}[\cdot]$.

†Each of these curves presumes the preprocessing assumption in (37). Using the simple approach in (36) (basically no a priori knowledge) resulted in only minor losses. Performance reductions were only observed at data rates of less than 500 bits/s and were typically less than 4%. The largest performance loss observed was less than 10%.

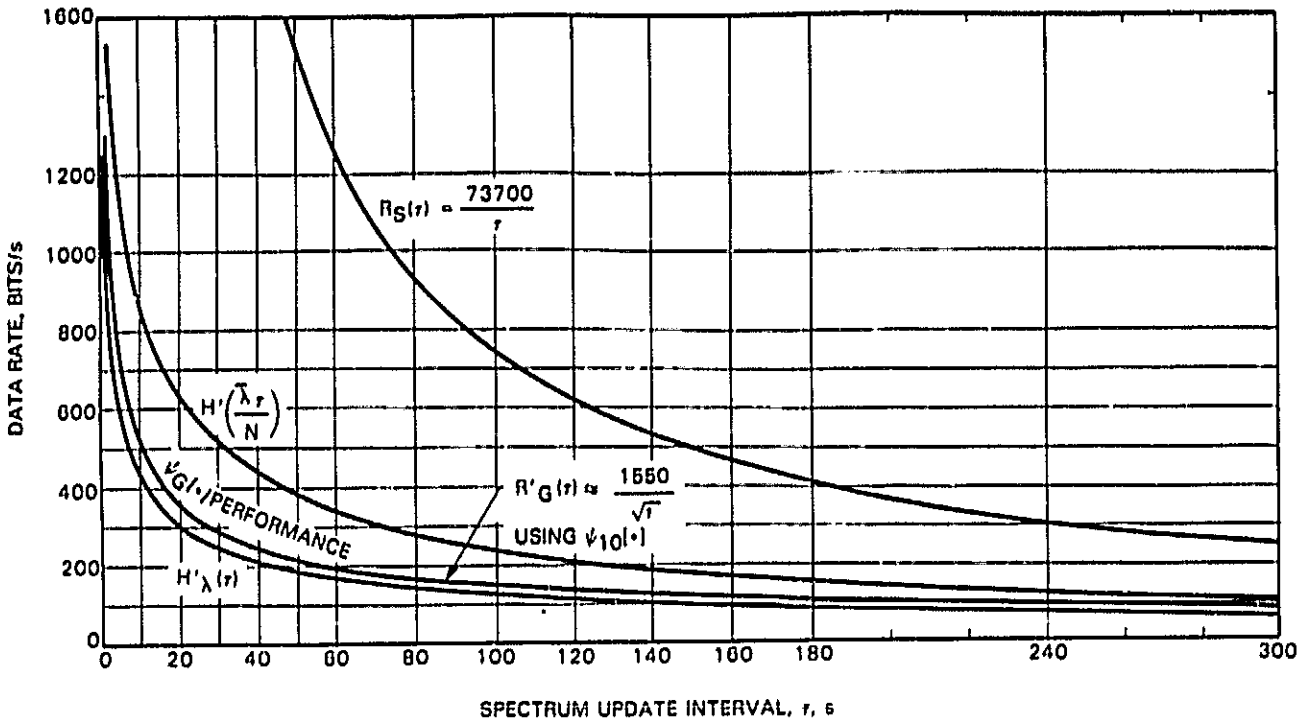

Fig. 7. $\psi_G[\cdot]$ Performance in bits/s Using $\psi_{10}[\cdot]$.

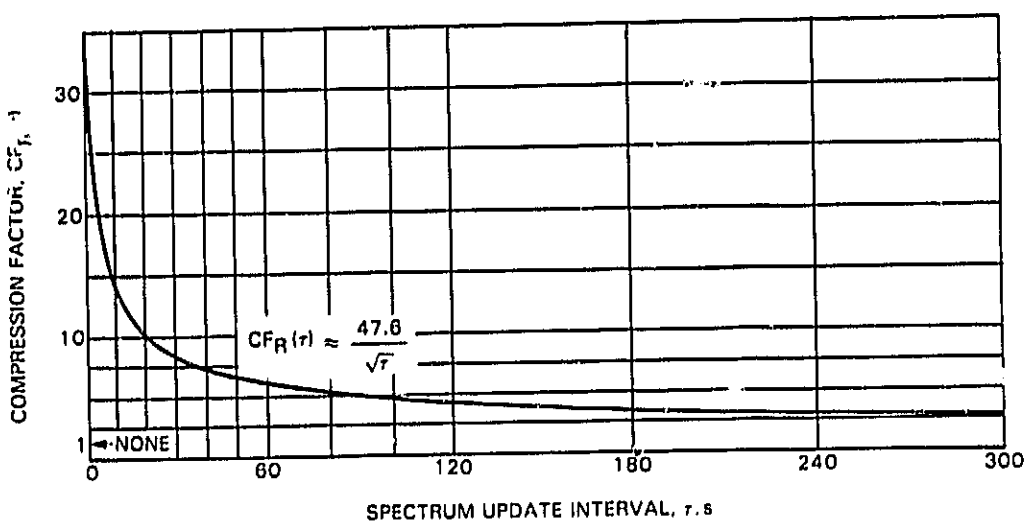

Fig. 8. $\psi_G[\cdot]$ Data Rate Compression Factor, $CF_R(\tau)$ Using $\psi_{10}[\cdot]$.

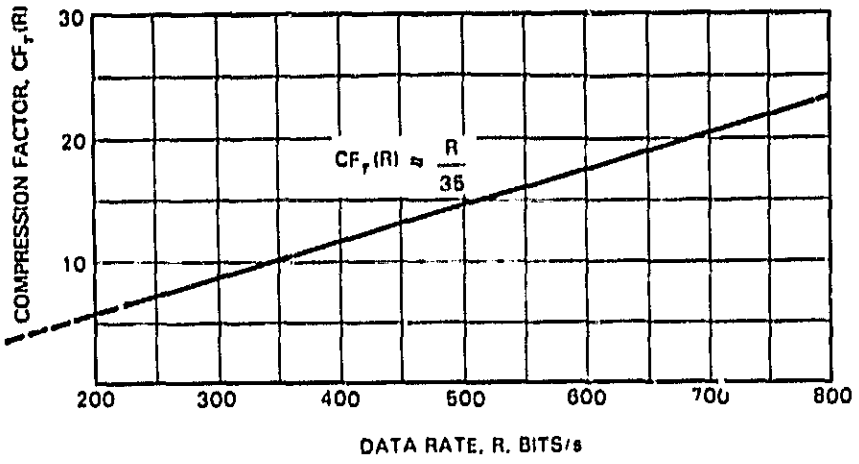

Fig. 9. $\psi_G[\cdot]$ Spectrum Update (τ) Compression Factor, $CF_\tau(R)$ Using $\psi_{10}[\cdot]$.

These curves have been translated to bits/s in Fig. 7 using the relation in (41). The corresponding performance for the Gamma Ray Spectrometer noiseless coder is denoted by $R'_G(\tau)$ where

$$R'_G(\tau) \approx \frac{1550}{\sqrt{\tau}} \text{ bits/s} \quad . \tag{46}$$

Additionally, the performance of $\psi_G[\cdot]$ relative to a standard representation is shown in Fig. 8 as a **data rate** compression factor $CF_R(\tau)$ given by

$$\begin{aligned} CF_R(\tau) &= \frac{9}{R_G(\tau)} \\ &= \frac{R_S(\tau)}{R'_G(\tau)} \\ &\approx \frac{47.6}{\sqrt{\tau}} \end{aligned} \tag{47}$$

when $\psi_{10}[\cdot]$ is used.

Figure 8 compares the average communication rates that can be achieved for a given spectrum update rate. Perhaps more important is a comparison of the spectrum update rates that can be achieved for a given data rate. For this we define a second compression factor $CF_\tau(R)$, where

$$CF_\tau(R) = \frac{\left\{\begin{array}{l}\tau \text{ such that the} \\ \text{standard spectrum} \\ \text{representation} \\ \text{requires R bits/s}\end{array}\right\}}{\left\{\begin{array}{l}\tau' \text{ such that } \psi_G[\cdot] \\ \text{performance} \\ \text{requires R bits/s}\end{array}\right\}} \tag{48}$$

which is very close to a straight line with intercept at R = 0 when $\psi_{10}[\cdot]$ is used. Under these assumptions we have

$$CF_\tau(R) \approx \frac{R}{35} \quad . \tag{49}$$

In fact it is this observation that led to the simpler but accurate approximations in (45) - (47). $CF_\tau(R)$ is plotted in Fig. 9.

Observations. Although these figures need to be studied closely, we can make several important observations.

- The $\psi_G[\cdot]$ coder is efficient. Its performance is close to the optimistic performance bound for multiple coders, $H_\lambda(\tau)$.

- Data rate compression factors **at a given spectrum update interval,** τ, range from 20:1 to 2.5:1 for 5s $\leq \tau \leq$ 300s.

- Compression factors for spectrum update intervals **at a given data rate,** R, range from 7.3:1 to 23:1 when data is constrained to 250 $\leq$ R $\leq$ 800 bits/s.

Of particular note are the following:

- Whereas spectrum updates every 30 seconds would require roughly 2500 bits/s by standard means, spectrum updates using noiseless coding could be accomplished every 10 seconds using only 500 bits/s.

- A data rate of 500 bits/s would allow spectrum updates no more often than once every 150 seconds if a standard representation were used. This is an update interval compression factor, $CF_{\tau}(R)$, of 15:1.

- $\psi_G[\cdot]$ coding data rate, $R'_G(\tau)$, with an update interval of 150 seconds drops to 100 bits/s.

- $\psi_G[\cdot]$ can code spectrums every 30 seconds using only 300 bits/s compared to approximately 2500 bits/s by standard means. This is a data rate compression of $CF_R(\tau) = 8.3:1$.

- A 300 bits/s data rate would allow spectrum updates no more frequently than once every 230 seconds. This is a spectrum update interval compression of $CF_{\tau}(R) = 7.6:1$.

Using a Simple Operator

Code Operator $\psi_{10}[\cdot]$ includes special modes to deal with data characteristics corresponding to very low entropies. For the Gamma Ray Spectrometer spectrums this situation occurs at the top end of a spectrum (small λ) when τ is small (less time to accumulate hits). A simpler code operator is feasible if one is willing to give up some performance at low values of τ.

An 8-option "FAST" compressor[3] operating on blocks of 10 bins was applied to the same test set.† We will denote this simpler algorithm by

$$\psi_F[\cdot] \quad . \tag{50}$$

A performance comparison of $\psi_F[\cdot]$ with $\psi_{10}[\cdot]$ is shown in Figs. 10-12. As the figures show, the advantage of $\psi_{10}[\cdot]$ becomes significant at spectrum intervals less than 1 minute. At $\tau = 30$ s, $\psi_F[\cdot]$ requires about 30% more data rate than $\psi_{10}[\cdot]$. The advantage of $\psi_{10}[\cdot]$ increases rapidly with smaller τ.

---

†This is the same algorithm which will be used to compress images at the 1986 Voyager II encounter of Uranus.

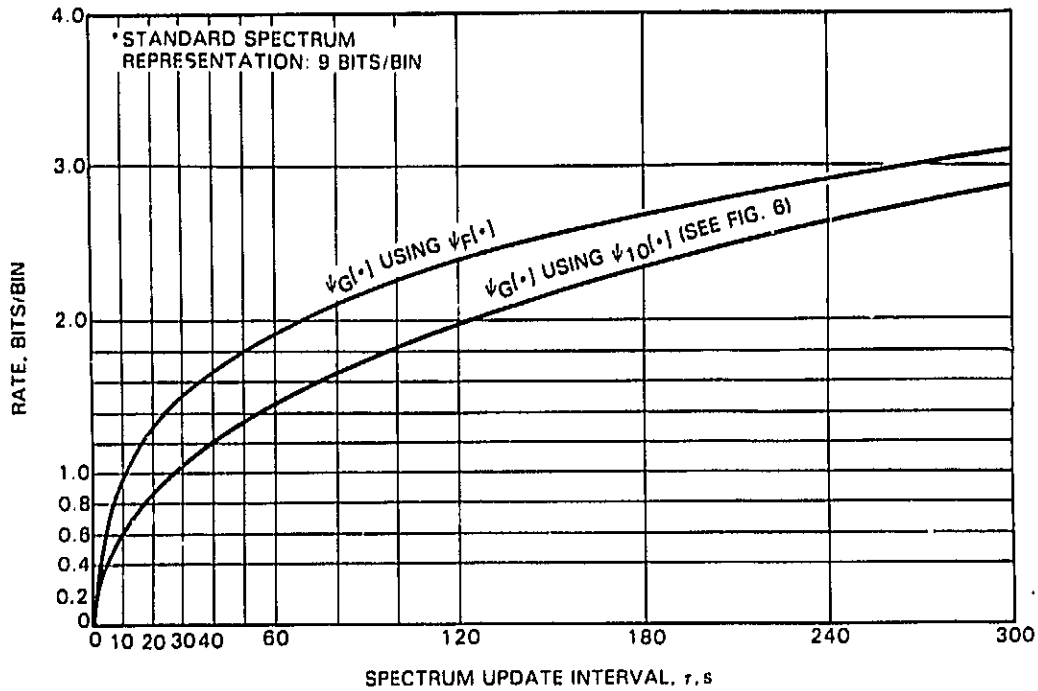

Fig. 10. $\psi_G[\cdot]$ Performance Comparison, $\psi_{10}[\cdot]$ vs. $\psi_F[\cdot]$ in bits/bin.

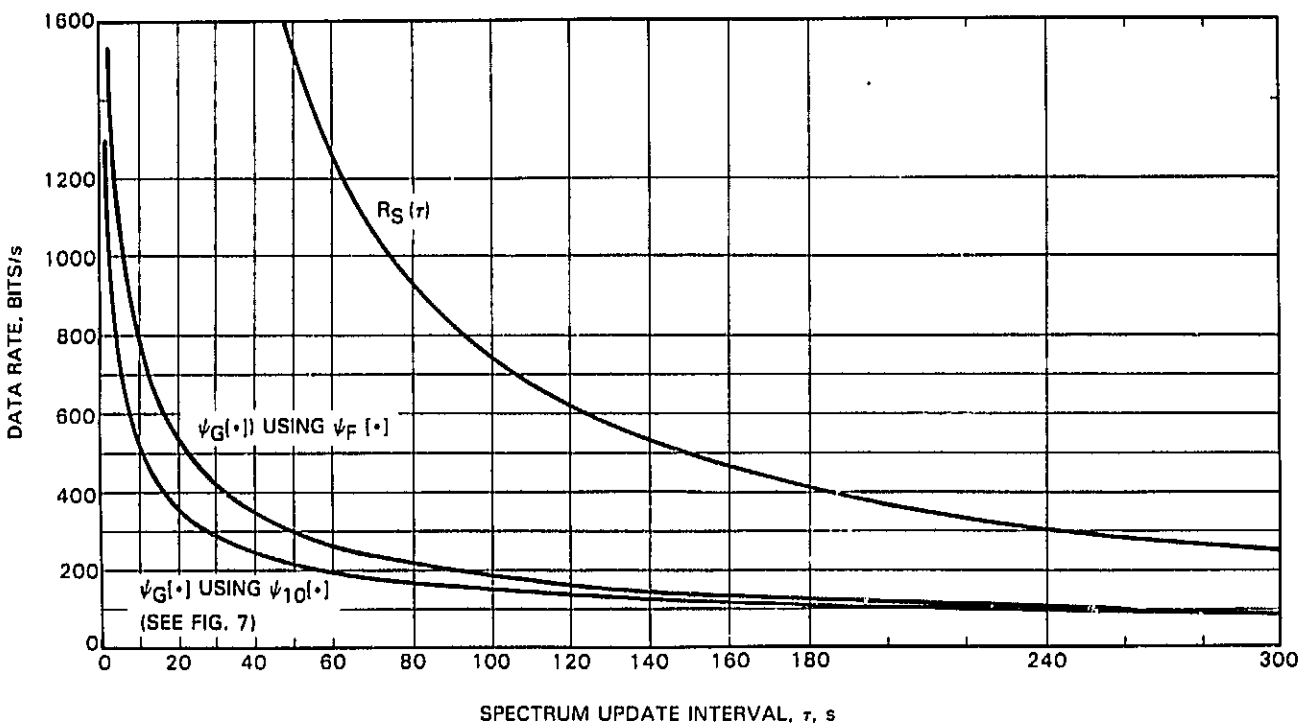

Fig. 11. $\psi_G[\cdot]$ Performance Comparison, $\psi_{10}[\cdot]$ vs. $\psi_F[\cdot]$ in bits/s.

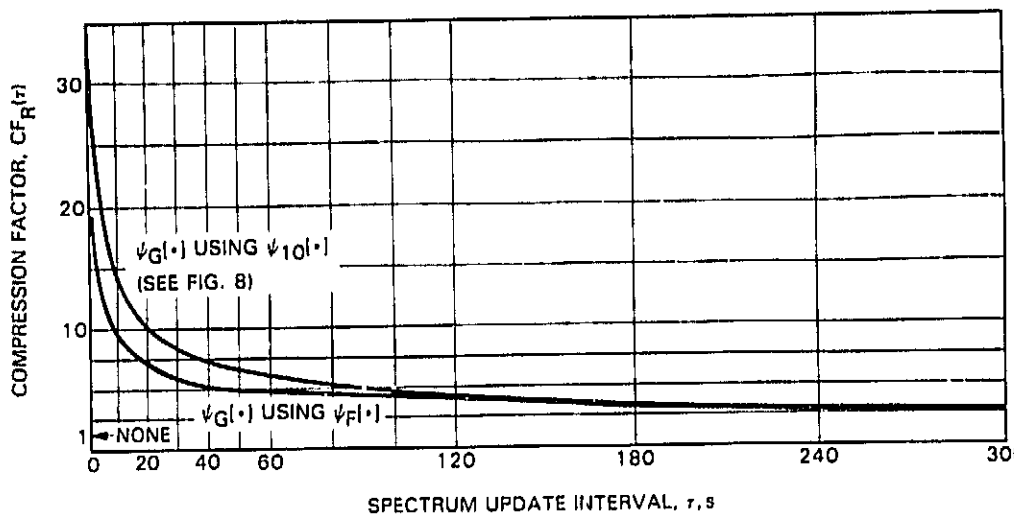

Fig. 12. $\psi_G[\cdot]$ Data Rate Compression Factor, $CF_R(\tau)$, Using $\psi_{10}[\cdot]$ or $\psi_F[\cdot]$.

III. BURST MODE CODER

Solar flares may cause a brief increase in the average arrival rate by as much as an order of magnitude. Under these conditions it is desirable to sample spectrums at update intervals as low as 0.05 s/spectrum.

The standard spectrum representation would require over 1 Mbit/s to accommodate this situation. On the other hand, the "raw data" mode in (5) is far more efficient, requiring, from (11)

$$R_R = 32740 \text{ bits/s} \quad . \tag{51}$$

ESTIMATING $\psi_G[\cdot]$ PERFORMANCE

We can use previous results to estimate the performance of $\psi_G[\cdot]$ under these burst conditions.

Let $\bar{\lambda}^N$ and $\bar{\lambda}_i^N$, $i = 1, 2, \ldots, 8192$ represent arrival rates under **nominal** conditions and $\bar{\lambda}^b$ and $\bar{\lambda}_i^b$, $i = 1, 2, \ldots, 8192$ represent arrival rates under **burst** mode conditions. Further, let τ^b denote the burst mode spectrum sampling interval.

Now if we satisfy

$$\bar{\lambda}^b = 10\,\bar{\lambda}^N \tag{52}$$

by assuming

$$\lambda_i^b = 10\,\lambda_i^N \quad \text{for all } i \tag{53}$$

we would expect the same "bits/bin" performance from $\lambda_G[\cdot]$ when

$$\left\{\bar{\lambda} = \bar{\lambda}^b\,,\ \tau = \tau^b\right\} \text{ as when } \left\{\bar{\lambda} = \bar{\lambda}^N,\ \tau = 10\,\tau^b\right\}$$

because, in both cases, each bin should (statistically) receive the same number of hits. From Eq. 6 the expected rate under burst mode conditions becomes

$$R_G^b(\tau^b) = 0.189\sqrt{10\,\tau^b} \text{ bits/bin} \quad . \tag{54}$$

Using Eq. 41 this translates to data rate as

$$R_G^{b'}(\tau^b) = \frac{(8192)(0.189)\sqrt{10}}{\sqrt{\tau^b}}$$

$$= \frac{4896}{\sqrt{\tau^b}} \text{ bits/s} \quad . \tag{55}$$

Substituting, we find that $\psi_G[\bullet]$ spectrum coding requires

$$R_G^{b'}(0.05) = 21896 \text{ bits/s} \tag{56}$$

under the stated burst mode conditions. This is roughly 11,000 bits/s less than required by a "raw data" representation (51). Subsequent paragraphs will seek to improve this result.

CODING THE DATA DIRECTLY

Rather than first collecting a spectrum we will consider coding the raw data sequence in (5), repeated here for convenience.

$$\tilde{B} = \alpha_\tau\, a_1\, a_2 \ldots . a_{\alpha_\tau} \tag{57}$$

where α_τ is the number of hits in τ seconds and a_i the value of the i^{th} hit.

By earlier discussions the contribution of α_τ to total rate is negligible compared to the sequence

$$\tilde{A} = a_1\, a_2 \ldots . a_{\alpha_\tau} \tag{58}$$

so we will concentrate on $\tilde{A}$.

Since we are ultimately interested in a spectrum from $\tilde{A}$, the order of arrivals is unimportant. Then we define the operation SORT[•] as a sorting algorithm such that

$$\text{SORT}[\tilde{A}] = \tilde{C} = c_1\, c_2 \ldots . c_{\alpha_\tau} \tag{59}$$

where the c_i are the samples of $\tilde{A}$, rearranged to place any a_i in front of any larger a_i. That is,

$$c_i \leq c_j \quad \text{if} \quad i < j \quad . \tag{60}$$

Now form the sequence

$$\tilde{D} = \text{DIFF}[\tilde{C}] = d_1\, d_2 \ldots . d_{\alpha_\tau}$$

where

$$d_1 = c_1 \geq 0$$

and (61)

$$d_i = c_i - c_{i-1} \geq 0 \quad \text{for} \quad i > 1 \quad .$$

We now define the coder of $\tilde{C}$ by

$$\psi_r[\tilde{C}] = \alpha_\tau * \psi_\ell[\tilde{D}] \tag{62}$$

where it is only necessary to select a proper coder for $\tilde{D}$.

We investigated the following noiseless coder for $\tilde{D}$

$\psi_\ell[\cdot] \equiv \psi_F[\cdot]$ = FAST COMPRESSOR (see (50) and Ref. 3):

- operating on blocks of 10 samples ;
- using 13 standard fast compressor modes; and (63)
- one special mode which identifies the presence of an all-zeroes block.

A block diagram of the overall coder of $\tilde{B}$ is shown in Fig. 13.

It should be understood that the SORT process is not reversible. Only $\tilde{C}$ can be recovered precisely. However, a spectrum can be derived without error from $\tilde{C}$.

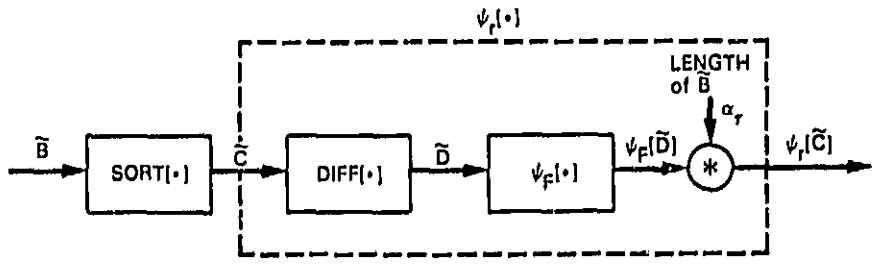

Fig. 13. Coder of Raw Data Sequence.

Performance

For notational purposes, let

$$r'(\tau^b) \tag{64}$$

denote the data rate required by the burst mode coding algorithm $\psi_r[\cdot]$, in (59)–(62) and Fig. 13, when the interval between the transmission of collected hits is once every $\tau = \tau^b$ seconds. A performance comparison of the burst mode coder and the spectrum coder $\psi_G[\cdot]$ is shown in Fig. 14 (under burst mode conditions).

Performance of the burst mode coder $\psi_r[\cdot]$ is better when $\tau^b < 0.25$ s. At the desired interval of $\tau^b = 0.05$ s, $\psi_r[\cdot]$ requires

$$r'(\tau^b = 0.05) \approx 16 \text{ kbits/s} \tag{65}$$

compared to about 22 kbits/s for $\psi_G[\cdot]$. This is a 2:1 reduction compared to a direct transmission of raw data (51).

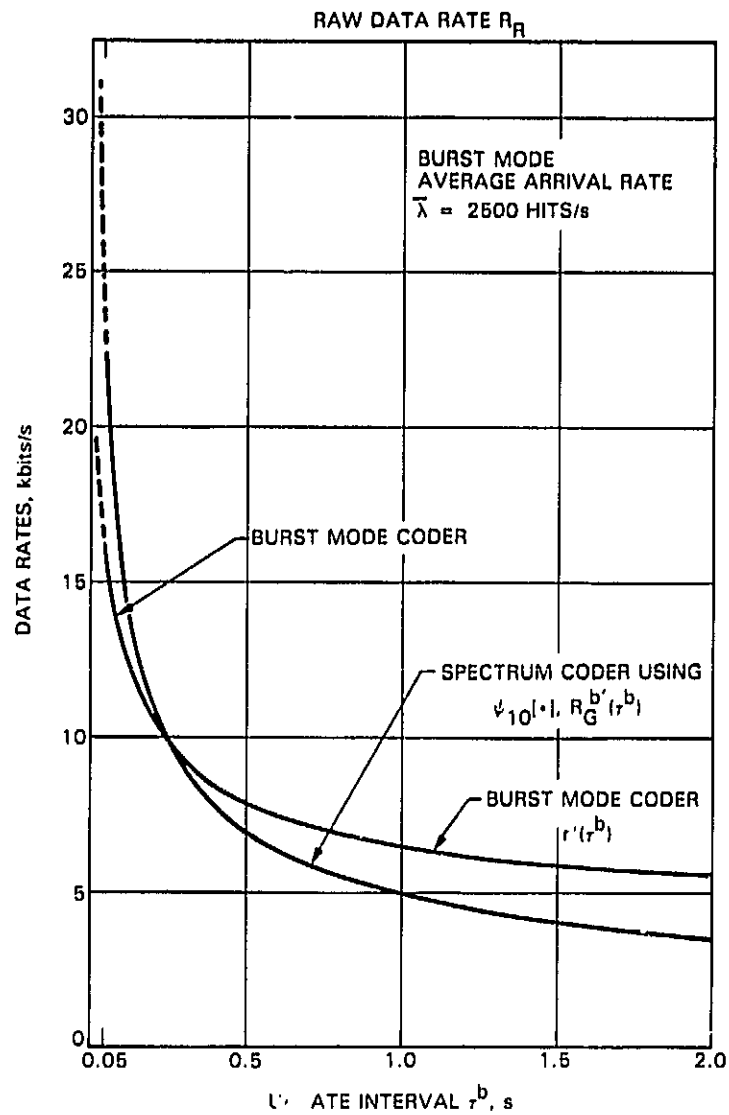

Fig. 14. Burst Mode Coding.

REFERENCES

(1) L. Kleinrock, "Queuing Systems," Volume 1, John Wiley & Sons, 1975.

(2) R. F. Rice, "Some practical universal noiseless coding techniques," **JPL Publication 79-22.** Jet Propulsion Laboratory, Pasadena, California, March 15, 1979.

(3) R. F. Rice, J. Lee, "Some practical universal noiseless coding techniques, part II," **JPL Publication 83-17.** Jet Propulsion Laboratory, Pasadena, California, March 1, 1983.

(4) J. Trombka, private communication.

(5) O. Parham, private communication.

NASA—JPL—Coml., L.A., Calif.